Machine Learning Approaches for Aircraft Stall and Flight Dynamics New Modeling: A Study on CRJ-700 and Cessna Citation X

by

Yvan Wilfried TONDJI CHENDJOU

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MAY 16TH, 2024

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mrs. Ruxandra Mihaela Botez, Thesis Supervisor
Department of Systems Engineering, École de technologie supérieure

Mr. Georges Ghazi, Thesis Co-supervisor
Department of Systems Engineering, École de technologie supérieure

Mr. Philippe Bocher, President of the Board of Examiners
Department of Mechanical Engineering, École de technologie supérieure

Mr. Tony Wong, Member of the jury
Department of Systems Engineering, École de technologie supérieure

Mr. Adrian Hiliuta, External Evaluator
CMC Electronics

THIS THESIS WAS PRENSENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

MAY 8TH, 2024

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENTS

Firstly, I am profoundly grateful to Professor Ruxandra Mihaela Botez, Head of the Laboratory of Applied Research in Active Control, Avionics, and AeroServoElasticity (LARCASE), for her immense support. Not only did she provide me the opportunity to conduct my thesis at the LARCASE, but her mentorship and financial support throughout this journey have been invaluable. I am also thankful for the chance to share my knowledge as a Teaching Assistant for the course "Introduction à l'avionique" (GPA745) under her guidance.

I express my gratitude to my co-supervisor, Professor Georges Ghazi, for his technical guidance and valuable advice throughout my thesis work. His mentorship has been instrumental in my research and academic growth.

Special thanks go to Professor Jérémy Laliberté and Dr. Salman Shafi for allowing me to join and the CREATE UTILI program. This program significantly enhanced my knowledge in Unmanned Aerial Systems and facilitated my collaboration with the Tri-Star multi-copters company team, in addition to providing financial support during my PhD.

I am grateful to FRQNT (Fonds de recherche du Québec) for their financial support, which was crucial in the pursuit of my PhD.

My PhD studies at LARCASE were enriched thanks to all my colleagues, particularly Elias Zohreh and Dr. Marine SEGUI, with whom I closely worked. I also acknowledge the intern students, Mouhamadou Wade and Benoit Diratchette, for their contributions to this work.

I would like to thank the CAE team, especially to Dr. Michel Nadeau Beaulieu, for guiding me during my internship and providing an invaluable opportunity to learn and grow professionally.

My deepest appreciation goes to my family for their invaluable support throughout my PhD studies. My parents have been a constant source of financial support and wise advice. Their sacrifices have enabled me to fulfill my dreams. I am also grateful to my siblings, Steve, Cynthia, Yann, for their emotional support. This PhD is not only my achievement but also theirs.

Lastly, I would like to thank those who became my second family during this PhD research studies: Dr. Ted Tchinde, Dr. Maxime Kuitche, Hassanatou Diakite, and Claudia Tatieze, for their friendship and support.

This PhD thesis has been achieved using a collaborative effort, and I am grateful to each one of you for your part in it.

APPROCHES D'APPRENTISSAGE AUTOMATIQUE POUR DES NOUVELLES MÉTHODES DE MODÉLISATION DU DÉCROCHAGE ET DE LA DYNAMIQUE DE VOL DES AVIONS: UNE ÉTUDE SUR LE CRJ-700 ET LE CESSNA CITATION X

Yvan Wilfried TONDJI CHENDJOU

RÉSUMÉ

Dans un contexte où l'industrie Aéronautique est confrontée aux défis de la durabilité environnementale et de l’efficacité opérationnelle, la poursuite d’une modélisation aérodynamique avancée est essentielle. Cette thèse présente une étude approfondie sur l'application des techniques avancées d'apprentissage automatique pour la modélisation aérodynamique et de la dynamique de vol des avions, en particulier pour les avions CRJ-700 et Cessna Citation X. La recherche s'appuie sur l'usage des réseaux de neurones artificiels « Artificial Neural Networks (ANN) » et de la régression par vecteurs de support « Support Vector Regression (SVR) » pour la modélisation aérodynamique des avions dans divers scénarios de vol, y compris pendant la phase critique de décrochage.

Les données essentielles pour cette étude ont été collectées grâce au simulateur de vol virtuel pour la recherche et l'ingénierie (VRESIM) du CRJ-700 et du simulateur de vol pour avions de recherche (RAFS) du Cessna Citation X. Ces deux simulateurs sont certifiés de niveau D par l'Administration fédérale de l'aviation (FAA) et ont été conçus par CAE pour la recherche au LARCASE. Une série de manœuvres diversifiées a été réalisées sur ces deux simulateurs de vol, permettant d'enregistrer un ensemble de paramètres de vols utilisés pour calculer les coefficients aérodynamiques. Cet ensemble de donnés a servi pour l'entraînement et la validation des modèles d'intelligence artificielle développés au cours de cette thèse.

La première contribution de cette recherche a mis en évidence l'efficacité des perceptrons multicouches « Multilayer Perceptrons (MLP) », une catégorie de ANN, et des SVR dans la prédiction de la dynamique longitudinale du CRJ-700 en utilisant des données de vol collectées sur le VRESIM. Grâce à l'optimisation bayésienne et la méthode de validation croisée « *k-fold* », la précision et la robustesse des modèles ont été assurés. Ces derniers ont été validés en comparant les coefficients aérodynamiques longitudinaux prédits avec des données réelles issues du VRESIM. Les modèles développés ont ainsi été capables de prédire les coefficients aérodynamiques avec des erreurs relatives moyennes absolues en dessous de 1% pour les coefficients de portance et de traînée, et en dessous de 5.74% pour les coefficients de moment de tangage. Les paramètres de vol longitudinaux simulés à partir de ces coefficients aérodynamiques prédits se sont avérés conformes aux normes de la FAA.

En s'appuyant sur la méthodologie établie pour la dynamique longitudinale, la seconde contribution aborde méthodiquement la modélisation de la dynamique latérale du CRJ-700, en utilisant les données recueillis du VRESIM. En utilisant les MLP et SVR, l’étude prédit méticuleusement les coefficients aérodynamiques latéraux, essentiels pour la manœuvrabilité et la stabilité de l'avion. Ces modèles ont démontré une très bonne précision, en affichant des erreurs relatives moyennes absolues inférieures à 5%. Les paramètres de vol latéraux simulés à partir de ces coefficients aérodynamiques prédits répondent également à tous les critères de tolérance de la FAA.

La troisième contribution a permis d’étendre cette approche à la modélisation du comportement en décrochage du CRJ-700. L'utilisation des MLP et des réseaux de neurones récurrents « Recurrent Neural Network (RNN) » a été efficace pour modéliser les phénomènes complexes du décrochage dynamique, enrichissant de ce fait notre compréhension globale du comportement de l'appareil dans ces phases critiques. Les modèles développés en utilisant la méthodologie proposée ont été capables de prédire la dynamique de vol du CRJ-700 dans des conditions de décrochage statique et dynamique avec des erreurs relatives moyennes inférieures à 1% pour les coefficients de portance et de trainée, et de 4.6% pour les coefficients de moment de tangage.

La méthodologie a été par la suite appliquée pour modéliser le décrochage dynamique de l’avion Cessna Citation X, en utilisant les données provenant du RAFS. Cette application a confirmé la flexibilité et l’adaptabilité de notre approche à différents types d’avions. Les résultats obtenus ont également montré une très grande précision, mettant en évidence la polyvalence et l'efficacité des méthodes d'apprentissage automatique développées pour prédire le décrochage des avions.

Mots-clés: modélisation, identification, aérodynamique, intelligence artificielle, apprentissage automatique, réseau de neurones, régression par les vecteurs de support.

MACHINE LEARNING APPROACHES FOR AIRCRAFT STALL AND FLIGHT DYNAMICS NEW MODELING: A STUDY ON CRJ-700 AND CESSNA CITATION X

Yvan Wilfried TONDJI CHENDJOU

ABSTRACT

In a context where the aviation industry faces the imperative challenge of environmental sustainability and operational efficiency, the pursuit of advanced aerodynamic modeling is essential. This thesis presents an in-depth study on the integration of advanced Machine Learning (ML) techniques for aerodynamic and flight dynamics modeling of aircraft, specifically focusing on the CRJ-700 and Cessna Citation X aircraft. The research leverages Artificial Neural Networks (ANN) and Support Vector Regression (SVR) for the estimation of aircraft aerodynamics across different flight scenarios, including the critical stall phase.

Essential data for this study was gathered using the CRJ-700 Virtual Research and Engineering Flight Simulator (VRESIM) and the Cessna Citation X Research Aircraft Flight Simulator (RAFS). Both simulators are Level D certified by the Federal Aviation Administration (FAA) and were designed by CAE for LARCASE research. A range of diversified maneuvers performed on these two flight simulators allowed for the recording of a comprehensive set of flight parameters data that were used to estimate aerodynamic coefficients, crucial for training and validating the artificial intelligence models developed in this thesis.

The first contribution to this research shown the effectiveness of Multilayer Perceptrons (MLP), a category of ANN, and SVR in predicting longitudinal aerodynamics of the CRJ-700 using data collected on the VRESIM. Bayesian Optimization (BO) and the *K-fold* cross validation methods ensured the accuracy and robustness of the models. They were validated by comparing the predicted longitudinal aerodynamic coefficients with experimental data obtained from the VRESIM. The developed models were able to predict aerodynamic coefficients with Mean Absolute Relative Errors (MARE) below 1% for lift and drag coefficients and below 5.74% for pitching moment coefficients. The longitudinal flight parameters simulated from these predicted aerodynamic coefficients were aligned with FAA criteria standards.

Building on the robust methodology established for longitudinal dynamics, the thesis methodically addressed the modeling of the CRJ-700's lateral dynamics, using data from the VRESIM. Employing MLP and SVR, the study meticulously predicts lateral aerodynamic coefficients, essential for the maneuverability and stability of the aircraft. These models showed very good accuracy, with MARE below 5%. The lateral flight parameters simulated from these predicted aerodynamic coefficients were also conformed with FAA guidelines.

The third contribution extended this approach to modeling the stall behavior of the CRJ-700. The innovative use of MLP and Recurrent Neural networks (RNN) was demonstrated in modeling the more complex and transient phenomena of dynamic stall, contributing to a comprehensive understanding of aircraft behavior during such critical phases. The models developed using the proposed methodology could predict the CRJ-700 flight dynamics in both static and dynamic stall conditions, with MARE below 1% for lift and drag coefficients, and 4.6% for pitching moment coefficients.

The methodology was subsequently applied to model the dynamic stall of the Cessna Citation X aircraft, using data from the RAFS. This application confirmed the flexibility and adaptability of our approach to different types of aircraft. The results obtained also showed very high accuracy, highlighting the versatility and effectiveness of the developed ML methods for predicting aircraft stall.

Keywords: modeling, identification, aerodynamics, artificial intelligence, machine learning, neural network, support vector regression

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BO	Bayesian Optimization
BR	Bayesian Regularization
CFD	Computational Fluid Dynamics
CRJ	Canadair Regional Jet
DM	Decomposition Method
ELLIOTSIG	Elliot Symmetric Sigmoid
FAA	Federal Aviation Administration
FE	Filter Error
GN	Gauss-Newton
GP	Gaussian Process
ICAO	International Civil Aviation Organization
LARCASE	Laboratory of Applied Research in Active Control, Avionics, and AeroServoElasticity

LM	Levenberg-Marquardt
LOGSIG	Log Sigmoid
LS	Least Square
MARE	Mean Absolute Relative Error
ML	Machine Learning
MIMO	Multi Input Multi Output model
MISO	Multi Input Single Output
MLP	Multilayer Perceptron
MSE	Mean Square Error
NN	Neural Networks
QP	Quadratic Programming
RAFS	Research Aircraft Flight Simulator
RMS	Root Mean Square
RNN	Recurrent Neural Network
SMO	Sequential Minimal Optimization

SVR	Support Vector Regression
TANSIG	Hyperbolic Tangent Sigmoid
VRESIM	Virtual Research and Engineering Flight Simulator

LIST OF SYMBOLS AND UNITS OF MEASUREMENT

List of Variables

a_x	x-axis coordinate of the aircraft acceleration
a_y	y-axis coordinate of the aircraft acceleration
a_z	z-axis coordinate of the aircraft acceleration
C	Support Vector Regression model regularization parameter
CD	Drag coefficient
Cl	Rolling moment coefficient
CL	Lift coefficient
Cm	Pitching moment coefficient
Cn	Yawing moment coefficient
CY	Side force moment coefficient
c_w	Mean aerodynamic chord length
H	Aircraft altitude
I_{xx}	Moment of inertia about the x body axis
I_{yy}	Moment of inertia about the y body axis

I_{zz}	Moment of inertia about the z body axis
m	Mass of the aircraft
M	Mach number
p	Aircraft roll rate
$\dot{p}$	Aircraft roll acceleration
q	Aircraft pitch rate
$\dot{q}$	Aircraft pitch acceleration
Q	Dynamic pressure
r	Aircraft yaw rate
$\dot{r}$	Aircraft yaw acceleration
R_e	Reynolds number
S_w	Wing reference area
T_x	Engine thrust force components along the x body axis
T_y	Engine thrust force components along the y body axis
T_z	Engine thrust force components along the z body axis

V_T	True airspeed
w	Weight matrix in a neural network
x_{cg}	x-axis coordinate of the aircraft center of gravity
x_{eng}	x-axis coordinate of the engine center of gravity
z_{cg}	z-axis coordinate of the aircraft center of gravity
z_{eng}	z-axis coordinate of the engine center of gravity

Greek Symbols

α	Angle of attack
β	Sideslip angle
ρ	Air density
θ	pitch angle
μ	Air dynamic viscosity
ξ_i, ξ_i^*	Slack variables
δ	Control surface deflection
ε	Support Vector Regression model precision

ϕ	Roll angle
ψ	Yaw angle
ζ	Damping ratio

INTRODUCTION

The aviation industry is currently facing significant challenges related to environmental sustainability, safety improvements, and operational efficiency. To address these concerns, the Federal Aviation Administration (FAA) has developed the United States Aviation Climate Action Plan with the goal of achieving net-zero emissions by 2050 (FAA, 2021). This global urgency to mitigate climate impacts has also prompted Boeing to invest in sustainable aviation fuels and technologies, aiming for carbon neutrality and compatibility of all commercial airplanes with sustainable aviation fuels by 2030 (Boeing, 2023).

0.1 Problem Statement and Motivations

Accurate aerodynamic modeling plays a crucial role in addressing these industry challenges. It provides a detailed understanding of airflow patterns around aircraft components, with an accurate evaluation of airfoil lift and drag, and simulation of complex internal and external geometries for airflow circulation (Abbas, De Vicente & Valero, 2013). This understanding of aerodynamics is vital in aircraft design where advancements in flow simulation, facilitated by high-performance computing, contribute to enhancing aircraft efficiency (Di Pasquale & Savill, 2022) – a key factor in meeting decarbonization goals. Furthermore, the development of comprehensive aerodynamic databases through Computational Fluid Dynamics (CFD) or wind tunnel testing remains a conventional approach that significantly benefits the conceptual design and overall improvement of aircraft performance. Overall, accurate aerodynamic modeling plays a pivotal role in addressing the challenges faced by the aviation industry, supporting the pursuit of environmental sustainability, safety improvements, and operational efficiency.

However, achieving accuracy in aerodynamic modeling presents inherent challenges. The dynamic and complex nature of the airflow around aircraft components requires precise modeling techniques to ensure reliable predictions. Although there have been advancements in aerodynamic modeling, further progress is required, especially in modeling complex phenomena like dynamic stall, which are not extensively covered in existing literature. Dynamic stall, characterized by its transient and complex nature, presents considerable challenges for accurate modeling using traditional techniques (Abbas et al., 2013; Tam, 1995). Furthermore, the growing demand for environmentally sustainable aviation technologies, as emphasized by the International Civil Aviation Organization (ICAO), underlines the importance of highly accurate aerodynamic models for designing and optimizing next-generation aircraft systems (ICAO, 2016).

In response to these challenges, Artificial Intelligence (AI) methods could present a promising opportunity to enhance aerodynamic modelling accuracy. With the ability to learn from data and generate precise models, AI might be capable of capturing complex aerodynamic behaviors that traditional methods find challenging to identify. Techniques such as ANN and SVR could potentially model complex aerodynamic phenomena accurately while reducing computational time. ANNs are recognised for learning and modelling complex dynamics through their layered structures, while SVR are known for robust and precise predictions even with limited data, potentially making them suitable for capturing nuanced aerodynamic behaviours like dynamic stall. The consideration of AI technologies in aerodynamic modelling suggests a shift towards data-driven, real-time modelling and prediction, which could be crucial for meeting the stringent performance and safety requirements of modern aviation systems.

0.2 Global Research Objective

Given challenges faced by traditional methods in complex aerodynamic modeling, a fundamental question emerges: Can AI methods transcend these constraints, thereby offering a new approach for more accurate and efficient aerodynamic and flight dynamics modeling? Despite the growing enthusiasm for AI, it is crucial to rigorously evaluate its actual potential in addressing the specific challenges of aerodynamics. Thus, the main objective of this research is to explore the effectiveness of AI-based methodologies, specifically ANN and SVR, in accurately modeling aircraft aerodynamic and flight dynamics models, while respecting FAA standards (FAA, 1991; ICAO, 2016). This exploration aims to determine how these technologies can potentially surpass conventional approaches by providing more accurate and robust modeling of lateral and longitudinal aerodynamics and flight dynamics under various operational conditions, including complex phenomena like stall.

More specifically, this objective involves the development of innovative methodologies to accurately model and predict lateral and longitudinal aerodynamic coefficients under various operating conditions, including stall, for specific aircraft models such as the Bombardier CRJ-700 commercial aircraft and the Cessna Citation X business aircraft. While AI typically requires substantial experimental data to train effective models, the use of highest-fidelity (level D) simulators such as the Virtual Research and Engineering Flight Simulator (VRESIM) for the CRJ-700 and the Research Aircraft Flight Simulator (RAFS) for the Cessna Citation X, becomes invaluable. They provide a wealth of accurate and simulated flight data for model training, validation, and optimization. Level D is the highest level given to the flight dynamics of flight simulators by the FAA. Additionally, the research explores both longitudinal and lateral aerodynamic aircraft modeling concepts to gain a comprehensive understanding of aircraft flight dynamics.

Finally, this research involves a comprehensive comparative analysis and optimization of AI techniques, including MLP, RNN, and SVR. The integration of these optimized AI models into a simulation platform that replicates flight scenarios and satisfies the FAA tolerance criteria (FAA, 1991; ICAO, 2016) represents a significant advancement toward real-time modeling and prediction in aerodynamics.

0.3 Thesis Organisation

The thesis comprises six chapters, each of these chapters addresses distinct aspects of the use of AI methodologies in aerodynamic modeling. The organizational structure ensures a coherent and comprehensive presentation of the research.

CHAPITRE 1 describes the state of art, providing an overview of relevant literature on aerodynamic modeling and AI methodologies. It establishes the current state of the field, identifies the research gaps, and outlines the specific objectives, approach, and contributions of this study.

CHAPITRE 2 to CHAPITRE 5 form the core of the thesis, including three published scientific journal articles and one submitted scientific journal article. **CHAPITRE 2** explores the application of MLP and SVR to model the longitudinal aerodynamics of the Bombardier CRJ-700. The research presented in this chapter aims to demonstrate the effectiveness of these AI techniques in accurately modeling an aircraft's longitudinal flight dynamics.

Building upon **CHAPITRE 2**, **CHAPITRE 3** extends the research to include the prediction of the lateral aerodynamic coefficients of the CRJ-700 using MLP and SVR. This chapter expands the scope of aerodynamic modeling and provides further evidence that both MLP and SVR can accurately model full flight dynamics (i.e., longitudinal and lateral motions).

CHAPITRE 4 focuses specifically on the stall dynamics modeling of the Bombardier CRJ-700, using MLP and RNN for identification purposes. Stall, a critical and dangerous phenomenon where the aircraft loses lift, presents significant risks to flight safety, making its accurate prediction and understanding essential. This chapter explores the complexity of stall dynamics, showcasing the adaptability of AI techniques in modeling complex aerodynamic phenomena.

In **CHAPITRE 5**, attention is given to the Cessna Citation X by employing MLP and RNN to identify and validate its longitudinal aerodynamic coefficients under stall conditions. The study presented in this chapter is mainly based on the method previously developed for the stall dynamics of the CRJ-700, demonstrating the versatility of AI methodologies for different aircraft models.

Finally, we consolidate the research findings from the 4 journal articles into a general discussion and conclusion. Connections are established between the articles, providing a conclusive understanding of the research objectives, followed by recommendations for future research.

CHAPITRE 1

STATE-OF-THE-ART AND RESEARCH CONTRIBUTION

1.1 Introduction to Aerodynamic Modeling

Aerodynamic modeling plays a crucial role in the field of aviation, having been through a significant evolution over time. Its origins can be traced back to the early aerodynamic principles proposed by pioneers such as Aristotle and Archimedes (Anderson, 1997). The 20th century saw a major shift in aerodynamics, evolving from a basic applied science to an academic field that deeply impacted aircraft design (Kármán, 2004). This period saw the emergence of structured methods and innovative strategies leading to significant advancements in aerodynamics, materials, and propulsion, contributing to substantial enhancements in aircraft performance (Anderson, 2006).

In modern aviation, the importance of aerodynamic modeling is highlighted by its direct impact on aircraft performance, environmental sustainability, and cost-effectiveness of operations. The effort to design aircraft for optimal performance while reducing its environmental impact is a key responsibility of aerodynamic designers (Anderson, 2000).

The quest for precise aerodynamic modeling in aviation is a challenging task, filled with complexities that arise from the inherent nature of aerodynamic phenomena (Hiliuta & Botez, 2007). The objective is to accurately represent aerodynamic forces and moments under a wide array of flight conditions. One of the main challenges comes from the complex and nonlinear behavior of aerodynamic phenomena, especially under extreme or unstable flight conditions, such as dynamic stall. These conditions often result in complex flow structures and interactions that are difficult to model accurately using traditional methods (Chernyshev, Lyapunov & Wolkov, 2019).

Furthermore, the complex nature of aerodynamic phenomena significantly increases the dimensionality of the related data, making modeling a difficult task. The aerodynamic properties of an aircraft are significantly impacted by many variables, such as the aerodynamic shape of the wing, the aircraft speed, the angle of attack, air density, surface roughness, and the aircraft configuration, among others. Each of these variables adds dimensions to the data, which, in turn, amplifies the challenges associated with accurate modeling (Xie, Huang, Song, Fu & Lu, 2022).

1.2 Aerodynamic Stall

The aerodynamic stall is a phenomenon that occurs when an aircraft wing exceeds a “critical angle of attack”. This angle varies depending on factors such as the wing airfoil shape, wing planform, and aspect ratio. In subsonic conditions, the critical angle of attack (known as stall angle of attack) typically ranges from 8 to 20 degrees (Clancy, 1975).

Under normal conditions, lift varies linearly with the angle of attack (for fixed Mach and Reynold numbers) for values below the critical (stall) angle. However, at high angles of attack beyond the critical angle, the flow over the airfoil separates, resulting in a sudden drop in lift due to an adverse pressure gradient on the airfoil surface. This phenomenon is called “static stall”. It is important to note that thick airfoils tend to have higher critical stall angles of attack compared to thinner airfoils with the same camber (Anderson, 2010). Additionally, symmetric airfoils generally have lower critical angles of attack compared to cambered airfoils, but they perform better in inverted flight.

At high angles of attack, when an aircraft airfoil is subjected to unsteady (time-varying) motions, it experiences more complex aerodynamic phenomena than static stall, which is called “dynamic stall”. Unlike static stall, which occurs gradually as the angle of attack increases, dynamic stall occurs when the angle of attack undergoes rapid variations. The wing can momentarily exceed the critical angle of attack and still generate significant lift, but this lift is not sustained and decreases rapidly. Dynamic stall is characterized by various complex aerodynamic phenomena, including shear layer instability, laminar-to-turbulent transition, and vortex formation.

Although significant progress has been made in recent years, dynamic stall remains a challenging area of research with various applications in the field of aeronautics. Researchers continue to explore and study dynamic stall to improve our understanding of its underlying mechanisms and its impact on aircraft performance and safety. Various semi-empirical methods developed by ONERA and MIT for the prediction of dynamic stall were studied and compared by Botez (1989).

1.3 Traditional Aerodynamic Modeling Techniques

CFD is a crucial tool for simulating fluid flows around an aircraft. This method uses Navier-Stokes equations to numerically model the interactions between fluid and aircraft surfaces. For example, Koreanschi et al. (2017) developed a global aerodynamic model using CFD, showcasing its ability to accurately predict aerodynamic coefficients under various conditions. Their research included a numerical simulation of a morphing wing-tip demonstrator aerodynamic performance, highlighting the potential of CFD in capturing aerodynamic behaviors. In a subsequent study by Gabor, Koreanschi & Botez (2012), CFD was used to simulate the aerodynamic characteristics of a morphing wing, equipped with a flexible upper surface and a controllable actuated aileron. The technology demonstrator in their study, resembling a real aircraft wing tip section, was developed through a complex and multidisciplinary design process. The results underscored the effectiveness of CFD in simulating intricate aerodynamic features, further emphasizing its importance in aerodynamic modeling.

CFD has the capacity to offer detailed insights into flow characteristics around and within various aerodynamic bodies. It facilitates the in-depth visualization and analysis of flow patterns, pressure distributions, and aerodynamic forces and moments, all of which are crucial for effective aerodynamic design and analysis (Mani & Dorgan, 2023). However, despite these benefits, CFD also faces several challenges. The most prominent issue is its computational time cost and substantial demands on computational resources, that require fine spatial resolutions (Drikakis, Kwak & Kiris, 2016). Additionally, accurately representing complex aerodynamic behaviors such as turbulence and dynamic stall is a significant challenge for CFD, which often requires advanced turbulence models and fine mesh resolutions, further increasing the computational requirements. Furthermore, the inherent assumptions and approximations in Navier-Stokes equations, especially in turbulence modeling, may lead to significant differences between CFD predictions and actual aerodynamic model results (Rizzi, 2023). Consequently, CFD models always need extensive validation against wind tunnel or flight test data to ensure reliability.

The panel method is another type of aerodynamic modeling that focuses on dividing the aircraft surface into smaller panels. Each panel contributes uniquely to the overall aerodynamic characteristics. This technique is known for its computational efficiency, being less demanding compared to CFD, making it an effective approach for predicting potential flows around the aircraft, particularly during the initial design phases (Katz & Plotkin, 2001). A specific application of the panel method can be seen in the aerodynamic analysis of the UAS-S4 Éhecatl airfoil, where a two-dimensional linear panel method was used. This method was successfully combined with an incompressible boundary layer model and a transition estimation criterion, resulting in the optimization of an unmanned aerial system using a flexible skin morphing wing (Gabor, Simon, Koreanschi & Botez, 2016). However, a significant limitation of the panel method is its inability to accurately capture viscous effects and complex flow behavior such as flow separation.

Wind tunnel testing is an experimental technique that involves testing a scaled model of an aircraft or its components within a controlled airflow environment to gather aerodynamic forces and moments data through various measurement techniques. These techniques include Pressure measurements (Piziali, 1994; Tang & Dowell, 1995), Laser Velocimetry (Braza et al., 2003), Laser Sheet Visualization (Wernert, Geissler, Raffel & Kompenhans, 1996) and Time-Resolved Particle Image Velocimetry (Hansman & Craig, 1987; Mulleners, Le Pape, Heine & Raffel, 2012). Koreanschi, Oliviu & Botez (2015) successfully carried out experimental validation of an ATR 42 optimized wing geometry using Price Païdoussis subsonic wind tunnel. Similarly, a novel control methodology for a morphing wing demonstrator was developed and tested by Tchatchueng Kammegne et al. (2018) in a subsonic wind tunnel, showing the contribution of wind tunnel testing in improving aircraft aerodynamic performance by using adaptive wing geometries. The airflow characterization of the wing was achieved using pressure sensors embedded in the flexible part of the wing upper surface.

Wind tunnel testing provides a controlled environment that conducts to the collection of repeatable and accurate aerodynamic data, playing an important role in verifying and validating theoretical and computational models prior to flight testing. Furthermore, wind tunnel testing enables the examination of flow patterns across a range of conditions that may be challenging to replicate in real-world settings. However, it can be expensive and time-consuming, demanding meticulous preparation and calibration. Furthermore, scale effects may introduce inaccuracies as the behavior of scaled models may not perfectly represent that of full-sized aircraft.

Semi-empirical methods play a critical role in aerodynamic modeling by integrating theoretical calculations with experimental results to predict aerodynamics. These methods have been considered in initial aircraft design and remain indispensable for preliminary assessments in today design practices. One widely adopted semi-empirical approach is the DATCOM, which facilitates the initial design and analysis of missile aerodynamics and performance (Rosema, Doyle & Blake, 2014). Tondji & Botez (2017) have used DATCOM for semi-empirical estimations of mass and center of gravity of the unmanned aerial system UAS-S4 by Hydra Technologies. Furthermore, The FDerivatives code, developed within our LARCASE laboratory, builds upon the methodology of the DATCOM code while incorporating advancements in aerodynamics theories. This code and its methodology have been successfully applied to various aircraft, including the UAS-S4 and UAS-S45 of Hydra Technologies (Kuitche & Botez, 2019), the military X-31 aircraft (Anton, Botez & Popescu, 2011), and the business aircraft Hawker 800 XP (Anton, Botez & Popescu, 2010).

Semi-empirical methods offer a balanced approach by leveraging the advantages of empirical data to enhance or correct theoretical models, resulting in more reliable predictions than either approach alone. These methods prove to be particularly advantageous in the early stages of design, significantly reducing the time and costs associated with extensive experimental testing. However, the reliance on empirical data introduces certain limitations. The accuracy of semi-empirical methods is inherently linked to the quality and relevance of the empirical data they rely on, which may not always be readily accessible or may be specific to particular conditions. Moreover, these methods may have limited flexibility in accurately predicting aerodynamic behavior beyond the range of available empirical data, which necessitates cautious extrapolation and may restrict their applicability to novel design concepts.

While traditional techniques have made notable contributions, the aviation industry is constantly seeking more precise and efficient modeling methods. The limitations inherent in these traditional approaches, such as the inability to accurately predict complex flow phenomena and the computational costs associated with certain simulations, highlight the need for innovative solutions. Consequently, there is a growing interest in exploring the potential of AI methods as valuable tools to enhance aerodynamic modeling.

1.4 Application of AI in Aerodynamic Modeling

The application of AI in the field of aerodynamics represents a significant advancement in addressing the limitations of traditional modeling techniques. With the rapid progress in computational power and algorithm development, AI has become an important and powerful tool for understanding the complex flow patterns and pressure distributions that characterize an aircraft aerodynamics. Notably, data-driven models in CFD have emerged, which establish relationships between inputs and outputs based on experimental data rather than relying solely on explicit physical mechanisms. This AI methodology signifies a fundamental change from traditional empirical methods (Wang & Wang, 2021).

Machine Learning (ML) algorithms, such as NN and SVR, leverage large datasets to “learn” the underlying physics of aerodynamic flows. This departure from empirical or semi-empirical methods allows these ML algorithms to discover relationships through pattern recognition and data inference, effectively handling the non-linearity and complexity inherent in aerodynamic systems (Zan, Han, Xu, Liu & Wang, 2022).

The integration of AI into aerodynamic modeling aligns with the industry's shift towards adaptive and resilient design approaches. AI's capacity to learn from real-time data and adapt to new scenarios presents opportunities for dynamic optimization of aircraft performance during flight, a paradigm shift from static models derived from wind tunnel tests and CFD simulations (Sabater, Stürmer & Bekemeyer, 2022; Zou & Sun, 2021). Nevertheless, applying AI methods to the well-known field of aerodynamics presents many challenges. Ensuring the reliability and validity of AI-derived models requires rigorous training, validation, and testing of experimental aerodynamic phenomena data. The "black box" nature of many AI algorithms needs transparency and interpretability to gain trust and acceptance within the aerospace community.

1.5 Artificial Neural Network in Aerodynamic Modeling

The field of ML provides a diverse set of tools for addressing the complexities of aerodynamics, while ANN are well known for their abilities to identify patterns in data that may be overlooked by traditional methods. ANN, designed to mimic the structure of the human brain, can effectively process and interpret large volumes of data, enabling them to make predictions based on their received inputs.

Among various types of ANN, MLPs which are feedforward artificial neural networks, have played a crucial role in modeling static and dynamic relationships using aerodynamic data. By employing multiple layers of interconnected neurons that sequentially process inputs, MLPs can approximate complex functions and capture both transitional and steady state of aerodynamic phenomena. These types of networks have been successful in estimating aerodynamic loads during gust conditions even in the presence of noisy and sparsely distributed surface pressure data (Chen et al., 2023). This capability of MLPs to extract meaningful patterns from such intricate datasets highlights their robustness and adaptability in dynamic and uncertain flight dynamics environments. The work carried out by Ghazi, Bosne, Sammartano & Botez (2017) serves as a precursor to our own research as their study presents a methodology for identifying static stall characteristics in the Cessna Citation X using MLPs. A Cessna Citation X RAFS developed by CAE Inc., serves as an excellent platform for conducting flight tests, and generating a comprehensive aerodynamic coefficients database. Building upon their achievements, our research presented in CHAPITRE 5 aims to advance aerodynamic modeling techniques even further, exploring their full potential on dynamic stall, a more complex phenomenon characterized by its transience and non-linearity.

RNN is another type of ANN that are designed to analyze sequential data, making them well-suited for capturing temporal relationships and dependencies (Suresh, Omkar, Mani & Guru Prakash, 2003). This characteristic gives high advantages in the field of aerodynamics, where flow properties often exhibit pronounced dependencies on the sequence of events, as seen in transient phenomena, such as dynamic stall. RNNs are excellent in capturing temporal dynamics by utilizing their internal memory to process data sequences. They learn and retain the sequential context within the flow, enabling a comprehensive understanding of underlying aerodynamic behavior. Notable advancements have been made by Mahajan, Kaur, Singh & Banerjee (2021) in the estimation of aerodynamic derivatives for an aircraft through the utilization of RNNs. Simulated flight data was employed to train RNN model, enabling the prediction of desired outputs based on given inputs. This predictive capability holds great significance in the dynamic analysis of aircraft, particularly in the assessment of stability and control across various flight conditions.

In CHAPITRE 4 and CHAPITRE 5, a comprehensive comparative analysis between MLP and RNN has been performed on their longitudinal aerodynamic coefficients modeling in the dynamic stall for both the CRJ-700 and Cessna Citation X. Through meticulous examination, the respective capabilities and constraints of each architecture in capturing the stall dynamics are thoroughly investigated and discussed.

1.6 Support Vector Regression in Aerodynamic Modeling

SVR, a robust ML technique derived from Support Vector Machines (SVM), has found widespread applications in modeling complex nonlinear systems. SVR offers a distinct approach compared to ANN, both in terms of its operational framework and expected outcomes. ANN rely on interconnected nodes in multiple layers, SVR is based on statistical learning theory and structural risk minimization. Its objective is to identify a hyperplane that optimally fits the data with a key emphasis on maximizing the margin between the closest data points, that are known as support vectors (Drucker, Burges, Kaufman, Smola & Vapnik, 1996). This methodology not only enhances the prediction robustness in the presence of noisy data, but it also mitigates the risk of overfitting, which can be a concern for Neural Networks (Brereton & Lloyd, 2010) applications. SVR ability to generalize effectively from limited training data is particularly advantageous in aerodynamic modeling where experimental data may be scarce and costly to obtain.

A notable application of SVR in the field of aerodynamics is described in the research conducted by Bagherzadeh (2020). In this study, a non-parametric system identification approach based on SVR was employed to develop a nonlinear aeroelastic model for aircraft. This methodology was applied to estimate forces and moments coefficients for the F/A-18 aircraft, utilizing both simulated and real flight data. The results demonstrated that the SVM-based approach outperformed ANN in this specific system identification task.

Moreover, the SVR has demonstrated its effectiveness in modeling unsteady aerodynamics at high angles of attack, a critical aspect of modern aircraft design that significantly impacts stability and control aspects (Wang, Qian & He, 2015). The capability of SVR to handle the dynamic characteristics of aerodynamics under such demanding flight conditions makes it an invaluable tool for ensuring aircraft safety and optimal performance.

In the field of optimization, SVR has been used as a sequential approximation method in conjunction with a hybrid cross-validation strategy (Yang & Zhang, 2013). This approach offers a solution for predicting aerodynamic coefficients, enabling the approximation of objective functions and constraints in aerodynamic optimization problems characterized by substantial computational complexity. By using the SVR as a sequential approximation technique, computational time is reduced without compromising prediction accuracy, thereby streamlining the optimization process.

Another approach has been proposed, that integrated SVR with Wavelet Transform and Gray Wolf Optimization, yielding a synergistic effect on the prediction capability of SVR. This innovative combination optimizes SVR using sophisticated algorithms, resulting in more accurate and efficient aerodynamic models (Zhang et al., 2023). The proposed method enhanced the overall performance of SVR, improving both the accuracy and efficiency of aerodynamic predictions.

In CHAPITRE 2 and CHAPITRE 3 of this thesis, a comprehensive comparative analysis is carried out, examining the performance of SVR in comparison to the previously discussed MLP models, on their applications in modeling longitudinal and lateral flight dynamics of CRJ-700.

In addition to the ANN and SVR discussed above, the application of fuzzy logic presents another promising avenue for enhancing the accuracy of aerodynamic modeling. A notable example is the work by Hiliuta, Botez & Brenner (2005), who explored the approximation of unsteady aerodynamic forces using Fuzzy techniques. Their study focuses on the F/A-18 aircraft, calculating unsteady aerodynamic forces in the frequency domain using sophisticated methods tailored for both subsonic and supersonic regimes. They demonstrate that by employing a combination of fuzzy clustering and shape-preserving techniques, it is possible to achieve a highly accurate approximation of these forces. This approach is particularly beneficial when the data are calculated for a range of unevenly spaced reduced frequencies, a common challenge in aerodynamic analysis. Their methodology not only complements the existing AI techniques discussed in this review but also underscores the versatility and potential of fuzzy logic in handling complex aerodynamic problems.

1.7 Research Objectives, Approach, and Contributions

The main objective of this thesis is to use AI techniques, especially ANN and SVR, to enhance the accuracy and predictive capability of flight dynamics and aerodynamics modeling, with a specific focus on two distinct aircraft: the Bombardier CRJ-700 and the Cessna Citation X. The Bombardier CRJ-700 is a regional jet that can accommodate 63 to 78 passengers, offering a maximum speed of Mach 0.85 and a flight range of up to 4,660 km. On the other hand, the Cessna Citation X is a high-speed business jet aircraft capable of carrying up to 12 passengers, flying at a maximum speed of Mach 0.92, and covering a range of 5,956 km. The use of these two aircraft types enables a comparative study of ML methods results. This comparative analysis serves to validate the universality of the developed models and expand their applicability within the aviation sector.

Conducted within the Laboratory of Applied Research in Active Controls, Avionics, and AeroServoElasticity (LARCASE), this PhD thesis builds upon the ongoing efforts of our laboratory team to incorporate AI into Aeronautical Engineering research. The selection of the Bombardier CRJ-700 and Cessna Citation X was based on the availability of high-fidelity level D simulators at the LARCASE, providing an ideal platform for the application and testing of our methodologies. According to the Federal Aviation Administration (FAA), level D is the highest qualification level given for dynamics and propulsion flight simulator models.

The main objective of this research is divided into three sub-objectives, presented in the following sections.

1.7.1 Longitudinal Fight Dynamics Modeling

The first contribution is to model the longitudinal flight dynamics of the CRJ-700 aircraft using a comprehensive AI-centric approach, incorporating advanced ML techniques and rigorous validation processes. The aim was to identify aerodynamic coefficients and predict short-period and phugoid dynamics of the CRJ-700 jet using two types of ML models: MLP and SVR.

Realistic flight test data was collected from the Level D CRJ-700 VRESIM, forming a solid database for the analysis. This dataset served as input for the training of MLP and SVR models. The Bayesian Optimization (BO) was applied to enable efficient parameter selection, and to enhance their prediction accuracy. In order to ensure the robustness of the models and avoid overfitting, cross-validation techniques were used.

The validation of the models was conducted with meticulous attention to detail through a comparison between the predicted longitudinal aerodynamic coefficients and their corresponding values from the VRESIM under the same flight conditions. In addition, the validation process ensured that the longitudinal flight dynamics parameters calculated from model predictions satisfied to the validation criteria established by the FAA (FAA, 1991; ICAO, 2016). This respect for FAA standards is crucial as it guarantees that the models meet the rigorous requirements necessary for their application in the aviation context.

This first study introduced an innovative approach that is different from traditional aerodynamic modeling techniques by leveraging the capabilities of MLP and SVR for system identification. In contrast to conventional methods such as CFD, semi-empirical methods or wind tunnel tests, the use of MLP and SVR offers a dynamic and adaptable framework to capture and represent the non-linear relationships existing within the aerodynamic coefficients and flight parameters, for a wide range of flight conditions, within a single prediction model. Furthermore, robust validation techniques, such as cross-validation, ensured the reliability and generalizability of these models. In addition, in contrast to time-consuming approaches such as CFD, AI models offer significantly faster predictions of aerodynamic coefficients, taking only fractions of a second. This high speed in prediction capability represents a substantial improvement, enabling the integration of accurate aerodynamic models into larger closed-loop control systems that required rapid response times. Consequently, the enhanced efficiency in aerodynamic modeling achieved through ANN and SVR not only provides accurate predictions, but it also supports real-time applications. This capability is particularly crucial in modern aerospace control and simulation systems, where real-time responsiveness is essential.

The results obtained in this study strongly support the main objective of the thesis, demonstrating the capability of accurately modeling longitudinal flight dynamics while respecting to FAA standards. This achievement not only reinforces the primary goal of the thesis, but also paves the way for future investigations into modeling lateral dynamics and more complex phenomena, such as stall behavior in the rest of this thesis. The results obtained in this study led to the publication of a first article (included in CHAPITRE 2, without modification):

Article 1: Tondji, Y., Ghazi, G., & Botez, R. M. (2024). Neural Networks and Support Vector Regression for the CRJ-700 Longitudinal Dynamics Modeling. AIAA *Journal of Aerospace Information Systems,* 21 (3), 263-278. DOI: https://doi.org/10.2514/1.I011332

This article was co-authored with both my PhD advisors: Dr. Ruxandra Mihaela Botez who supervised the research and provided guidance through regular meetings; Dr. Georges Ghazi, also as co-author, who contributed with his technical expertise, offering essential assistance throughout the research process.

1.7.2 Lateral Flight Dynamics Modeling

The second contribution of this research focuses on the modeling of the lateral dynamics of the Bombardier CRJ-700 aircraft, using an AI-centric methodology similar to that of the first sub-objective. The primary goal was to predict the aircraft lateral aerodynamic coefficients, which play a crucial role in ensuring aircraft maneuverability and stability during flight. To accomplish this goal, experimental flight parameters of the lateral flight dynamics were collected using the CRJ-700 VRESIM. This data was then used to train the MLP and SVR models. In order to optimize the performance of both models, Bayesian Optimization technique was employed to fine-tune the model hyperparameters. To validate the accuracy of both models, a set of predicted lateral flight dynamics parameters was compared to experimental data obtained from the VRESIM, by means of FAA tolerance criteria (FAA, 1991; ICAO, 2016), ensuring that the models met the required standards for aviation applications.

This study introduces an innovative approach by integrating ML techniques into the modeling of lateral flight dynamics. Traditionally, aerodynamic studies have focused primarily on longitudinal flight dynamics, while lateral dynamics has received relatively less attention, probably due to its complexity. This novel application of ML in the domain of lateral flight dynamics brings several advantages, as those discussed in the first sub-objective, such as the ability to effectively process complex and non-linear data, which is a characteristic of lateral flight dynamics behavior, and the rapid prediction of aerodynamic coefficients, which are crucial for real-time control systems. Moreover, by extending modeling efforts to lateral flight dynamics, the study achieves a complete representation of aircraft dynamics (consisting of both longitudinal and lateral flight dynamics models) and consequently advances the PhD thesis' main objective. The results obtained from this study led to the writing of a second article (included in CHAPITRE 3, unmodified):

Article 2: Tondji, Y., Ghazi, G., & Botez, R. M. Predicting Lateral Dynamics of CRJ700 Using Multilayer Perceptron and Support Vector Regression. This article was submitted in the AIAA *Journal of Aerospace Information Systems* in November 2023.

This article was co-authored by Dr. Ruxandra Mihaela Botez, who supervised the research and provided guidance through regular meetings. Dr. Georges Ghazi, as co-supervisor, contributed with his technical expertise, offering essential assistance throughout the research process.

1.7.3 Dynamic Stall Modeling

The third contribution of the thesis focuses on the aerodynamic modeling of more complex dynamics than the two preceding sub-objectives, which is the stall. It outlines a novel methodology to model the variations of aerodynamic coefficients and predict aircraft longitudinal flight dynamics during stall maneuvers, including their hysteresis nonlinear cycle. The research was applied to two distinct aircraft, the Bombardier CRJ-700 and the Cessna Citation X. Level D flight simulators of both aircraft were used to gather flight data in both linear and non-linear stall phases. This methodology uses MLP and RNN models to correlate flight data with aerodynamic coefficients, supported by a novel approach for tuning ANN parameters. The validation of these models, respecting FAA standards (FAA, 1991; ICAO, 2016), demonstrated high accuracy in predicting the flight dynamics of the CRJ-700 and Cessna Citation X under stall conditions.

The originality of these studies lies in their pioneering methodology of using ANN to model aircraft flight dynamics under stall conditions, which is not extensively explored in the current state of the art. Few literatures existing on stall phenomena has predominantly focused on the static stall, which is characterized by a sudden loss of lift when the angle of attack exceeds its critical value. Our research introduces an innovative approach by dynamically modeling stall conditions through AI methods. This dynamic aspect considers the aircraft response over time, capturing transient behaviors and hysteresis effects. Traditionally, stall modeling has been achieved using empirical or physics-based methods, which often have limitations in accurately capturing the complex and nonlinear dynamics associated with stall maneuvers. The use of ANN introduces a novel and innovative approach to address these challenges.

The focus on studying stall dynamics is particularly important as it addresses a critical gap in aerodynamic modeling, especially in the context of general aviation safety. Loss of control incidents during stall conditions remains a significant concern in aviation, and developing accurate models to understand and predict these dynamics is crucial. This study contributes to the PhD thesis by extending the modeling to dynamic stall, which enables a comprehensive understanding of aircraft dynamics during critical flight phases.

This research is presented in two distinct journal articles, where each article specifically addresses a distinct type of aircraft. The third article focuses on the Bombardier CRJ-700 (included in CHAPITRE 4, unmodified), while the fourth article highlights the investigation on the Cessna Citation X (included in CHAPITRE 5, unmodified).

Article 3: Tondji, Y., Wade, M., Ghazi, G., & Botez, R. M. (2022). Identification of the Bombardier CRJ-700 Stall Dynamics Model Using Neural Networks. AIAA *Journal of Aerospace Information Systems,* 19 (12), 781-798.
DOI: https://doi.org/10.2514/1.I011104

Article 4: Tondji, Y., Wade, M., Ghazi, G., & Botez, R. M. (2022). Identification and Validation of the Cessna Citation X Longitudinal Aerodynamic Coefficients in Stall Conditions using Multi-Layer Perceptrons and Recurrent Neural Networks. *INCAS BULLETIN*, 14 (2), 103-119.
DOI:http://dx.doi.org/10.13111/2066-8201.2022.14.2.9

These articles were co-authored by Dr. Ruxandra Mihaela Botez, who supervised the research and provided guidance through regular meetings. Dr. Georges Ghazi, also as co-supervisor, contributed with his technical expertise, offering essential assistance throughout the research process. Mr. Wade Mouhamadou, Master’s Student, was also included as co-author as he contributed to the development and testing of the proposed methodology during his internships.

CHAPITRE 2

NEURAL NETWORKS AND SUPPORT VECTOR REGRESSION FOR THE CRJ-700 LONGITUDINAL DYNAMICS MODELING

Yvan Tondji [a], Georges Ghazi [b], Ruxandra Mihaela Botez [c]

[a,b,c] Department of System Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3

Paper published online in the *AIAA Journal of Aerospace Information Systems*, January 2024, pp. 1-16.
DOI: https://doi.org/10.2514/1.I011332

Résumé

Cet article présente une nouvelle méthodologie pour identifier les coefficients aérodynamiques et prédire les dynamiques de courte période et de phugoïde d'un aéronef en utilisant deux types de modèles d'apprentissage supervisé : les réseaux de neurones artificiels « Artificial Neural Network (ANN) » et la régression par vecteurs de support « Support Vector Regression (SVR) ». L'étude a été validée sur le jet régional CRJ-700. Des données de tests de vol simulés ont été collectées lors de diverses manœuvres effectuées sur un simulateur de vol virtuel pour la recherche CRJ-700 de niveau D (VRESIM) conçu par CAE et Bombardier. Le niveau D est la qualification la plus élevée pour les modèles de dynamique de vol et de propulsion donnée par la « Federal Aviation Administration (FAA) ». Les modèles ANN et SVR ont été entraînés en utilisant les données recueillies à partir du VRESIM pour développer des modèles multidimensionnels capables de prédire les coefficients aérodynamiques de l'appareil pour toutes les conditions dans l'enveloppe de vol, définies par l'altitude, la vitesse, le poids et la position du centre de gravité. Le choix des solveurs et l'optimisation des hyperparamètres sont détaillés pour les deux types de modèles. Ces modèles ont été validés en comparant les paramètres de vol prédits avec les données expérimentales obtenues à partir du VRESIM de niveau D CRJ-700 en considérant les mêmes entrées de pilote. Les résultats ont montré que les deux types de modèles (SVR et ANN) étaient capables de reproduire avec une excellente précision le comportement non linéaire des dynamiques de période courte et de phugoïde.

Abstract

This paper presents a new methodology to identify the aerodynamic coefficients, and to predict the short period and phugoid dynamics of an aircraft using two types of supervised learning models: Artificial Neural Network (ANN), and Support Vector Regression (SVR). The study was validated on the CRJ-700 regional jet. Simulated flight tests data was collected during various maneuvers performed on a Level-D CRJ-700 Virtual Research Simulator (VRESIM) designed by CAE and Bombardier. Level D is the highest qualification for flight dynamics and propulsion models given by the FAA. Both ANN and SVR models were trained using the data collected from the VRESIM to develop multidimensional models capable of predicting the aerodynamic coefficients of the aircraft for any conditions in the flight envelope, defined by altitude, speed, weight, and center of gravity position. The choice of solvers and the optimization of hyperparameters are detailed for both types of models. These models were validated by comparing predicted flight parameters with experimental data obtained from the CRJ-700 Level D VRESIM considering the same pilot inputs. The results showed that both types of models (SVR and ANN) were able to reproduce with excellent accuracy the nonlinear behavior of the short period and phugoid dynamics.

2.1 Introduction

Improving the safety, performance, and efficiency of aircraft has always been a major concern of airlines and the transportation industry. Novel technologies such as “morphing wings” (Koreanschi, Sugar Gabor & Botez, 2016a; Sugar Gabor, Koreanschi, Botez, Mamou & Mebarki, 2016), aimed at reducing aircraft fuel consumption, are currently being designed and contribute to the industrial objective of developing green aircraft. However, the implementation of these technologies requires the development of innovative aircraft control and command techniques (Abdolhosseini, Zhang & Rabbath, 2013; Botez, Kammegne &

Grigorie, 2015; Moncayo, Perhinschi, Wilburn, Wilburn & Karas, 2012; Shao, Chen & Zhang, 2019; Tchatchueng Kammegne et al., 2018; Wang & Zhang, 2018) that have interesting potentials but require an accurate and detailed modeling of the aircraft aerodynamics.

Furthermore, precise modeling of the longitudinal dynamics of aircraft can have significant implications for the aviation industry. It can provide a more accurate representation of system dynamics, enabling more effective feedback model of air transportation system change (Mozdzanowska, Weibel & Hansman, 2008). This can also simplify approaches to mitigate complexity-driven issues in commercial auto flight systems (Vakil & John Hansman, 2002), thereby enhancing system reliability and safety. Additionally, precise aircraft dynamic models can aid in the analysis of flight data (Li, Das, John Hansman, Palacios & Srivastava, 2015), enabling more accurate detection of abnormal operations and providing valuable insights into aircraft behavior under various conditions. Today, the most common techniques for determining the aerodynamic coefficients of an aircraft can be divided into four categories: Computational Fluid Dynamics methods (CFD) (Koreanschi et al., 2016; Tinoco, 1998), semi-empirical and empirical models (Bierbooms, 1992; Botez, 1989), experimental methods, and system identifications techniques (Ghazi et al., 2017; Tondji, Wade, Ghazi & Botez, 2022b).

Empirical codes are generally adapted and used in the context of preliminary design. They are helpful for the definition and optimization of a new concept or architecture. These codes are based on theoretical models and/or aerodynamic databases obtained from numerical simulation results and wind tunnel tests (Botez, 1989). They have the advantage of being applicable to a wide range of configurations and allow to obtain results quickly. However, the determination of aerodynamic properties from these codes depends on the quality of the database. In general, to obtain sufficiently accurate estimates, the aircraft configuration studied must be close to the one used to populate the database.

CFD methods are used to predict the three-dimensional structure of flows around a wing under a given flight configuration, by numerically solving the fundamental equations of fluid dynamics. The modeling of aerodynamic coefficients has seen significant advancements due to the contributions of CFD methods, solidifying their status as a crucial tool within the aerospace sector. Ekaterinaris & Platzer (1998) have provided a comprehensive overview of the application of CFD methods in the context of dynamic stall.

However, the accuracy of CFD method calculations is usually limited by the assumptions of the mathematical equations used by their algorithms, especially the linearization about a steady flight condition (defined in terms of altitude, Mach and incidence) using small disturbance theory (Nelson, 1998). Therefore, the determination of the coefficients should be performed in parametric form for various fixed flight configurations; such simulations to cover the entire flight envelope can be very costly in terms of computing time.

Experimental methods based on wind tunnel tests are often used to reproduce flight conditions over a wide range of the envelope of an aircraft. The aerodynamic coefficients are modeled based on measurement techniques such as wing pressure measurements (Piziali, 1994; Tang & Dowell, 1995), laser velocimetry (Braza et al., 2003), Laser Sheet Visualization (Wernert et al., 1996) or even Time-Resolved Particle Image Velocimetry (Hansman & Craig, 1987; Mulleners et al., 2012). These techniques have the advantage of being able to determine aerodynamic coefficients based on “real” fluid measurement. However, they are prohibitively expensive and time-consuming (Botez, 2018; Moir & Coton, 1995), depending on the scale of the model, and they may not properly account for aeroservoelasticity effects.

An alternative method for predicting aircraft aerodynamic coefficients of an aircraft is by using system identification techniques. System identification is the process of identifying, from a certain class of systems, a mathematical structure to which the system under test is equivalent, using observations of the inputs and outputs (Klein & Morelli, 2006). For example, Perhinschi et al. (2002) devised a versatile simulation instrument for the identification of aircraft parameters, employing techniques of system identification. This tool was subsequently utilized to establish new control strategies. The primary objective of these techniques was to construct a mathematical representation of a physical system based on data derived from measurements. Essentially, it involved understanding the system dynamics and transcribing them into mathematical formulas. In this scenario, the application of artificial intelligence can prove to be highly beneficial.

The most commonly used approaches for system identification are polynomial regression and spline functions (Hosder et al., 2001; Reisenthel & Bettencourt, 1999). However, using these methods to accurately determine the variation of aerodynamic coefficients from multidimensional data can be a very difficult task. Therefore, other approaches must be considered. Artificial Intelligence (AI) is a rapidly advancing technology that significantly impacts several industries and engineering processes (Fan, Dulikravich & Han, 2005; Ghazi et al., 2017; Perhinschi, Wayne, Clark & Lyons, 2007; Tondji et al., 2022b). AI includes several system identification techniques suitable for modeling nonlinear multivariable systems from data. Among them, Artificial Neural Networks (ANN) and Support Vector Regression (SVR) have demonstrated their ability to model complex nonlinear systems. For instance, Peyada & Ghosh (2009) used an ANN to estimate the aerodynamic coefficients of a HANSA-3 and an ATRAS aircraft using measured flight parameter data, such as speed and acceleration. These calculated values were then compared with results derived from the Least Square (LS) and Filter Error (FE) methods. Their ANN model gave better approximations than the LS and FE methods. Nevertheless, some inherent drawbacks, such as slow training convergence speed, usually make ANN difficult in their application to practical problems. Support Vector Regression (SVR) is a type of Support Vector Regression (SVM) used in system modeling when the problem to be solved is equivalent to a regression problem. SVM was proposed by Vapnik (1995) as a new ML tool based on statistical learning theory. Initially, they were developed for pattern recognition problems. They have recently been effectively expanded to nonlinear system modeling and nonlinear function approximation (Drucker et al., 1996). SVR algorithms can be used to design relatively complex non-linear models, as can ANN (Fan et al., 2005). In addition, they are mathematically simple to analyse because they correspond to a linear method in a high-dimensional feature space that is nonlinearly related to the input space.

This article discusses two distinct approaches to modeling an aircraft movement, specifically employing two forms of artificial intelligence technologies: ANN and SVR. These methods were tested on the Bombardier CRJ-700 regional jet using a high-quality flight simulator, the Virtual Research Equipment Simulator (VRESIM) for the Bombardier CRJ-700 (see Figure 2.1). The premise of our experiment was that the flight simulation quality was sufficient for our testing purposes and the data obtained could stand comparison with that gathered by Bombardier's flight test engineering team. This article is organised as follows: section 2.2 shows the methodology, including the simulated flight tests, the data processing to estimate aerodynamic coefficients from measurable flight parameters, and the SVR and ANN models training and hyperparameter optimization. section 2.3 presents the results obtained and their comparison with experimental data gathered from the CRJ-700 VRESIM. The article concludes with our deductions and future considerations.

Figure 2.1 Validating the proposed methodology with a Bombardier CRJ-700 Virtual Research Equipment Simulator

2.2 Methodology

2.2.1 Simulated Flight Test Procedure

The first step in system identification is to observe the behavior of the system of interest (by measuring its outputs) in response to external influences. For this purpose, a series of simulated flight tests were performed on the CRJ700 VERSIM under different flight conditions in order to analyze the variation of the longitudinal aerodynamic coefficients of the CRJ700 aircraft as a function of external inputs, such as flight conditions and pilot commands. It should be noted that the simulation data used in this study were generated under ideal conditions, excluding factors such as sensor noise or atmospheric turbulence. By adopting this approach, we can focus on understanding the fundamental aerodynamic behavior of the aircraft, in the absence of intricate effects of external perturbations.

The entire flight envelope of the aircraft was considered. The flight envelope refers to the regions in terms of altitude/speed or weight/center of gravity location where the aircraft remains stable. For the Bombardier CRJ-700, the center of gravity location varied within an interval between the ‘‘*Take-off, Landing & Approach lower limit*’’ (14 % of the chord) and the “*Take-off, Landing & Approach upper limit”* (35 % of the chord), while the aircraft weight ranged between the ‘‘*Minimum Flight Weight*’’ ($42{,}000\ lbs$) and the ‘‘*Maximum Landing Weight*’’ ($67{,}000\ lbs$). The indicated airspeed varied from $260\ kts$ to $330\ kts$, while the pressure altitude varied from $5000\ ft$ to $35{,}000\ ft$.

Figure 2.2 and Figure 2.3 show the 49 flight conditions considered in the weight/center of gravity flight envelope (Figure 2.2) and the 48 flight conditions considered in the altitude/speed flight envelope (Figure 2.3). The combination of these two flight envelopes thus led to a total of 2352 flight conditions.

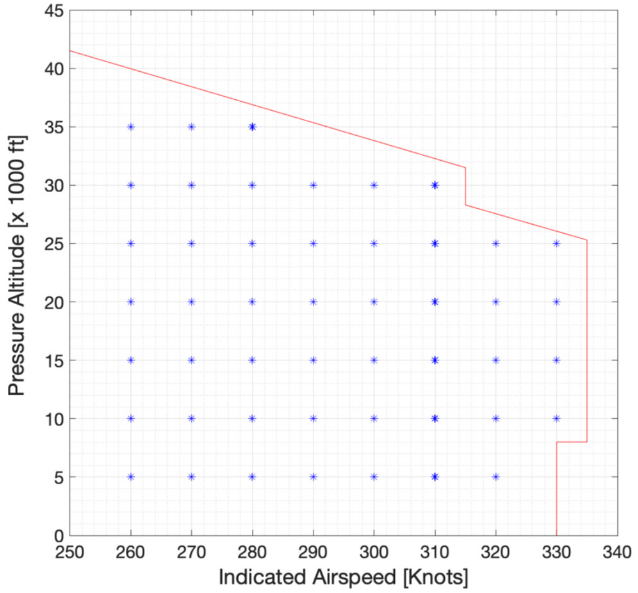


Figure 2.2 Pressure altitude versus IAS flight envelope of Bombardier CRJ-700 VRESIM

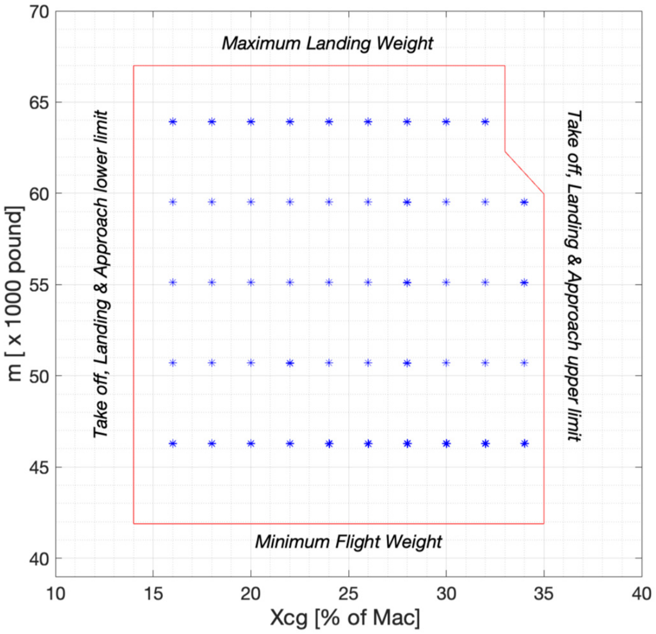
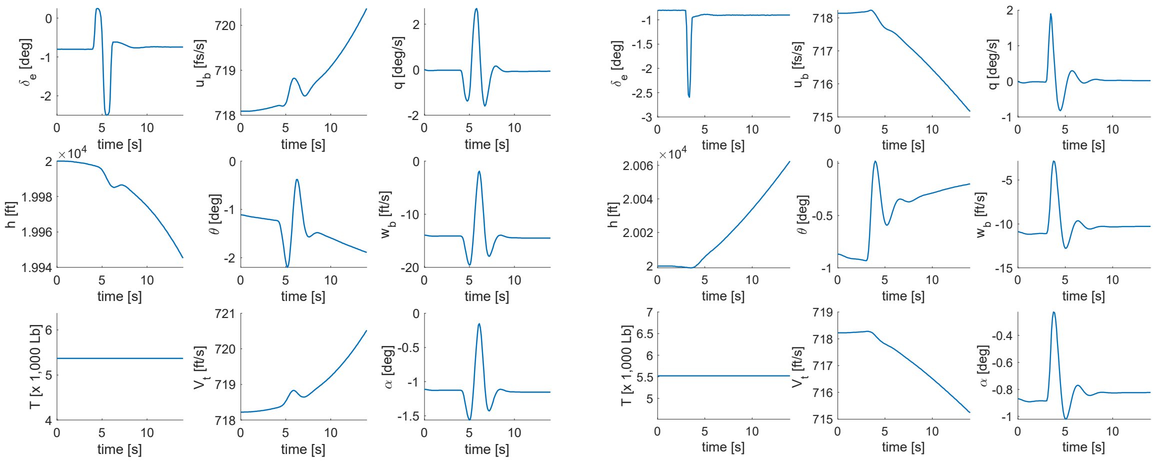


Figure 2.3 Weight versus center of gravity position flight envelope of the Bombardier CRJ-700 VRESIM

All simulated flight tests were performed automatically using pre-coded scripts. These simulated flight tests aimed to highlight the short period and phugoid dynamics of the aircraft. Thus, for each set of flight condition, the following procedure was applied. First, the aircraft was trimmed in cruise configuration at the selected flight condition expressed in terms of altitude, speed, weight, and center of gravity position. The aircraft was maintained in trim conditions for approximately 3 seconds, then an elevator input was applied while the thrust remained constant. Finally, the aircraft was left in free oscillations for 260 seconds to allow observation of the short period and phugoid modes.

Two types of elevator inputs were used to excite these two longitudinal modes: doublet inputs and pulse inputs. The doublet input is a suitable elevator excitation for observing the short period, while pulse inputs tend to excite both phugoids and a short period. Having the two types of motion was necessary so that the AI models could learn data information about both short period and phugoid dynamics. Fifty doublet inputs and fifty pulse inputs of different magnitudes and natural frequencies were generated from real pilot commands. For each simulated flight test, one type of input was randomly selected from the 100 available generated inputs.

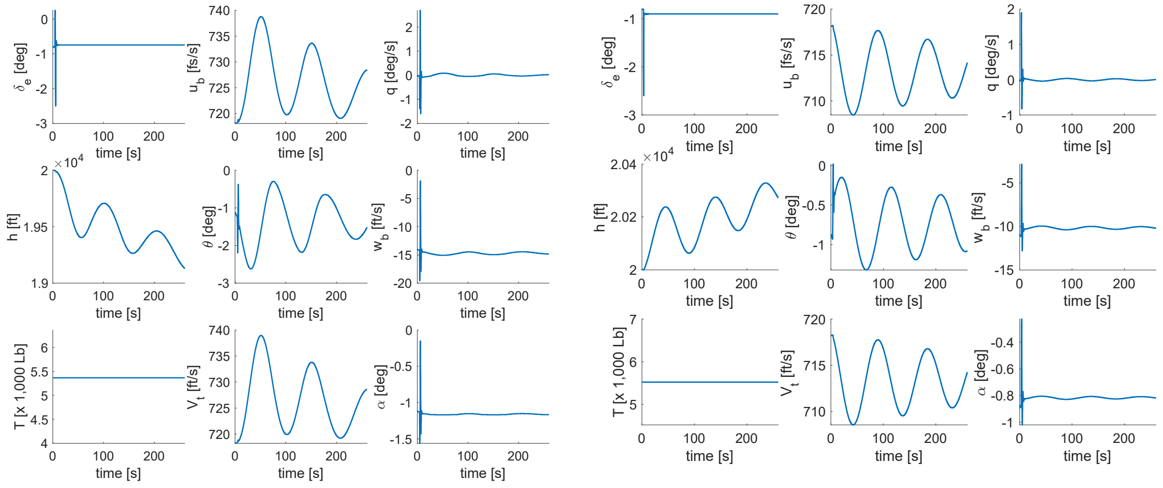
During the simulated flight tests, we recorded various critical flight parameters, including altitude, Mach number, true airspeed, angular velocities, accelerations, engine thrust, and control surface deflections, at a sampling rate of 30 Hz. Figure 2.4 and Figure 2.5 show examples of the data recorded from the VRESIM for two simulated flight tests (A and B). Figure 2.4 shows a 14-second simulation where the short-period dynamics can be observed, while Figure 2.5 shows a 260-second simulation that further highlights the phugoid dynamics.



Simulated flight test A: $m = 55,120\ lb \mid X_{cg} = 16\% \mid h = 20,000\ ft \mid IAS = 320\ kts$

Simulated flight test B: $m = 63,930\ lb \mid X_{cg} = 18\% \mid h = 20,000\ ft \mid IAS = 320\ kts$

Figure 2.4 Short period mode data from 14-sec tests A & B on CRJ-700 VERSIM



Simulated flight test A: $m = 55,120\ lb \mid X_{cg} = 16\% \mid h = 20,000\ ft \mid IAS = 320\ kts$

Simulated flight test B: $m = 63,930\ lb \mid X_{cg} = 18\% \mid h = 20,000\ ft \mid IAS = 320\ kts$

Figure 2.5 Phugoid data from 260-sec tests A & B on CRJ-700 VERSIM

The two simulated flight tests were carried out at an altitude $h = 20{,}000$ ft, and at an indicated air speed $IAS = 320\ kts$. For simulated flight test A, the aircraft weight was set at $m = 55{,}120$ lbs and the center of gravity at $x_{cg} = 16\%$, while for simulated flight test B, the aircraft weight was set at $m = 63{,}930$ lbs, and the center of gravity position at $x_{cg} = 18\%$.

In addition, the elevators were excited with a doublet input in simulated flight test A, and with a pulse input in simulated flight test B. The short-period motion is clearly visible on the pitch rate q, the vertical component of the airspeed w_b, and the angle of attack α, and is almost imperceptible on the longitudinal component of the airspeed u_b. This mode is characterized by a low-damped oscillation (about 2.5 seconds) appearing instantaneously when excitation begins at 3 seconds for simulated flight test B and at 4 seconds for simulated flight test A. In contrast, the phugoid is characterized by a change in altitude h, pitch attitude θ and longitudinal speed u_b at a nearly constant angle of attack α and pitch rate q. This mode is also characterized by a very low-damped, and low-frequency dynamics with a period of about 100 seconds.

2.2.2 Data Processing and Aerodynamic Coefficients Determination

After performing simulated flight tests, the next step was to estimate the longitudinal aerodynamic coefficients of the aircraft in the stability axes. These coefficients were mapped with their corresponding flight parameters and pilots' inputs to be used for supervised learning of the models. They are expressed in the stability axes $R_s\ (o, x_s, y_s, z_s)$. These orthogonal axes are derived from the body axis reference $R_b\ (o, x_b, y_b, z_b)$, which is defined so that the x_b-axis is aligned with the nose of the aircraft, the y_b-axis points to the right-wing side, and the x_b-z_b is a symmetrical plane. Starting from the body axis reference, the stability axes are obtained by rotating the body axes around the y_b-axis, so that the x_s-axis is aligned with the velocity vector. The three longitudinal aerodynamic coefficients in the stability axes system, i.e, CD_s, CL_s and Cm_s , were obtained based on the following equations (Ghazi et al., 2017) :

$$CL_s = CL_b \cos(\alpha) - CD_b \sin\ (\alpha) \tag{2.1}$$

$$CD_s = CD_b \cos(\alpha) + CL_b \sin(\alpha) \tag{2.2}$$

$$Cm_s = Cm_b - CD_b z_{cg} - CL_b x_{cg} \tag{2.3}$$

where $\{x_{cg}, z_{cg}\}$ are respectively the horizontal and vertical distances from the aircraft center of gravity to its aerodynamic center, and CL_b, CD_b and Cm_b are the lift, drag and pitching moment following equations (Ghazi et al., 2017):

$$CL_b = \frac{ma_z - T_z}{1/2\rho V_T^2 S_w} \tag{2.4}$$

$$CD_b = \frac{ma_x - T_x}{1/2\rho V_T^2 S_w} \tag{2.5}$$

$$Cm_b = \frac{I_{yy}\dot{q} - T_x z_{eng} - T_z x_{eng}}{1/2\rho V_T^2 S_w c_w} \tag{2.6}$$

where ρ is the air density, $\{T_x\ , T_z\}$ are the components of the engine thrust, I_{yy} is the aircraft moment of inertia about the lateral axis, S_w is the wing reference area, c_w is the wing mean aerodynamic chord, and a_x and a_z are the aircraft longitudinal and vertical accelerations, respectively.

2.2.3 Neural Network and Support Vector Regression models' implementation

Artificial Neural Networks (ANN) and Support Vector Machines (SVMs) are two different types of artificial intelligence models that have proven to be effective in assimilating historical data and generating approximate representations of a system without requiring a mathematical model or solving its dynamic equations (Fan et al., 2005; Ghazi et al., 2017; Tondji et al., 2022b). These models have the particularity of being parametric, which means that their performance depends significantly on the choice of their parameters.

This section explains how the optimization of the parameters for the ANN and SVM was done to develop the longitudinal aerodynamic coefficient models of the CRJ700.

2.2.3.1 Definition of Models' Inputs and Outputs

The first step in designing an AI model to solve a multidimensional regression problem is to define a data structure, where variables considered as inputs (x) of the model are mapped to target variables (o), also called output variables. The data structure is usually composed of n pair of (x_i, o_i), where n is the number of sampled data pairs in the structure.

In the same way as for the dataset, it is necessary to define a structure for the model. A neural network model is a Multi Input Multi Output model (MIMO). Therefore, this type of model can be directly configured to predict all three aerodynamic coefficients at the same time. Equation (2.7) defines a sampled output vector for the designed neural network model.

$$o_i = \left\{\widehat{CL_s}, \widehat{CD_s}, \widehat{Cm_s}\right\}_i \tag{2.7}$$

In contrast, an SVM is a Multi Input Single Output (MISO) model, which means that this type of model can only predict one aerodynamic coefficient at a time. Therefore, three different SVM models were designed, each model associated with one of the three output variables, i.e., $\left\{\widehat{CL_s}\right\}$, $\left\{\widehat{CD_s}\right\}$, $\left\{\widehat{Cm_s}\right\}$.

The inputs vector is composed evidently of flight parameters that can influence the outputs variations. Anderson (2010) performed a dimensional analysis on aerodynamic coefficients using Buckingham's π theorem. He showed that for a given aircraft weight and center of gravity position, the aerodynamic coefficients of an aircraft in steady flight are functions of the of following independent variables: angle of attack α, Mach number M, and Reynolds number R_e. However, since the Reynolds number is not directly measurable in flight, it can be approximated according to Equation (2.8).

$$R_e = \frac{\rho V_T c_w}{\mu} = \frac{2c_w}{\mu} \times \frac{Q}{V_T} \tag{2.8}$$

where V_T is the true airspeed, Q is the dynamic pressure, ρ is the air density, μ is the air dynamic viscosity and c_w is the mean aerodynamic chord length. In Equation (2.8), the term $\frac{2c_w}{\mu}$ remains constant across different flight cases because the aircraft chord length does not change. Our trial-and-error tests have revealed that the $\frac{Q}{V_T}$ ratio significantly influences the model accuracy and convergence rate. Furthermore, despite variations in the aircraft weight across different flight cases, our analysis through trial and error has led us to conclude that its impact on the prediction of aerodynamic coefficients can be deemed negligible. This could be explained by the fact that for business aircraft, weight variations are usually small.

Mccroskey (1981) demonstrated that the pitch rate q also greatly influences the variation of the aerodynamic coefficients. In addition, the elevator angle deflection δ_e and the horizontal stabilizer angle deflection δ_s were also considered as inputs, because they are used to control the longitudinal motion of the aircraft. Finally, since the model must be defined over the entire weight/center of gravity of the flight envelope, the position of the center of gravity must also be considered as an input because it significantly influences the aerodynamic pitching moment coefficient. Therefore, an input vector sample of the model is defined as follows:

$$x_i = \left\{\alpha, M, \frac{\mathrm{Q}}{V_T}, q, \delta_e, \delta_s, X_{cg}\right\}_i \tag{2.9}$$

2.2.3.2 Data management

This section presents a procedure for selecting suitable data for training AI models (ANN and SVM). This approach reduces the amount of data required for the optimal training of our models.

In this study, two types of AI models were implemented to model the longitudinal dynamics of an aircraft using data obtained from simulated flight tests conducted on a VRESIM Bombardier CRJ700. To minimize the number of simulated flight tests required, the model should be trained using the minimum amount of data when replicating the methodology with actual aircraft flight test data. Figure 2.6 shows the flight conditions on the weight/center of gravity position in the flight envelope used for training and testing the model. The training dataset was used to optimize the weight and bias of the AI models, while the test dataset was utilized to evaluate the model performance based on data that was not considered during training. This dataset is primarily used to adjust key parameters of the model, such as the learning function, activation function, and number of hidden layers for the ANN or solver algorithm, and the precision and penalty factor for the SVM. Data of the selected points circled in red will be further divided into training and testing sets using k-fold cross validation, highlighting the distinction between the two datasets.

It was assumed that a well-trained model should be able to interpolate the desired outputs between two flight conditions with different center of gravity positions (with the same mass) and between two masses if the center of gravity positions were close. Therefore, only 15 of the 49 configurations in terms of mass/center of gravity position were used for training and testing. These configurations were selected as shown in Figure 2.6.

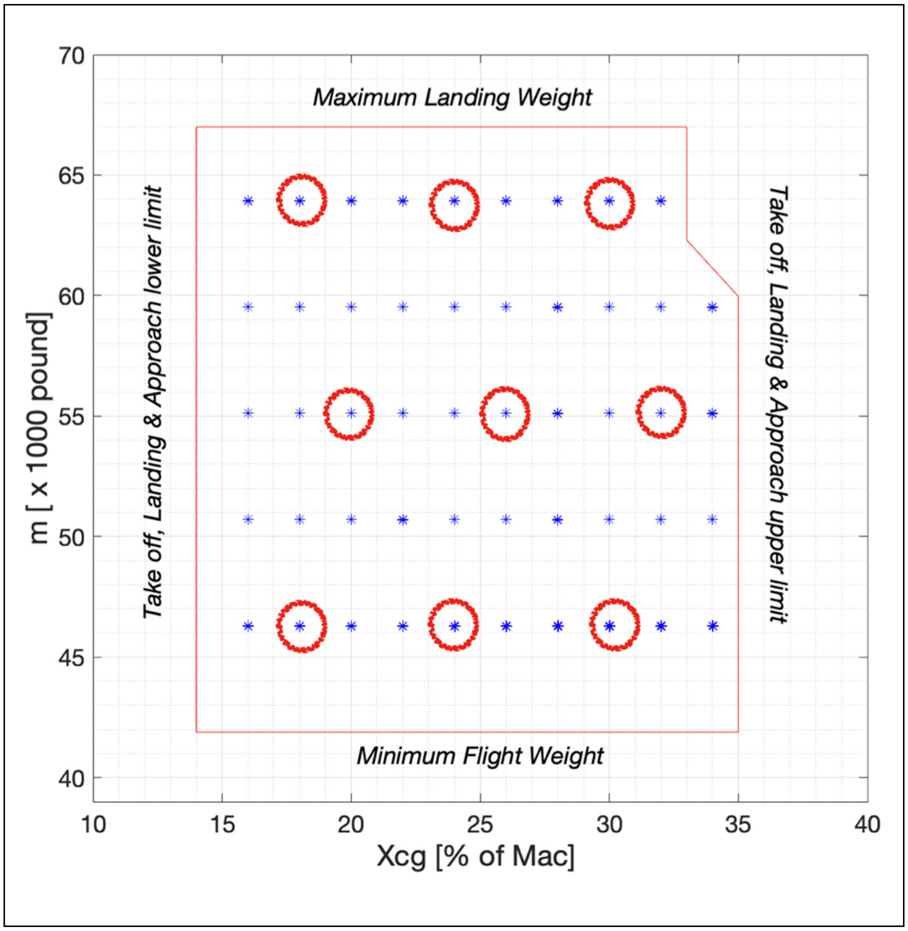


Figure 2.6 Weight and center of gravity conditions selected for models training and testing

As a reminder, for each point selected in Figure 2.6, there are 48 different flight conditions in terms of altitude and speed. However, only 24 of them were used for training and testing the model. These conditions were selected as shown in Figure 2.7 Therefore, a total of 360 of the 2352 flight cases available in the database were used to build (train and test) the AI models. In other words, only 15.3 % of the database was used to train the models.

The rest of the data (i.e., 84.7 %) was used for validation. The validation set was used to demonstrate the accuracy of the trained AI models in predicting the three aerodynamic coefficients based on new data that were not used in the training and optimization processes. In addition, only short period motion data (the first 14 seconds of each flight case) were used for training. It was expected that the AI models would be able to generalize their "knowledge" to accurately predict both short-period and phugoid dynamics, although they were only trained with data collected from short-period maneuvers.

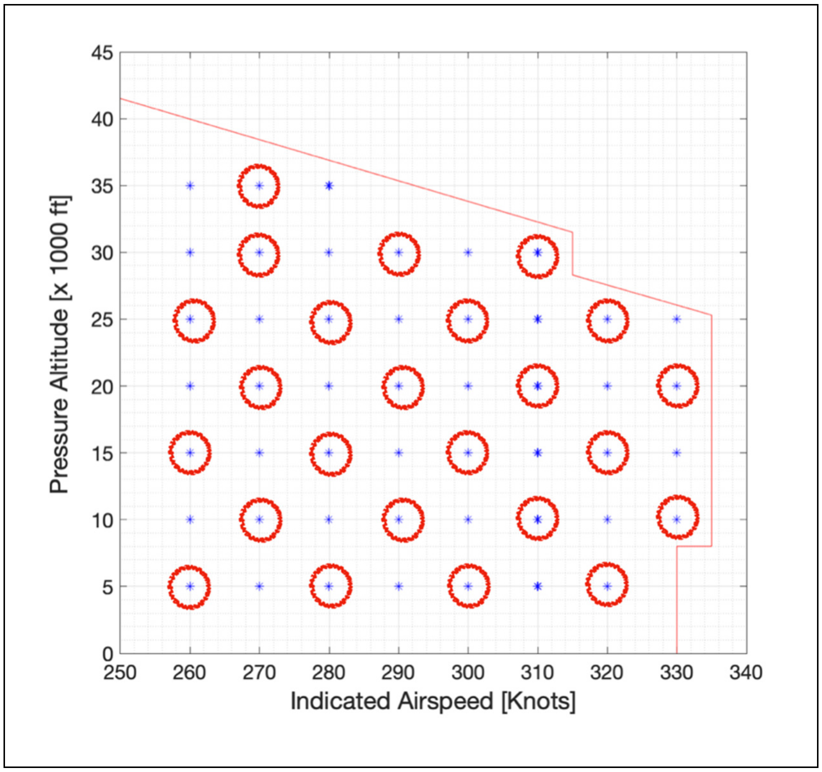


Figure 2.7 Flight conditions selected for training on pressure altitude versus IAS flight envelope the Bombardier CRJ-700 VRESIM

To split the data between the training set and the test set, the “k-fold” cross-validation method was used. Indeed, during the training phase, an algorithm was used to minimize a predefined cost function by varying the unknown parameters of the model in order to achieve the best possible performance. In the testing phase, the cost function was evaluated to confirm that the model performance during training was representative of its ability to generalize to new data. By doing it, we aim to mitigate the risk of overfitting (Yeom, Giacomelli, Fredrikson & Jha, 2018). It is important to note that while overfitting can occur when the model is trained on a dataset that does not capture the full range of conditions, other factors, such as model complexity and lack of regularization, can also lead to overfitting. In such cases, the model may perform well on the training data, but may fail to predict new (untrained) data. To address this issue, the k-fold method divides the 360 flight case data into k subsets, or folds, of approximately equal size. Each data point is assigned to a specific fold and remains there for the entirety of the process. The ANN or SVR models are trained k times, with a unique subset of the k folds serving as the testing dataset in each iteration while the remaining subsets $(k - 1)$ constitute the training dataset. This ensures that each data point is used once as test data and is utilized to train models $(k - 1)$ times. In practical ML problems, a value of k = 5 is commonly used, provided that each training or test fold is sufficiently large to represent the entire dataset (Garvin, Daniela, Trevor & Robert, 2013). As the database used in this study was sizable, this condition was met. The performance of the trained model was evaluated as the average of the performances from each round of training.

2.2.3.3 Artificial Neural Networks

Designing neural networks models for system identification applications involves several processes. The first step is to determine the most appropriate type for the problem of interest. Artificial Neural Networks (ANN) have been shown to be effective in dealing with regression and function approximation problems, where real-valued parameters, such as aerodynamic coefficients, are predicted from a given dataset (Haykin, 1998).

The primary building block of ANN is the artificial neuron, which is also referred to as a "node" or "perceptron". An artificial neuron has multiple connections from either a set of ANN inputs or from neurons in another hidden layer. Figure 2.8 provides a schematic representation of an artificial neuron.

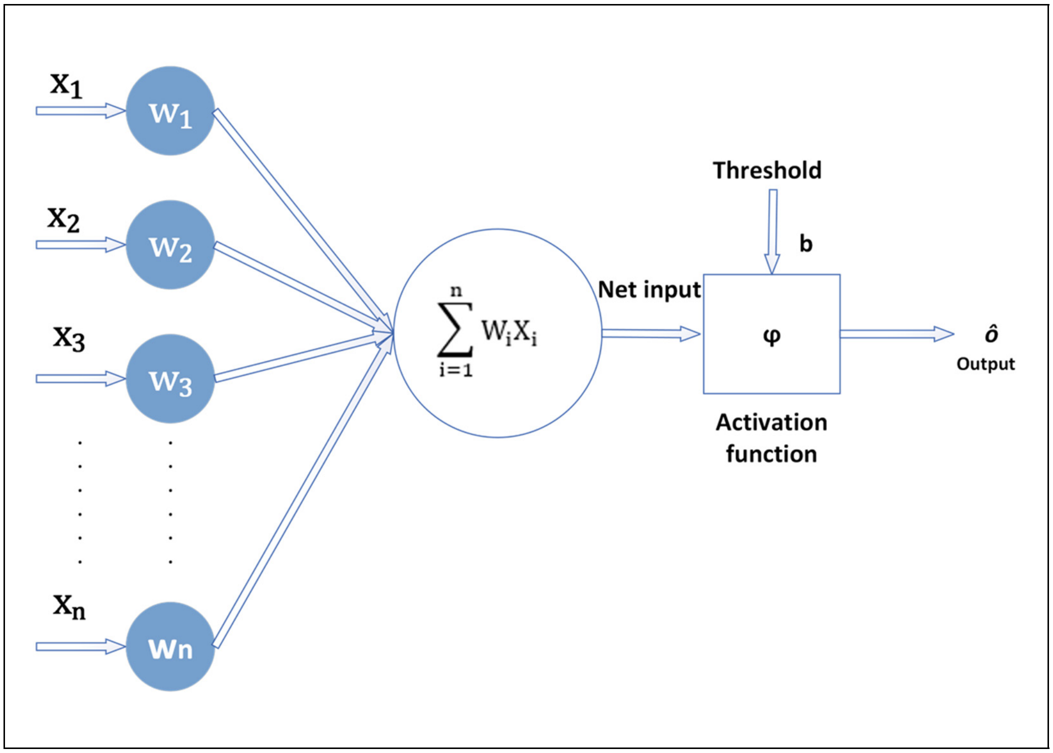


Figure 2.8 Graphical representation of an artificial neuron

A perceptron is a basic processing unit that computes an output from a given set of inputs, as seen in Figure 2.8 To compute the value of the neuron output $\hat{o}$, the neuron input signal $x = [x_1, x_2, x_3, \dots, x_n]$ is multiplied by its corresponding weights $w = [w_1, w_2, w_3, \dots, w_n]$, summed, and then fed to a "transfer" or "activation" function φ. This activation function decides whether the neuron should be activated or not. In the literature, several forms of activation functions exist, such as the linear function, the sigmoid function, and the rectified linear unit activation function (Linse & Stengel, 1993). The choice of the activation function for aerodynamic coefficient prediction is described later in this section. The mathematical expression for the output of a neuron is as follows:

$$\hat{o} = \varphi(x, w, b) = \varphi\left(\sum_{i=1}^{i=n} x_i w_i + b\right) \tag{2.10}$$

where b is a constant bias that defines the activation threshold of the neuron.

As shown in Figure 2.9, ANN consist of a collection of neurons that are interconnected and arranged in layers. The first layer, also called the “input layer”, is designed to receive signals from a set of inputs, while the last layer, also called the “output layer”, is determined by the number of outputs of the model. An arbitrary number of hidden levels exist between these two layers. The number of hidden layers and the number of neurons per layer are key parameters for ANN and influence their performance.

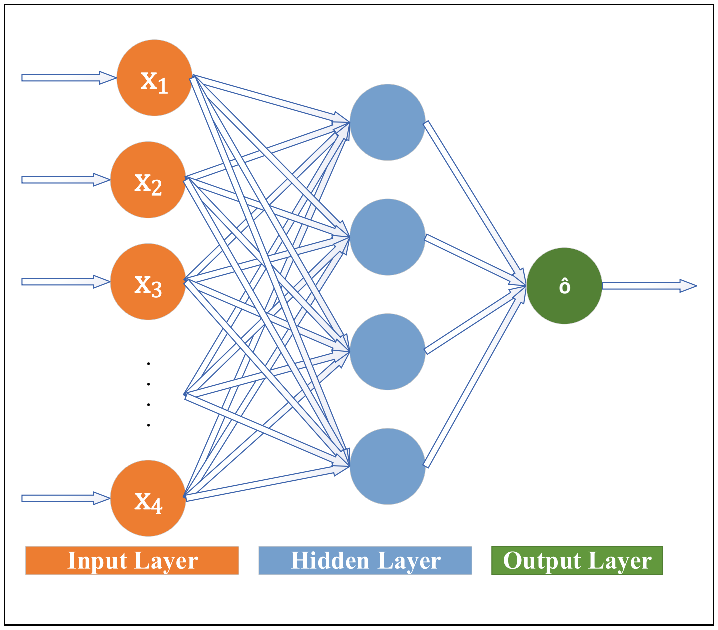


Figure 2.9 Illustration of an ANN neural network

The predicted output $\hat{o}$ of the ANN can be calculated using Equation (2.11) (Haykin, 1998):

$$\hat{o} = \varphi_m\left(\sum_{k=1}^{k=n_m} w_{m,k} \times \ldots \times \varphi_2\left[\sum_{i=1}^{i=n_2} w_{2,i} \times \varphi_1\left(\sum_{j=1}^{n_1} x_{1,j}w_{1,j} + b_{1,j}\right) + b_{2,i}\right] + b_{m,k}\right) \quad (2.11)$$

In Equation (2.11), $\hat{o}$ represents the ANN predicted output. The activation function of the final layer is φ_m, and n_m is the number of neurons in that layer. Similarly, φ_2 and φ_1 are the activation functions for the second and first hidden layers, respectively, while n_2 and n_1 represent the number of neurons in these layers. The weights $w_{m,k}$, $w_{2,i}$, $w_{1,j}$ and biases $b_{m,k}$, $b_{2,i}$, $b_{1,j}$ are the parameters to be learned during the training phase. The choice of the solver (i.e., the training function) and of the hyperparameters considerably influences the performance of a neural network. These hyperparameters are the "activation function" of the neurons, the number of hidden layers, and the number of neurons per hidden layer.

Choice of Solver (Training Function)

The training algorithm is the solver that carries out the ANN learning process. It defines a procedure for updating the weights and biases of a network with the goal of finding a combination that minimizes the cost function. Given an ANN with fixed hyperparameters, it is necessary to identify the solver that provides the highest performance.

MATLAB software features nine gradient-based local optimization techniques that can be used for training neural networks. These techniques are listed in Table 2.1.

Table 2.1 Training algorithms considered to train the network

Algorithm	**Description**
OSS	One-step secant backpropagation
CGP	Conjugate gradient backpropagation with Polak-Ribiére updates
CGB	Conjugate gradient backpropagation with Powell-Beale restarts
CGF	Conjugate gradient backpropagation with Fletcher-Reeves updates
BFG	BFGS[1] quasi-Newton backpropagation
RP	Resilient backpropagation
SCG	Scale conjugates gradient method
LM	Levenberg-Marquardt optimization.
BR	Bayesian regularization backpropagation

The network was trained with these nine algorithms, according to the 5-k folds cross validation procedure defined in section 2.2.3.2. For this first analysis, the activation function, and the structure of the ANN (i.e., the number of hidden layers and neurons) were fixed. The *logsig* activation function was arbitrarily chosen according to (Tondji, Wade, Ghazi & Botez, 2022a), where the authors conducted a similar study for modeling the aerodynamic coefficients of the Cessna Citation X in a stall condition. The size of two hidden layers and five neurons per hidden layer was also arbitrarily chosen. Each training algorithm was then used to determine the weights and biases that minimized the training error. The performance evaluation was done during the training and test phases. The training and test performances (i.e., the cost function) of the ANN model were calculated using the Mean Square Error (MSE). Thus, for a given set of n data points, the MSE of an ANN was calculated according to Equation (2.12):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[\hat{o}_i(w) - o_i]^2 \tag{2.12}$$

[1] Broyden-Fletcher-Goldfarb-Shanno

where o_i is the ith output vector associated with the ith input vector in the dataset, and $\hat{o}_i$ is the ith output vector predicted by the AI model.

In Figure 2.10, the training and test performance values for each training algorithm are presented in blue and orange, respectively. It is evident that the Bayesian Regularization (BR) and Levenberg-Marquardt (LM) algorithms resulted in the lowest (and comparable) MSE values. This result aligns with expectations, given the well-established proficiency of the BR and LM algorithms in managing nonlinear regression problems (MacKay, 1992). It can also be observed that the training and test performances obtained were very close for all the training functions. For example, the performance obtained on the training set using the LM algorithm was $MSE = 3.47 \times 10^{-4}$, while the performance obtained on the test set using the same algorithm was $MSE = 3.9 \times 10^{-4}$. This aspect allows to conclude that the model was well trained without overfitting. Moreover, as the two performances (training and test) are close, and as their variations are correlated, it is possible to consider only one of them for the rest of the comparative analyses.

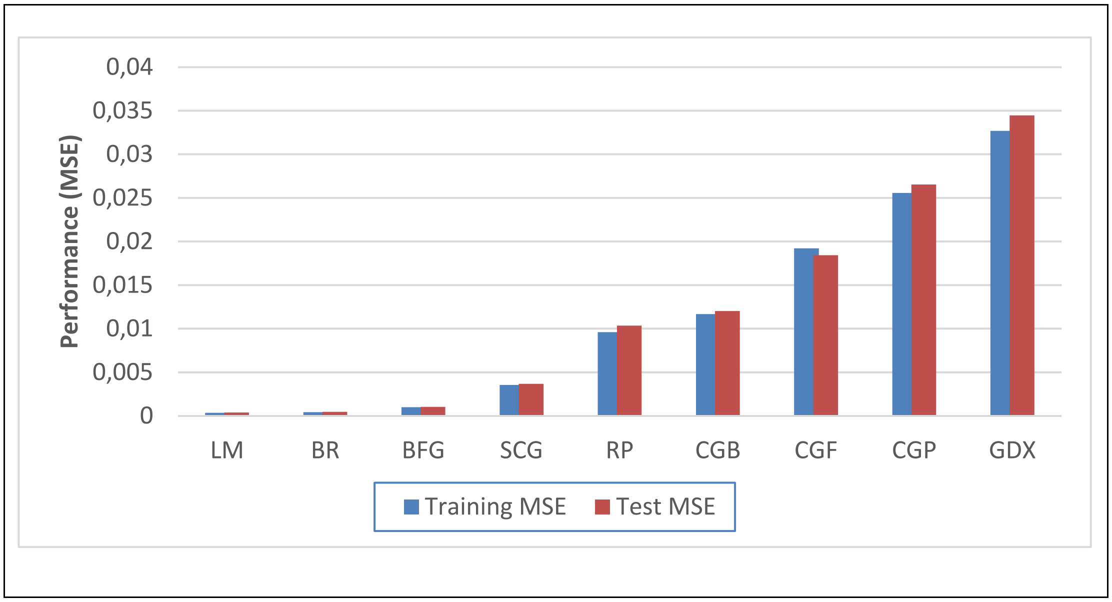


Figure 2.10 ANN training and test performances for different training algorithms

Choice of ANN Hyper Parameters

After identifying the two most effective solvers (LM and BR), a second study was conducted to determine the values of the ANN hyperparameters. To do this, the activation function that provided the best performance was first identified, and then the size of the network, defined by the number of neurons and the number of layers, was determined.

The activation function evaluates whether a neuron should be triggered, based on the relevance of each neuron contribution to the model prediction. The influence of various activation functions on the network performance was examined by testing a variety of activation functions available in the MATLAB software and listed in Table 2.2. For each activation function listed in Table 2.2, a neural network model was trained with both the LM and BR training algorithms.

Table 2.2 Implemented activation functions: a is the neuron activation, y is the neuron output

Activation Function	**Mathematical Equation**
Log Sigmoid (logsig)	$y(a) = \frac{1}{1 + \exp\,(-a)}$
Hyperbolic Tangent Sigmoid (tansig)	$y(a) = \frac{2}{(1 + \exp(-2 * a))} - 1$
Elliot Symmetric Sigmoid (elliotsig)	$y(a) = \frac{a}{(1 + \lvert a \rvert)}$
Radial basis (radbas)	$y(a) = \exp\,(-a^2)$
Normalized radial basis (radbasn)	$y(a)_i = \frac{\exp(-a_i^2)}{\sum_{j=1}^{n} \exp\,(-a_i^2)}$ where a is the input vector to a soft max function that consists of n elements of n classes, and a_i is the i-th element of the input vector.

Activation Function	Mathematical Equation
Soft max (softmax)	$y(a)_i = \frac{\exp(a_i)}{\sum_{j=1}^{n} \exp\,(a_j)}$ where a is the input vector to a soft max function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Saturating linear (satlin)	$y(a) = \begin{cases} 0, & \text{if } a \leq 0 \\ a, & \text{if } 0 \leq a \leq 1 \\ 1, & \text{if } 1 \leq a \end{cases}$
Symmetric saturating linear (Satlins)	$y(a) = \begin{cases} -1, & \text{if } a \leq -1 \\ a, & \text{if} - 1 \leq a \leq 1 \\ 1, & \text{if } 1 \leq a \end{cases}$
Triangular basis (Tribas)	$y(a) = \begin{cases} 1 - \lvert a \rvert, & \text{if} - 1 \leq a \leq 1 \\ 0, & \text{otherwise} \end{cases}$
Positive linear (Poslin)	$y(a) = \begin{cases} a, & \text{if } a \geq 0 \\ 0, & \text{if } a \leq 0 \end{cases}$

Figure 2.11 shows the test performance achieved for each activation function when the neural network was trained using the LM and BR algorithms to estimate the longitudinal aerodynamic coefficients. It should be noted that the performance obtained with the *Tribas* and *Poslin* activation functions are not shown in Figure 2.11 for reasons of scale, as their values were excessively high compared to those of the other activation functions (the MSE was 0.87 for *Tribas* and 0.91 for *Poslin*). The best performances were obtained for the *sigmoid* type of activation functions, namely *tansig*, *logsig* and *elliotsig*. Table 2.3 shows the values obtained. Any of those three sigmoid functions could be used to correctly learn and predict the aerodynamic coefficients of the aircraft, as their corresponding performances are somewhat similar. However, it was recommended to select the LM training algorithm in association with the 'tansig' activation function for the neural network training, since this combination provided the lowest MSE value.

Table 2.3 Test performances obtained for various activation functions [× 10^{-4}]

	tansig	**logsig**	**elliotsig**
LM	3.58	4.12	5.06
BR	4.11	4.00	3.75

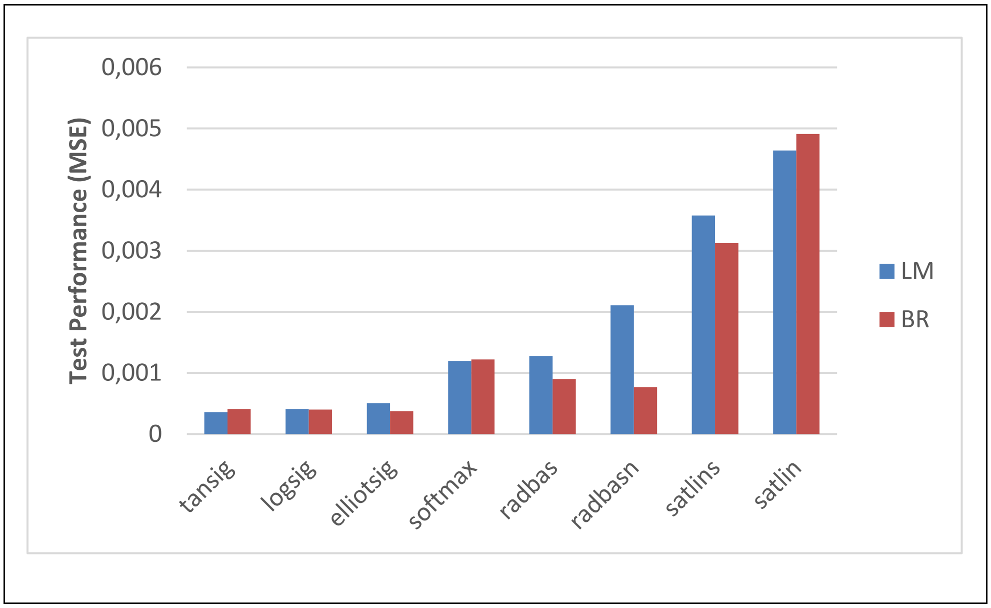


Figure 2.11 ANN performance variation for various activation functions

The last step in the neural network design process was to determine the appropriate number m of hidden layers of the network, and the number n of neurons per hidden layer required to achieve the best performance (i.e., the lowest MSE). Indeed, an insufficient number of neurons or hidden layers can result in the inability of the neural network to correctly identify correlations in a complex dataset. In contrast, too many neurons in the hidden layer can lead to overfitting problems or unnecessarily long training time. A trade-off must be made between too many and too few hidden neurons/layers.

To accomplish this, a similar approach to the one described for the choice of the training algorithm and activation function, was adopted. This approach aimed to evaluate the performance of several neural network structures (i.e., with different numbers of layers and neurons), and to identify the one that gave the best performance within an appropriate training time. In reference (Tondji et al., 2022b), a similar study was conducted using neural networks to predict longitudinal aerodynamic coefficients of the Cessna Citation X in stall conditions. Three different neural networks were used to predict each aerodynamic coefficient. The study showed that a network of $m = 5$ hidden layers and $n = 9$ on each hidden layer was necessary for modeling the lift coefficient. The drag coefficient, on the other hand, required a network of size $(m, n) = (2, 14)$. The pitching moment coefficient required an ANN of size $(m, n) = (4, 12)$.

In the current study, we use a single neural network to model the three longitudinal aerodynamic coefficients. For this reason, the modeling should require a more complex neural structure than that used in (Tondji et al., 2022b). However, we are modeling aerodynamic coefficients of an aircraft in stabilized flight conditions, with less complex dynamics than those of an aircraft in stall conditions, as was modeled in (Tondji et al., 2022b).

Based on this observation, a minimum of $m_{min} = 1$ hidden layers, and a maximum of $m_{max} = 5$ hidden layers was set. Similarly, a range between $n_{min} = 3$ and $n_{max} = 14$ was defined for the number of neurons per hidden layer. Therefore, $12^5 = 2.48 \times 10^5$ different structures would be needed to train to find the optimal one. Given this high number, it was assumed that all the hidden layers would have the same number of neurons to reduce the number of potential structures and, thus, reduce the time required for this optimization. This assumption was considered reasonable, as changing the number of neurons from one layer to another did not significantly improve the network performance (Heaton, 2008). By making this assumption, the number of possible structures drops to 60.

Using the 5-fold cross-validation approach described in section 2.2.3.2, each of the 60 structures was trained 5 times. The test performance (MSE) obtained for each (m, n) combination is shown in Figure 2.12.

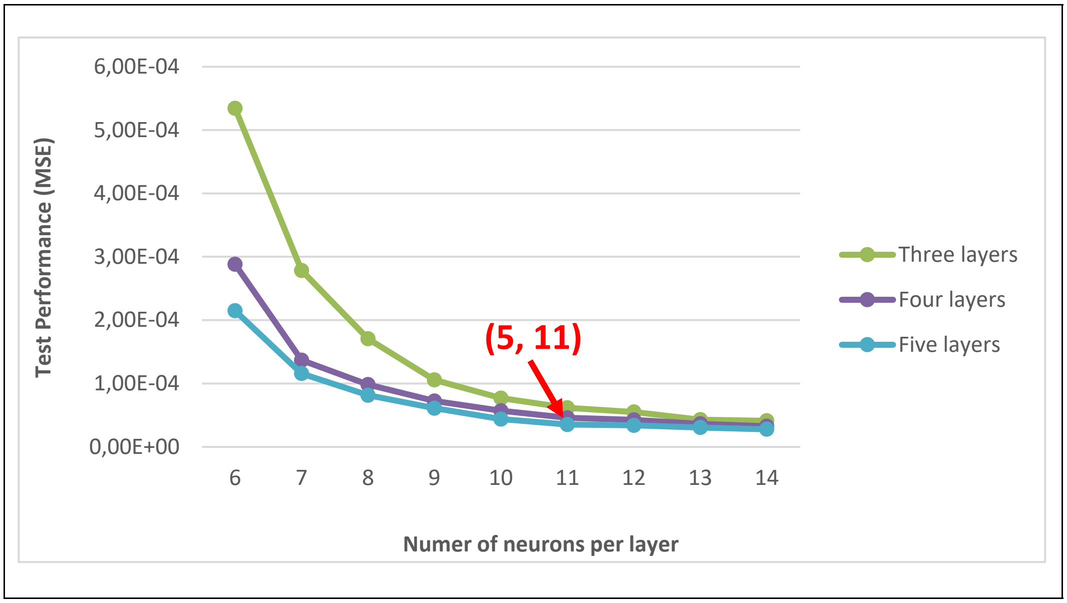


Figure 2.12 Performances of various ANN structures for their estimation of the CL_s of the Bombardier CRJ-700

Results obtained using ANN topologies with 1 to 5 neurons per hidden layer and topologies with 1 and 2 hidden layers have been deleted for scaling reasons because their $MSEs$ were too large compared to the results obtained with the other topologies. Figure 2.12 shows that for all three number of layers, the MSE value significantly decreases when the number of neurons per layer increases, reflecting an improvement in the model performance. For the same number of neurons per layer, increasing the number of layers also decrease the model MSE. We can see that structures with 5 layers perform better than those with fewer layers. The MSE values of 5-layer typologies decreased significantly when the number of neurons increased until reaching n=11, from where improvement in performance was no longer significant. Thus, the optimal ANN structure in this case, with a good trade-off between performance and learning time, is the typology $(m, n) = (5, 11)$.

2.2.3.4 Support Vector Regression

Support Vector Regression (SVR) is a powerful ML technique based on statistical learning theory that can be used for approximating nonlinear functions and modeling nonlinear systems (Wang et al., 2015). In a MISO nonlinear system modeling problem, the SVR approximates the non-linear dynamics of the system using a linear regression formulation in a high-dimensional feature space F. This is achieved by applying a nonlinear transformation $\phi\ (x)$ from the inputs space R_m to the feature space F, as shown in Equation (2.13). This transformation function ϕ plays a crucial role in the SVR by mapping input data into a higher-dimensional feature space, commonly known as a Hilbert space. This mapping is fundamental to the SVR effectiveness because it enables linear regression to be conducted within this “expanded space”, even when the relationship in the original input space is nonlinear.

$$\hat{o}(x) = \mathrm{w}^T \phi\ (x) + b \tag{2.13}$$

The SVR objective is to minimize the absolute errors between the predicted and actual values, unlike ANN regression, which aims to minimize the sum of squared residuals. To accomplish this objective, the SVR introduces a margin around the hyperplane of best fit that allows for a certain precision ε. For each data point i, the absolute difference between the actual value o_i and the predicted value $(\mathrm{w}^T \phi\ (x_i) + b)$ should be less than or equal to the allowed error ε (Drucker et al., 1996), as stated in Equation (2.14).

$$|o_i - (\mathrm{w}^T \phi\ (x_i) + b)| \leq\ \varepsilon, \quad i = 1, \dots, n \tag{2.14}$$

Data points within the margin are predicted correctly, while those outside of the margin are considered errors. However, there may be no function that satisfies this constraint for all points (x_i, o_i), so that slack variables ξ_i and ξ_i^* are introduced to allow some flexibility in the predictions. The slack variables are non-negative and are used to measure the violation of the constraint defined in Equation (2.14), allowing for a certain amount of error greater than the ε value to be tolerated in the predictions.

The introduction of slack variables in Equation (2.14) leads to the following constraints (Drucker et al., 1996):

$$\begin{cases} o_i - \mathrm{w}^T\phi\,(x_i) - b \le\; \varepsilon + \xi_i \\ \mathrm{w}^T\phi\,(x_i) + b - o_i \le\; \varepsilon + \xi_i^* \\ \qquad\qquad \xi_i \ge 0 \\ \qquad\qquad \xi_i^* \ge 0 \end{cases} ,\forall\, i = 1, \dots, n \tag{2.15}$$

The objective function for the SVR is developed based on the problem of minimizing the prediction errors while also minimizing the weights. The objective function is given by Equation (2.16), also known as the primal formula (Vapnik, 1995):

$$J(\mathrm{w}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2.16}$$

The first part of the objective function, $\frac{1}{2}\|w\|^2$, is the same as the traditional linear regression function and is designed to minimize the sum of squared weights to prevent overfitting by avoiding large weight values. The second part of the objective function, $C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$, is specific to the SVR and aims to minimize the total slack variables. The sum of the slack variables $\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ gives the total error that is allowed in the predictions. The regularization parameter C controls the trade-off between minimizing the total slack variables and minimizing the weights.

The choice of a solver and the optimization of the SVR hyperparameters are presented below only for the prediction of the lift coefficient of the Bombardier CRJ700. The described procedure was applied also to model the drag and pitching moment coefficients, and the results for all aerodynamic coefficients are presented in section 2.3.

Choice of the SVR Solver

In the SVR context, the optimization problems can be formulated as Quadratic Programming (QP) problems. Quadratic Programming involves minimizing a quadratic objective function subjected to linear constraints (Boggs & Tolle, 2000). Solving the QP problem is a critical step in training a SVR model. For modeling the lift coefficient (CL_S) using a fixed SVR model, we tested three different solvers: the Decomposition Method (DM), Sequential Minimal Optimization (SMO), and Improved SMO Decomposition Algorithm (ISDA). Each of these solvers has its advantages and disadvantages. The hyperparameters were set as follows: Kernel function: $Gaussian$, Gaussian Parameter $\sigma = 20$, penalty factor $C = 3.09,$ and a precision $\varepsilon = 0.001$.

The decomposition method is efficient in handling large-scale problems by decomposing the original QP problem into a series of smaller QP subproblems (Chih-Chung Chang, Chih-Wei Hsu & Chih-Jen Lin, 2000). However, it may not be suitable for all problems, as it can require a significant amount of memory. In our specific case, the decomposition method did not go through the entire optimization process because it ran out of memory. Considering our training set size of 49,680 seven-dimensional data points, the required memory is at least 36 GB. This is a significant increase compared to our available resources, which stand at 32 GB of memory. This limitation presents a notable disadvantage of the decomposition method for our specific problem.

The SMO solver is an iterative algorithm that solves the QP problem by breaking it into smaller subproblems that can be solved analytically (Platt, 1998). It is designed to handle large-scale problems efficiently. In our case, the SMO solver reached the convergence goal faster than DM and ISDA, in only 2.7×10^6 iterations.

The ISDA solver is an improvement over the traditional SMO algorithm, as it employs an adaptive selection of working set size and search direction (Kecman, Huang & Vogt, 2005). This adaptive approach can provide faster convergence in some cases. However, for our problem, the ISDA solver has not reached the convergence goal within the maximum allowed 10^8 iterations, which suggests that the ISDA algorithm might not be the most suitable choice for this optimization problem. This result indicates that the SMO is the most appropriate choice for solving our optimization problem, considering its efficiency and ability to converge within the given constraints.

SVR Hyperparameter Optimization using Bayesian Optimization

The selection of SVR hyperparameters, including ε, C, Kernel function and Kernel specific parameters, plays a crucial role in determining the performance of the Support Vector Regression (SVR) model. These choices have a significant impact on the accuracy and robustness of the predictions generated by the model. The Kernel function is responsible for transforming the input data into a higher-dimensional space, where the SVR model can more effectively capture the underlying relationship between the inputs and the target output. Among the commonly used Kernel functions are Gaussian and polynomial Kernels, which have been tested and proven effective in modeling a wide range of nonlinear systems (Fan et al., 2005; Wang et al., 2015). Gaussian and polynomial Kernels functions expressions are shown respectively in Equation (2.17) and Equation (2.18), where σ is the Gaussian Kernel scale and q is the polynomial order of the polynomial Kernel function.

$$K\left(x_i, x_j\right)_G = \exp\,(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}) \tag{2.17}$$

$$K\left(x_i, x_j\right)_q = \left(1 + x_i^T x_j\right)^q \tag{2.18}$$

The Kernel specific parameters have a significant impact on the shape and complexity of the Kernel function, which, in turn, affects the model performance. For instance, if we use a polynomial Kernel function, we must set its polynomial degree q, and capture the complex relationships between input variables and the target outputs. However, a higher polynomial degree also increases the risk of overfitting. Similarly, if we use Gaussian Kernel function, the Gaussian Kernel scale σ should be set appropriately to control the shape of the Gaussian Kernel function in order to avoid overfitting or underfitting.

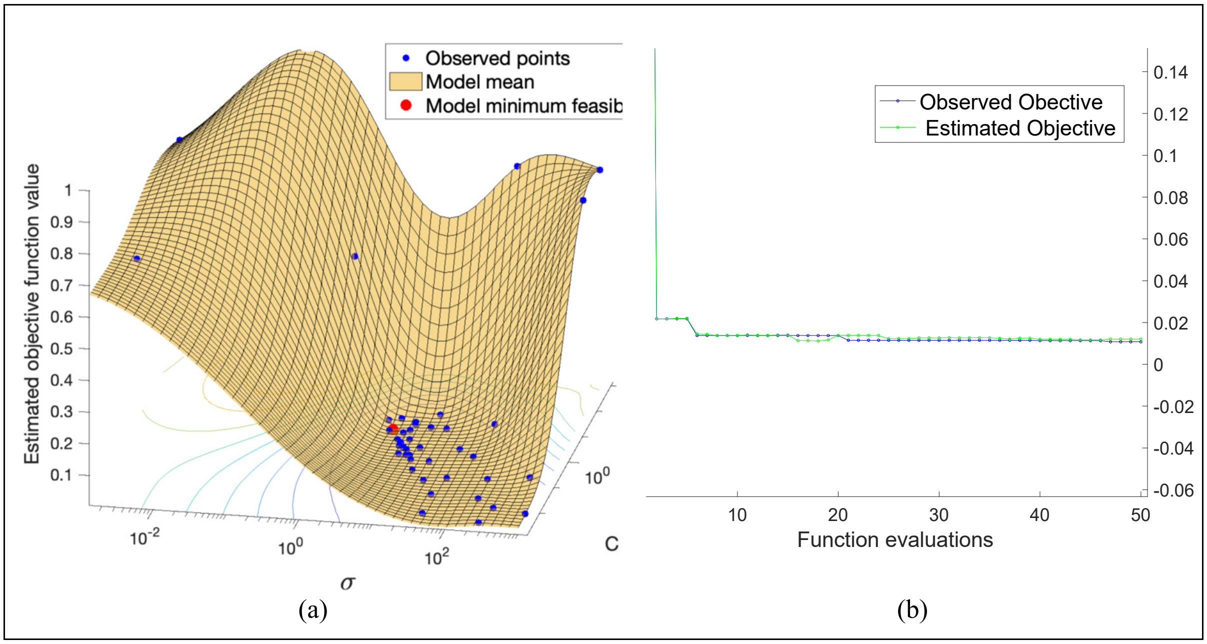
The regularization parameter C is another crucial parameter that controls the trade-off between the model complexity and its ability to fit the training data. A highvalue of C prioritizes minimizing the training error, resulting in a model that fits the training data very closely.

However, this could also lead to capturing noise and overfitting. In contrast, a low value of C prioritizes minimizing the model complexity and could result in a model that generalizes well. However, if C is set too low, the model may not capture the underlying relationship between the input variables and the target outputs, resulting in underfitting.

The precision ε is an user-defined parameter that controls the errors tolerance in the model predictions. It defines a margin or "insensitive region" around the true target output, within which errors are considered acceptable and do not contribute to the model training error. This parameter should be not too small, so that the minimization of the objective function in Equation (2.15) can converge to an existing solution. However, it must be not too high for the aerodynamic model to satisfy the FAA validation tolerances. For the lift aerodynamic coefficient CL_s modeling, trial-and-error experiments showed that a precision of 10^{-3} provided the best results.

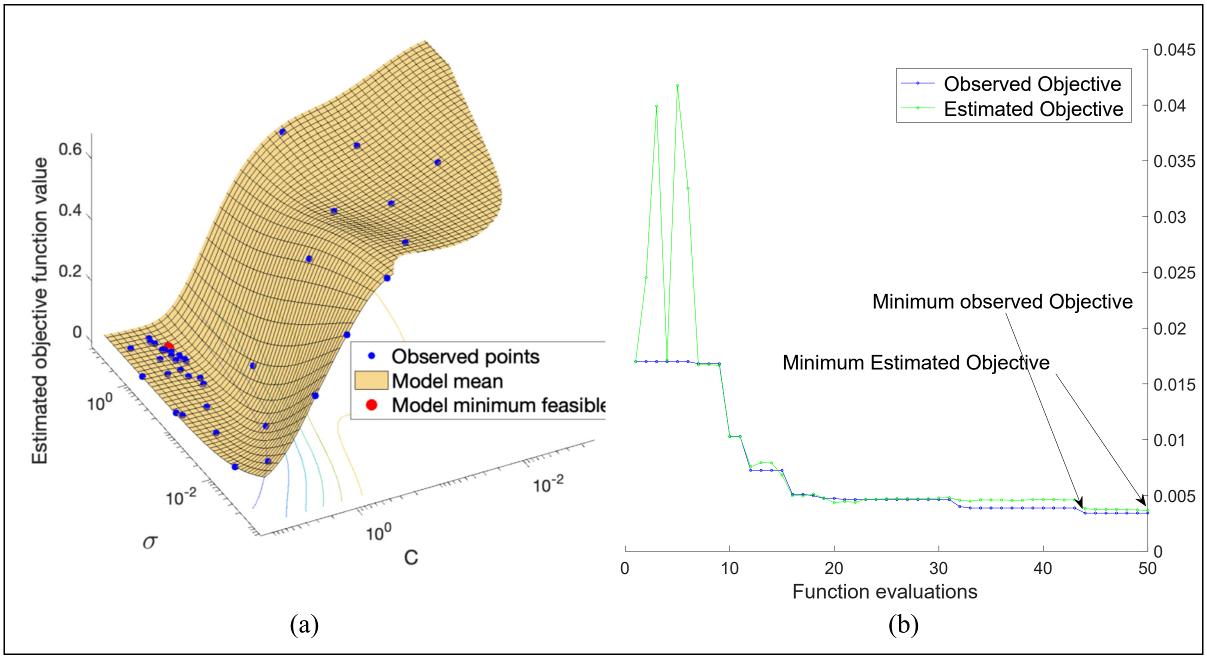
Bayesian Optimization (BO) (Shahriari, Swersky, Wang, Adams & de Freitas, 2016) was used to determine the optimal pair of parameters (C, σ) or (C, q). The BO is a powerful and efficient method for finding the best parameter values for a model while minimizing the number of times in which the objective function needs to be evaluated. This is particularly useful when the objective function is expensive to evaluate or lacks an analytical expression, as is the case with the SVR objective function. To achieve this evaluation, the BO employs a surrogate probabilistic model to approximate the true objective function. This surrogate model is a statistical model that estimates the structure of the objective function and quantifies the uncertainty in the predictions. It plays a critical role in the BO by efficiently exploring the parameter space and identifying its optimal parameters. Indeed, during the optimization process, the surrogate model serves as a substitute for the actual objective function, as it is cheaper to evaluate and provides valuable insights into the uncertainty of its predictions (Pelikan & Goldberg, 1999).

Figure 2.13 and Figure 2.14 illustrate the first and second steps of the BO grid for the pair of parameters $(C,\ \sigma)$ when the Gaussian Kernel function is used. In the first step, five optimizations were performed using a broad search range of $(C, \sigma) \in [10^{-3}\ 10^{3}], [10^{-3}\ 10^{3}]$. These optimizations yielded similar minimum values estimated by the surrogate model (Minimum Estimated Objective Function Values), but which were located at different evaluation points $(C_{\min est}, \sigma_{\min est})$. However, the $C_{\min est}$ and $\sigma_{\min est}$ values were found to be between 10^{-3} and 10. For example, Figure 2.13.(a) displays the objective function model plot of the first optimization among the five optimizations. The plot shows all evaluated points (in terms of (C, σ)) and their corresponding objective values estimated by the surrogate model in "blue" circles. The "red" circle represents the Minimum Estimated Objective Function Value called $obj_{\min est} = 0.0121$ at $(C_{\min est}, \sigma_{\min est})$ = (0.58, 6.17). Figure 2.13.(b) shows the comparison between the Estimated Objective Function Value and the observed value of the actual objective function (Observed Objective Function Value) during the optimization process. Both values decreased over iterations and were always close, thus indicating that the surrogate probabilistic model approximated well the objective function. In the second step, an optimization was performed with a refined search range of $(C, \sigma) \in [10^{-3}\ 10], [10^{-3}\ 10]$. This optimization yielded a better result than the first step optimization, with a Minimum Estimated Objective Function Value of $obj_{\min est} = 0.0036$ at $(C_{\min est}, \sigma_{\min est})$ = (1.78, 3.02). Furthermore, Figure 2.14.b clearly shows that the Minimum Estimated Objective Function Value and Minimum Observed Objective Function Value were different, although they were close. In fact, the Minimum Observed Objective Function Value was $obj_{\min obs} = 0.0034$ at $(C_{\min obs}, \sigma_{\min obs})$ = (2.28, 2.98). Therefore, $(C_{\min obs}, \sigma_{\min obs})$ was considered the optimal hyperparameters, as they led to a minimum objective function value of $obj_{\min obs} = 0.0034$.



(a) (b)

Figure 2.13 1st step Bayesian Optimization plot for (C, σ) estimation to model CL_S using SVR



(a) (b)

Figure 2.14 2nd step Bayesian Optimization plot for the (C, σ) estimation to model CL_S using SVR

The same optimization procedure was repeated for parameters $(C,\ q)$ when the polynomial Kernel function was considered. The BO results are presented in Table 2.4. We can conclude that performances achieved with Gaussian Kernel function are better than performances obtained with Polynomial Kernel function, as the $obj_{\min obs}$ is smaller. Similar studies have been conducted to model the drag and pitching moment coefficients. All results are presented in section 2.3.

Table 2.4 Optimal hyperparameters obtained via the BO of SVR model for the determination of CL_S

Kernel function	$obj_{\min obs}$	$C_{\min obs}$	$\sigma_{\min obs}$	$q_{\min obs}$
Gaussian	0.0034	2.28	2.98	-
Polynomial	0.0036	0.0407	-	5

2.3 Results

This section presents the results of the proposed methodology. Two analyses were conducted to show the efficiency and accuracy of the ANN and SVR models.

The first analysis was designed to determine how well both AI models predicted the aerodynamic coefficients. For this purpose, Bombardier CRJ-700 VRESIM aerodynamic coefficients dataset from 1,992 flight cases used for validation were compared to the outputs of the ANN and the SVR models, while using Mean Absolute Relative Error (MARE) as the metric. For a given set of n data points, the $MARE$ was calculated according to Equation (2.19), where o_i is the ith output vector associated with the ith input vector in the dataset, and $\hat{o}_i$ is the ith output vector predicted by the model.

$$MARE = \frac{1}{n}\left(\sum_{\mathrm{i}=1}^{n}\left|\frac{\hat{o}_i - o_i}{\hat{o}_i}\right|\right) \times 100 \tag{2.19}$$

However, due to the absence of established criteria in the literature for validating the aerodynamic coefficients of an aircraft, a secondary analysis was conducted. The trained ANN and SVR models were integrated into a CRJ-700 simulation platform, designed to replicate the simulated flight test scenarios conducted with the VRESIM. This validation was performed by comparing several flight parameters predicted by the developed platform with those measured with the VRESIM, in accordance with the tolerance criteria specified by the FAA in the Manual of Criteria for the Qualification of Flight Simulators (FAA, 1991; ICAO, 2016).

2.3.1 Validation of the Aerodynamic Coefficients

Two types of artificial intelligence models have been developed to predict the longitudinal aerodynamic coefficients of the Bombardier CRJ700: ANN and SVR. The ANN model was developed to predict the three coefficients CL_S, CD_S, and Cm_S, while three distinct SVR models (one for each aerodynamic coefficient) were designed. These models were trained with 360 (15.3 %) flight cases and validated on 1,992 (84.7 %) flight cases. As results, the optimal hyper parameters for the ANN were the training function LM, activation function 'tansig', 5 hidden layers and 11 neurons on each hidden layer. The results obtained were identical to those obtained for similar studies of aerodynamic coefficients modeling using ANN (Tondji, Ghazi & Botez, 2022; Tondji et al., 2022b), confirming their relevance. The size of the resulting network was also acceptable, considering that only one network was used to predict the three aerodynamic coefficients.

All three SVM models performed better with the SMO solver and Gaussian Kernel. The values of the Regularization parameter C, the Kernel scale σ and the precision ε are given in Table 2.5. In addition, Kernel scale values were of the same order, given that the research domain interval was $[10^{-3}\ 10^{3}]$. This aspect demonstrated the similarities between the aerodynamic coefficients data structure and pathway. The precision of each model was chosen depending on the order of the aerodynamic coefficients' values. A precision of $\varepsilon = 10^{-3}$ was chosen for CL_S and CD_S because the values of the coefficients were positive and of the order of 10^{-1} or 10^{-2}. For the pitching moment coefficient Cm_S, a more restrictive precision of $\varepsilon = 10^{-4}$ was selected because the values of this coefficient could change their signs, meaning that some values were too small and close to zero.

Table 2.5 Optimal hyperparameters ε, σ and C obtained for the SVR by use of Gaussian Kernel function and SMO solver

	Precision $\boldsymbol{\varepsilon}$	**Kernel scale $\boldsymbol{\sigma}$**	**Regularization parameter $\boldsymbol{C}$**
$\boldsymbol{CL_S}$	10^{-3}	2.98	2.28
$\boldsymbol{CD_S}$	10^{-3}	2.05	18.98
$\boldsymbol{Cm_S}$	10^{-4}	3.12	107.71

Figure 2.15 to Figure 2.17 present examples of estimated aerodynamic coefficients plotted with respect to the angle of attack, juxtaposed with their corresponding experimental data. The data predicted by the ANN (depicted in red) and SVR (shown in black) models closely align with the experimental data obtained with the VRESIM (illustrated in blue).

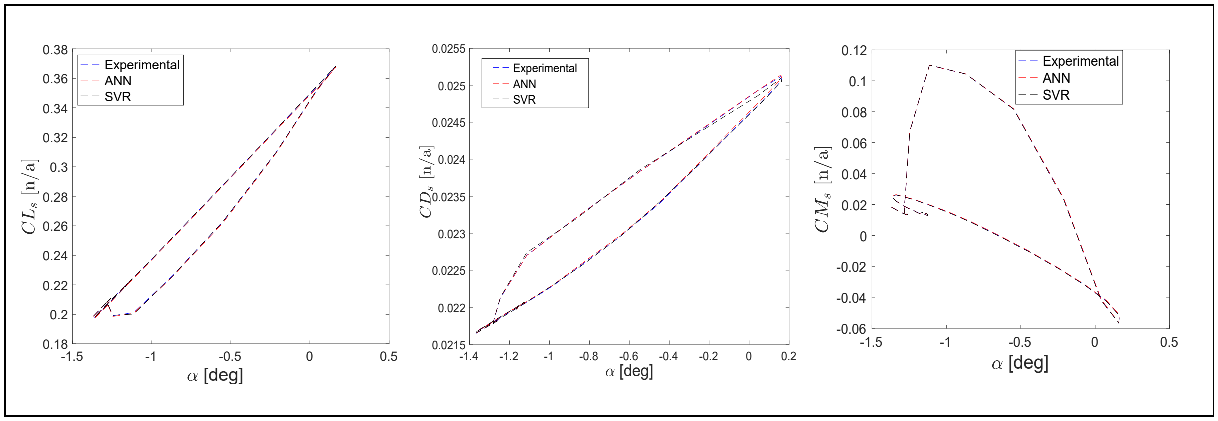


Figure 2.15 Predicted coefficients for flight test at $m = 46{,}300\ lb \mid Xcg = 24\% \mid h = 5000\ ft \mid IAS = 310\ kts$

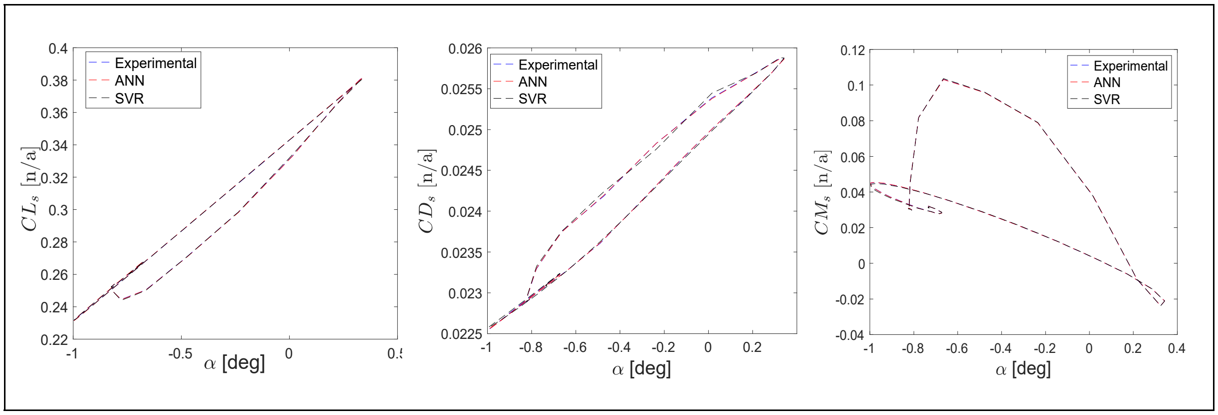


Figure 2.16 Predicted coefficients for flight test at $m = 59{,}520\ lb \mid Xcg = 18\% \mid h = 10{,}000\ ft \mid IAS = 310\ kts$

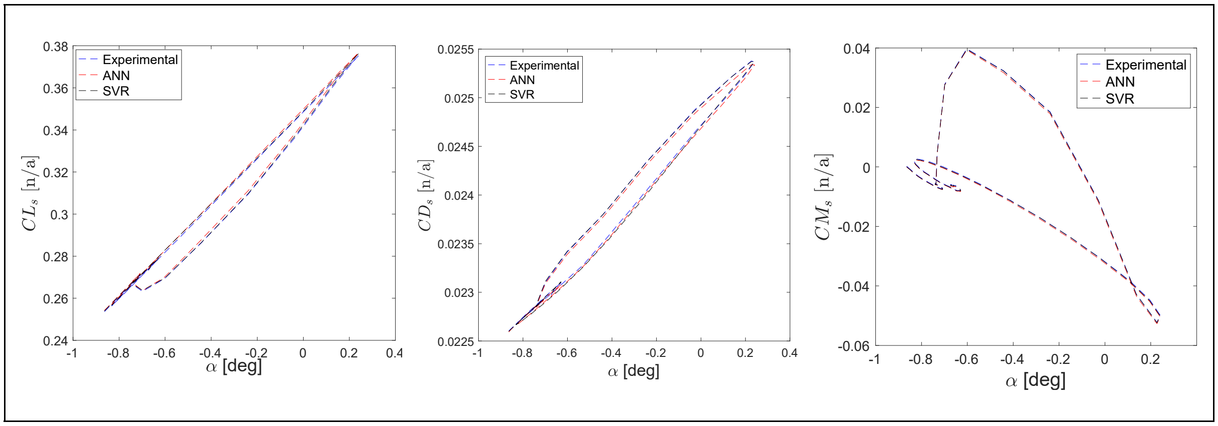


Figure 2.17 Predicted coefficients for flight test at $m = 66{,}930\ lb \mid Xcg = 32\% \mid h = 5000\ ft \mid IAS = 320\ kts$

The average errors obtained for all validation flight scenarios are presented in Table 2.6. As indicated in this table, the ANN model predicted lift and drag coefficients with an average MARE of less than 0.1%, while SVR models predicted them with an average MARE of less than 1%. Similarly, the ANN predicted the pitching moment coefficients with an average MARE of 0.5%, while SVR did it with an average MARE of 5.74 %. From a general perspective, we can conclude that the ANN model was ten times more accurate than the SVR models. However, SVR models achieved the precision specified during the learning phase. Table 2.6 shows that the average residual error obtained with SVR was equal to 2.4×10^{-3} for CL_S, and 0.206×10^{-3} for CD_S which is close to the specified precision of $\varepsilon = 10^{-3}$. Similarly, the average residual error for Cm_S was equal to 8×10^{-4}, which is on the same order as the specified precision of $\varepsilon = 10^{-4}$. Therefore, we can conclude that both SVR and ANN models performed very well. In addition, specifying the desired model performances allow us to significantly reduce the model training time. We can also observe that the performances obtained in terms of $average\ MARE$ for Cm_S were approximately ten time higher than those obtained for CL_S and CD_S with both ANN and SVR models. This result can be explained by the fact that the pitching moment coefficient changes its sign during the flight simulation. Consequently, the Cm_s has relatively low values, around zero, which leads to large relative errors ($MARE$). However, the results obtained remain very good, as the residual errors are very low, even negligible, on the order of 10^{-4}.

Table 2.6 Average $MARE$ and Average Residual Error (and the corresponding Standard Deviation) obtained for the prediction of CL_s, CD_s and Cm_s using ANN and SVR

	Average $MARE$ [%]		**Standard deviation of the $MARE$ [%]**		**Average Residual Error [× 10^{-4}]**		**Standard deviation of Residuals Error [× 10^{-4}]**	
	ANN	SVR	ANN	SVR	ANN	SVR	ANN	SVR
CL_S	0.08	0.82	± 0.06	± 1.22	8.27	24	± 1.67	± 12
CD_S	0.03	0.76	± 0.02	± 1.25	0.19	2.06	± 0.08	± 3.70
Cm_S	0.50	5.74	± 0.42	± 6.69	4.09	8	± 1.1	± 8.04

2.3.2 Flight Dynamics Model Validation

A secondary analysis was conducted to further validate the accuracy of the ANN and SVR in modeling the longitudinal dynamics (short period and phugoid motions) of the CRJ700. This analysis involved incorporating both models into a simulation platform of the Bombardier CRJ-700 that was designed in Matlab/Simulink (see Figure 2.18). The simulation platform was then used to replicate the 1,992 validation flight scenarios, and the longitudinal flight parameters predicted by the platform were compared with those derived from the CRJ-700 VRESIM.

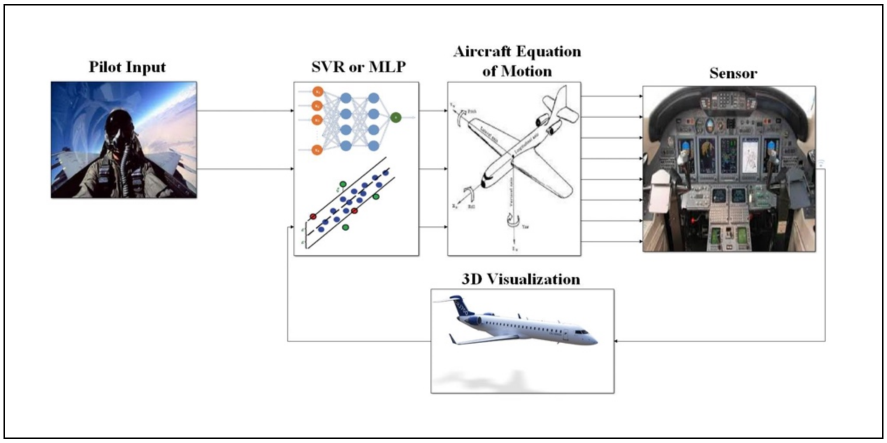


Figure 2.18 Bombardier CRJ-700 flight dynamics platform.

The FAA has defined criteria to validate the developed models by specifying tolerances for short-period and phugoid dynamics. For the short-period dynamics model, the pitch rate must be estimated within a tolerance of ± 2°/s, and the normal acceleration within a tolerance of ± 0.1 g. For the phugoid dynamics model, the oscillation period of the pitch angle should be estimated within a tolerance of ± 10 %, and the damping ratio within a tolerance of ± 0.02.

Figure 2.19 to Figure 2.21 show the comparative simulation results for the short-period dynamics for three (3) different flight cases. The variables simulated by the ANN model are represented in “red” color. The variables simulated by the SVR model are depicted with “black” lines. The dotted “blue” lines represent the tolerance limits specified by the FAA, which were calculated using the VRESIM experimental data. These results demonstrate that for the three flight cases presented in Figure 2.19 to Figure 2.21, pitch rate and vertical acceleration values were obtained within the specified tolerances. Indeed, the flight parameters calculated by the simulation platform with the two models (i.e., ANN (red) and SVR (black)) were almost exactly superimposed on the parameters obtained by the VRESIM, thereby fulfilling all the tolerances specified by the FAA. The same type of comparison was performed for all the 1,992 flight cases used for model validation, and a success rate of 100 % was achieved, meaning that the models were able to predict the short-period flight parameters very well within the tolerances and for the entire CRJ700 flight envelope.

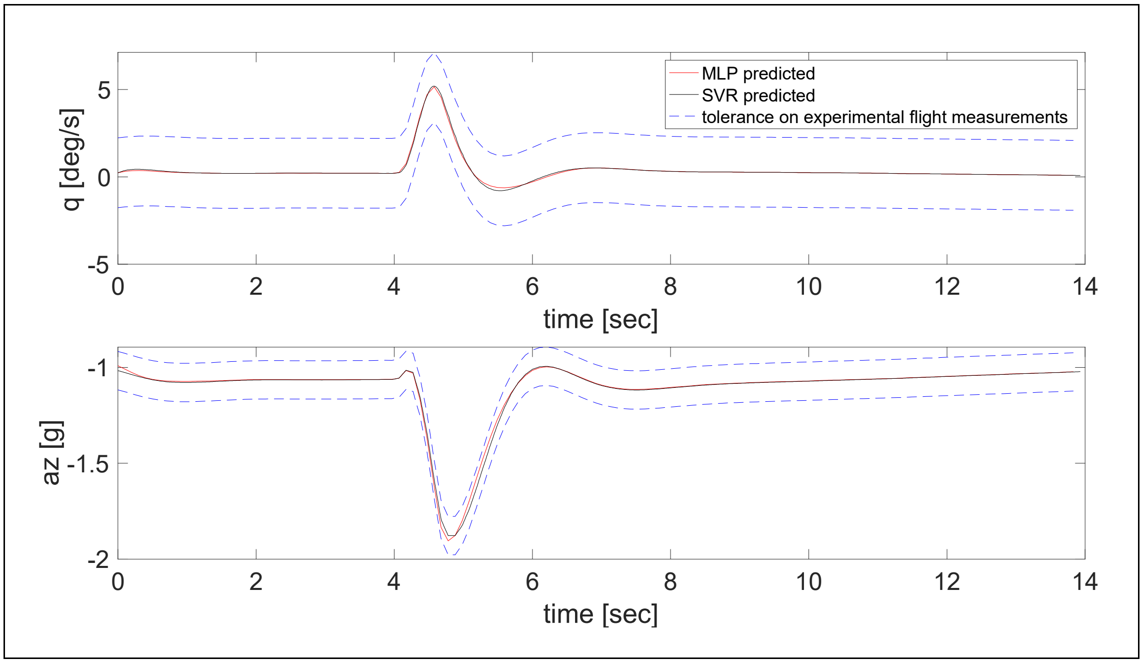


Figure 2.19 Short period parameters for flight test
at $m = 46{,}300\ lb \mid Xcg = 24\% \mid h = 5000\ ft \mid IAS = 310\ kts$

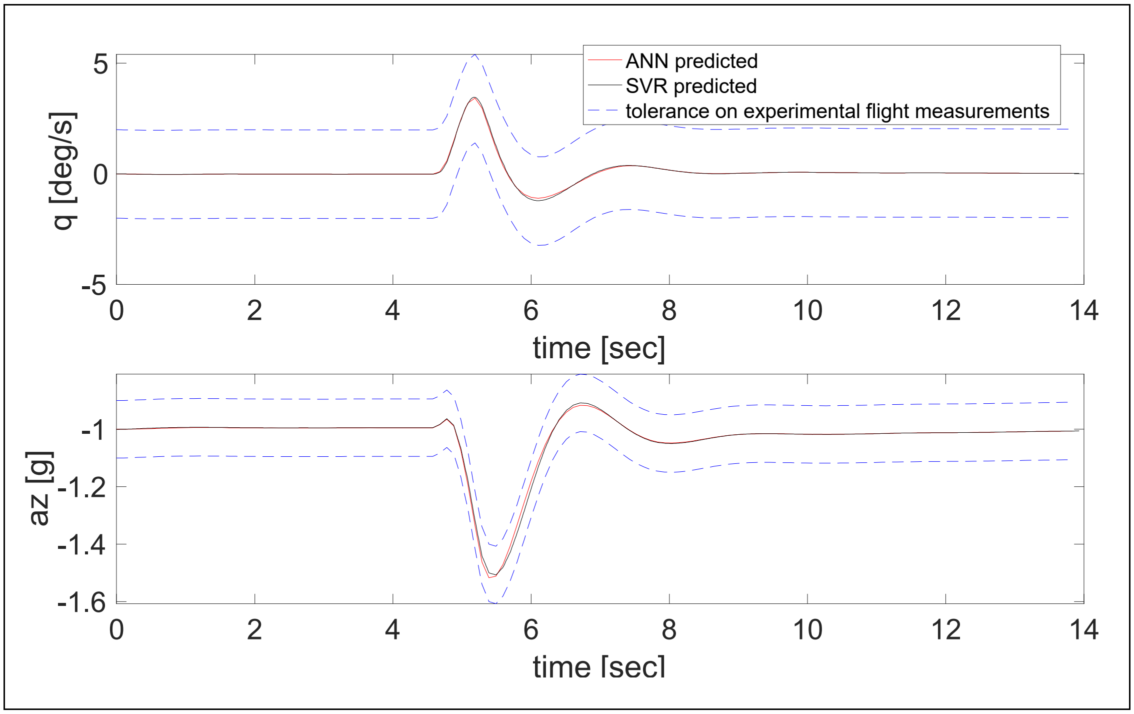


Figure 2.20 Short period parameters for flight test
at $m = 59{,}520\ lb \mid Xcg = 18\% \mid h = 10{,}000\ ft \mid IAS = 310\ kts$

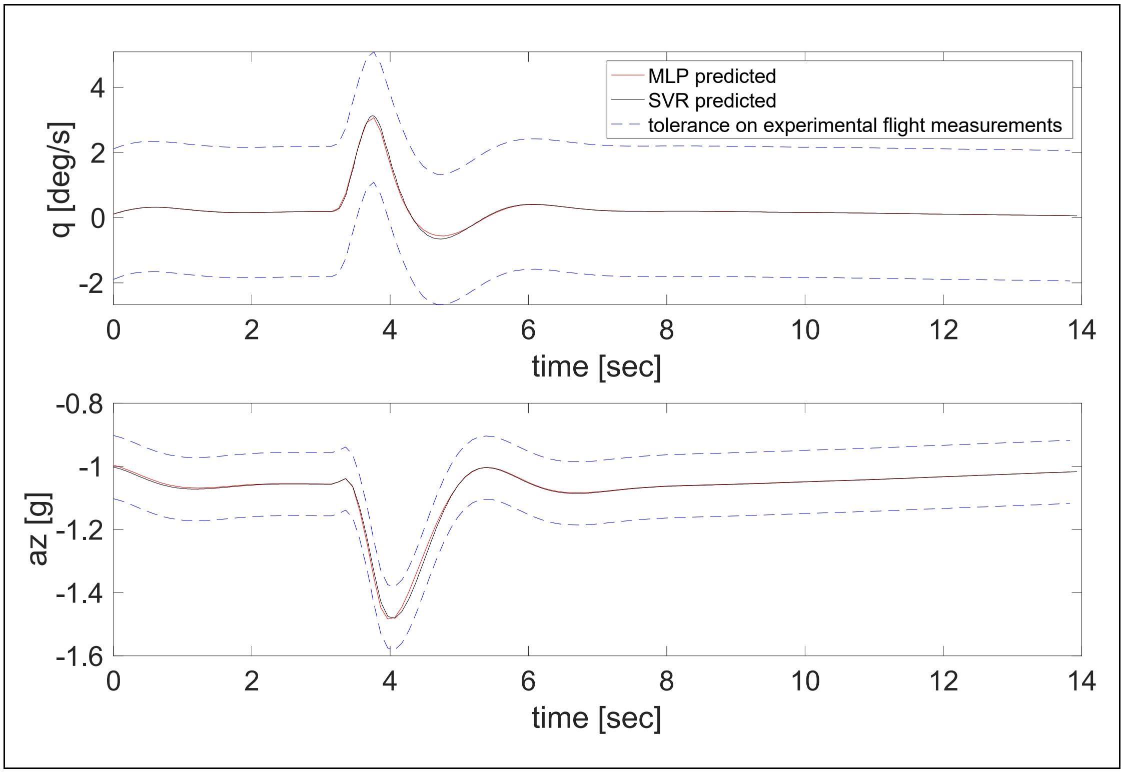


Figure 2.21 Short period parameters for flight test
at $m = 66{,}930\ lb$ | $Xcg = 32\%$ | $h = 5000\ ft$ | $IAS = 320\ kts$

For the phugoid model validation, the pitch angle period and the damping ratio were determined by finding the phugoid oscillation peaks, as shown in Figure 2.22 and in Equation (2.20).

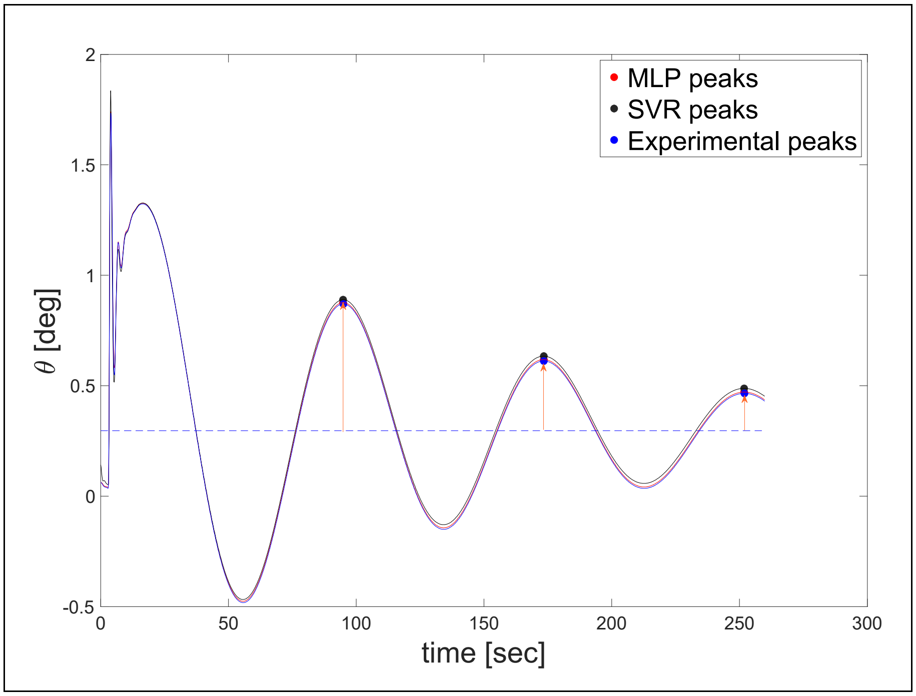


Figure 2.22 Example of pitch angle prediction for a simulated flight test at $m = 59{,}520\ lb \mid Xcg = 18\% \mid h = 10{,}000\ ft \mid IAS = 310\ kts$

Once the peaks were determined, as shown in Figure 2.22, the period of the pitch angle θ was found by calculating the mean of the time difference between successive peaks. The damping ratio was determined by using the following Equation (2.20):

$$\zeta = \frac{1}{\sqrt{1 + \left(\frac{2\pi}{\delta}\right)^2}} \tag{2.20}$$

where δ is the logarithmic decrement and is defined as the natural logarithmic of the ratio of two successive altitude peaks, as given by Equation (2.21),

$$\delta = \frac{1}{n}\log\left(\frac{p(i)}{p(i + nT)}\right) \tag{2.21}$$

where $p(t)$ and $p(t + nT)$ are the amplitudes of two peaks separated by n periods of value T.

For all validation flight cases, the predicted pitch angles periods and damping ratios were found to be within the FAA tolerances. The results of the calculations are given in Table 2.7 for the same flight cases presented in Figure 2.19 to Figure 2.21. The relative errors of the period T estimations were all less than 10 %, and the absolute residual errors of the damping ratio estimations were all less than 0.02. Based on these results, we can conclude that a 100% success rate was obtained for both the ANN and the SVR models.

Table 2.7 Error on the prediction of pitch angle period and damping ratio

	Relative error of the period T estimation [%]			**Residual error of the damping ratio ζestimation**		
	ANN	SVR	Tolerance limit	ANN	SVR	Tolerance limit
m = 46,300 lb | Xcg = 24% | h = 5000 ft | IAS = 310 kts	0	0	± 10	0.0054	0.0054	± 0.02
m = 59,520 lb | Xcg = 18% | h = 10,000 ft | IAS = 310 kts	0	0	± 10	-0.0021	-0.0019	± 0.02
m = 59,520 lb | Xcg = 18% | h = 10,000 ft | IAS = 310 kts	0	0.11	± 10	-0.0052	-0.0052	± 0.02

2.4 Conclusions

This study explored the use of Artificial Neural Networks (ANNs) and Support Vector Regression (SVRs) models to predict the longitudinal aerodynamic coefficients of the CRJ700 aircraft. Both models were tested and compared with experimental data obtained from a Level-D CRJ-700 VRESIM, where Level-D refers to the highest certification level issued by the FAA for the flight dynamics and propulsion models. In our approach, the Bayesian optimization was used. That is a strategy known for efficiently finding the best model parameters, ensuring that our models were finely tuned. Additionally, K-folds cross-validation was employed, providing a robust measure of our models performance across different data subsets. The aerodynamic coefficients were predicted as functions of several factors such as: flight conditions, which included the center of gravity position variation, Mach number, and the ratio of dynamic pressure to the true airspeed; flight parameters, which included the angle of attack and pitch rate; additionally, pilot inputs, such as elevator and horizontal stabilizer angle deflections, which were also incorporated into the models to enhance their accuracy.

Both ANN and SVR methodologies gave excellent performance across 1,992 simulated validation flight tests. They were able to predict the aerodynamic coefficients with average mean absolute relative errors smaller than 1% for the lift and drag coefficients, and smaller than 5.74 % for the pitching moment coefficient. They achieved a 100% success rate in predicting longitudinal flight parameters with an accuracy within the FAA tolerances corresponding to the Level-D. This high success rate represents the AI models potential in the Aerospace field. It is worth noticing that the learning speed of the SVR model is particularly higher compared with that of the ANN model, which could be useful for real-time applications.

This study has demonstrated the potential of ANN and SVR models in predicting aerodynamic coefficients in the Aerospace field. Building upon the outcomes of this study, future research can focus on modeling complex phenomena, such as aerodynamic stall (Tondji et al., 2022b) and ground effect during low-altitude flight phases. Additionally, exploring the application of AI techniques to capture the dynamic motion of aircraft across different flight conditions is promising. The findings of this research provide a foundation for advancing our understanding of aircraft dynamics through the use of advanced AI methodologies.

In conclusion, the application of AI techniques in Aerospace Engineering has given promising results. This research paves the way for a future in which flight systems can benefit even more from AI integration, making flight operations better and more efficient.

CHAPITRE 3

PREDICTING LATERAL DYNAMICS OF CRJ700 USING MULTILAYER PERCEPTRON AND SUPPORT VECTOR REGRESSION

Yvan Tondji [a], Georges Ghazi [b], Ruxandra Mihaela Botez [c]

[a,b, c] Department of System Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3

Paper submitted for publication, November 2023

Résumé

Dans le domaine en rapide évolution de l'aérodynamique, la prédiction précise des coefficients aérodynamiques latéraux est essentielle pour améliorer le contrôle de vol et assurer la sécurité. Cette étude utilise des techniques d'apprentissage automatique, spécifiquement les perceptrons multicouches « Multilayer Perceptron (MLP) » et la régression par vecteurs de support « Support Vector Regression (SVR) », pour prédire les coefficients aérodynamiques latéraux de l'avion Bombardier CRJ-700, un jet régional largement utilisé dans l'industrie aéronautique. Les données expérimentales pour cette étude ont été collectées en utilisant le simulateur de vol virtuel pour la recherche CRJ-700 de niveau D (VRESIM), qui possède la qualification la plus élevée pour les modèles de dynamique de vol et de propulsion selon la « Federal Aviation Administration » (FAA). Le VRESIM a été conçu par CAE et Bombardier. L'ensemble de données comprend une large gamme de conditions de vol et les coefficients aérodynamiques latéraux correspondants, servant de fondement à la formation et aux tests des deux modèles d'apprentissage automatique. De plus, l'Optimisation Bayésienne (BO) a été utilisée pour affiner nos modèles. Les modèles ont été validés en comparant un ensemble de paramètres de vol prédits avec des données expérimentales obtenues à partir du VRESIM, basés sur les critères de tolérance de la FAA. Cela assure la fiabilité et l'applicabilité des modèles dans des scénarios du monde réel. Nous avons obtenu de très bons résultats, démontrant le potentiel du MLP et du SVR dans la prédiction précise de la dynamique latérale des aéronefs, offrant une voie prometteuse pour l'amélioration des systèmes de contrôle de vol.

Abstract

In the rapidly evolving field of aerodynamics, accurate prediction of lateral aerodynamic coefficients is essential for enhancing flight control and ensuring safety. This study uses machine learning techniques, specifically Multilayer Perceptron (MLP) and Support Vector Regression (SVR), to predict the lateral aerodynamic coefficients of the Bombardier CRJ-700 aircraft, a regional jet widely used in the aviation industry. Experimental data for this study were collected using the Level-D CRJ-700 Virtual Research Simulator (VRESIM), which has the highest qualification for flight dynamics and propulsion models according to the Federal Aviation Administration (FAA). The VRESIM was designed by CAE and Bombardier. The dataset comprises a wide range of flight conditions and corresponding lateral aerodynamic coefficients, serving as the foundation for training and testing both machine learning models. In addition, Bayesian Optimization (BO) was used to fine-tune our models. The models were validated by comparing a set of predicted flight parameters with experimental data obtained from the VRESIM, based on FAA tolerance criteria (FAA, 1991; ICAO, 2016). This ensured the reliability and applicability of the models in real-world scenarios. We obtained very good results, demonstrating the potential of both MLP and SVR in accurately predicting aircraft lateral dynamics, offering a promising avenue for improving flight control systems.

Introduction

3.1 Introduction

In recent years, the aerospace industry has greatly benefited from advancements in computational power and Artificial Intelligence (AI) techniques. In particular, machine learning models such as the Multilayer Perceptron (MLP) and Support Vector Regression (SVR) have been widely used to predict various aircraft dynamics characteristics, such as engine performance (Andrianantara, Ghazi & Botez, 2021; Zaag & Botez, 2017) and longitudinal aerodynamic coefficients (Ghazi et al., 2017; Tondji, Ghazi & Botez, 2023; Tondji et al., 2022a; Tondji et al., 2022b), which are essential for understanding the forces and moments acting on an aircraft during maneuvers. Accurate modeling of lateral dynamics plays a fundamental role in the design and testing of aircraft, as it allows the comprehensive analysis and prediction of aircraft behavior under various flight conditions, paving the way for safer and more efficient aircraft designs and controls systems (Abdolhosseini et al., 2013; Botez et al., 2015; Moncayo et al., 2012; Shao et al., 2019; Tchatchueng Kammegne et al., 2018; Wang & Zhang, 2018).

Traditional methods for determining lateral aerodynamic coefficients can be grouped into three main categories: empirical or semi-empirical methods, Computational Fluid Dynamics (CFD), and Wind Tunnel Testing. Empirical methods are based on well-established aerodynamic theories and use a series of equations derived from experimental data to estimate aerodynamic coefficients. Examples include the USAF Stability and Control DATCOM (Data Compendium), a renowned semi-empirical method (Blake, 1985). An advancement in empirical methods is the FDerivatives code developed at our LARCASE laboratory, which is also based on DATCOM code methodology with aerodynamic improvements theories. This code and its associated methodology were applied to the UAS-S4 and UAS-S45 from Hydra Technologies (Kuitche & Botez, 2019), the military X-31 aircraft (Anton et al., 2011), and the business aircraft Hawker 800 XP (Anton et al., 2010). These methods offer simplicity, low computational cost, and broad applicability. However, a limitation is their potential lack of precision for detailed flight simulation because they depend on idealized assumptions and generalized datasets, which may not represent the specific aircraft configuration under study.

CFD methods use Navier-Stokes equations to simulate the fluid flow around an object, enabling the determination of aerodynamic coefficients (Anderson, 2010). These methods provide a detailed visualization of the aerodynamic phenomena and can accurately predict the coefficients for a wide range of flight conditions. For example, Islas-Narvaez et al. investigated the use of CFD methods for modeling the lateral aerodynamic coefficients of an aircraft (Islas-Narvaez, Ituna-Yudonago, Ramos-Velasco, Vega-Navarrete & Garcia-Salazar, 2022). Their approach involved using a 3D steady state model based on the equations of mass, momentum, and turbulence to simulate flow around a tail-sitter aircraft. The Spalart Allmaras model was chosen to analyze the turbulent behavior of the airflow. The mathematical model was validated with experimental data, and the results showed good agreement. The CFD simulations revealed that the proposed tail-sitter aircraft had improved lateral aerodynamic performance compared to a baseline aircraft. The main improvements were a reduction in the sideslip angle at which the aircraft became unstable, and an increase in the maximum sideslip angle that the aircraft could maintain. The results also showed that the proposed aircraft had a lower lateral drag coefficient than the baseline aircraft.

Unfortunately, CFD requires significant computational resources and technical expertise, limiting its usability for large-scale or real-time applications. Moreover, the precision of CFD calculations is often constrained by the assumptions inherent in the mathematical equations used by their algorithms, especially by the linearization around steady flight conditions.

Wind tunnel testing is a direct experimental approach in which scaled aircraft models are tested in a controlled environment (Hansman & Craig, 1987; A. Koreanschi, Sugar Gabor & Botez, 2016b). This method has been employed to determine the lateral aerodynamic coefficients for various types of aircraft. Jones et al., for instance, conducted wind tunnel testing of a tail-sitter UAV to obtain its lateral aerodynamic coefficients (Mei, Zhou, Su, Shan & Wang, 2023). The UAV was tested in a low-speed and open-jet wind tunnel. Pressure measurements were made on the UAV surface using a total pressure probe and a static pressure probe. Flow visualization was also performed using “smoke”. The results of the wind tunnel testing were then used to calculate the lateral aerodynamic coefficients of the UAV. The coefficients were calculated for a range of angles of attack, sideslip angles, and Mach numbers, and the results showed that the UAV had good lateral dynamics performance. The UAV was stable in both the longitudinal and lateral axes and had a good roll rate. While wind tunnel methods can provide accurate and reliable results, they can be time-consuming, costly, and may not fully represent real-world flight conditions due to scaling effects and the steady nature of the testing environment.

Machine Learning, with its ability to learn complex patterns and make predictions from large datasets, offers a promising alternative to traditional methods (Perhinschi et al., 2002; Perhinschi et al., 2007). MLP and SVR, as supervised learning models, can be trained on historical flight data to accurately predict lateral aerodynamic coefficients. In a previous study, we proposed a novel methodology for predicting the longitudinal aerodynamic coefficients of an aircraft using Support Vector Regression (SVR) (Tondji et al., 2023). This method exhibited remarkable results, with average relative errors of less than 1% for the lift and drag aerodynamic coefficients, and an average error of 5.74% for the pitching moment aerodynamic coefficients. The methodology presented in this paper is an extension of the aforementioned study and focuses on predicting lateral aerodynamic coefficients using Machine learning (MLP and SVR).

To apply machine learning models effectively in this field, several challenges need to be addressed. Firstly, the complex and non-linear nature of aerodynamic coefficients, which varies according to different factors such as airspeed, altitude, and aircraft configuration, present a challenging task for machine learning models. Effectively capturing these nuances is essential for accurate predictions. Secondly, the performance of the models is strongly influenced by their hyperparameters, such as the number of hidden layers in the MLP and the kernel function in the SVR. While selecting the right hyperparameters is vital, finding their optimal values through this high-dimensional search space can be both time-consuming and computationally demanding.

The objective of this study is to optimize the prediction of lateral aerodynamic coefficients of the Bombardier CRJ-700 regional jet using MLP and SVR models, thus employing BO for hyperparameters tuning. The optimization considered various critical parameters, such as the MLP's activation function, its number of hidden layers and neurons per layer, and the SVR's kernel function, kernel scale or polynomial order, and its regularization parameter. Data for model training and validation was obtained from a Level-D Virtual Research Equipment Simulator (VRESIM) for the Bombardier CRJ-700 (see Figure 3.1).

Figure 3.1 Validating the proposed methodology with a Bombardier CRJ-700 Virtual Research Simulator (VRESIM)

Furthermore, the study seeks to integrate these optimized AI models into a CRJ-700 simulation platform built in MATLAB - SIMULINK software, replicating flight test scenarios carried out using the Virtual Research Environment for Simulation (VRESIM). This integration demands careful consideration of multiple components, including aerodynamic models, equations of motion, sensors and pilot inputs. The performance of integrated models will be assessed based on their ability to predict flight parameters within the tolerance criteria set by the Federal Aviation Administration (FAA). In addition to the technical aspects of model integration, our study aligns with industry-wide initiatives to modernize the air transportation system through the implementation of innovative aviation information systems. For instance, Mozdzanowska et al. (2008), discusses the importance of overcoming stakeholder barriers and streamlining safety approval for new technologies. These considerations are critical for the successful adaptation of our AI models and other such systems into operational use.

The scope of this research includes the design of the optimization process, implementation of the optimized models, and their integration into the simulation platform. In addition, the study will provide a comprehensive analysis of the results, emphasizing the comparison of model predictions with flight data from the VRESIM and evaluating the model performance in terms of the FAA tolerances.

3.2 Methodology

3.2.1 Data Collection

The envelope considered in this study to perform the flight tests is the four-dimensional space in which an aircraft can operate efficiently and safely, defined by its limits of altitude, airspeed, weight, and center of gravity location. For this study, the flight envelope was characterized by varying the center of gravity locations, rangig from the "Take-off, Landing & Approach lower limit" (14% of the chord) to the "Take-off, Landing & Approach upper limit" (35% of the chord), aircraft weight ranging from the "Minimum Flight Weight" (42,000 lb) to the "Maximum Landing Weight" (67,000 lb), and indicated airspeeds ranging from 260 to 330 knots, as well as the pressure altitude varying from 5000 to 35,000 ft. This wide range of flight conditions provided a comprehensive and robust dataset for training and validating both MLP and SVR models. By testing the aircraft performance within this flight envelope, we ensure its safe and effective operation in various situations, including takeoff, landing, and high-altitude cruising. The flight scenarios considered in the weight/center of gravity flight envelope, and in the altitude/speed flight envelope are shown in Figure 3.2(a) and Figure 3.2(b), respectively. There were 49 flight conditions considered in the weight/center of gravity flight envelope, and 48 in the altitude/speed flight envelope, resulting in a total of 2,352 flight conditions.

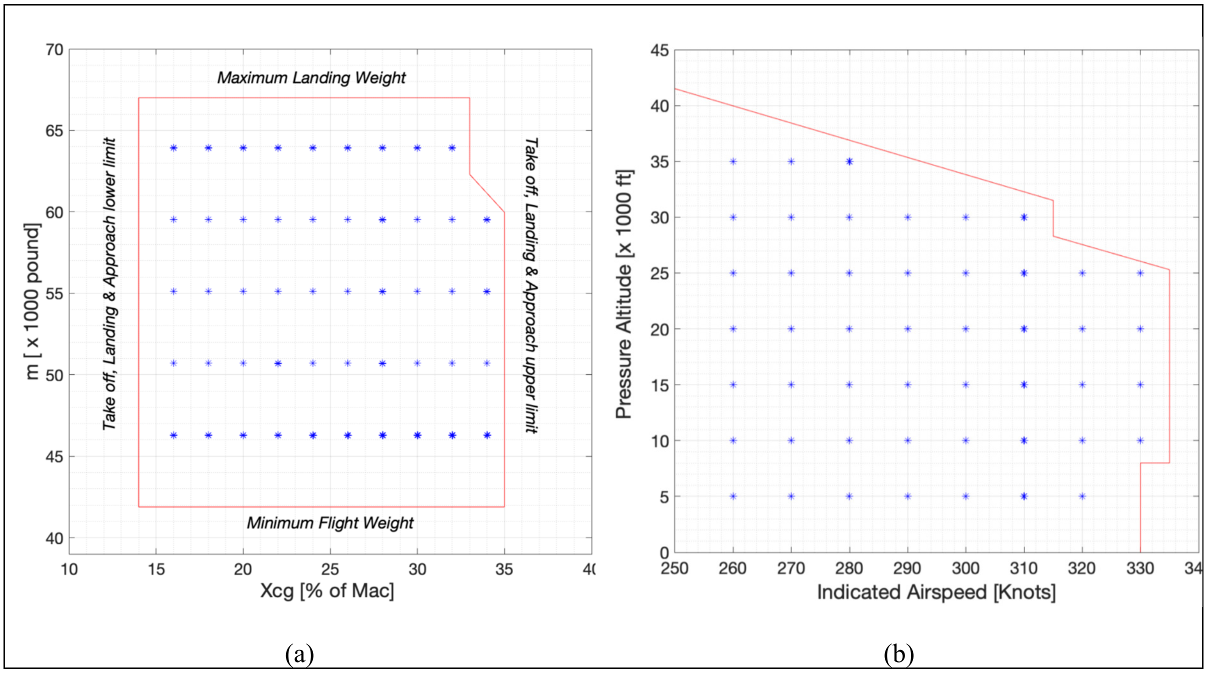


Figure 3.2 Weight/Center of gravity location flight envelope (a), and Altitude/Speed flight envelope (b) of the Bombardier CRJ-700 VRESIM

Pre-coded scripts were used to automatically perform all flight tests. These flight tests aimed to observe the lateral modes (roll, Dutch Roll, and spiral modes) when the aircraft was subjected to disturbance. The following procedure was used for each flight condition to induce lateral modes. First, the aircraft was trimmed and maintained in a straight and level flight with a constant altitude, airspeed, weight, and angle of attack. At predetermined times, symmetric impulses were applied to the rudder (at the 5th second) and ailerons (at the 23rd second) to stimulate the roll, Dutch Roll and spiral modes and allow the aircraft to oscillate freely until the 60th second. During each test, various flight parameters, such as altitude, Mach number, true airspeed, angular velocities, accelerations, engine thrust, and control surface deflections were recorded at a sampling rate of 30 Hz.

Typical examples of data recorded from the VRESIM for a given flight test are illustrated in Figure 3.3. The flight test was carried out at an altitude $h = 20{,}000$ ft, with an indicated air speed $IAS = 320$ kn. The weight of the aircraft was set at $m = 55{,}120$ lbs and the center of gravity at $x_{cg} = 16\%$.

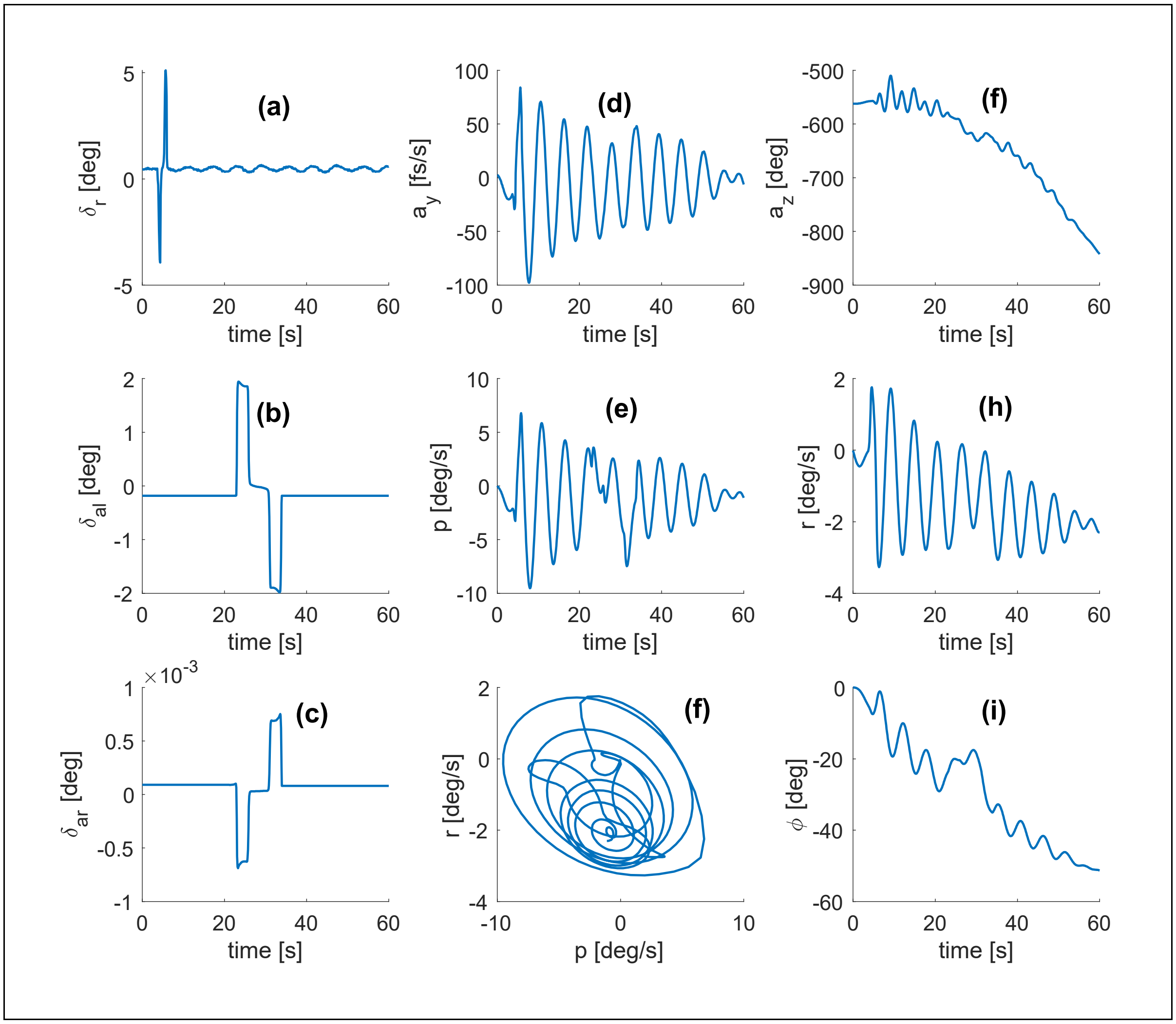


Figure 3.3 Example of flight parameters recorded for flight tests at $m = 55{,}120\ lbs$ | $x_{cg} = 16\%$ | h = $20{,}000\ ft$ | $IAS = 320$ kts

After applying the command on the rudder (Figure 3.3.(a)) at 5 seconds, the aircraft experienced oscillations in both yawing and rolling moments, leading to the Dutch Roll mode. Figure 3.3.(e) and Figure 3.3.(h) illustrate these oscillations for rolling and yawing motions, respectively. As shown in Figure 3.3.(f), these oscillations followed a cyclical pattern and decreased in amplitude over time. A command to the ailerons (Figure 3.3.(b) and Figure 3.3.(c)) was next applied at 23 seconds, resulting in a rapid damped motion observed on the roll rate variation (Figure 3.3.(e)), which characterizes the roll mode.

3.2.2 Data Pre-Processing

3.2.2.1 Definition of Model's Inputs and Outputs

Parameters selection is a critical step in the data preprocessing phase, as it helps to identify the most relevant input variables for a model, thereby reducing noise, computational cost, and enhancing model performance. The input parameters considered in this study included aircraft operating conditions such as weight m, center of gravity location x_{cg}, altitude h, and Mach number M, as well as lateral flight parameters such as roll and yaw rates q, r, roll and yaw accelerations $\dot{q}$ $\dot{r}$, lateral accelerations and moment of inertia a_y, I_{yy}, sideslip angle β, roll angle ϕ, and control surface deflections such as aileron deflection δ_a, and rudder deflection δ_r. Some of them are linearly correlated with the aerodynamic coefficients, having an absolute value of the Pearson correlation coefficient greater than 0.5. This coefficient measures the linear relationship between potential input variables and target outputs (Asuero, Sayago & González, 2006), with values ranging from -1 to 1. The correlation matrix (in absolute value) of input variables relative to target outputs (CY_s, Cl_s or Cn_s) is shown in Figure 3.4. A value of -1 indicates a perfect negative linear correlation, while a value of 1 indicates a perfect positive linear correlation. Conversely, a value of 0 indicates no linear correlation. However, even if the linear correlation is weak, some variables might exhibit quadratic or higher-order correlations with aerodynamic coefficients. The following input vector was considered:

$$x = \{x_{cg}, m, h, M, \beta, \phi, r, p, \dot{r}, \dot{p}\ , a_y, I_{yy}\delta_a, \delta_r\} \tag{3.1}$$

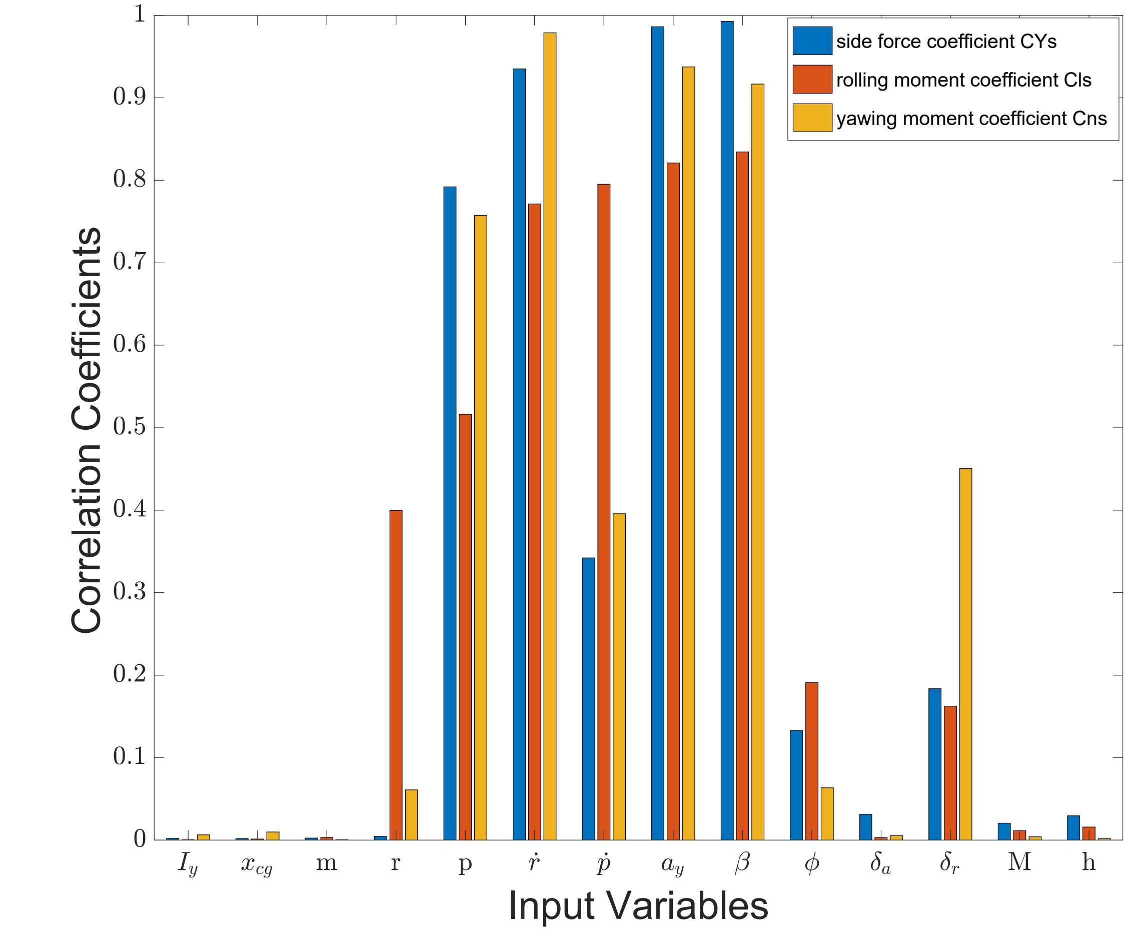


Figure 3.4 Linear correlation of input variables with output variables using the Pearson correlation coefficient

The output variables, which comprise rolling and yawing moment coefficients Cl_s and Cn_s as well as side force coefficient CY_s, are key parameters that describe an aircraft response to lateral inputs or disturbances. The three lateral aerodynamic coefficients in the stability axis system were determined from recorded flight parameters using the following equations (Ghazi et al., 2017):

$$CY_s = CY_b \quad (3.2)$$

$$Cl_s = Cl_b \cos(\alpha) + Cn_b \sin(\alpha) \quad (3.3)$$

$$Cn_s = Cn_b \cos(\alpha) - Cl_b \sin(\alpha) \quad (3.4)$$

In Equation (3.2) to Equation (3.3), CY_b, Cl_b and Cn_b represent the side force, rolling moment and yawing moment coefficients, respectively, expressed in the aircraft body axis system. They were computed using the following equations (Ghazi et al., 2017):

$$CY_b = \frac{ma_y - T_y}{1/2\rho V_T^2 S_w} \tag{3.5}$$

$$Cl_b = \frac{I_{xx}\dot{p} - I_{xz}\dot{r} + qr(I_{zz} - I_{yy}) - I_{xz}pq}{1/2\rho V_T^2 S_w b_w} \tag{3.6}$$

$$Cn_b = \frac{-I_{xz}\dot{p} + I_{zz}r + pq\left(I_{yy} - I_{xx}\right) + I_{xz}qr}{1/2\rho V_T^2 S_w b_w} \tag{3.7}$$

where ρ denotes the air density, T_y represents the lateral component of engine thrust, I_{xx}, I_{yy} and I_{zz} denote the aircraft moment of inertia about the longitudinal, lateral, and vertical axes, respectively, while I_{xz} represents the moment of inertia about the xz-plane. Additionally, S_w and b_w are the wing reference area and span, respectively, while a_y, p, q and r denote the aircraft lateral acceleration, roll, pitch, and yaw rates, respectively.

3.2.2.2 Data Normalization

To ensure the quality and reliability of the data used to train the machine learning models, data normalization is an essential preprocessing step. This process involves scaling the parameters to a uniform range, typically [0, 1] or [-1, 1], ensuring each parameter has an equivalent impact on the model output calculations. This is particularly important given that the input parameters in this study have diverse units or scales. Normalizing the data helps preventing parameters with larger values from dominating the model and skewing the results.

For this study, we employed Z-score normalization (Al-Faiz, Ibrahim & Hadi, 2019), which standardizes the variables by subtracting the mean and then dividing them by the standard deviation. This method ensures all variables have a mean value of 0 and a standard deviation of 1. By applying this normalization process to the input data, we were able to effectively eliminate any biases that may have resulted from the use of different units or scales.

3.2.3 Data Management

In the context of developing machine learning models, particularly for models such as the Multi-Layer Perceptron (MLP) or Support Vector Regression (SVR), the dataset must be partitioned into training, testing, and validation sets. This procedure is essential to assess the model performance and avoid overfitting. The training set is used to optimize the model weights and biases, while the testing set evaluates the model performance using data that was not used for training. The testing set is crucial in adjusting the model hyperparameters. Once the model is fine-tuned, the validation set is used to provide an unbiased estimate of the model performance on new and unseen data.

It should be noted that in this study, AI models were designed to predict lateral aerodynamic coefficients based on flight test data obtained from a Level-D CRJ700 VRESIM. If this methodology were to be applied using data from aircraft real flight tests, it would be essential to train the model with minimal data to reduce the number of flight tests needed. Given that reality, a strategy was employed to reduce the amount of data required for effective model training. Figure 3.5 illustrates the flight envelope in terms of weight and center of gravity position used for training and testing the model. The selected points for training and testing are highlighted in red. It was assumed that a well-trained model should consider the interpolation between two flight conditions with different center of gravity positions (with the same mass) and between two masses if the center of gravity positions were identical. Thus, only 9 of the 49 available flight conditions were used for training and testing, distributed as shown in Figure 3.5.

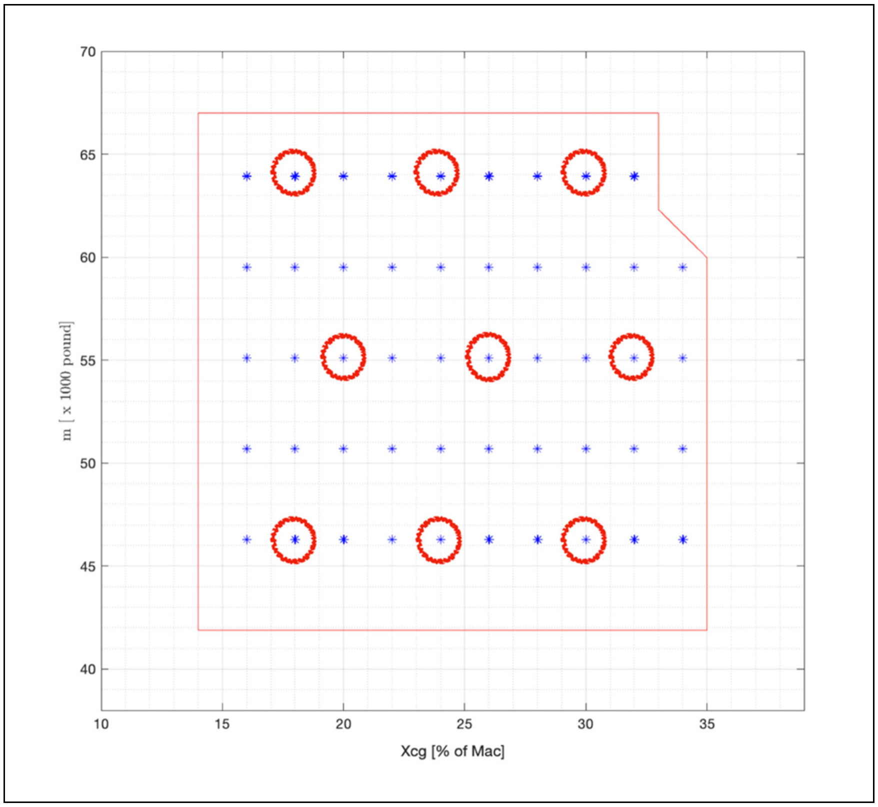


Figure 3.5 Flight conditions selected for model training and testing on a Weight versus Center of gravity position flight envelope of the Bombardier CRJ-700 VRESIM

Each circled point in Figure 3.5 corresponds to 48 distinct altitude and airspeed flight conditions. However, only 24 of these conditions were selected for model training and testing, as highlighted in red in Figure 3.6. This means that from the 2352 flight cases available in the database, only $9 \times 24 = 216$ were used for model training and testing, representing a 9.2% ratio. The remaining 90.8% of the data was allocated for model validation.

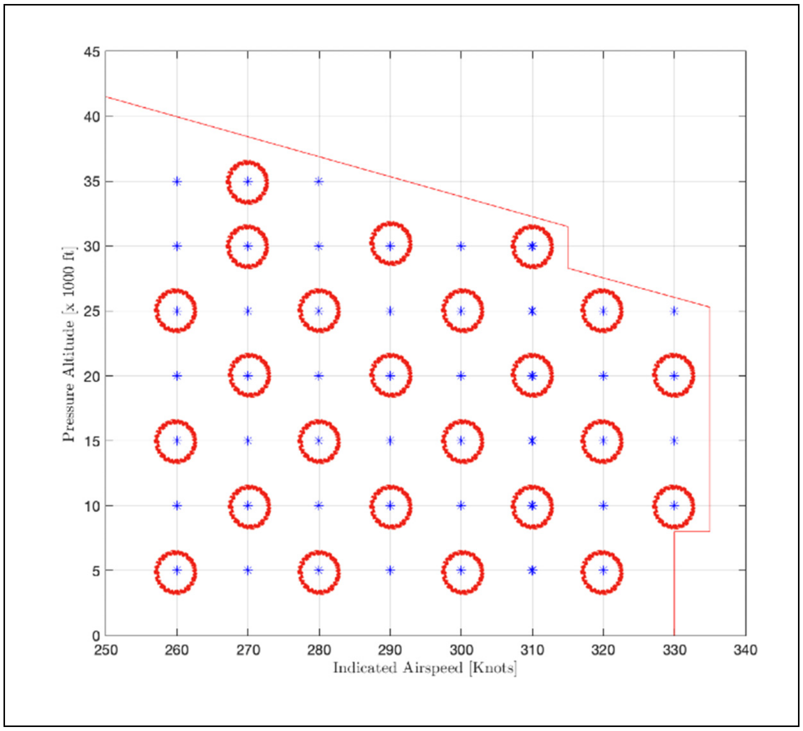


Figure 3.6 Flight conditions selected for model training and testing on the Pressure Altitude versus IAS flight envelope flight envelope of the Bombardier CRJ-700 VRESIM

To separate the data into training and test sets, we used the k-fold cross-validation technique (Stone, 1974). This approach is widely used for model evaluation and selection in machine learning. The K-fold cross-validation helps to provide a more reliable and stable performance estimate for a model with unseen data, reducing the potential bias and variance associated with a single train-test split. For this purpose, the data from 216 flight cases were randomly divided into k −groups or folds of approximately equal dimensions. The model was then trained k times, each time using a different fold as the test set and the remaining $k-1$ folds as the training set. Each element in the dataset is designated to a particular fold and remains in that fold for the duration of one complete training. This process ensures that each element in the dataset is used exactly once as part of the test set and $(k-1)$ times in the training set. The overall model performance is then evaluated as the mean of the errors resulting from the k training.

Following recommendations in (Haykin, 1998) for applying k-fold cross-validation to regression model training, $k = 5$ is chosen for the number of folds in the cross-validation procedure. This number indicates that the dataset was randomly divided into five groups, with each group being used as the test set once, and the other four groups as training sets.

3.2.4 Multi-Layer Perceptron Theory

A MLP is a feedforward artificial neural network consisting of multiple layers of nodes (neurons) connected by weighted edges (Haykin, 1998). MLPs are widely used for supervised learning tasks, such as regression and classification. The architecture of an MLP typically includes an Input layer, one or more Hidden layers, and an Output layer. The Input layer contains nodes that represent the input parameters of the dataset, with the number of nodes corresponding to the number of input parameters (Haykin, 1998). Hidden layers consist of multiple nodes that perform non-linear transformations of the input data. The number of hidden layers and the number of nodes in each hidden layer are user-defined hyperparameters that affect the complexity and performance of the MLP. The Output layer contains nodes representing the predicted output values, where the number of nodes correspond to the number of output variables.

Figure 3.7 shows a graphical representation of the MLP, with the Input layer defined by the input vector in Equation (3.1) and the Output layer representing the lateral aerodynamic coefficients.

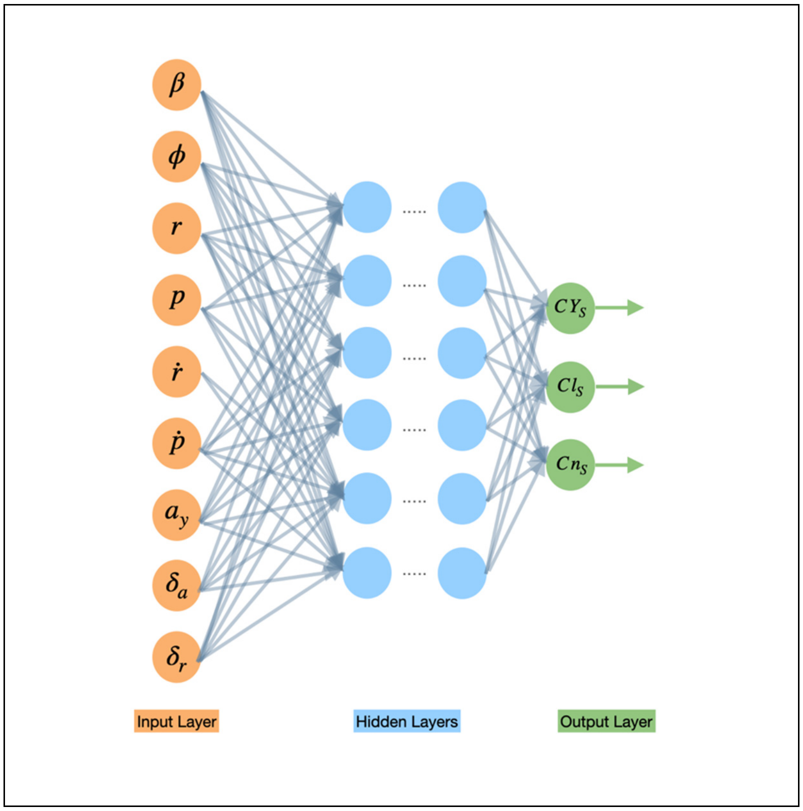


Figure 3.7 MultiLayer Perceptron for predicting the lateral aerodynamic coefficients of the bombardier CRJ-700 VRESIM from flight parameters

Each hidden neuron in Figure 3.7 receives inputs from the previous layer and applies a weighted sum to the inputs, adds a bias term, and then processes the result through an activation function (Linse & Stengel, 1993) to produce an output. The output feeds into to the next layer in the network. Both weights and bias terms are adjustable parameters in the network, with the weight representing the strength of the connections between neurons and the bias allowing the neuron to shift its activation function horizontally and improve flexibility in fitting the data. The activation function introduces non-linearity into the network and determines the output of a neuron based on the weighted sum of its inputs and the bias term. It plays a crucial role in enabling neural networks to learn complex and non-linear patterns in the data.

To learn patterns, MLP models adjust the weights of the connections between nodes to minimize a loss function, which measures the discrepancy between the predicted outputs of the model and the actual target values. The Mean Squared Error (MSE) is commonly used as a loss function for regression problems, as with our MLP. For a given set of n data points, the MSE of an MLP is calculated according to Equation (3.8):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[\hat{o}_i(w) - o_i]^2 \tag{3.8}$$

where o_i is the ith output vector associated with the ith input vector in the dataset, and $\hat{o}_i$ is the ith output vector predicted by the MLP model.

In MLP models, the learning process uses backpropagation, an algorithm that calculates the gradients of the loss function with respect to the network weights. After determining the gradients, they are used to update the weights of the network through an optimization algorithm (a training function). This iterative process is repeated until the loss function reaches a minimum value. In this study, the optimization of the MLP hyperparameters was done using the Bayesian Optimization (BO) technique and is presented in section 3.2.6.

3.2.5 Support Vector Regression Theory

Support Vector Regression (SVR) is a powerful supervised learning algorithm used for regression tasks. It is based on the principles of Support Vector Machines (SVM), which are primarily used for classification (Drucker et al., 1996). The main goal of SVR is to find a function $f(x)$ that can accurately predict the output values o for a given set of input data points x, while minimizing the generalization error.

$$f(x) = \mathrm{w}^T\phi\ (x) + b \tag{3.9}$$

In Equation (3.9), the variables w and b represent the weights vector and the bias term, respectively. The function $\phi\ (x)$ is known as a mapping function or Kernel function, which transforms the input data vector x from its original space, where data cannot be separated linearly, into a higher-dimensional feature space where linear separation becomes possible. The main objective of Support Vector Regression (SVR) is to locate a hyperplane that best fits the training data while maintaining a balance between minimizing the error and achieving the largest margin possible. This objective is achieved by introducing a margin of tolerance ε around the hyperplane, which allows for a certain degree of error in predictions. Any data point (x_i, o_i) within this ε-insensitive tube is considered to be correctly predicted, regardless of its actual distance from the hyperplane, and it should satisfy the following inequality:

$$|o_i - (\mathrm{w}^T \phi\ (x_i) + b)| \leq\ \varepsilon, \quad i = 1, \ldots, n \tag{3.10}$$

From an optimization perspective, the SVR aims to minimize the optimization problem given by Equation (3.11) (Vapnik, 1995):

$$J(\mathrm{w}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{3.11}$$

subject to the following constraints:

$$\forall\ i = 1, \ldots, n \begin{cases} o_i - \mathrm{w}^T \phi\ (x_i) - b \leq\ \varepsilon + \xi_i \\ \mathrm{w}^T \phi\ (x_i) + b - o_i \leq\ \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{3.12}$$

The term $\|w\|^2$ in Equation (3.11) is introduced as a means to find a simpler model with a larger margin, which can lead to better generalization performance. The slack variables ξ_i and ξ_i^* allow for some degree of error in the predictions and measure the distance between data points that are outside the ε-insensitive tube and the tube margin, representing the prediction error. The regularization parameter C determines the trade-off between minimizing the margin (by reducing $\|w\|^2$) and minimizing the prediction error (by reducing $\sum_{i=1}^{n}(\xi_i + \xi_i^*)$). A higher value of C puts more emphasis on minimizing the prediction error, while a smaller value of C prioritizes finding a larger margin.

The optimization problem in Equation (3.11) is solved by transforming the problem into its “dual form” and then solving this dual problem using techniques such as Quadratic Programming (QP) or Sequential Minimal Optimization (SMO). The loss function used in this study is the ε-insensitive loss function, which, for a given set of n data points, is defined as (Vapnik, 1995):

$$L_{\varepsilon} = \sum_{i=1}^{n} \max(0, \ |o_i - f(x_i)| - \varepsilon) \tag{3.13}$$

In the same way as for the MLP, the optimization of the SVR hyperparameter was done using the Bayesian Optimization (BO) technique and is presented in the next section 3.2.6.

3.2.6 Choice of Solver and Hyperparameters Optimization using Bayesian Optimization

Bayesian Optimization (BO) is a global optimization technique designed for optimizing expensive-to-evaluate objective functions (Shahriari et al., 2016). It is particularly well-suited for hyperparameter tuning in machine learning models, such as MLP and SVR. The underlying theory of BO is based on probabilistic modeling and decision theory, making it a highly effective method for exploring complex search spaces.

The process of BO consists of two main components: a probabilistic surrogate model and an acquisition function. The surrogate model is used to model the expensive-to-evaluate objective function by capturing the relationship between the input parameters and the objective function value. Typically, a Gaussian Process (GP) is used as the surrogate model due to its ability to provide a flexible, non-parametric representation of the function and its uncertainty estimates. The acquisition function is a key component of Bayesian Optimization, as it determines which points in the search space should be explored or exploited next. It balances exploration (searching for regions with high uncertainty) and exploitation (focusing on regions with high predicted performance). Common acquisition functions include Expected Improvement (EI)

(Jones, Schonlau & Welch, 1998), Probability of Improvement (PI) (Kushner, 1964), and Upper Confidence Bound (UCB) (Srinivas, Krause, Kakade & Seeger, 2009).

The Bayesian Optimization process iteratively updates the surrogate model based on the new function evaluations and optimizes the acquisition function to determine the next point to evaluate. This procedure is repeated until a stopping criterion is met, such as reaching a predefined number of iterations. The K-fold cross validation technique is applied on each optimization iteration.

3.2.6.1 Multi-Layer Perceptron Hyperparameters Fine-Tuning

In this study, the BO was employed for hyperparameter optimization of Multi-Layer Perceptrons (MLPs) to efficiently tune the model for better performance. The hyperparameters optimized using BO included the activation function, the number of hidden layers, and the number of neurons per hidden layer. The training function was also included as an optimization variable.

In the case of MLP, the training function is crucial in determining which algorithm to use to update weights and biases during the learning process. The choice of training function significantly impacts the convergence speed, the overall training time, and the model's ability to generalize. In our study, we consider three gradient-based optimization algorithms, listed in Table 3.1: TRAINBFG (BFG), TRAINLM (LM), and TRAINBR (BR).

Table 3.1 Training algorithms considered to train the MLP

Algorithm	**Description**
TRAINBFG	Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi-Newton
TRAINLM	Levenberg-Marquardt optimization.
TRAINBR	Bayesian regularization

TRAINLM is an efficient and fast algorithm that combines the strengths of gradient descent with the Gauss-Newton method, making it particularly suitable for small to medium-sized problems (Marquardt, 1963). However, it may require a significant amount of memory. *TRAINBR*, on the other hand, introduces a regularization term to control the complexity of the network and prevent overfitting (Dan Foresee & Hagan, 1997). This feature makes it particularly useful for problems with limited data or when the model needs to generalize well to unseen data. Lastly, *TRAINBFG* is a memory-efficient optimization algorithm that offers a good balance between convergence speed and memory requirements, making it suitable for a wide range of regression problems, especially those with a large number of weights and biases (Fletcher, 1970). All three training functions have proven their efficiency in similar studies, where longitudinal aerodynamic coefficients were estimated using MLPs (Tondji et al., 2022b).

Tondji et al. (2022b) demonstrated the effectiveness of using the Hyperbolic tangent (*tansig*), Logistic sigmoid (*logsig*), and Elliot sigmoid (*elliotsig*) functions for regression problems, particularly in aerodynamic coefficients modeling. These three activation functions are commonly used in neural networks to introduce non-linearity and enable the learning of complex relationships between input and output variables (LeCun, Bottou, Orr & Müller, 2012). The *tansig* function provides a smooth transition and good sensitivity around the origin, producing outputs in the range of -1 to 1 (Bishop, 2006). The *logsig* function ensures that the output is positive, ranging from 0 to 1 (Graupe, 2013), and the *elliotsig* function offers a computationally efficient alternative to the *tansig* function, with an output range between -1 and 1 (Elliott, 1998). All three functions have the advantage of having bounded outputs and being differentiable, which can help prevent extreme output values during training. While the *tansig* and *logsig* functions are based on exponential functions and may require more computational resources, the *elliotsig* function, due to it simpler mathematical structure, enables faster computation (Elliott, 1998). The formulas for *tansig*, *logsig* and *elliotsig* activation functions are given in Equation (3.14), to Equation (3.16):

$$y(a) = \frac{2}{(1 + \exp(-2 * a))} - 1 \tag{3.14}$$

$$\mathrm{y(a)} = \frac{1}{1 + \exp\,(-\mathrm{a})} \tag{3.15}$$

$$y(a) = \frac{a}{(1 + |a|)} \tag{3.16}$$

The performance of a multilayer perceptron (MLP) is influenced by the number of hidden layers m and the number of neurons per hidden layer n. These two factors allow the network to capture complex and non-linear relationships between input and output variables (Cybenko, 1989). However, determining the optimal number of hidden layers and neurons for an MLP with 9 inputs and 3 outputs (lateral aerodynamic coefficients) presents a challenge. It is generally recommended to start with a small number of hidden layers (e.g., one or two) and progressively increase this number to enhance model performance. While increasing the number of layers and neurons can improve performance, there is a risk of overfitting, especially if the size of the training dataset is small. A study was conducted using neural networks to predict longitudinal aerodynamic coefficients of the Cessna Citation X in stall conditions (Tondji et al., 2022a). Three different neural networks were used to predict three different longitudinal aerodynamic coefficients. The study found that a network with $m = 5$ hidden layers and $n = 9$ neurons per layer was enough to model the lift coefficient, while the drag coefficient required a network of size $(m, n) = (2,14)$, and the pitching moment coefficient required an MLP of size $(m, n) = (4,12)$.

In this study, a single neural network is used to model the three lateral aerodynamic coefficients, which should have fewer complex dynamics than longitudinal aircraft coefficients in stall conditions. Based on this observation, a range of $m_{min} = 1$ to $m_{max} = 6$ hidden layers, and $n_{min} = 3 \; to \; n_{max} = 15$ neurons per layer was defined to obtain the optimal network structure. This resulted in $13^6 = \; 2.82 \times 10^6$ possible structures, which is relatively large. For the sake of simplicity, it was decided to reduce the number of possible structures to 78 by assuming that all hidden layers should have the same number of neurons. This assumption was made because changing the number of neurons from one layer to another did not significantly improve network performance (Heaton, 2008).

Regarding the hyperparameter optimization process, the maximum number of training epochs for the MLP was limited to 1000 to minimize the optimization time. The objective function used for this optimization was the mean squared error (MSE) calculated on the test set.

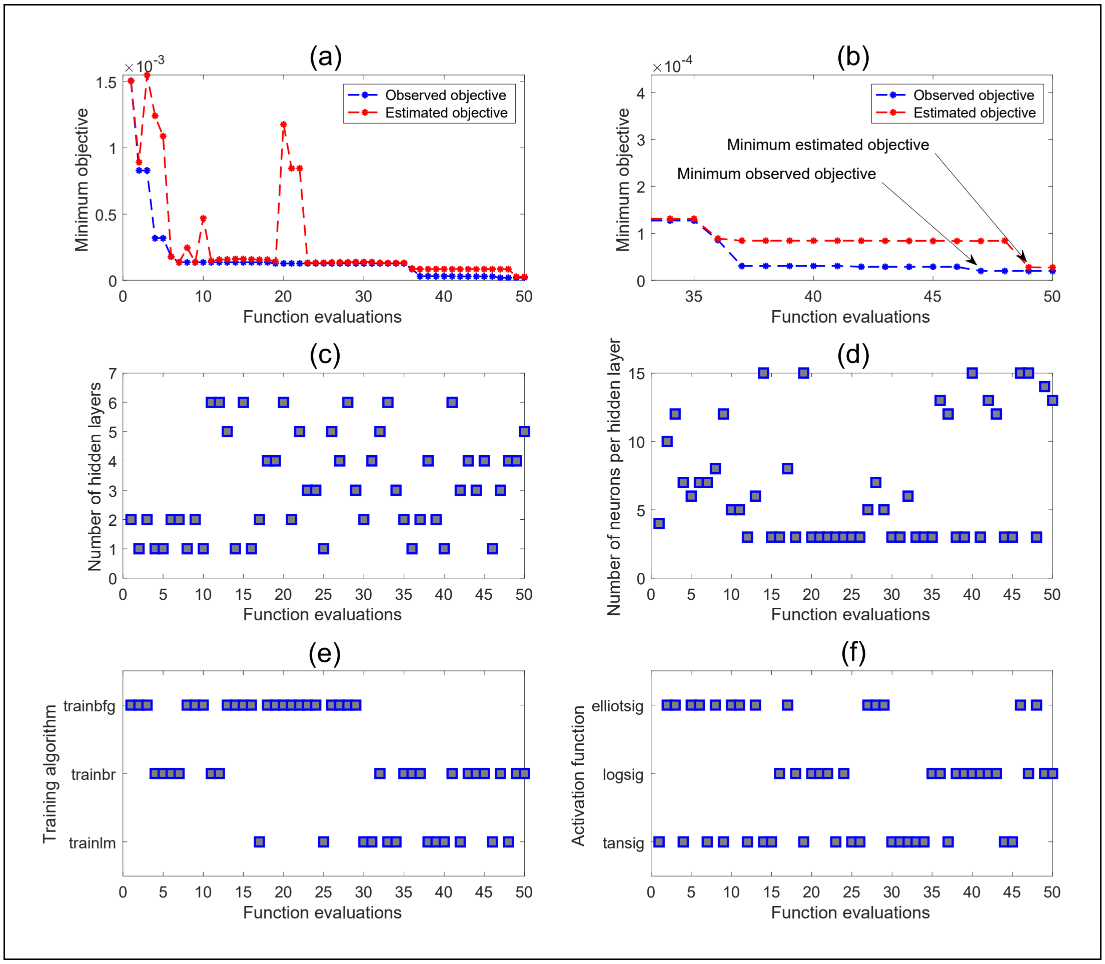


Figure 3.8 Bayesian Optimization convergence plots of an MLP

To demonstrate the effectiveness of the BO process, the convergence plots are presented in Figure 3.8. Figure 3.8(a) shows the improvement in both the observed objective function value (performance metric) and the estimated objective function value (estimated by the surrogate model) over the iterations. Both values decrease over time and are in close agreement, indicating that the probabilistic model accurately approximates the objective function. A close-up view of Figure 3.8(a) is presented in Figure 3.8(b), and reveals that the minimum observed objective of 1.96×10^{-5} was achieved at iteration 47, while the minimum estimated objective of 2.73×10^{-5} was achieved at iteration 49.

The search space for the four variables, namely the number of hidden layers, the number of neurons, the training algorithm, and the activation function, are presented in Figure 3.8(c), Figure 3.8(d), Figure 3.8(e), and Figure 3.8(f), respectively. It is worth noting that during the optimization process, all possible values for these four variables were tested, enabling the probabilistic model to gather the necessary data to choose subsequent evaluation points and ultimately converge to the best solution.

The minimum observed function value of 1.96×10^{-5} was obtained for the optimized parameter values: training function = *TRAINBR*, activation function = *logsig*, number of hidden layers $m = 4$, and number of neurons per hidden layer $n = 14$.

3.2.6.2 Support Vector Regression Hyperparameters Fine Tuning

The SVM models are designed in MATLAB software as Multi Input Single Output (MISO) models, which means that this type of model can only predict one aerodynamic coefficient at a time. Therefore, three different SVM models were designed, each model associated with one of the three output variables, i.e., $\{CY_s\}$, $\{Cl_s\}$, $\{Cn_s\}$. The SVR hyperparameter optimization is presented in this section only for the side force coefficient $\{CY_s\}$. However, the described procedure was also applied to the rolling and yawing moment coefficients, and the results are presented later in this section.

We first focused on selecting the appropriate solver for the optimization problem. This solver plays a critical role in determining the performance and convergence of the SVM model. Various solvers, such as the Decomposition Method (Chih-Chung Chang et al., 2000), the

Sequential Minimal Optimization (SMO) (Platt, 1998), and the Improved SMO Decomposition Algorithm (ISDA) (Kecman et al., 2005) employ different optimization techniques to find the optimal solution to the SVM problem, which can significantly impact the model's training time and overall accuracy. Previous research, shown in the study conducted by (Tondji et al., 2023), trained similar SVR models for determining the longitudinal aerodynamic coefficients of the CRJ700 aircraft. They found that the SMO solver demonstrated efficiency and effectiveness in achieving optimal results compared to other solvers. Thus, the SMO solver was selected for this research to ensure that the SVM model is optimized efficiently and effectively, leading to improved performance and accuracy.

The BO was then employed for SVR hyperparameter tunning. The hyperparameters optimized using the BO included the regularization parameter C, the kernel function $\phi(x)$, and the kernel-specific parameters. The precision ε was set to 10^{-3} for CY_s, as it was large enough to satisfy the FAA validation tolerances.

The choice of kernel function plays a crucial role in the SVR performance, as it determines how the input data is transformed in the feature space. Gaussian and polynomial kernels are popular choices for the SVR due to their abilities to handle non-linear data (Fan et al., 2005; Wang et al., 2015). The equations for Gaussian (also known as Radial Basis Function) and polynomial kernels are shown in Equation (3.17) and Equation (3.18), respectively, where σ and q are the kernel-specific parameters of the Gaussian kernel scale and the polynomial order, respectively.

$$K(x_i, x_j)_G = \exp\,(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}) \tag{3.17}$$

$$K(x_i, x_j)_q = \left(1 + x_i^T x_j\right)^q \tag{3.18}$$

The kernel scale σ plays a crucial role in determining the flexibility of the Gaussian function. Specifically, it controls the width of the function, and therefore determines how flexible or rigid a model is. If the kernel scale is too small, the model may overfit the data, while a kernel scale that is too large may cause underfitting. Similarly, the polynomial order q determines the complexity of the model and its ability to handle non-linear data. Increasing the polynomial order can improve the model flexibility, but it may also lead to overfitting if the order is too high.

We optimized the float values C and σ within default search ranges of $C \in [10^{-3}, 10^{3}]$ and $\sigma \in [10^{-3}, 10^{3}]$, respectively, and the integer value q within a search range of $q \in [1, 5]$. Due to the large number of parameters and the wide search range (especially for C and σ), we repeated the optimization process five times to check whether the results obtained were similar.

Figure 3.9 shows the convergence plot of the first optimization for the CY_s SVR model. The observed objective function and the estimated objective function values decrease over time, and are in close agreement, indicating that the probabilistic model accurately approximates the objective function. The polynomial order values of q was evaluated during iterations where the polynomial kernel function was used (refer to Figure 3.9(c) and Figure 3.9(e)), while kernel scale values of σ were tested during iterations where the Gaussian kernel function was used (refer to Figure 3.9(c) and Figure 3.9(d)). The plot in Figure 3.9(b) shows that the minimum observed objective of 1.13×10^{-5} was achieved at the 35th iteration.

A summary of the results obtained for the five optimizations in terms of minimum observed objectives ($obj_{\min obs}$) and optimized hyperparameters is presented in Table 3.2.

Table 3.2 Optimal hyperparameters obtained for five distinct BOs of the CY_s SVR model using SMO solver

	$\boldsymbol{obj_{\min obs}}$ $[\mathbf{10^{-5}}]$	**Regularization parameter** $\boldsymbol{C_{min\ obs}}$	**Kernel function**	**Kernel scale** $\boldsymbol{\sigma_{min\ obs}}$
$\mathbf{1^{st}\ BO}$	1.13	3.2175	Gaussian	12.51
$\mathbf{2^{nd}\ BO}$	0.31	46.53	Gaussian	5.57
$\mathbf{3^{rd}\ BO}$	0.36	7.25	Gaussian	7.01
$\mathbf{4^{th}\ BO}$	0.47	13.66	Gaussian	14.8
$\mathbf{5^{th}\ BO}$	0.43	16.22	Gaussian	10.06

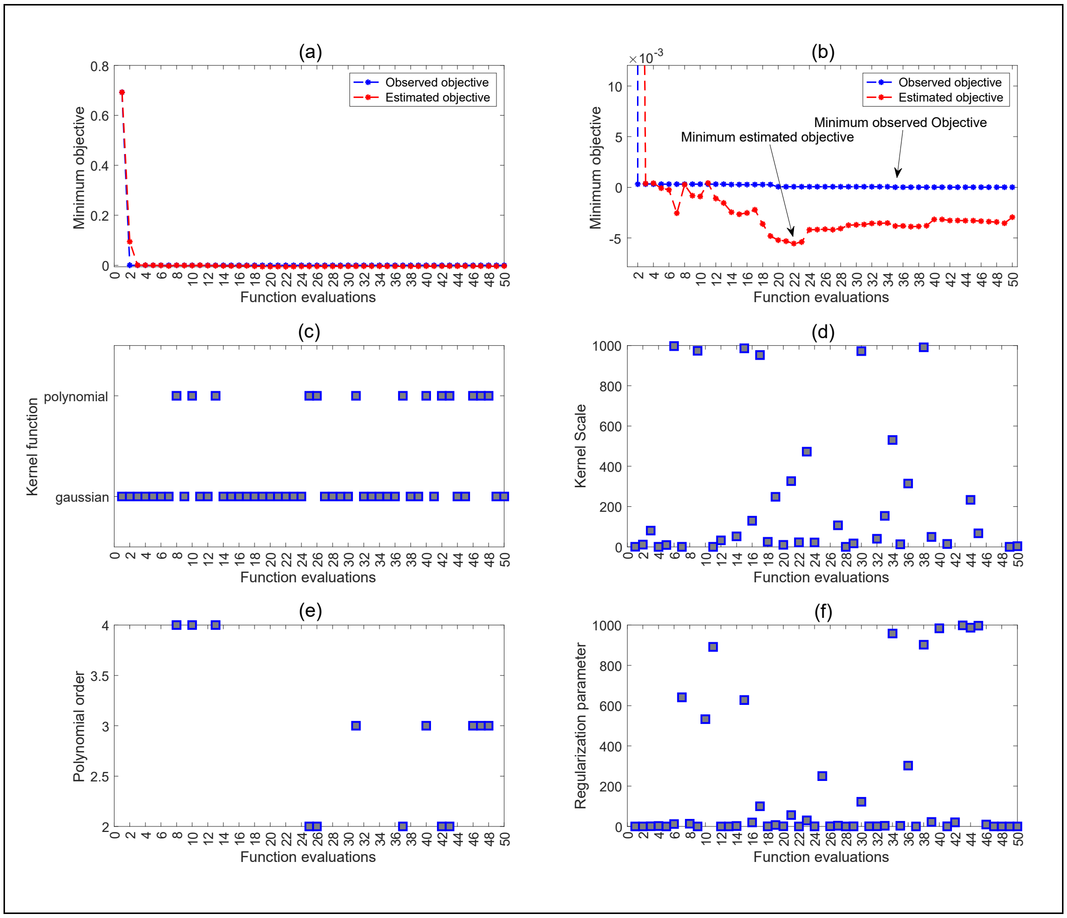


Figure 3.9 Bayesian Optimization convergence plot of the CY_s SVR model

As shown in Table 3.2, for the optimization runs, the Gaussian kernel function was chosen, indicating its suitability for the CY_s SVR model, while the polynomial kernel function and order q were found to be not applicable in this context. The optimal regularization parameter and the kernel scale, $C_{min\ obs}$ and $\sigma_{min\ obs}$, showed variability across different runs, suggesting that the model performance is sensitive to these hyperparameters. However, the values of $C_{min\ obs}$ ranged from 0 to 50, and the values of $\sigma_{min\ obs}$ ranged from 0 to 20 for all five runs. Thus, for a more refined optimization, the search field of $C_{min\ obs}$ was narrowed from $[10^{-3}, 10^{3}]$ to [0, 50], and the search field of $\sigma_{min\ obs}$ was narrowed from $[10^{-3}, 10^{3}]$ to [0, 20]. In addition, since the polynomial kernel function was found not to be applicable, the kernel function can be fixed to the Gaussian function. The convergence plot obtained from this refined search field optimization is shown in Figure 3.10.

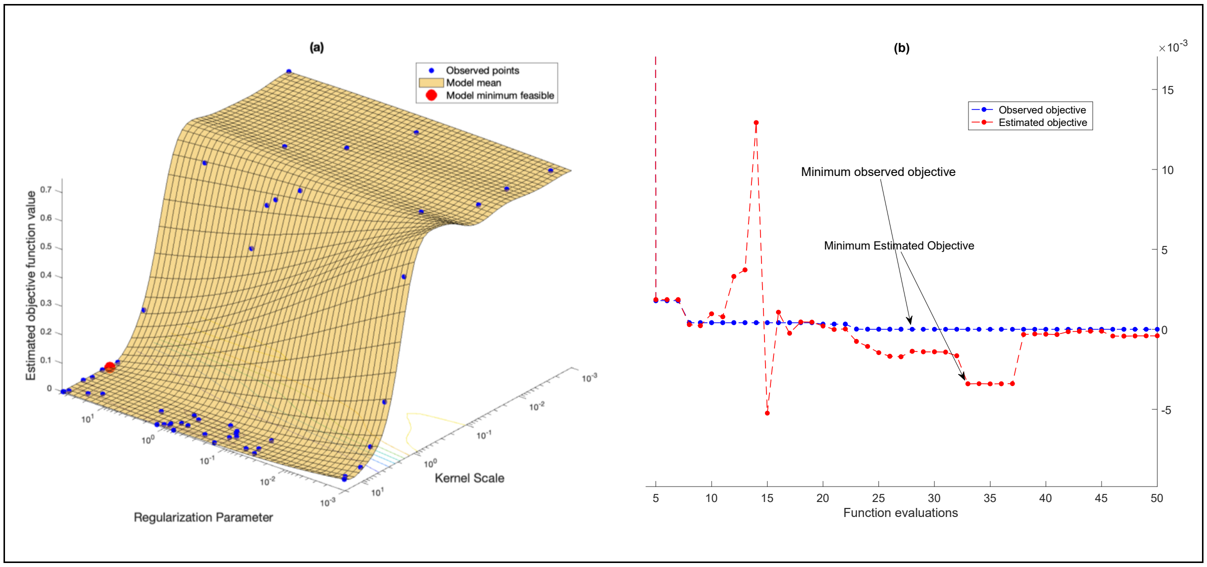


Figure 3.10 Refined search field for Bayesian Optimization convergence plot of the CY_s SVR model

This optimization yielded a better result than the previous five optimizations. Figure 3.10(a) displays all the evaluated points (in terms of (C, σ)), and their corresponding objective values estimated by the surrogate model in blue circles. The red circle in Figure 3.10(a) represents the minimum estimated objective $obj_{\min est} = -4 \times 10^{-4}$ found at $(C_{\min est}, \sigma_{\min est}) =$ (0.15, 7.16). While the minimum estimated objective and the minimum observed objective were close, they were not identical. Figure 3.10(b) magnifies this difference, with the minimum observed objective being $obj_{\min obs} = 9 \times 10^{-7}$ at $(C_{\min obs}, \sigma_{\min obs}) =$ (49.26, 8).

Therefore, the hyperparameters ($C_{\min obs}$, $\sigma_{\min obs}$) = (49.26, 8) were deemed optimal, as they led to a minimum objective function value of 9×10^{-7}.

3.3 Results

We performed two distinct analyses to evaluate the effectiveness of our AI models in predicting aerodynamic coefficients and their performance when integrated into a flight simulation platform. For validation purposes, 2,136 flight cases (90.8% of all collected data) were used, ensuring an unbiased assessment of the model performance.

The first analysis seeks to determine the accuracy of both MLP and SVR models in predicting lateral aerodynamic coefficients by comparing their predicted values with experimental data obtained from the CRJ700 VRESIM. To quantify the model performance, we calculated the Mean Absolute Relative Error (MARE) for each model using Equation (3.19) where o_i represents the i$^{\text{th}}$ output vector associated with the i$^{\text{th}}$ input vector in the dataset, and $\hat{o}_i$ represents the i$^{\text{th}}$ output vector predicted by the model. A lower MARE indicates better model performance and higher prediction accuracy.

$$MARE = \frac{1}{n}\left(\sum_{\mathrm{i}=1}^{n}\left|\frac{\hat{o}_i - o_i}{\hat{o}_i}\right|\right) \times 100 \tag{3.19}$$

The second analysis consisted of integrating the trained MLP and SVR models into a CRJ-700 simulation platform designed to replicate flight test scenarios conducted using the VRESIM. Through this analysis, we evaluated how well the AI models performed when integrated into a practical application, such as a flight simulator, and assessed their potential for real-world use. We compared several flight parameters predicted by the developed platform with those measured using the VRESIM for the 2,136 validation flight cases, adhering to the tolerance criteria specified by the Federal Aviation Administration (FAA) in the Manual of Criteria for the Qualification of Flight Simulators (FAA, 1991; ICAO, 2016). This allowed us to determine the effectiveness and reliability of the integrated AI models in simulating realistic flight conditions and scenarios.

3.3.1 Validation of Lateral Aerodynamic Coefficients

The hyperparameter optimization process for the MLP model yielded the following parameters: the training function, set to *trainbr*, the activation function set to *logsig*, the number of hidden layers set to $m = 4$, and the number of neurons per hidden layer set to $n = 14$. Comparing these results to the literature on similar problems, we found that these parameters are consistent with successful implementations of MLP models for other complex prediction tasks. In particular, previous studies have reported favorable results with the *trainb*' training function and *logsig* activation function in comparable applications (Tondji et al., 2022a; Tondji et al., 2022b), further validating our model configuration.

To improve the predictive accuracy of our optimized MLP model, we trained it again using a higher number of epochs (5,000) compared to the initial 1,000 epochs used during the hyperparameter optimization process. This extended training period aimed to more effectively capture the inherent patterns in the training data, thereby improving the model overall performance.

Table 3.3 presents the hyperparameter optimization results for SVR models for predicting the three lateral aerodynamic coefficients. A precision of $\varepsilon = 10^{-4}$ was chosen for the three aerodynamic models because some values of the coefficients were of the order of 10^{-2} or 10^{-3}. All three SVR models achieved better performance when they used the Gaussian kernel function. The optimized kernel scales for all three models were found to be within the same order, despite a wide search range of $[10^{-3}, 10^{3}]$. This suggests that the data structures and complexities of the problems associated with predicting each aerodynamic coefficient are somewhat analogous, and the optimized kernel scales reflect this similarity. In addition, the optimized regularization parameter C for CY_S was higher than for Cl_S and Cn_S, suggesting that the complexity and sensitivity of the models to overfitting may differ.

Table 3.3 Optimal hyperparameters ε, σ and C obtained for the SVR using the Gaussian kernel function and SMO solver

	Precision $\boldsymbol{\varepsilon}$	**Kernel scale $\boldsymbol{\sigma}$**	**Regularization parameter $\boldsymbol{C}$**
$\boldsymbol{CY_S}$	10^{-4}	8	49.25
$\boldsymbol{Cl_S}$	10^{-4}	16.40	5.08
$\boldsymbol{Cn_S}$	10^{-4}	10.4	1.53

Figure 3.11 to Figure 3.13 show examples of estimated aerodynamic coefficients plotted with respect to the roll angle. The data predicted by the multilayer perceptron (MLP) model (represented in red dots) and the support vector regression (SVR) model (shown in black crosses) closely align with the experimental data obtained with the VRESIM (illustrated in blue).

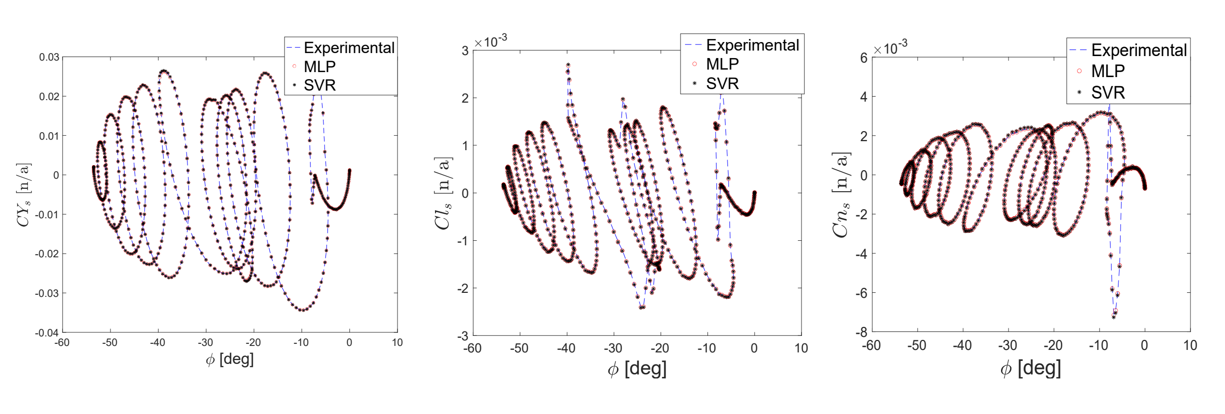


Figure 3.11 Examples of predicted lateral aerodynamic coefficients for a flight test at m = 46,300 lb | Xcg = 24% | h = 20,000 ft | IAS = 260 kts

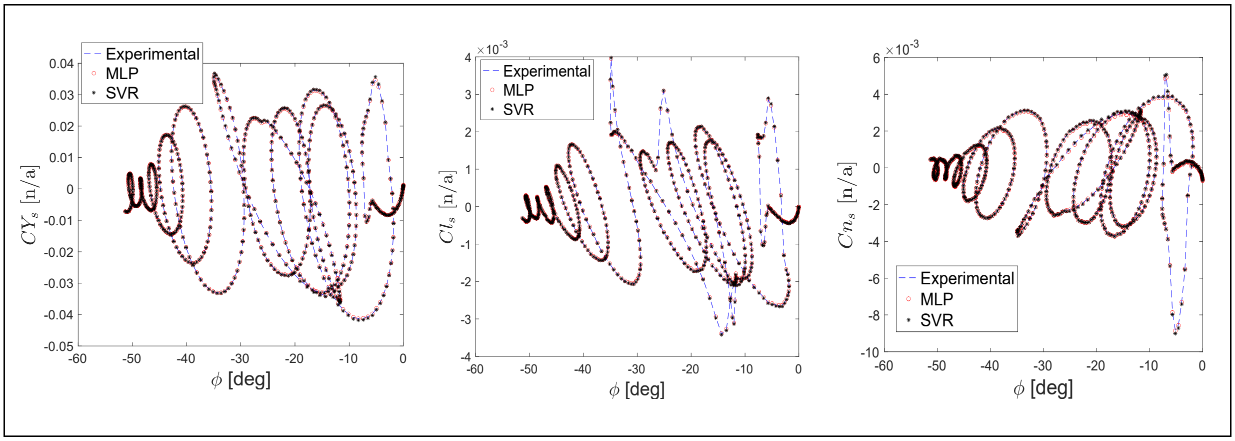


Figure 3.12 Examples of predicted lateral aerodynamic coefficients for a flight test at m = 55,120 lb | Xcg = 18% | h = 20,000 ft | IAS = 280 kts

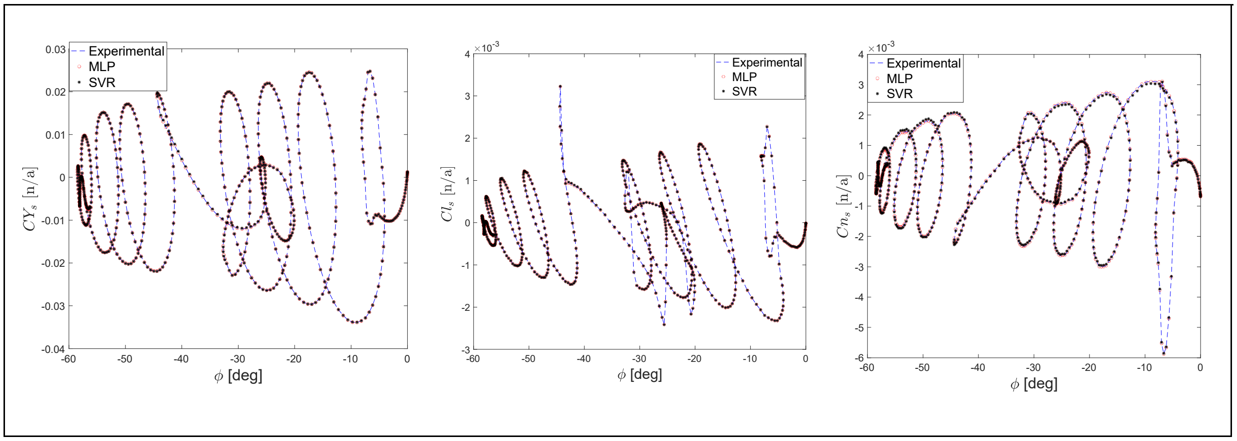


Figure 3.13 Examples of predicted lateral aerodynamic coefficients for a flight test at m = 63,930 lb | Xcg = 30% | h = 30,000 ft | IAS = 280 kts

The average errors obtained for all validation flight scenarios are listed in Table 3.4. In addition, Figure 3.14 presents histograms comparing the residual error distribution of both the MLP and SVR models in predicting the lateral aerodynamic coefficients. The MLP model shows higher accuracy for $C\mathrm{Y}_S$, as indicated by a lower average MARE and a narrow error distribution centered around zero. However, for $C\mathrm{l}_S$ and $C\mathrm{n}_S$, the SVR model demonstrates a greater accuracy with a lower average MARE and a higher degree of prediction consistency, with a smaller error distribution compared to the MLP. It is important to note that the hyperparameter ε was set to 10^{-4} for all three SVR models. Despite this constraint, the observed residual errors were of the order of 10^{-5}, indicating that the SVR model exceeded the initial expectations and provided predictions with a level of precision higher than the defined margin of error (ε).

Table 3.4 Average $MARE$ and Average Residual Errors
obtained for the prediction of CY_s, Cl_s and Cn_s using MLP and SVR

	Average $MARE$ [%]		**Average Residual [× 10^{-5}]**	
	MLP	SVR	MLP	SVR
CY_S	0.17	0.70	1.84	9.78
Cl_S	2.74	0.78	4.19	1.42
Cn_S	4.24	1.12	7.95	2.60

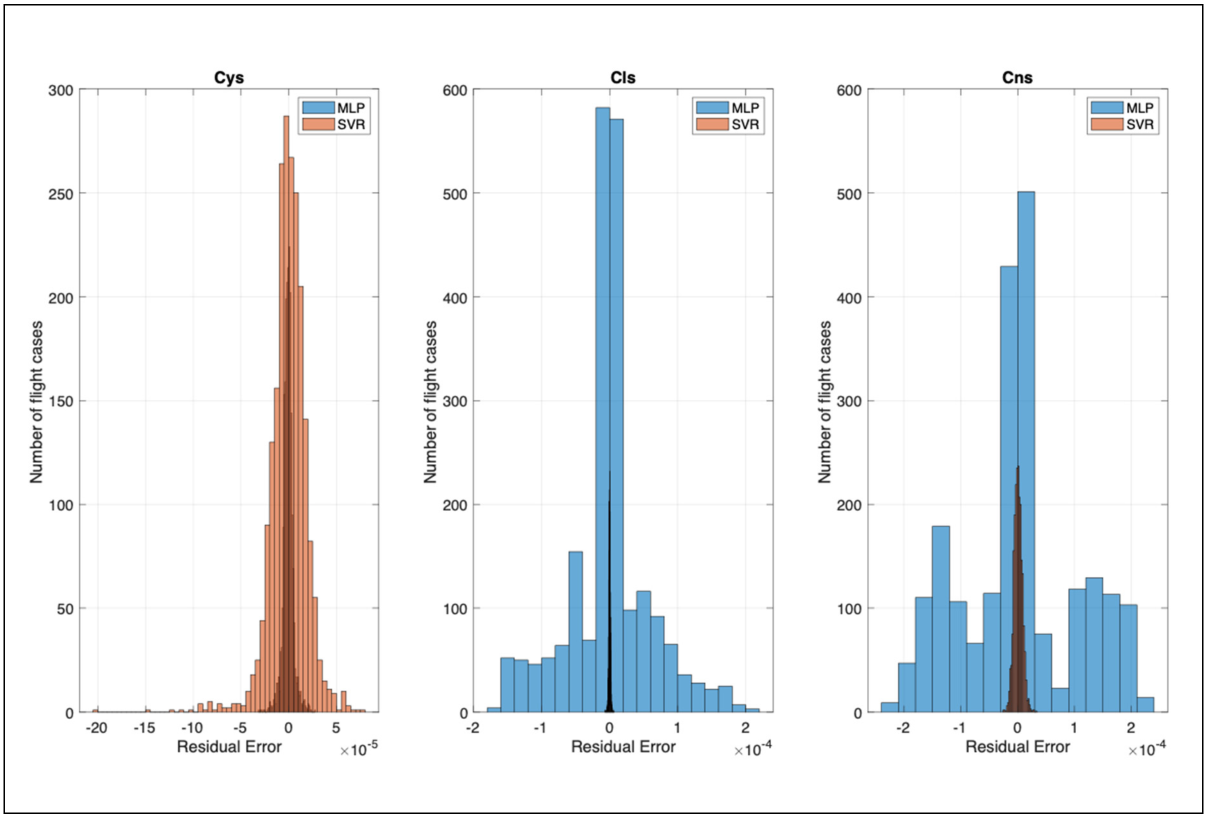


Figure 3.14 Comparison of residual error distributions for MLP and SVR in the prediction of lateral aerodynamic coefficients

3.3.2 Flight Dynamics Model Validation

The MLP and SVR models developed in this work were integrated into a MATLAB-based CRJ-700 flight simulation platform. A functional diagram of this platform is shown in Figure 3.15. Designed to accurately reproduce flight test scenarios, this simulation platform can replicate conditions encountered using the VRESIM. The simulator contains several key components. The pilot inputs were used to mimic the specific inputs used in the VRESIM scenarios, ensuring consistency between the simulated flight and the reference data. The lateral aerodynamic models (shown in green) were obtained using the MLP and SVR trained to predict the lateral coefficients, and a SVR model developed and validated in a previous study (Tondji et al., 2023) at LARCASE was used to calculate the longitudinal aerodynamic coefficients. These AI models (lateral and longitudinal) replace the traditional look-up tables often used in flight simulators, and compute the forces and moments based on the current flight conditions and aircraft state. The equations of motion describe the physical behavior of the aircraft based on the forces and moments calculated by the aerodynamic models. These equations govern the aircraft motion and changes in its state over time. The sensors in the simulation platform mimic those on an actual aircraft, providing various flight parameters data. This sensor data is used to assess the performance of the simulation and to compare it with the reference data from VRESIM. Finally, the 3D visualization translates the motion and state of the aircraft, calculated by the equations of motion and aerodynamic models, into a graphical representation, thereby rendering the flight scenarios in a visual display.

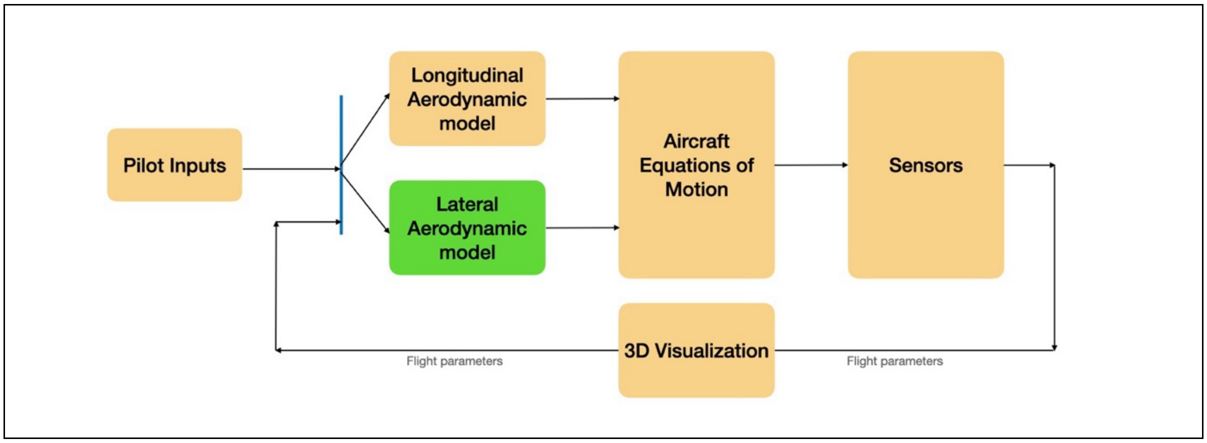

Pilot Inputs
Longitudinal Aerodynamic model
Lateral Aerodynamic model
Aircraft Equations of Motion
Sensors
3D Visualization
Flight parameters
Flight parameters

Figure 3.15 Bombardier CRJ-700 flight dynamics platform.

To validate the accuracy of the developed MLP and SVR models, a series of flight test scenarios were conducted, which replicated those performed with the VRESIM. The simulator-generated flight parameters were then compared with the corresponding parameters recorded by the VRESIM. These comparisons were conducted according to the FAA tolerance, for validating the developed models by use of tolerances for Dutch Roll. These criteria specify that the roll rate oscillation period should be estimated within a tolerance of $\pm 10\%$, while the damping ratio should be within a tolerance of ± 0.02. Furthermore, there should be a maximum discrepancy of ±1 second in the time between the roll angle peaks.

For illustrative purposes, comparison simulations of roll rate and roll angle are shown in Figure 3.16 and Figure 3.17, respectively. These simulations correspond to the flight case at $m = 46{,}300\, lb$, $Xcg = 24\%$, $h = 20{,}000\, ft$ and $IAS = 260\, kts$. The flight parameters obtained by the simulation platform using the MLP and SVR models are denoted in red and black, respectively, while the parameters derived from the VRESIM are represented in blue. The peak points are marked by dotted circles. The plots reveal oscillations that occur post-aileron impulse, approximately at 40 seconds. At this time, the damping ratio can be measured effectively.

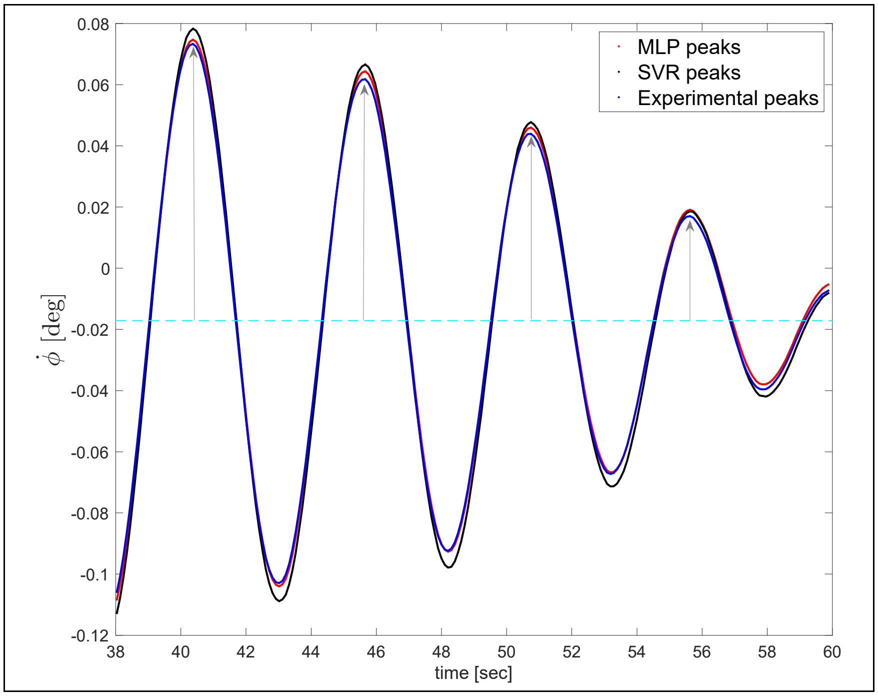


Figure 3.16 Example of roll rate prediction for a flight test at m = 46,300 lb | Xcg = 24% | h = 20,000 ft | IAS = 260 kts

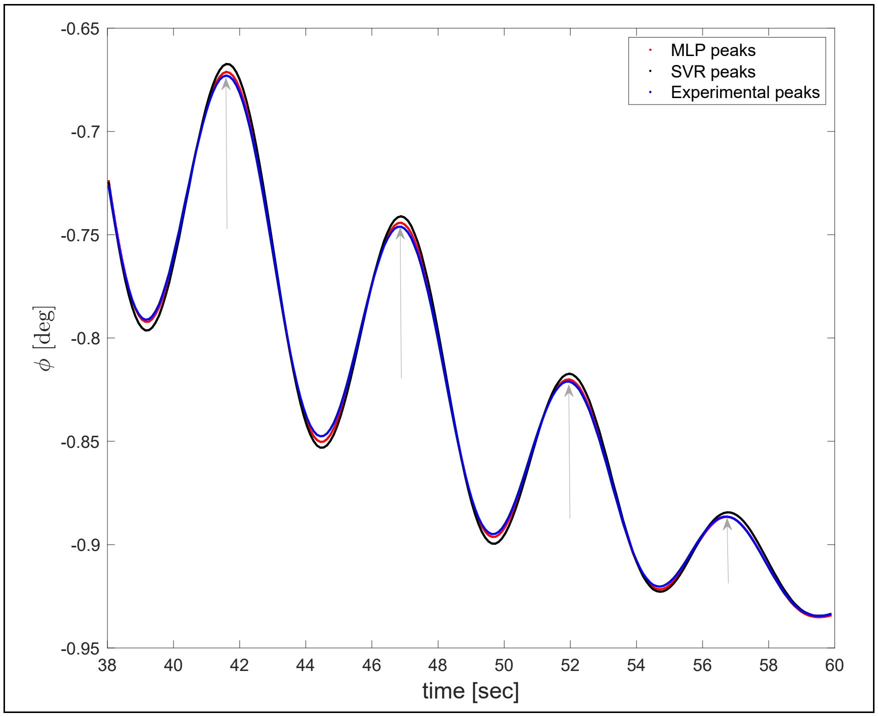


Figure 3.17 Example of roll angle prediction for a flight test at m = 46,300 lb | Xcg = 24% | h = 20,000 ft | IAS = 260 kts

To determine the roll rate period, we calculated the mean time difference between successive peaks for the three signals. The damping ratio of the roll rate was then computed using Equation (3.20) (Alexandersson, Mao & Ringsberg, 2021):

$$\zeta = \frac{1}{\sqrt{1 + \left(\frac{2\pi}{\delta}\right)^2}} \tag{3.20}$$

where δ represents the logarithmic decrement. This decrement is defined as the natural logarithm of the ratio between two successive altitude peaks, as shown in Equation (3.21) (Alexandersson et al., 2021):

$$\delta = \frac{1}{n} \log\left(\frac{p(i)}{p(i + nT)}\right) \tag{3.21}$$

where $p(i)$ and $p(i + nT)$ denote the amplitudes of two peaks separated by n periods of duration T. The peaks values and the locations of the roll angles were also measured.

After analyzing all validation flight cases, both the MLP and SVR models satisfied the FAA-specified tolerances for roll rates and roll angles. Examples showing the comparison of FAA tolerances between experimental results and AI predictions for roll angle and roll rate are provided in Table 3.5 and Table 3.6, respectively. These comparisons correspond to the flight cases illustrated in Figure 3.11 to Figure 3.13.

Table 3.5 Errors of the prediction of roll rate, damping ratio, and roll angle peaks' location

	Relative error of the period *T* estimation			**Residual error of the damping ratio *ζ* estimation**		
	MLP [%]	SVR [%]	Tolerance limit [%]	MLP [$\times 10^{-2}$]	SVR [$\times 10^{-2}$]	Tolerance limit [n/a]
m = 46,300 lb | Xcg = 24% | h = 20,000 ft | IAS = 260 kts	< 0.4	< 0.4	± 10	0.31	0.17	± 0.02
m = 55,120 lb | Xcg = 18% | h = 20,000 ft | IAS = 280 kts	< 0.4	< 0.4	± 10	0.11	0.038	± 0.02
m = 63,930 lb | Xcg = 26% | h = 25,000 ft | IAS = 330 kts	< 0.4	0.5	± 10	0.55	0.45	± 0.02

Table 3.6 Errors of the prediction of roll angle peaks' location appearing after 40 seconds

	1st peak		2nd peak		3rd peak		4th peak		
	MLP [sec]	SVR [sec]	MLP [sec]	SVR [sec]	MLP [sec]	SVR [sec]	MLP [sec]	SVR [sec]	Tolerance limit [sec]
m = 46,300 lb | Xcg = 24% | h = 20,000 ft | IAS = 260 kts	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	1
m = 55,120 lb | Xcg = 18% | h = 20,000 ft | IAS = 280 kts	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	0.11	1
m = 63,930 lb | Xcg = 26% | h = 25,000 ft | IAS = 330 kts	< 0.1	0.11	< 0.1	< 0.1	< 0.1	< 0.1	0.11	< 0.1	1

As shown in Table 3.5 for the relative error of the period T estimation, both the MLP and SVR models showed excellent performances. They maintained an error of less than or equal to 0.5%, which is significantly lower than the tolerance limit of ±10%. Similarly, for the residual error of the damping ratio ζ estimation, both models performed well within the tolerance limit of

±0.02. The SVR model performed slightly better than the MLP model in all given cases. Regarding the error of the prediction of roll angle peaks' location, both MLP and SVR models produced excellent results, with errors of less than 0.12 sec, within the tolerance limit of 1 sec.

3.4 Conclusions

The main objective of this research was to accurately predict lateral aerodynamic coefficients, which are critical for improving flight control and ensuring safety in the aviation industry. Specifically, we focused on the Bombardier CRJ-700 aircraft, a widely used regional jet, and trained and optimized artificial neural network (Multi-Layer Perceptron, MLP) and Support Vector Regression (SVR) models to predict these coefficients. The data for this study was collected using the Level-D CRJ-700 (VRESIM) designed by CAE Inc and Bombardier. To optimize the performance of these models, we used Bayesian Optimization techniques to determine the optimal hyperparameters for both models. For the MLP, we found that the optimal training function was *trainbr*, the activation function was *logsig*, and the architecture consisted of 4 hidden layers with 14 neurons each. For the SVR, the optimization revealed that the Gaussian kernel function performed the best with different kernel scale and regularization parameters values across the three lateral aerodynamic coefficients.

To evaluate the performance of the models, we compared their predictions against experimental data from 2,136 flight cases. The SVR models outperformed the MLP model for the prediction of Cl_S and Cn_S, while the MLP achieved better results than the SVR in predicting CY_S. However, both models demonstrated very good precision overall, with a Mean Absolute Relative Error (MARE) below 5%. The residual errors achieved by the SVR models were of the order of 10^{-5}, indicating a level of accuracy well beyond the initial precision set by the ε hyperparameter (10^{-4}). Furthermore, all models met 100% of the Federal Aviation Administration (FAA) tolerance criteria for all validation flight cases, confirming their reliability for practical application.

Our study has significant practical implications for the aviation industry, as the enhanced accuracy in predicting lateral aerodynamic coefficients using SVR models can directly contribute to the development of more efficient, safe, and cost-effective aircraft designs. Additionally, incorporating our AI (MLP and SVR) models into flight simulators can create highly realistic training environments, potentially improving training outcomes and flight safety. Our findings can also help improve the control logic of aircraft autopilot systems, thus leading to smoother and more efficient flights. Similarly, proactive risk management techniques like the cluster-based anomaly detection method (Li et al., 2015), which utilizes data from the flight data recorder to identify operational anomalies, could complement these improvements by providing another layer of safety through operational monitoring.

However, it is important to acknowledge the limitations of our study. One primary constraint is that the models were trained and tested on a limited dataset, which may not encompass all possible flight scenarios and conditions. Future research could involve expanding the scope of the training dataset to include more diverse flight conditions and scenarios, as well as investigating the use of other advanced machine learning techniques, such as deep learning algorithms, for aerodynamic coefficients prediction. By addressing these limitations and expanding upon the current work, we believe that future research can continue to drive forward the integration of AI techniques in aerodynamics.

CHAPITRE 4

IDENTIFICATION OF THE BOMBARDIER CRJ-700 STALL DYNAMICS MODEL USING NEURAL NETWORKS

Yvan Tondji [a], Mouhamadou Wade [b], Georges Ghazi [c], Ruxandra Mihaela Botez [d]

[a,b,c,d] Department of System Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3

Paper published in the *AIAA Journal of Aerospace Information Systems*, Vol. 19, No. 12,
December 2022, pp. 781-798.
DOI: https://doi.org/10.2514/1.I011104

Résumé

Cet article a pour objectif de présenter une méthodologie innovante destinée à modéliser les coefficients aérodynamiques et à prédire la dynamique des aéronefs en conditions de décrochage, y compris le cycle d'hystérésis, en utilisant les réseaux de neurones. Les variations des coefficients aérodynamiques, essentielles au processus d'identification, ont été déduites à partir des données de vol recueillies au cours de diverses manœuvres de décrochage. Par la suite, simulateur de vol virtuel pour la recherche CRJ-700 (VRESIM), qualifié de niveau D par la « Federal Aviation Administration » (FAA) et conçu par CAE, Inc. et Bombardier, a été utilisé pour la collecte de données de vol tant dans les phases de décrochage statique et dynamique. Selon la FAA, le niveau D représente le plus haut degré de qualification pour les modèles de dynamique de vol et de propulsion. Les perceptrons multicouches « Multilayer perceptron (MLP) » et les réseaux de neurone récurrents « Recurrent Neural Networks (RNN) » ont été entrainés pour prédire les coefficients aérodynamiques et leur corrélation avec les paramètres de vol. Une nouvelle méthodologie pour le l'optimisation des hyperparamètres des modèles MLP et RNN, incluant le nombre optimal de couches et de neurones, a été développée. Les modèles résultants ont été validés en comparant les données de vol prédites avec les données expérimentales obtenues à partir du CRJ-700 VRESIM de niveau D, en prenant en compte les mêmes commandes de pilotage. Les modèles élaborés grâce à cette méthodologie

ont réussi à prédire la dynamique de vol du CRJ-700 dans les conditions de décrochage statique et dynamique avec une précision remarquable, respectant les tolérances établies par la FAA.

Abstract

This paper aims to present a new methodology to model the aerodynamic coefficients and predict the aircraft dynamics under stall conditions, including the hysteresis cycle, using neural networks. The aerodynamic coefficients variations required for the identification process were estimated from flight data collected during different stall maneuvers. Then, a level-D-qualified Bombardier CRJ-700 virtual research equipment simulator (VRESIM) developed by CAE, Inc. and Bombardier was used to gather flight data in both linear and nonlinear stall phases. According to the Federal Aviation Administration (FAA), level D is the highest qualification level for flight dynamics and propulsion models. Multilayer perceptron (MLP) and recurrent neural networks were trained for the aerodynamic coefficients learning and their correlation with flight parameters. A new methodology for tuning the neural network parameters, such as the optimal number of layers and neurons, was developed. The resulting models were validated by comparing predicted flight data with experimental data obtained from the level D Bombardier CRJ-700VRESIMby considering the same pilot inputs. The models developed using the proposed methodology were able to predict the CRJ-700 flight dynamics in both static and dynamic stall conditions, with very good precision, within the tolerances of the FAA (FAA, 1991; ICAO, 2016).

4.1 Introduction

or reasons of safety and passenger comfort, aircraft are normally designed to fly in conditions within a defined envelope in order to avoid dangerous situations, such as stalling. Stall is an aerodynamic phenomenon that occurs at high angles of attack and at low speeds, and results in a significant suddenly reduction in the lift of the wing (Spentzos et al., 2005). Although it represents a certain risk for commercial aircraft, flying at a high angle of attack and in near-stall conditions has several advantages, such as increasing the aircraft's lift capacity or reducing landing distances. Thus, given the industry's desire to constantly improve the performance and safety of aircraft, aerodynamic phenomena such as stall are of great interest. Understanding stall allows manufacturers to explore new control strategies (Abdolhosseini et al., 2013; Moncayo et al., 2012; Shao et al., 2019; Wang & Zhang, 2018) to extend the operational envelope of the next generation of aircraft. It may also help the airline industry in the development of new approaches for proactive risk management (Li et al., 2015; Vakil & John Hansman, 2002), aiming to identify and mitigate risks prior eventual accidents, or to develop feedback models of air transportation system (Mozdzanowska et al., 2008). However, modeling the aircraft dynamics in the stall region remains a challenge for engineers and researchers (Tinoco, 1998).

When an aircraft reaches stall conditions, it is subjected to multiple complex and nonlinear aerodynamic phenomena, such as boundary layer instabilities, vortex instabilities, early transition from laminar to turbulent flow and critical flow separation (McCroskey, 1981). The stall can be divided into two regions: static and dynamic. A static stall occurs when the aircraft’s angle of attack exceeds a critical value, and the airflow begins to separate from the wing. In this case, the aircraft remains controllable despite the loss of lift, and the pilot can return to the aircraft stable configuration by reducing its angle of attack. Dynamic stall, on the other hand, is more complex, and occurs when the airflow has completely separated from the wing. In this case, the pilot may temporarily lose control of the aircraft, which exhibits a complex and non-linear behavior. Mathematically, the aerodynamic coefficients of an aircraft in stall conditions can be described by two distinct trends: a static trend for static stall, and a hysteresis trend for dynamic stall.

Today, the most common techniques for determining the aerodynamic coefficients of an aircraft include the use of semi-empirical models (Bierbooms, 1992; Botez, 1989; Fischenberg, 1995) or Computational Fluid Dynamics (CFD) methods (Spentzos et al., 2005; Tinoco, 1998). Semi-empirical methods allow the variation of the coefficients to be modelled as a function of the flight conditions by using mathematical equations. These equations are based on theoretical models coupled with aerodynamic databases obtained from wind tunnel and flight tests. Botez

(1989) presents a comparative study of semi-empirical models for the prediction of dynamic stall. Although very practical, these models are too simple to be generalized to all aircraft, or to model non-linear phenomena such as those governing aircraft stall.

CFD methods, on the other hand, are more powerful and generalizable methods. They are based on the resolution of fluid dynamic equations by numerical calculations. Over the past decades, CFD methods have greatly contributed to improve the modeling of unsteady flows, to the point of becoming an essential tool for the aerospace industry. A review of CFD methods on dynamic stall was presented by Ekaterinaris & Platzer (1998). For example, Spentzos et al. (2005) performed numerical simulations of three-dimensional dynamic stall using CFD. The full Navier-Stokes equations, coupled with a two-equation turbulence model, were solved, and the results were presented for square-shaped wings having a NACA 0012 airfoil. The obtained results were validated using experimental data obtained from both the surface pressure distribution on the wing and the flow topology. Barakos & Drikakis (2003) also used Navier-Stokes and turbulence transport equations to computationally investigate subsonic and transonic turbulent flows around oscillating and ramping airfoils under dynamic stall conditions. They investigated the effects of various parameters, such as Reynolds number, Mach number and pitch rate on the stall angle.

CFD methods allow for cost-effective simulations leading to the understanding of critical parameters for aerodynamic modeling in stall conditions. They can also provide detailed information anywhere in the flow field, which allows the accurate characterization of the laminar to turbulent transitions region of the wing under tests. However, their calculation accuracy is usually limited by the assumptions of the mathematical equations used by their algorithms. Even though numerical simulations allow a first representation of the studied stall models, experimental tests must often confirm the obtained results before being considered as industrial improvements.

Many researchers have investigated experimental techniques to obtain the aerodynamic coefficients of a wing under stall conditions from wind tunnel tests. Piziali (1994), for instance, performed an analysis to evaluate the aerodynamic characteristics of a NACA 0015 finite wing with an aspect ratio of 10 for different values of pitch rates and angles of attack, and for a Reynolds number of 10^6. In his study, Piziali (1994) used pressure measurements at various spanwise wing locations to visualize the variations of aerodynamic loads on a wing. Tang & Dowell (1995) carried out a similar experiment on a NACA 0012 airfoil square wing model with an aspect ratio of 1.5. The model oscillated around the pitch axis at several reduced pitch rates and angles of attack. They also analyzed pressure fluctuations at three different spanwise locations of the wing to characterize the flow. Later, Braza et al. (2003) performed a laser velocimetry technique to obtain accurate measurements of flow velocity during dynamic stall. Wernert et al. (1996) used a Laser Sheet Visualization (LSV) technique to visualize the flow around a NACA 0012 airfoil. The wing oscillated with an amplitude of 10 degrees and a mean incidence angle of 15 degrees. The aspect ratio of the wing was 2.8, the Reynolds number was 3.73×10^5, and a reduced frequency of 0.15 was used. Mulleners et al. (2012) and Hansman & Craig (1987) investigated flow separation dynamics using a Time-Resolved Particle Image Velocimetry (TR-PIV) technique to visualize the velocity field on a steady airfoil. In Mulleners et al. (2012), experiments were performed on an OA209 airfoil at a Mach number of 0.16 and a Reynolds number of 1.8×10^6, while in Hansman & Craig (1987), tests on the NACA 64-210 and NACA 0012 at Reynolds number of $3.1\ \times 10^5$ were performed.

Wind tunnel experiments have the advantage of dealing with a “real” fluid and can be used to reproduce flight conditions over a wide range of the envelope of an aircraft. However, they have some disadvantages: they are prohibitively expensive and time-consuming (Botez, 2018; Moir & Coton, 1995), depending on the scale of the model, and they may not properly consider the effects of aeroservoelasticity. In addition, wind tunnels are limited because the Reynolds number that can be reproduced in such an environment is typically restricted to a range of 0.5×10^6 to 1×10^6, whereas in actual flight conditions it can vary between 20×10^6 and 50×10^6 (Mulleners et al., 2012).

Another alternative to model aircraft aerodynamic coefficients is to use system identification techniques. They combine the advantages of the previously presented methods and overcome their weaknesses. For example, in (Perhinschi et al., 2002), a flexible simulation tool for aircraft parameters identification was developed using system identification techniques. The developed tool was further used to implement new control strategies. These techniques aim to design mathematical model of a physical system from measurement data. In other words, the objective is to learn the dynamics of the system and translate it into mathematical equations. In this context, the use of artificial intelligence can be very practical.

Artificial Intelligence (AI) is a rapidly advancing technology that could soon have a significant impact on several industries and engineering processes. It refers to an artificial design of intelligence based on the human model; it can learn from past data and predict natural language and information. Technologies based on AI in (Haykin, 1998) are currently being developed for flight simulation, and they could solve a wide range of complex problems in the aeronautical field (Ben Mosbah, Botez & Dao, 2016; Ben Mosbah, Botez, Medini & Dao, 2020; Ben Mosbah, Flores Salinas, Botez & Dao, 2013; Boely, Botez & Kouba, 2011; Boely & Botez, 2010; De Jesus Mota & Botez, 2011; De Jesus Mota & Botez, 2009; Mosbah, Botez & Dao, 2013; Perhinschi et al., 2007). These studies have demonstrated that “past data” can be used to build a generalized mathematical model of a system being tested (Al-Shareef, Mohamed & Al-Judaibi, 2008).

Machine learning (ML), which is a subset of AI, has already successfully been used for solving complex system identification problems. Several system identification methods have already been applied to identify aerodynamic coefficients. For example, kriging (Appleby, Liu & Liu, 2020) and convolutional neural network (Chen, He, Qian & Wang, 2020) methods have been used to predict longitudinal aerodynamic coefficients in linear flight regimes through graphical image analysis of wing airfoils. However, these visual methods have the disadvantage to assume the steadiness of aerodynamic coefficients variation with angle of attack, which is not valid in the stall regime. Other ML methods such as Support Vector Machines (Paulete-Perianez, Andres-Perez & Lozano, 2019), Maximum Likelihood, Extreme Learning Machine (ELM) networks (Verma & Peyada, 2020; Hari Om Verma & Peyada, 2020) have also demonstrated their effectiveness in the aircraft parameter estimation field, especially for aerodynamic coefficients variations with the angle of attack. However, these methods have difficulties in solving optimization problems with a large number of variable dimensions and have limited accuracy for solving highly nonlinear engineering problems such as stall modeling. To our best knowledge among all these ML techniques, we believe that Neural Networks could be the most suitable presented in the literature for stall modeling as they can handle multi variables dimension modeling of nonlinear phenomena (Basappa & Jategaonkar, 1995; Ghazi et al., 2017; Linse & Stengel, 1993; Moin, Khan, Mobeen & Riaz, 2021; Peyada & Ghosh, 2009; Punjani & Abbeel, 2015).

Punjani & Abbeel (2015) defined a rectified linear unit Neural Network (NN) model and proved its effectiveness for modeling an aerobatic helicopter's dynamics. A helicopter is a complex system combining a rigid body model with aerodynamics, vibrations, engines, etc. Consequently, their system identification is a complex issue, especially when unsteady flight regimes are considered. The prediction of the vertical acceleration of the helicopter is empirically one of the most challenging problems; the NN-developed model improved this prediction by 60% over the previous state-of-the-art methods (Punjani & Abbeel, 2015). The overall Root Mean Square (RMS) acceleration was improved by 58%. Peyada & Ghosh (2009) had a similar challenge of modeling aircraft parameters using Neural Networks (NN). Their Feed Forward Neural Network (FFNN) was coupled with Gauss-Newton (GN) optimization. It used a black-box approach to estimate the aerodynamic coefficients of an aircraft using measured flight variables, such as speeds and accelerations. The algorithm was validated using both HANSA-3 aircraft and ATTAS aircraft flight data. The measured values were compared with those obtained using the Least Square (LS) and Filter Error (FE) methods. Their FFNN gave better approximations than the LS and FE methods. Other researchers, including Linse & Stengel (1993), Moin et al. (2021), Basappa & Jategaonkar (1995), Singh & Ghosh (2007) and Ghazi et al. (2017) also contributed to the application of neural networks for aircraft aerodynamic coefficients identification.

The advantages of neural network methods are their ability to learn through past data and to provide function approximations of the tested system, without any representation of a mathematical model or need to solve its equations of motion. In addition, once the model is trained with past data, the solutions are computed very quickly (milliseconds in software and microseconds in hardware) (Haykin, 1998). However, NNs require access to a significant amount of data to be able to train adequately. Numerical models, such as those encoded in high-certification level flight simulators, can represent the flight dynamics of an aircraft under near-reality flight conditions with excellent accuracy, to the point of being a reference for researchers (Ghazi et al., 2017; Hamel, Sassi, Botez & Dartigues, 2013; Zaag & Botez, 2017). Aircraft flight simulators present the advantage of enabling the rapid gathering of data that can be used to build the large database needed to feed a system identification model.

This paper aims to present a new methodology to model the aerodynamic coefficients and identify the dynamics of an aircraft in both static and dynamic stall conditions, using neural networks. The methodology was applied on the Bombardier CRJ-700 regional jet. This aircraft was selected because a highly qualified flight simulator was available for the study: a Virtual Research Equipment Simulator (VRESIM) for the Bombardier CRJ-700 (see Figure 4.1). The VRESIM was designed and manufactured by CAE Inc and has a level D qualification for its flight dynamics and propulsion model. According to the Federal Aviation Administration (FAA), the level D is the highest qualification level for flight simulators. Therefore, it was assumed that the VRESIM was accurate enough to be considered as a test aircraft, and that all the collected data was similar to the actual data collected by the Bombardier's flight test engineers' team.

Figure 4.1 Bombardier CRJ-700 Virtual
Research Equipment Simulator (VRESIM) used
for the validation of the proposed methodology

The rest of the paper is structured as follows: section 4.2 presents the methodology, which includes the procedure for acquiring flight data from the flight simulator and the data preprocessing needed to estimate the aerodynamic coefficients from measurable flight parameters. The method developed to determine the optimal neural network structure, as well as the training process are also discussed in this section. Section 4.3 presents the results obtained and their comparison with experimental data collected from the CRJ-700 VRESIM. Finally, the paper ends with conclusions and remarks regarding future work.

4.2 Methodology

Modeling a physical system consists of postulating a mathematical model that is supposed to reflect the structure of the system, and in determining the parameters defining that model. Such a model is very useful, especially when it is impossible or very expensive to create an experimental environment to test and analyze the system behavior. Within this context, the objective of this section is to present the methodology developed at LARCASE for modeling the aerodynamic coefficients of the Bombardier CRJ-700 aircraft in stall conditions using neural networks.

4.2.1 Flight Test Procedure and Data Gathering

The first step in this methodology was to conduct a series of flight tests and collect a large amount of flight data. The objective was to create a database reflecting the variation of the aircraft aerodynamic coefficients over a wide range of flight conditions. For this purpose, several flight tests were performed with the Bombardier CRJ-700 VRESIM, following the procedure described in Figure 4.2.

As illustrated in Figure 4.2, the procedure used in this study involved several steps. The first step was to stabilize (or trim) the aircraft at a given flight condition expressed in terms of airspeed and altitude. This maneuver was accomplished by engaging the altitude hold mode from the autopilot panel to maintain the altitude, while the airspeed was stabilized manually by adjusting the throttle position.

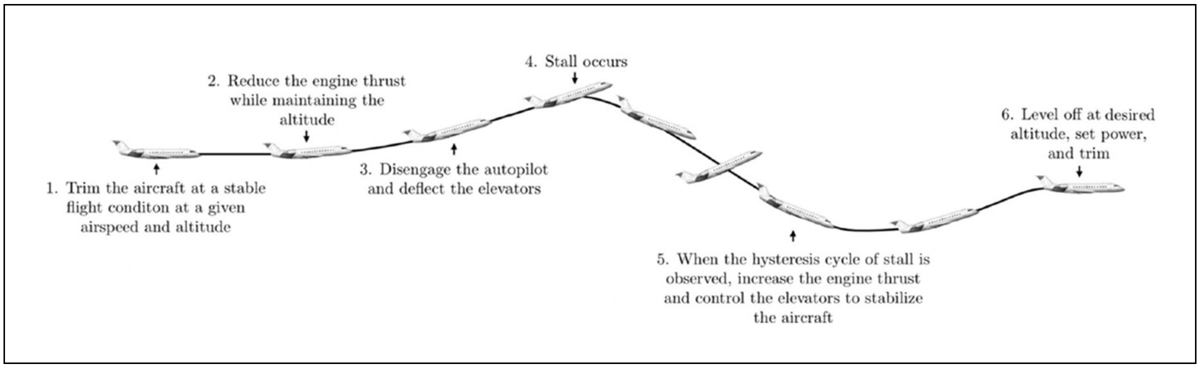

4. Stall occurs
2. Reduce the engine thrust while maintaining the altitude
6. Level off at desired altitude, set power, and trim
1. Trim the aircraft at a stable flight conditon at a given airspeed and altitude
3. Disengage the autopilot and deflect the elevators
5. When the hysteresis cycle of stall is observed, increase the engine thrust and control the elevators to stabilize the aircraft

Figure 4.2 Stall flight test procedure illustration

Once the aircraft was trimmed (step 1), the next step was to stall it. For this purpose, the engine thrust was reduced by returning the throttles to the idle position (step 2). This action resulted in a gradual reduction of the aircraft airspeed, which was compensated by the autopilot by increasing the angle of attack to maintain the altitude. When the aircraft airspeed was relatively low, close to stall speed, the autopilot was then disengaged, and the yoke was pulled back manually to deflect the elevators (step 3). This second action caused the angle of attack to suddenly increase until reaching the stall angle (α_{stall}). During this part of the flight test, the aircraft was maintained in stall conditions as much as possible by controlling the elevators in order to observe the dynamic stall, and at least one hysteresis cycle (step 4). Finally, the last maneuver was to recover the aircraft. The engine thrust was increased to gain airspeed, and the elevators were manually controlled to re-stabilize the aircraft (step 5), and then they returned to normal level flight conditions (step 6).

During the flight tests, all the required flight parameters, such as the altitude, Mach number, true airspeed, angular velocities, accelerations, engine thrust, and control surface deflections, were recorded at a sampling rate of 30 Hz. A typical example of data recorded from the VRESIM for a flight test conducted at an altitude of 7500 ft, a Mach number of 0.20, and with the slats fully retracted (i.e., 0^o) is shown in Figure 4.3. In this case, the pilot suddenly deflected the elevators δ_e at about 120 seconds (see Figure 4.3.a), which caused the angle of attack α to immediately increase above the stall angle (see Figure 4.3.f). The lift force then decreased significantly, resulting in a change in vertical acceleration a_z (see Figure 4.3.h). Similarly, the drastic change in longitudinal acceleration a_x reflects the increase of the drag force that occurs during the stall (see Figure 4.3.g). The combination of these two phenomena led to a decrease in the altitude h.

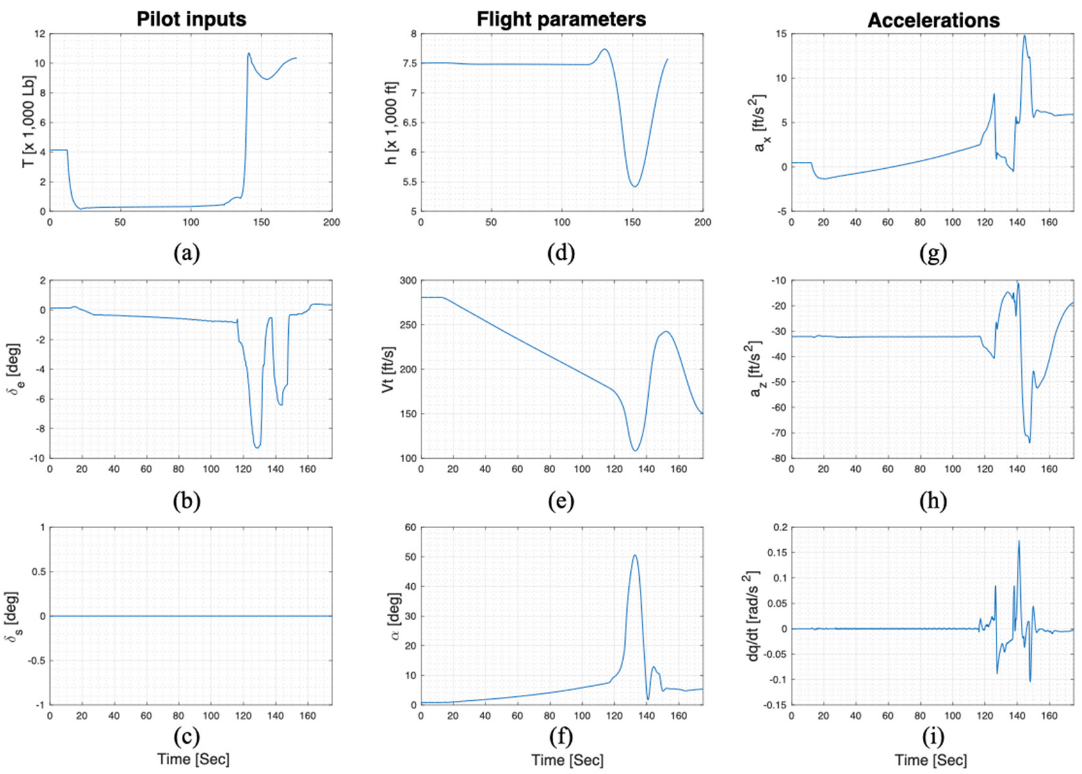


Figure 4.3 Example of data recorded for a flight test at 7500 ft, Mach 0.20, and slats δ_s retracted

Starting from the procedure described in Figure 4.2, a total of 39 flight cases (i.e., scenarios) were conducted with the Bombardier CRJ-700 VRESIM. These flight cases were determined by selecting 13 different altitudes ranging from 5000 to 35,000 ft, with a step of 2500 ft. In addition, as slats affect the shape of a wing by locally increasing its camber, which has the effect of delaying the stall phenomenon, three slat configurations were considered: 0^o (i.e., fully retracted), 20^o (i.e., intermediate position), and 45^o (i.e., fully extended). Table 4.1 to Table 4.3 summarize the flight conditions in terms of altitude, Mach number and angle of attack, considered for each of the three slat configurations.

Table 4.1 Flight test conditions with slats at 0°

Flight Case Number	**Altitude [ft]**	**Mach Number [at stall]**	**Angle of attack [at stall, in °]**
1	5000	0.21	17.02
2	7500	0.20	17.01
3	10,000	0.26	17.00
4	12,500	0.24	17.01
5	15,000	0.31	17.03
6	17,500	0.31	17.02
7	20,000	0.30	16.84
8	22,500	0.36	17.10
9	25,000	0.37	17.00
10	27,500	0.38	17.00
11	30,000	0.40	17.01
12	32,500	0.34	17.00
13	35,000	0.45	17.12

Table 4.2 Flight test conditions with slats at 20°

Flight Case Number	**Altitude [ft]**	**Mach Number [at stall]**	**Angle of attack [at stall, in °]**
14	5000	0.18	17.62
15	7500	0.19	17.55
16	10,000	0.19	17.58
17	12,500	0.20	17.60
18	15,000	0.21	17.50
19	17,500	0.23	17.60
20	20,000	0.24	17.56
21	22,500	0.30	17.60
22	25,000	0.30	17.54
23	27,500	0.31	17.70
24	30,000	0.36	17.70
25	32,500	0.36	17.60
26	35,000	0.37	17.70

Table 4.3 Flight test conditions with slats at 45°

Flight Case Number	**Altitude [ft]**	**Mach Number [at stall]**	**Angle of attack [at stall, in °]**
27	5000	0.16	17.72
28	7500	0.17	17.66
29	10,000	0.19	17.71
30	12,500	0.19	17.51
31	15,000	0.21	17.64
32	17,500	0.22	17.55
33	20,000	0.23	17.88
34	22,500	0.23	17.77
35	25,000	0.28	17.66
36	27,500	0.28	17.63
37	30,000	0.29	17.69
38	32,500	0.29	17.61
39	35,000	0.31	17.56

4.2.2 Data Processing and Aerodynamic Coefficients' Determination

Once all the flight tests were completed, the next step was to estimate the aerodynamic coefficients of the aircraft, and then to create a database to prepare the neural network training process.

Starting from the gathered data, the three aerodynamic coefficients in the stability axes of the aircraft, CD_s, CL_s and Cm_s , were estimated based on the following equations (Ghazi et al., 2017):

$$CL_s = CL_b \cos(\alpha) - CD_b \sin(\alpha) \tag{4.1}$$

$$CD_s = CD_b \cos(\alpha) + CL_b \sin(\alpha) \tag{4.2}$$

$$Cm_s = Cm_b - CD_b z_{cg} - CL_b x_{cg} \tag{4.3}$$

where $\{x_{cg}, z_{cg}\}$ are the distances of the aircraft center of gravity relative to its aerodynamic center, and CL_b, CD_b and Cm_b are the lift, drag and pitching moment coefficients, respectively, expressed in the aircraft body axis. These coefficients were determined from the following equations:

$$CL_b = \frac{ma_z - T_z}{1/2\rho V_T^2 S_w} \tag{4.4}$$

$$CD_b = \frac{ma_x - T_x}{1/2\rho V_T^2 S_w} \tag{4.5}$$

$$Cm_b = \frac{I_{yy}\dot{q} - T_x z_{eng} - T_z x_{eng}}{1/2\rho V_T^2 S_w c_w} \tag{4.6}$$

where ρ is the air density, $\{T_x , T_z\}$ are the components of the engine thrust, I_{yy} is the aircraft moment of inertia about the lateral axis, S_w is the wing reference area, c_w is the mean aerodynamic chord of the wing, and a_x and a_z are the longitudinal and vertical accelerations of the aircraft, respectively.

Figure 4.4 to Figure 4.6 show the aerodynamic coefficients estimated from flight test data for Bombardier CRJ-700 VRESIM, and for the three slat configurations. Note that for confidentiality reasons, the data presented in these figures are normalized to the mean value and standard deviation of each data sample.

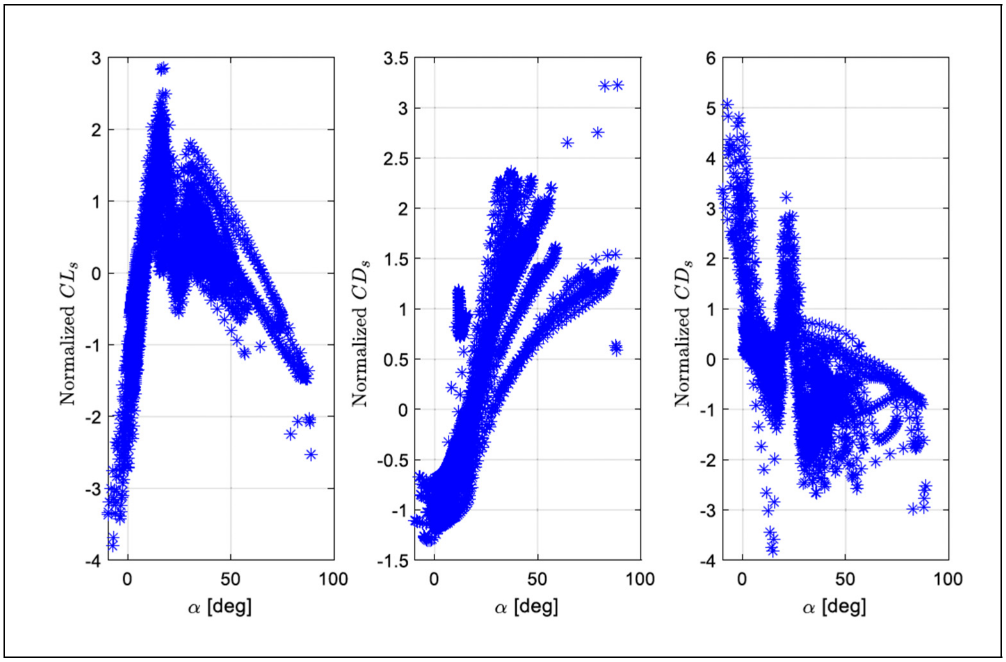


Figure 4.4 Aerodynamic coefficients' estimation from flight test data obtained from the Bombardier CRJ-700 VRESIM (slats at 0°)

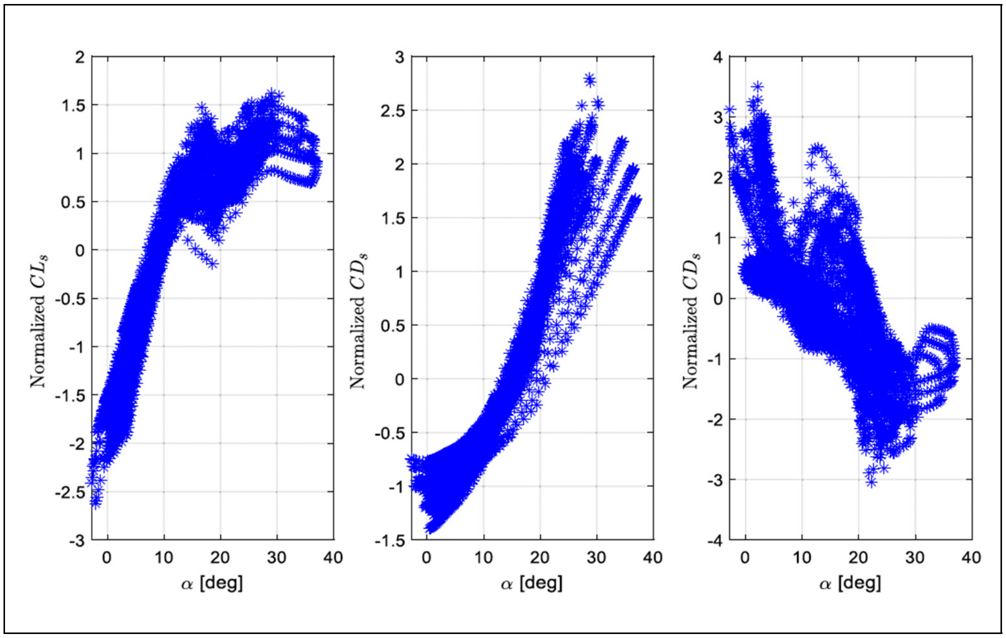


Figure 4.5 Aerodynamic coefficients' estimation from flight test data obtained from the Bombardier CRJ-700 VRESIM (slats at 20°)

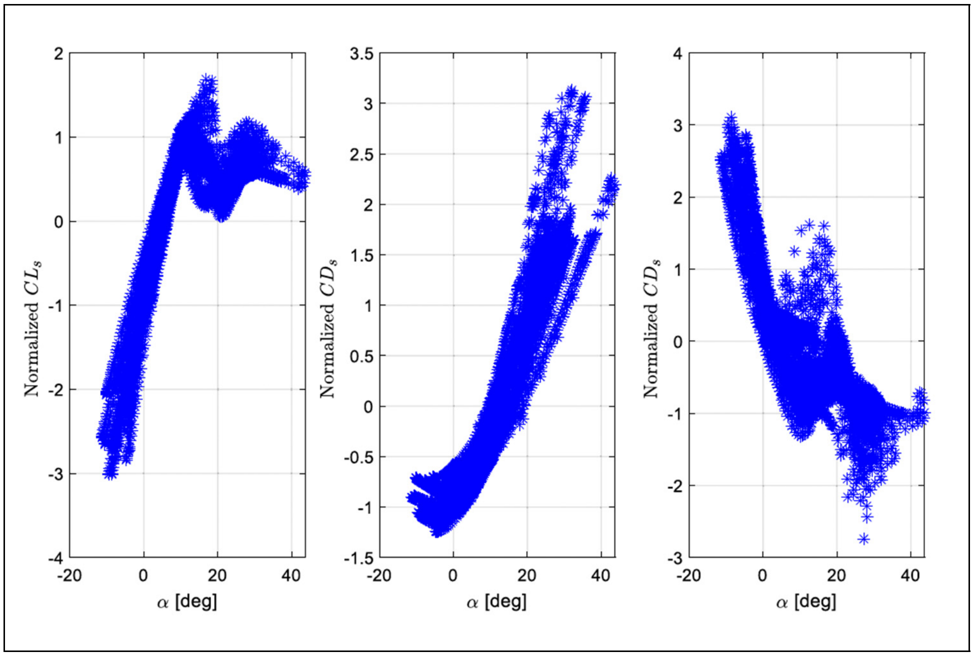


Figure 4.6 Aerodynamic coefficients' estimation from flight test data obtained from the Bombardier CRJ-700 VRESIM (slats at 45°)

It is interesting to note the effect of the slats on the lift and drag variations during the stall phase. Indeed, when the slats were fully retracted (see Figure 4.4), a very fast reduction in lift and an increase in drag could be observed, and the angle of attack could increase up to 80°. However, when the slats were extended (see Figure 4.5 and Figure 4.6), the reduction in lift was smaller, and the angle of attack did not exceed 45°. This observation highlights the importance of taking the slats into account in the model, as they have a great influence on the stall.

4.2.3 Neural Network Modeling

There are typically three categories of methods commonly used to identify a model for nonlinear systems: block-oriented (Baldelli, Lind & Brenner, 2005), functional time series (Yao, Müller & Wang, 2005), and black-box (Ghazi et al., 2017). Neural networks are part of the black-box methods. Given their ability to successfully approximate continuous and discontinuous functions, neural networks are convenient for identifying complex nonlinear systems (Haykin, 1998), such as the one considered in this paper.

4.2.3.1 Selection of the Type of Neural Networks

The design of neural networks for system identification problems involves several steps. The first step is to determine the best suited type of network to solve the problem of interest. Indeed, there are a very large number of network types, and new types are still being developed. Among all the existing types, two have especially proven their effectiveness in various fields of application: Multi-Layer Perceptrons (MLPs) and Recurrent Neural Networks (RNNs). MLPs are typically used to solve regression and function approximation problems, where real-value parameters such as aerodynamic coefficients are predicted from a given data set (Haykin, 1998). However, in this study, RNNs were also considered, because they have been demonstrated to be effective in designing models from time-series data by using information from previous state during the learning process (Williams & Zipser, 1989).

The following sections offer a brief description of the architecture of MLPs and RNNs.

Multilayer Perceptron

The fundamental element of a neural network, whatever its type, is the artificial neuron. Figure 4.7 shows a schematic representation of an artificial neuron, also called a “node” or “perceptron”, with multiple inputs from either a set of inputs or from neurons in another hidden layer.

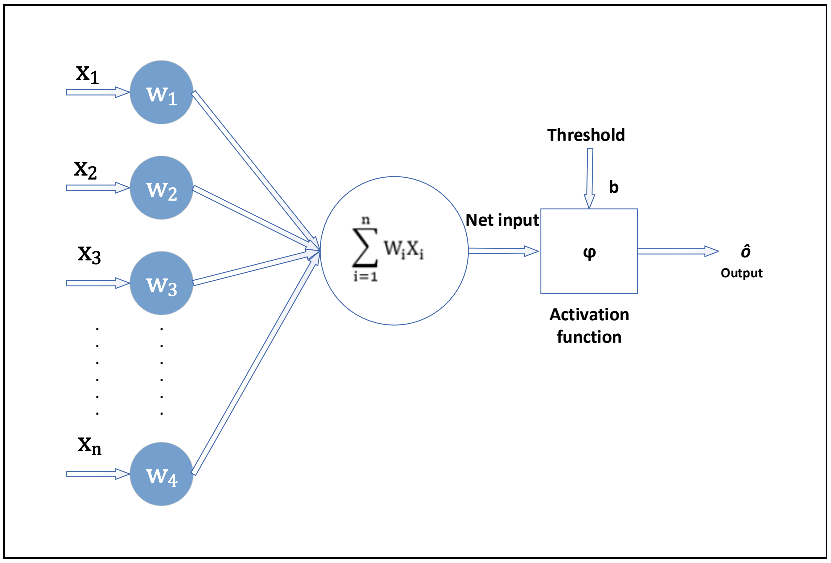


Figure 4.7 Graphical representation of an artificial neuron

As illustrated in Figure 4.7, a perceptron is a simple processing unit that computes an output from a given set of inputs. To compute the value of the neuron's output $\hat{o}$, the input signal of the neuron $X = [x_1, x_2, \dots, x_n]$ is multiplied by its corresponding weights $W = [w_1, w_2, \dots, w_n]$, summed up, and then fed to a "transfer" function or to an "activation" function. An activation function is associated with each neuron in a network and determines whether the neuron should be activated or not. Several types of activation functions exist in the literature, such as the linear function, the sigmoid function, or the rectified linear unit activation function (Linse & Stengel, 1993). Mathematically, the output of a perceptron can be written as follows:

$$\hat{o} = \varphi(X, W, b) = \varphi\left(\sum_{i=1}^{i=n} x_i w_i + b\right) \tag{4.7}$$

where φ is the activation function and b is a constant bias that defines the activation threshold of the neuron.

MLPs are composed of a set of neurons, connected to each other, and organized in layers, as shown in Figure 4.8. The first layer, also called the "input layer", aims to receive signals from the input data, while the last layer, also called the “output layer”, is defined according to the model’s number of outputs. Between these two layers, there is an arbitrary number of hidden layers. The number of hidden layers, as well as the number of neurons per layer, are essential parameters for MLPs, and in some way, determine their performance.

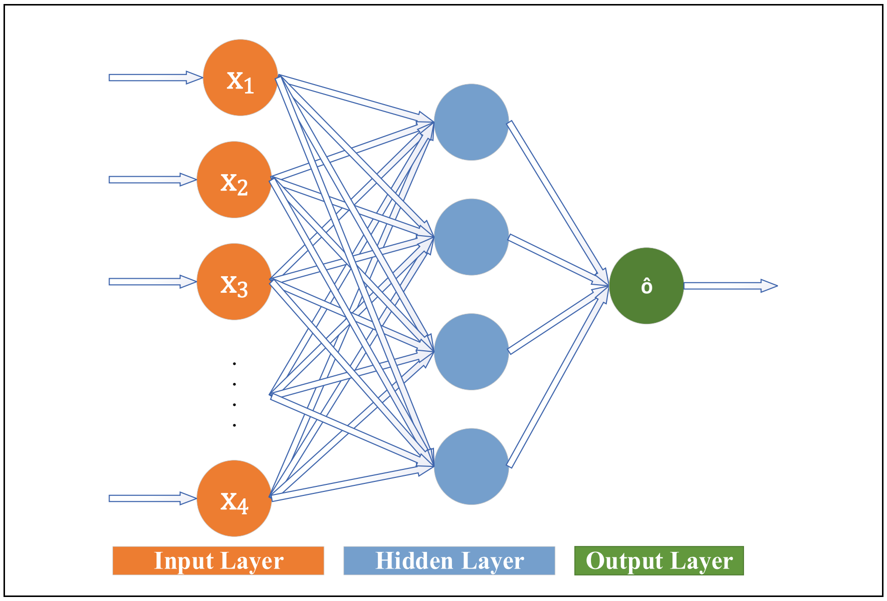


Figure 4.8 Graphical representation of a MLP Neural Network

The predicted output $\hat{o}$ of the MLP can be computed according to Equation (4.8) (Tondji et al., 2022a):

$$\hat{o} = \varphi_m \left(\sum_{k=1}^{k=n_m} w_{m,k} \times \ldots \times \varphi_2 \left[\sum_{i=1}^{i=n_2} w_{2,i} \times \varphi_1 \left(x_{1,j} w_{1,j} + b_{1,j} \right) + b_{2,i} \right] + b_{m,k} \right) \quad (4.8)$$

where X is the input vector, m is the number of layers of the neural network, φ_i is the activation function of the layer i, n_i is the number of neurons of layer i, and $W_{i,j}$ and $b_{i,j}$ are the weights and bias, respectively, of the j^{th} neuron of the layer i.

Recurrent Neural Network

As the name suggests, RNNs are a type of neural network that have recurrent interconnections. The main objective is to preserve the information of the neurons' variation with time. The main advantage of this type of networks is that they are, in theory, more efficient for processing time-series signals, such as voice, semantic analysis of videos or sentences, and others. Thus, since the aerodynamic coefficients vary in some way over time, it was assumed that the information from previous states might be relevant for approximating present or future states. Indeed, Suresh et al. (2003) demonstrated that a RNN was more effective than a MLP for learning the behavior of complex dynamic systems, such as the behavior of an aircraft in the stall region. In fact, for most dynamic systems, the output can be expressed as function of past inputs and outputs. Therefore, based on this observation, it was decided to improve the MLP model, as proposed by Elman (1990), by adding output feedback delayed by a time step Δt on each neuron. In this way, the prediction of the aerodynamic coefficients at a given time t depends on the aircraft's state at a previous time $(t - \Delta t)$. The Elman Neural Network (ENN) is a particular type of RNN, defined as a feed-forward network with additional memory neurons (called the "copied layer") and local feedback. A graphical architecture of the ENN is given in Figure 4.9.

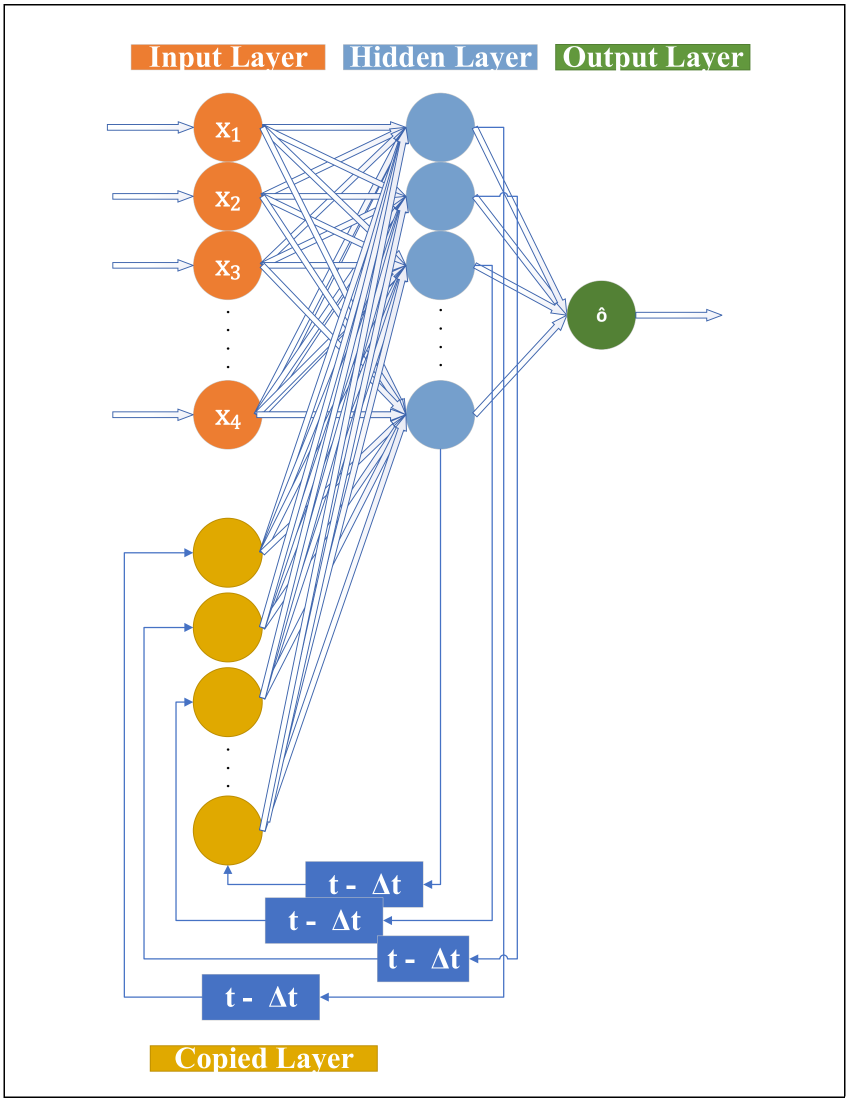


Figure 4.9 Graphical representation of a Recurrent Neural Network

The hidden layer receives its information from both the input and the copied layers during the training cycle, which are then combined and analyzed simultaneously. On each hidden layer neuron, the activation function is fed back to the copied layers at every time step to provide additional input combined with other information. The process is repeated for the successive training cycles, such that each time, the copied layer contains the outputs values of the hidden layer from the previous training cycle. This recurrence gives the network its dynamic properties. For more details on the Elman Neural Network (ENN) algorithm, refer to Elman (1990).

4.2.3.2 Data Labeling

Many experiments have shown that the airflow around the wing of an aircraft during a stall maneuver can be characterized by three main regions (McCroskey, 1981). These three regions are highlighted in Figure 4.10 for each aerodynamic coefficient, and with three different colors: blue, green and red.

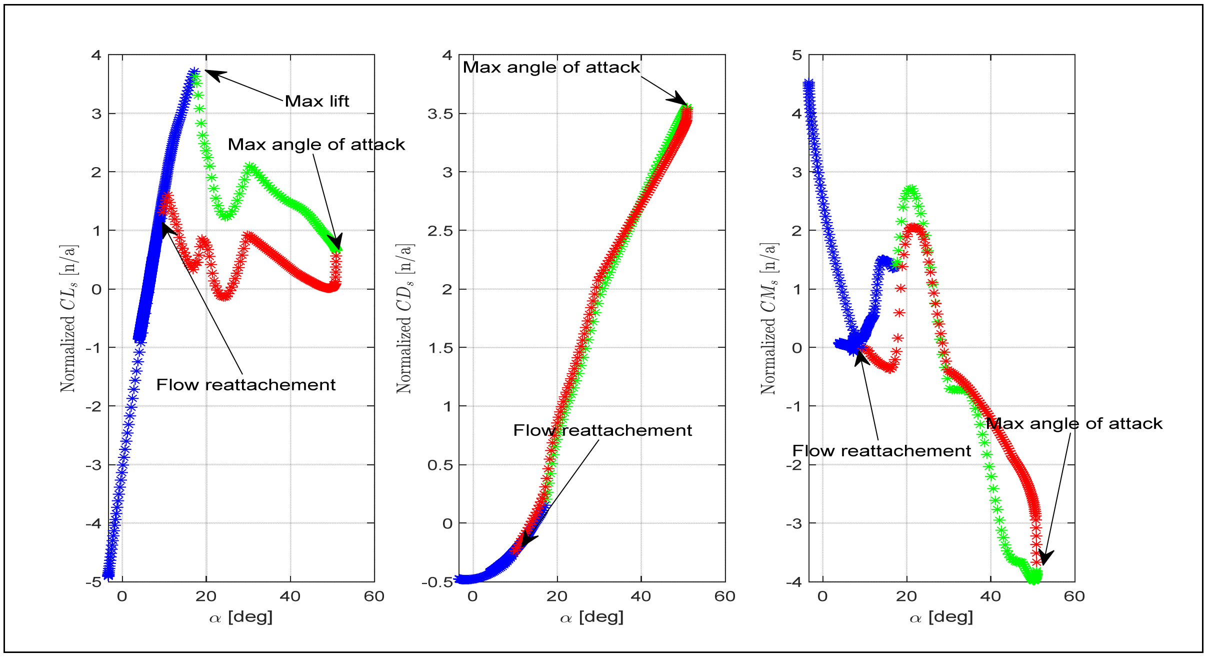


Figure 4.10 Dynamic stall sages highlighted on aerodynamic coefficients' variation with respect to the angle of attack for the flight test at 17,500 ft with slats retracted

In the first region (in blue), the boundary layer is attached to the wing. The lift, drag, and pitching moments follow approximately the unsteady linear thin airfoil theory trends (Spentzos et al., 2005). Then, the reversed flow starts at the rear of the airfoil and moves forward to the leading-edge region until the lift coefficient reaches its maximum value. In the second region (in green), which represents the static stall, vortices begin to appear near the leading edges and gradually move towards the trailing edge as the angle of attack increases. This is the beginning of the flow separation. The aircraft remains controllable despite the loss of lift. The third region (in red) (the dynamic stall regime), is characterized by a succession of secondary vortices, leading to highly nonlinear pressure fluctuations (McCroskey, 1981) and complete separation of the flow.

Due to the highly nonlinear nature of the aerodynamic coefficients in the dynamic stall regime, it was found necessary to label the data so that the NN models could understand whether they were predicting aerodynamic coefficients inside or outside the dynamic stall region. For this reason, all data outside the dynamic stall region were labeled with 1, while all the data inside this area were labeled with 2. Mathematically, this notation was equivalent to define a new vector of labels $L = [l_1, l_2, \dots, l_n]$, such as:

$$l_i = \begin{cases} 1 & \text{if the data is in dynamic stall region} \\ 2 & \text{otherwise} \end{cases} \qquad i = 1, \dots, n \tag{4.9}$$

4.2.3.3 Definition of the Neural Networks' Inputs and Outputs

In this study, three different MLPs and three different RNNs were developed for the prediction of the aerodynamic coefficients of the Bombardier CRJ-700 (i.e., one MLP, and one RNN for each aerodynamic coefficient). This strategy was used because it was found that single output neural networks were more accurate and did not need a complex architecture to learn the correlation between the input data and the target values. Similarly, the prediction of aerodynamic coefficients does not necessarily require similar neural network structures (same number of hidden layers and neurons per hidden layer), as some coefficients are easier (e.g., CL_s and CD_s) to model than others (e.g., Cm_s).

Basically, the aerodynamic coefficients of an aircraft in stall conditions depend on the following variables: angle of attack α, Mach number M, pitch rate q, rate of change of the angle of attack $\dot{\alpha}$, true airspeed V_T (McCroskey, 1981), and the surface control deflections, such as the elevator angle δ_e, the horizontal stabilizer angle δ_H, and slats angle δ_s. In addition, depending on the aircraft configuration, the wing airflow may also be affected by the air coming from the engines. Thus, the aerodynamic coefficients may also depend on the engine thrust T (or thrust coefficient). Finally, it is worth noting that since the variation of aerodynamic coefficients with respect to the input variables is nonlinear, a better correlation can usually be found by adding the square or cube of those variables as inputs to the neural networks. Based on all these observations and assumptions, the input vector X was defined as follows:

$$X = \left[\alpha, \alpha^2, \alpha^3, \dot{\alpha}, M, M^2, M^3, \frac{q}{V_T}, \left[\frac{q}{V_T}\right]^2, T, \delta\right]^T \tag{4.10}$$

where δ represents all control surface deflections (i.e., δ_e, δ_H, and δ_s.).

The output $\hat{o}$ is one of the three aerodynamic coefficients that needs to be estimated:

$$\hat{o} = \{\widehat{CL}_s, \widehat{CD}_s, \widehat{Cm}_s\} \tag{4.11}$$

where $\widehat{CL}_s$, $\widehat{CD}_s$, and $\widehat{Cm}_s$ are the predicted lift, drag and pitching moment coefficients in the stability axis.

4.2.3.4 Data Organization and Model Performance Evaluation

Out of the 39 flight cases conducted on the Bombardier CRJ-700 VRESIM, 27 were used as training and test datasets, while the remaining 12 cases were used for validation purposes. Note that the training dataset was used to optimize the neural network weights, while the test dataset was used to determine the network performance. In addition, to assess the ability of the networks to learn the inherent relationships among the data, a training error was required. In this study, the training error, also called training performance, was calculated based on the Mean Square Error (MSE), and according to the following equation:

$$MSE_{TR}\ (w) = \frac{1}{n}\sum_{k=1}^{n}\left[\hat{o}_k(w_{i,j}) - o_k\right]^2 \tag{4.12}$$

where the subscript TR refers to the training set data, i is the position of the neuron on the layer j, o_k is the k^{th} training data, and $\hat{o}_k$ is the k^{th} value predicted by the network.

As mentioned above, the test dataset was used to evaluate the performance of the model based on data that was not considered during training. This dataset is mainly used to adjust the key parameters of the model, such as the learning function, the activation function, or the number of hidden layers. Thus, for a given set of n data points and a given set of values of weights $w_{i,j}$, the test error (or test performance) was calculated in the same way as for the MSE_{TR}, and according to the following equation:

$$MSE_{TE}\ (w) = \frac{1}{n}\sum_{k=1}^{n}\left[\hat{o}_k\left(w_{i,j}\right) - o_k\right]^2 \tag{4.13}$$

Finally, the validation dataset, consisting of 12 flight cases, was used to validate the final model. This validation set was used to demonstrate the accuracy of the trained network in predicting the aerodynamic coefficients based on new data that has not been used to train or optimize the network. For a given set of n validation data points, and a given set of values of weights $w_{i,j}$, the validation error (or validation performance) was calculated based on the Mean Absolute Relative Error ($MARE$), as follows:

$$MARE = \frac{1}{n}\left(\sum_{k=1}^{n}\left|\frac{\hat{o}_k - o_k}{\hat{o}_k}\right|\right) \times 100 \tag{4.14}$$

where o_k is the k^{th} experimental data used for validation, and $\hat{o}_k$ is the k^{th} value predicted by the network.

When evaluating the model performances, it is important to ensure that the data used to train the network is representative enough of the data used to test the network. Otherwise, the network might be able to correctly interpolate the trained data but would fail to predict the aerodynamic coefficients for a new data set, which was not used for training. Such a problem is called “overfitting” (Yeom et al., 2018). To avoid this situation, it was decided to use a k-fold cross-validation to appropriately split the data from the 27 flight cases into training and test sets. The k-fold cross-validation method randomly divides the data into k-groups (or folds) of approximately equal sizes. Each element in the data sample is assigned to an individual fold and stays in that fold for the duration of the procedure. The network is then trained k times. At each iteration, a unique group between the k^{th} folds is used as the test dataset. The remaining $(k-1)$ folds constitute the training dataset, which means that each element is used as test data one time and is used to train the network $(k-1)$ times.

Figure 4.11 shows an example of data management for a 4-fold cross-validation procedure.

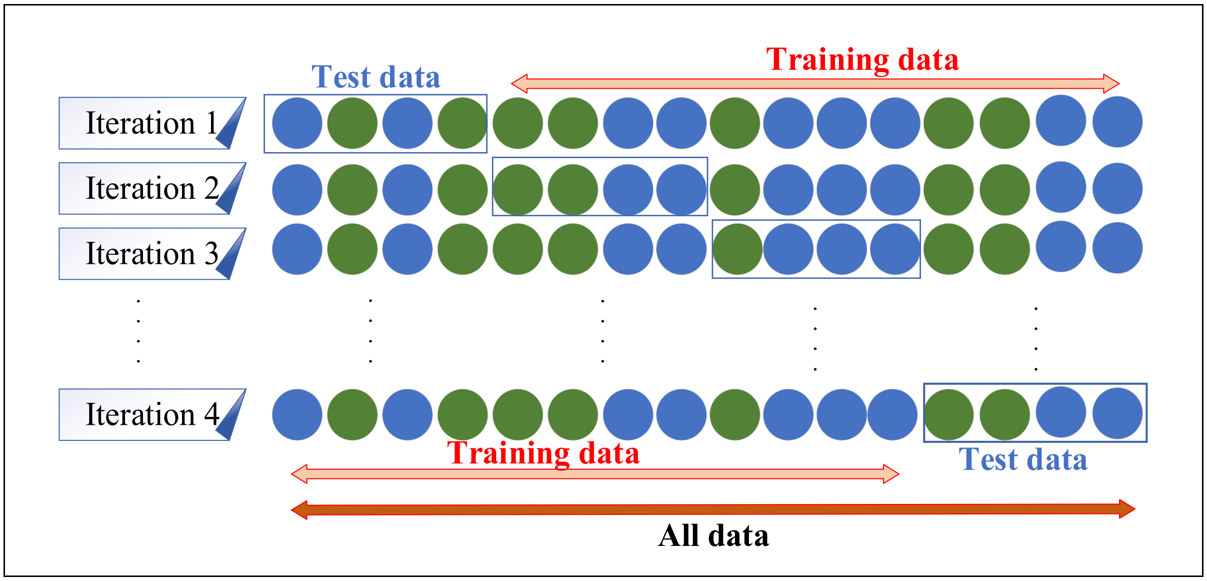


Figure 4.11 Example of data management for a 4-fold cross-validation procedure

The value for k should be chosen such that each train/test fold of data is large enough to be statistically representative of the general dataset. For most applied machine learning problems, a value of 5 to 10 is recommended (Garvin et al., 2013). Since the database used in this study was relatively large, a value of $k = 10$ was chosen. The performance of the trained model was computed as the mean of the performances obtained at each iteration.

4.2.3.5 Choice of Training Algorithm and Activation Function

The activation function, combined with the choice of training algorithm, influences the overall performances of the network and its training time. Thus, in order to determine the best activation function/learning algorithm combination, two analyses were performed.

The first analysis consisted of evaluating the performance of the network for different training functions. For this purpose, nine gradient-based local optimization algorithms were used to train the network. These algorithms are listed in Table 4.4.

Table 4.4 Training algorithms considered to train the networks

Algorithm	**Description**
OSS	One-step secant backpropagation
CGP	Conjugate gradient backpropagation with Polak-Ribiére updates
CGB	Conjugate gradient backpropagation with Powell-Beale restarts
CGF	Conjugate gradient backpropagation with Fletcher-Reeves updates
BFG	BFGS[2] quasi-Newton backpropagation
RP	Resilient backpropagation
SCG	Scale conjugates gradient method
LM	Levenberg-Marquardt optimization.
BR	Bayesian regularization backpropagation

Both MLP and RNN networks were trained with these nine training algorithms. For this first analysis, the activation function and the structure of the neural network were assumed to be the same. Each training algorithm was then used to determine the weights and biases that minimized the training error (MSE_{TR}). The performance of each neural network was evaluated using the test error (MSE_{TE}).

Figure 4.12 shows the test error MSE_{TE} value obtained for each training algorithm. Note that, for the sake of clarity, the results are presented in this figure for the MLP network, and for the

[2] Broyden-Fletcher-Goldfarb-Shanno

prediction of the lift coefficient CL_s of the Bombardier CRJ- 700. Similar types of results were obtained using the RNN and for the other coefficients.

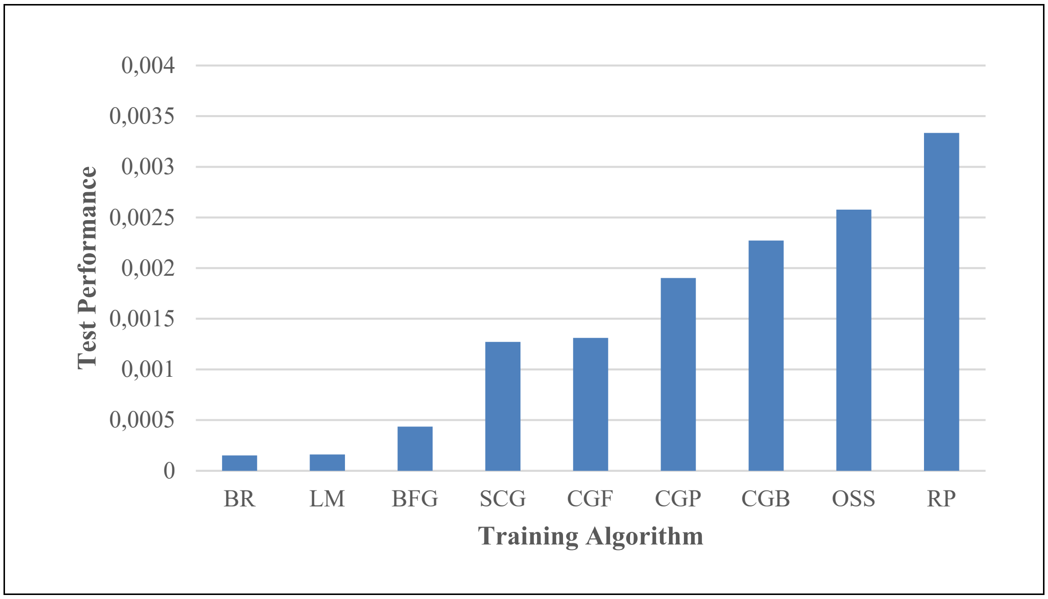


Figure 4.12 Network performance variation for different training algorithms

The results in Figure 4.12 clearly show that both the Bayesian Regularization (BR) and Levenberg-Marquardt (LM) and) algorithms provided the lowest MSE_{TE} value. This result was expected, as both the BR and LM algorithms are well known for their excellent performances in solving nonlinear regression problems. Both algorithms operate using the same procedure, except that in the BR algorithm, a backpropagation is used to compute the Jacobian of the network performance with respect to the weights and bias variables (MacKay, 1992).

Once the two best training algorithms were identified, a second analysis consisted in testing several activation functions, and evaluating their impact on the network performance. Based on previous results, it was decided to train a unique NN structure model with several activation functions, using both the BR and LM algorithms, as they gave the best and very similar performances ($MSE_{TE} = 1.55 \times 10^{-4}$ for the BR, and $MSE_{TE} = 1.63 \times 10^{-4}$ for the LM). In addition, although the BR performed relatively better than the LM, we were wondering if it will still be the case when changing the activation function.

Following the work of Maca, Pech & Pavlasek (2014), the tested activation function and their respective formulas are presented in Table 4.5.

Table 4.5 Implemented activation function: a is the neuron's activation, y is the neuron's output

Activation Function	**Mathematical Equation**
Log Sigmoid (logsig)	$y(a) = \frac{1}{1 + exp\ (-a)}$
Hyperbolic Tangent Sigmoid (tansig)	$y(a) = \frac{2}{(1 + exp(-2 * a))} - 1$
Elliot Symmetric Sigmoid (elliotsig)	$y(a) = \frac{a}{(1 + \lvert a \rvert)}$
Radial basis (radbas)	$y(a) = exp\ (-a^2)$
Normalized radial basis (radbasn)	$y(a)_i = \frac{exp(-a_i^2)}{\sum_{j=1}^{n} exp\ (-a_i^2)}$ where a is the input vector to a soft max function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Soft max (softmax)	$y(a)_i = \frac{exp(a_i)}{\sum_{j=1}^{n} exp\ (a_j)}$ where a is the input vector to a soft max function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Saturating linear (satlin)	$y(a) = \begin{cases} 0, & if\ a \leq 0 \\ a, & if\ 0 \leq a \leq 1 \\ 1, & if\ 1 \leq a \end{cases}$
Symmetric saturating linear (satlins)	$y(a) = \begin{cases} -1, & if\ a \leq -1 \\ a, & if -1 \leq a \leq 1 \\ 1, & if\ 1 \leq a \end{cases}$
Triangular basis (tribas)	$y(a) = \begin{cases} 1 - \lvert a \rvert, & if -1 \leq a \leq 1 \\ 0, & otherwise \end{cases}$
Positive linear (poslin)	$y(a) = \begin{cases} a, & if\ a \geq 0 \\ 0, & if\ a \leq 0 \end{cases}$

Figure 2.11 shows the test error MSE_{TE} value obtained for each activation function when MLP was trained for predicting the lift coefficient CL_s of the Bombardier CRJ-700 with the LM and BR algorithms. As expected, the performance of each algorithm depends on the activation function. As illustrated in Figure 2.11, both algorithms, associated with a $Tansig$ transfer function, gave the best results in terms of their smallest values. Indeed, the MSE_{TE} obtained for the two combinations ($LM, Tansig$) and ($BR, Tansig$) are 3.18×10^{-4} and 2.59×10^{-4}, respectively. Thus, any of these two combinations can be used to learn and predict the lift coefficient accurately. However, to strictly respect the optimization process, it was decided to use the BR training algorithm associated with the $Tansig$ activation function for the determination of CL_s with MLP, as that combination gave the minimum MSE_{TE} value.

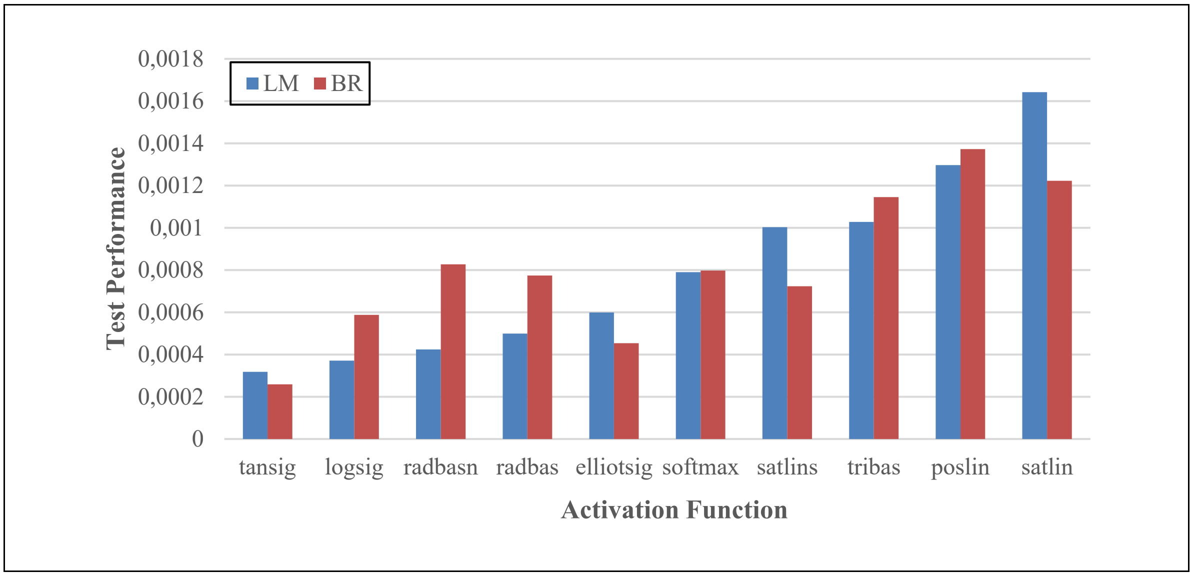


Figure 4.13 Network performance variation for various activation functions

The procedure presented in this section for determining the optimal training algorithm and activation function for the of CL_s estimation was repeated for the other two aerodynamic coefficients, as well as for the RNN. The resulting training algorithms and activation functions for all trained models are presented in section 4.3.

4.2.3.6 Neural Network Structure Optimization

Finally, the last step of the neural network design process was to determine the optimal structure that provides the best performance while ensuring a relatively acceptable learning time. The "optimal" structure can be defined in terms of the (m, n) combination, where m is the number of layers of the neural network, and n is the number of neurons on each layer resulting in the minimum test error MSE_{TE}. To do this, a procedure similar to the one presented in the previous section was considered. This procedure aimed to test different neural network structures, and to determine which one provided the best performance in an appropriate learning time.

In Ghazi et al. (2017) and Suresh et al. (2003), similar studies were conducted. However, in these studies, the authors considered only the static stall and bounded the number of layers to 3, and the number of neurons per layer to 10 and 3, respectively. Since the study presented in this paper includes the dynamic stall, it was assumed that the network structure would be more complex than the ones used in Ghazi et al. (2017) and Suresh et al. (2003), because of the highly non-linear nature of the data. Based on this observation, a minimum number of hidden layers $m_{min} = 1$, and a maximum number of $m_{max} = 5$ was assumed. Similarly, the minimum number of neurons per layer was set to $n_{min} = 1$, and the maximum number was set to $n_{max} = 15$. This range of parameters leads to 15^5 possible structures, which is clearly very large. To reduce the number of possible structures, it was assumed that all hidden layers should have the same number of neurons. Such an assumption was considered because it was found that varying the number of neurons from one hidden layer to another did not significantly improve the network performance. This assumption reduces the number of possible structures from 15^5 to 75.

The optimization procedure for determining the optimal network structure is summarized in Figure 4.14.

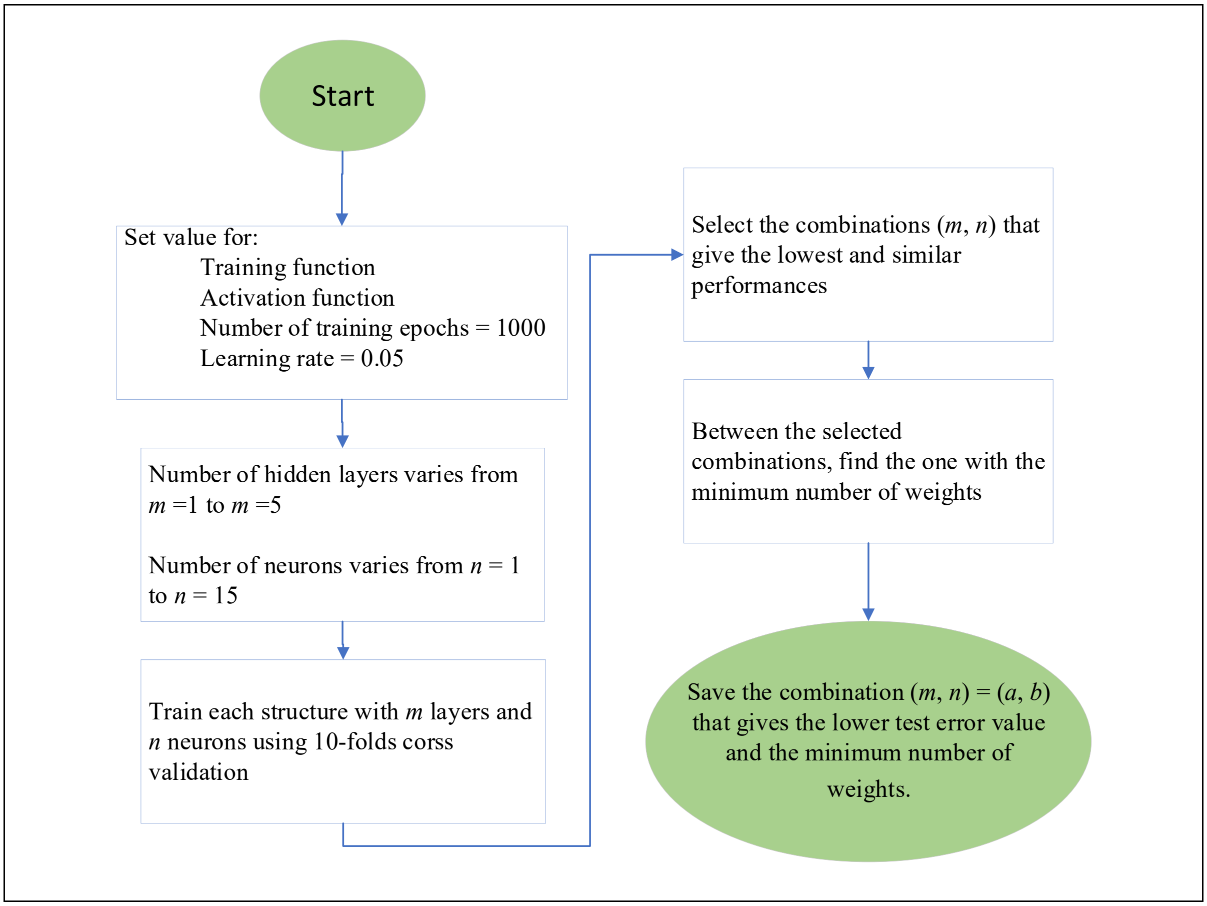

Start
Set value for:
Training function
Activation function
Number of training epochs = 1000
Learning rate = 0.05
Number of hidden layers varies from *m* =1 to *m* =5
Number of neurons varies from *n* = 1 to *n* = 15
Train each structure with *m* layers and *n* neurons using 10-folds corss validation
Select the combinations (*m*, *n*) that give the lowest and similar performances
Between the selected combinations, find the one with the minimum number of weights
Save the combination (*m*, *n*) = (*a*, *b*) that gives the lower test error value and the minimum number of weights.

Figure 4.14 Neural network structure optimization procedure.

The number of hidden layers m varies from 1 to 5, and the number of nodes n varies from 1 to 15. Each of the 75 structures is trained 10 times using the 10-fold cross validation procedure described in section 4.2.3.4. Figure 4.15 shows the variation of the MSE_{TE} for each (m, n) combination. These results were obtained for the estimation of the lift coefficient for the Bombardier CRJ-700 using MLPs. Note that structures with 1, 2 and 3 neurons have been removed for scaling purposes, because their test errors were too high compared to the errors obtained with the other structures. Figure 4.15 clearly shows that the MSE_{TE} decreases with the number of neurons and layers. However, it can be observed that the MSE_{TE} of the networks has negligeable variations for structures with more than $m = 2$ layers, and more than $n = 9$ neurons per layer; except for some cases ((4, 13), (5,9) and (5, 11)) that gives unexpected high MSE_{TE} values probably due to overtraining. Figure 4.16 shows the MSE_{TE} (on the left axis) and the number of weights and biases (on the right axis) obtained for networks with 2 to 5 layers, and with 9 to 15 neurons. The x-axis represents the network structure (m, n), from the lowest to the highest MSE_{TE} value. The lowest performance is obtained for the structure $(5, 14)$, with a MSE_{TE} of 1.39×10^{-5}. However, a similar MSE_{TE} of 2.39×10^{-5} was obtained for the structure $(3, 15)$, with a significant reduction of the weights and biases (from 1079 to 736). A high number of weights and biases means that the network will take more time to be trained, as it should update all the weights and biases to find their optimal values that minimize the training errors. In addition, a network with a high number of weights and biases will require more storage (memory) to be implemented in a real system. Thus, the optimal MLP structure in this case, with a good trade-off between performance and learning time, turned out to be a 3 hidden-layer structure, with 15 neurons per hidden layer.

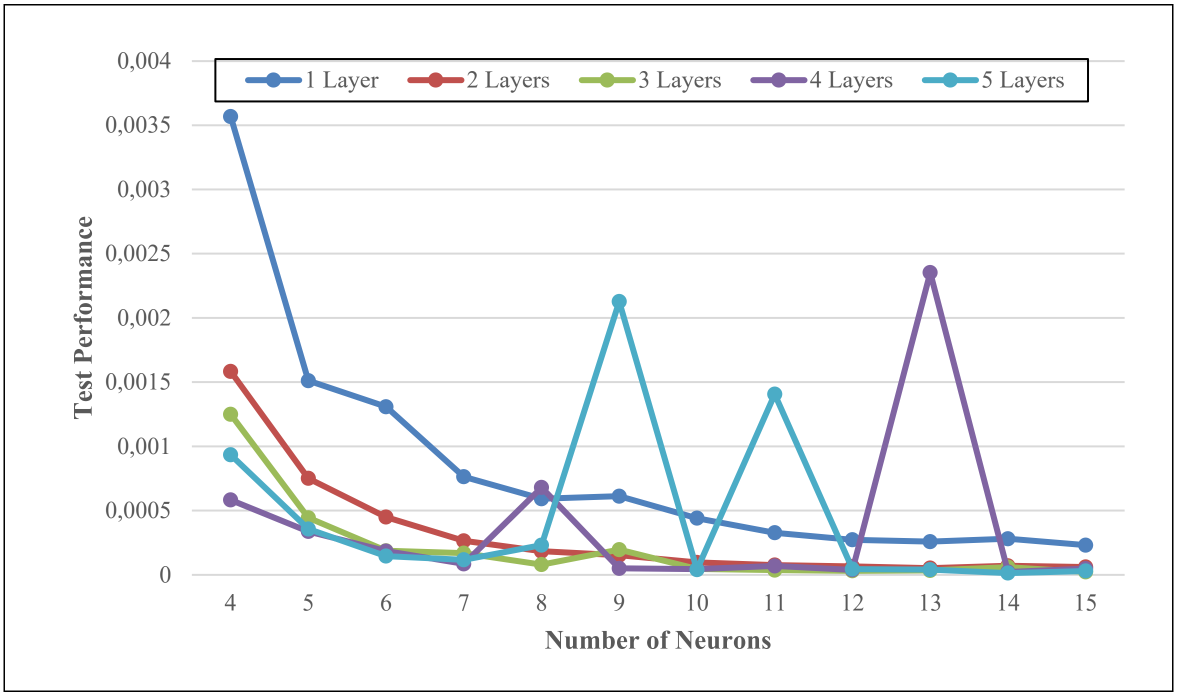


Figure 4.15 Performances for various MLP structures for the estimation of the CL_s of the Bombardier CRJ-700

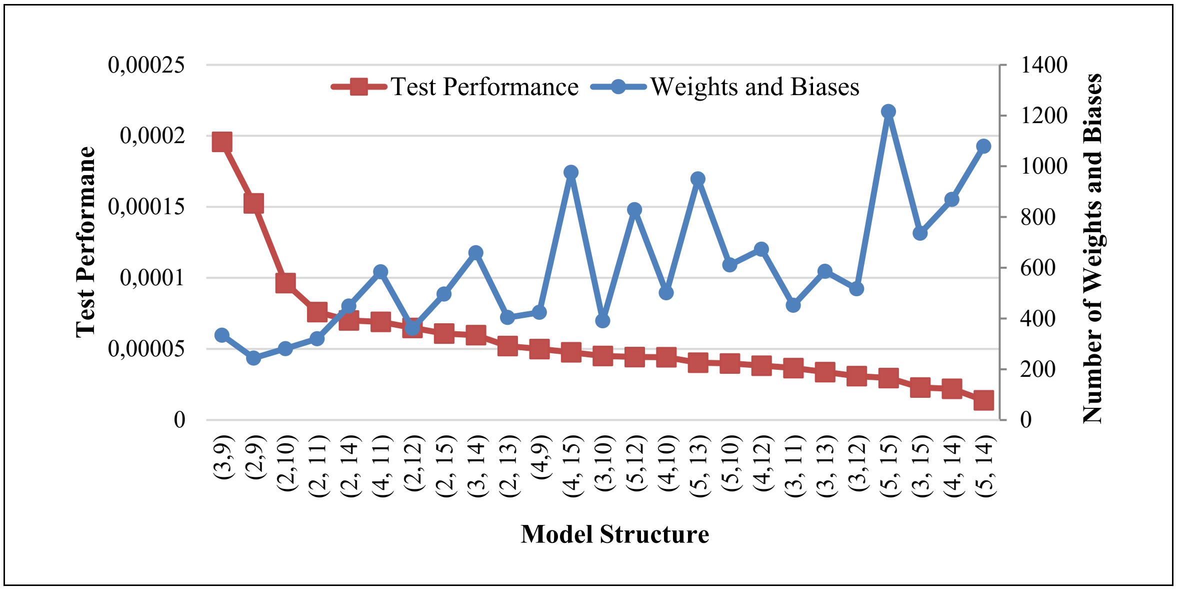


Figure 4.16 Performances and corresponding number of weights and biases for various MLP structures for the estimation of the CL_s of the Bombardier CRJ-700.

4.3 Results

This section presents the validation results of the proposed methodology. To demonstrate the effectiveness of the two neural networks obtained (i.e., MLP and RNN), two analyses were performed. The first analysis consisted in evaluating the accuracy of the neural networks in terms of the prediction of aerodynamic coefficients. For this purpose, the outputs of both the MLP and the RNN models were compared with experimental aerodynamic coefficients' data obtained from the Bombardier CRJ-700 VRESIM. However, since there are no criteria in the literature for the validation of aircraft aerodynamic coefficients, a second analysis was performed. This analysis aimed at implementing the trained neural networks models in a developed CRJ-700 simulation platform to reproduce several stall scenarios performed with the VRESIM. The validation in this case was realized by comparing several flight parameters predicted by the developed platform with those measured with the VRESIM. The tolerances established to validate these models were based on criteria defined by the Federal Aviation Administration (FAA) in the Manual of Criteria for the Qualification of Flight Simulators, (FAA, 1991; ICAO, 2016) corresponding to the level D.

4.3.1 Validation of the Aerodynamic Coefficients

As explained in section 4.2, out of the 39 flight cases performed on the CRJ700 VRESIM, 27 (70%) were used to train the neural networks, while the remaining 12 (30%) were used for validation purposes. Two models were developed – one based on a MLP structure, and another based on a RNN structure – and trained using the procedure described in section 4.2 for the determination of lift, drag and pitching moment aerodynamic coefficients. The optimal parameters obtained for each of the two models are presented in Table 4.6.

Table 4.6 Optimal parameters obtained for the MLP and RNN

Parameters	**MLP**			**RNN**		
	$\boldsymbol{CL_s}$	$\boldsymbol{CD_s}$	$\boldsymbol{Cm_s}$	$\boldsymbol{CL_s}$	$\boldsymbol{CD_s}$	$\boldsymbol{Cm_s}$
Training algorithm	*BR*	*LM*	*BR*	*BR*	*LM*	*BR*
Activation function	*tansig*	*logsig*	*tansig*	*tansig*	*logsig*	*tansig*
Number of hidden layers	3	5	5	3	3	5
Number of nodes per hidden layer	15	12	13	11	13	11
Number of epochs	1000	1000	1000	1000	1000	1000
Test error $\boldsymbol{MSE_{TE}}$	2.39×10^{-5}	4.31×10^{-5}	6.1×10^{-5}	2.95×10^{-5}	6.24×10^{-5}	5.83×10^{-5}

As shown in Table 4.6, the best performances in terms of MSE_{TE} for the estimation of the lift, drag and pitching moment coefficients were obtained with the BR and LM algorithms, combined with the *tansig* and *logsig* activation functions. Furthermore, it can be noted that the RNN model performs as well as the MLP model, but with a less complex structure, as the total number of neurons of the hidden layers ($m \times n$) are sometimes smaller. Indeed, for the prediction of the drag coefficient, for instance, the MLP model requires $m = 5$ hidden layers and $n = 12$ neurons per hidden layer to achieve a test error of $MSE_{TE} = 4.31 \times 10^{-5}$, while the RNN model needs a structure of only $m = 3$ hidden layers and $n = 13$ neurons per hidden layer, resulting in a test error of $MSE_{TE} = 6.24 \times 10^{-5}$. This observation can be explained by the fact that the RNNs, by their nature, are structurally designed to consider the dynamic behavior of complex systems, such as the variation of aerodynamic coefficients under stall conditions. However, the MLP structure is still able to give similar type of results, as the dynamics of stall phenomena could be considered by adding the pitch rate q, as well as the rate of change of the angle of attack $\dot{\alpha}$, as inputs to the network. Globally, most of all the tested errors MSE_{TE} obtained with MLP and RNN models are very small of order of 10^{-5} or 10^{-4}. Therefore, we can globally say that the obtained results in Table 4.6 are excellent.

Figure 4.17 to Figure 4.19 show examples of the estimated aerodynamic coefficients compared with their experimental data (plotted with respect to the angle of attack) for three different slat angles. From a general perspective, the data predicted by the MLP (in red) and by the RNN (in blue) models match the experimental data obtained by VRESIM (in black) quite well, especially during the hysteresis cycle. The results revealed that all coefficients were estimated with a residual error of less than 10^{-3}. This error is very small and allows to conclude that both models were able to successfully predict the aerodynamic coefficients for these three flight cases.

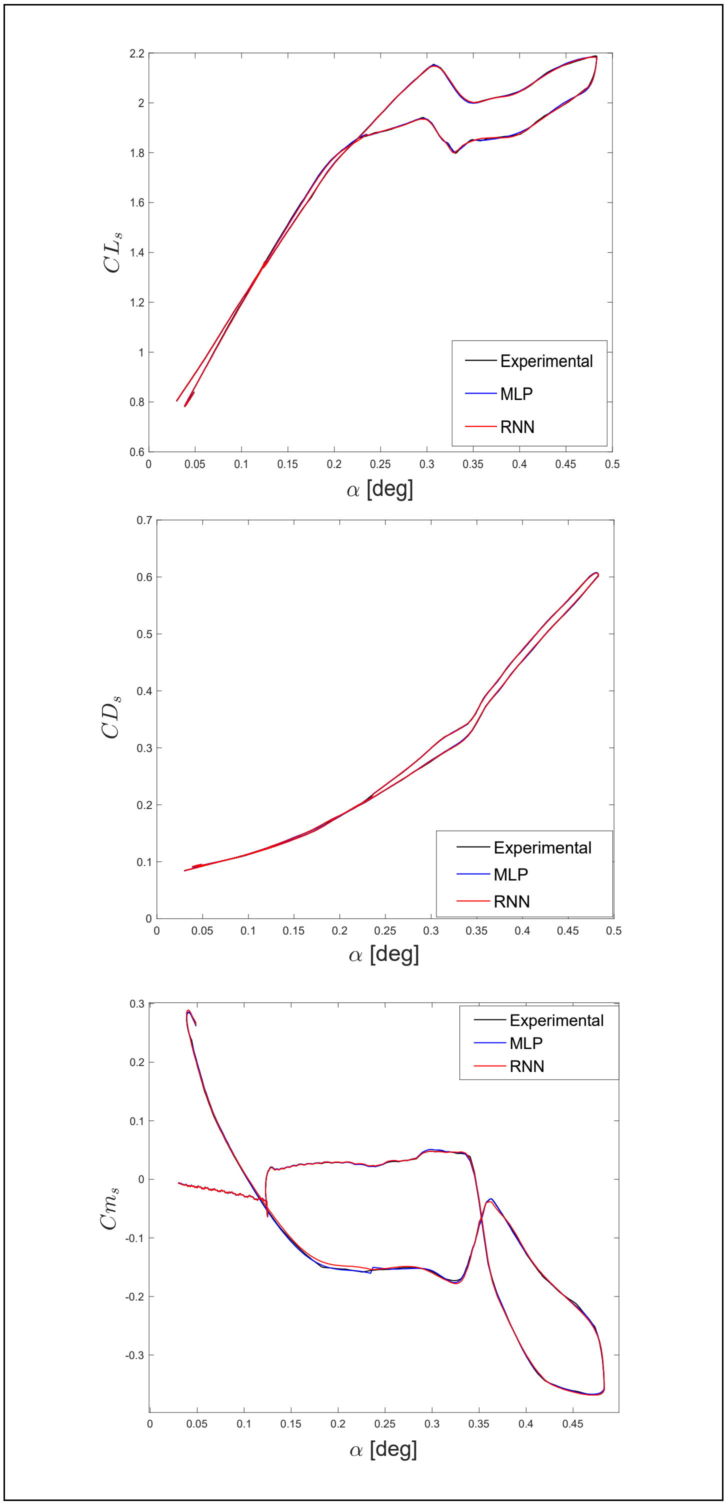


Figure 4.17 Examples of predicted aerodynamic coefficients for a flight test at 22,500 ft with slats at 20

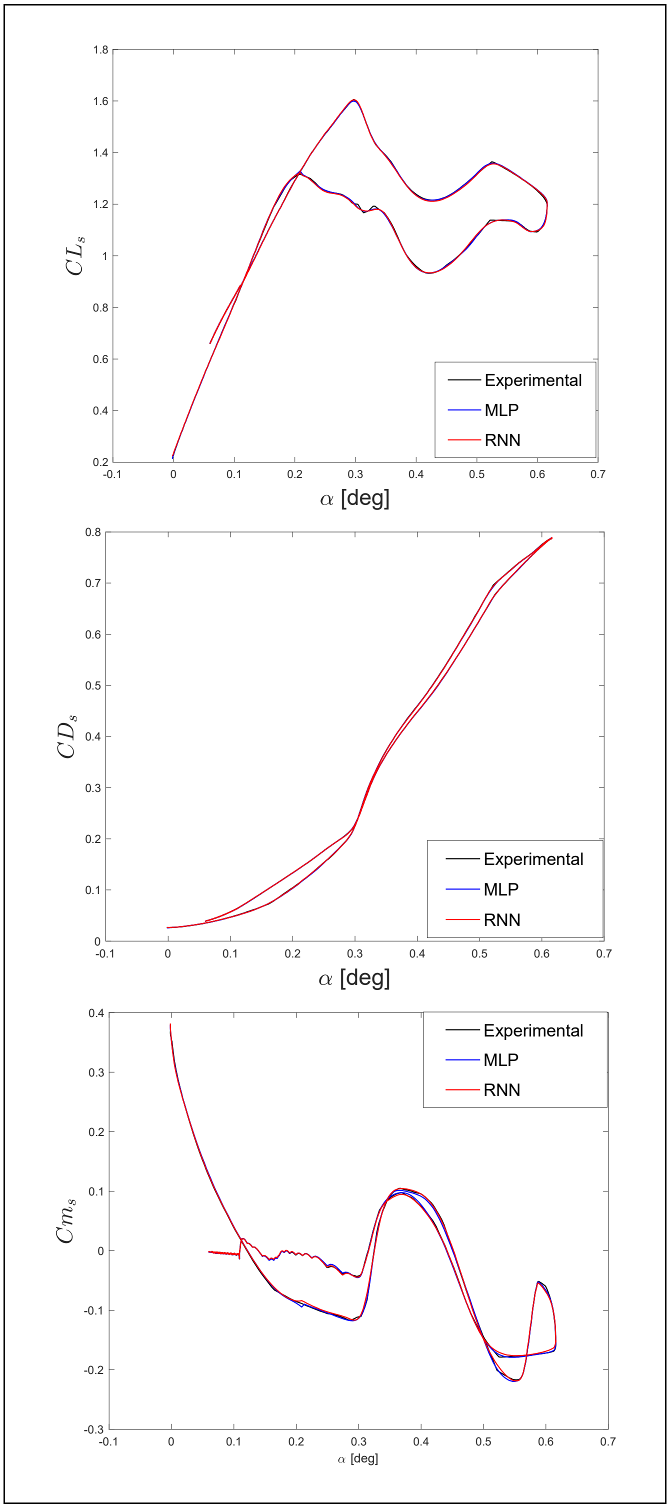


Figure 4.18 Examples of predicted aerodynamic coefficients for a flight test at 25,000 ft with slats retracted

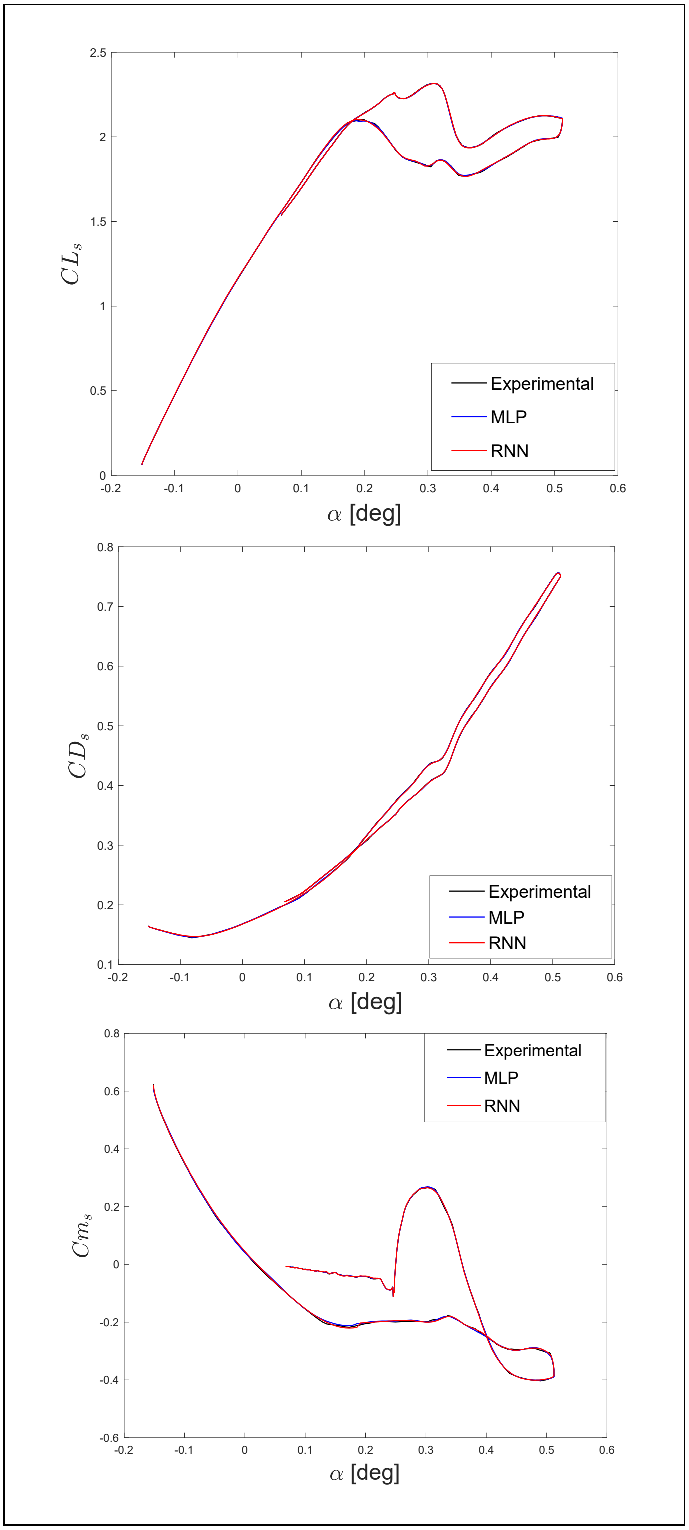


Figure 4.19 Examples of predicted aerodynamic coefficients for a flight test at 30,000 ft with slats at 45

The analysis presented in the previous three examples was repeated for all the other validation flight cases. The results obtained in terms of Mean Absolute Relative Error ($MARE$) for each aerodynamic coefficient for the MLP and RNN models are presented in Table 4.7 and Table 4.8, respectively.

Table 4.7 Mean Absolute Relative Error ($MARE$) and Mean Absolute Residual obtained between experimental data and predicted values with MLP models

Flight Case		CL_s		CD_s		Cm_s	
Altitude [ft]	**Slat angle [in º]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-4}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-4}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-4}]**
7500	0	0.07	5.72	0.19	1.29	1.45	4.70
15,000	0	0.06	5.22	0.16	1.25	0.73	5.61
25,000	0	0.09	9.01	0.16	1.76	7.90	7.10
35,000	0	0.13	8.35	0.21	1.46	6.41	4.14
15,000	20	0.07	9.40	0.07	1.28	10.93	5.07
17,500	20	0.05	7.55	0.07	1.05	7.59	6.08
22,500	20	0.07	10.7	0.07	1.26	1.66	5.82
32,500	20	0.07	8.57	0.11	1.51	10.34	5.38
17,500	45	0.06	10.1	0.08	2.59	1.12	8.32
20,000	45	0.11	12.4	0.08	2.71	1.45	9.46
30,000	45	0.08	10.3	0.08	2.39	1.55	9.59
35,000	45	0.06	9.80	0.08	2.34	1.64	9.27

Table 4.8 Mean Absolute Relative Error ($MARE$) and Mean Absolute Residual obtained between experimental data and predicted values with RNN models

Flight Case		CL_s		CD_s		Cm_s	
Altitude [ft]	**Slat angle [in º]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-4}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-4}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-4}]**
7500	0	0.07	7.47	0.25	1.48	1.51	5.84
15,000	0	0.12	6.31	0.19	1.29	0.96	6.87
25,000	0	0.13	10.8	0.14	1.51	4.68	5.14
35,000	0	0.10	7.60	0.15	1.23	9.54	4.78
15,000	20	0.08	10.7	0.13	2.13	10.65	1.19
17,500	20	0.07	9.32	0.11	1.87	6.88	5.55
22,500	20	0.08	10.9	0.11	2.13	1.94	8.72
32,500	20	0.06	7.52	0.15	2.07	9.81	6.85
17,500	45	0.08	11.4	0.10	3.38	2.16	12.3
20,000	45	0.15	14.8	0.12	3.84	1.65	11.3
30,000	45	0.07	11.2	0.10	3.16	2.90	12.7
35,500	45	0.06	10.8	0.10	3.12	2.33	11.4

As respectively shown in Table 4.7 and Table 4.8, both MLP and RNN methodologies can globally estimate the lift, drag and pitching moment coefficients for each validation flight case, and with similar precision. However, it can be seen that for five flight cases, in each Table 4.7 and Table 4.8 the $MARE$ for the pitching moment coefficient Cm_s is higher than > 5%. After analyzing the results in detail for these flight cases, it was found that this aspect could be explained by the fact that the pitching moment coefficient changes its sign when the aircraft stalls (see Figure 4.17, Figure 4.18 and Figure 4.19). Consequently, the Cm_s has relatively low values, around zero, which lead to larger relative errors ($MARE$) than 5%. However, even if the $MARE$ errors on the Cm_s seem relatively large (around 10%) for some flight cases, the results obtained remain very good as the mean residual errors are very low, even negligible.

Finally, Table 4.9 summarizes the average errors $MARE$ obtained for all validation flight cases. As seen on Table 4.9, both MLP and RNN models were able to predict the lift and drag coefficients with an average $MARE$ of less than 0.14%. The pitching moment coefficient, on the other hand, was predicted with an average $MARE$ of about 4.5%. Nevertheless, as explained above, this percentage remains acceptable because it is due to the fact that the pitching moment coefficient can take a very small value, around zero. In terms of residual errors, the order of the average residual errors for the 12 validation flight cases is between 10^{-3} and $10^{-4}$for all three coefficients, which is much lower with respect to the order of the experimental data, which is 10^{-1}. This aspect therefore shows that the $MARE$ errors are negligible.

Table 4.9 Average $MARE$ and standard deviation of $MARE$
obtained for the prediction of CL_s, CD_s and Cm_s using MLP and RNN

	Number of validation cases	**Average $MARE$ [in %]**	**Standard deviation of the $MARE$ [in %]**
Estimation of C_{L_s} with MLP	12	0.08	0.02
Estimation of C_{D_s} with MLP	12	0.11	0.05
Estimation of C_{m_s} with MLP	12	4.40	3.92
Estimation of C_{L_s} with RNN	12	0.09	0.03
Estimation of $C_{D_s}$with RNN	12	0.14	0.04
Estimation of C_{m_s} with RNN	12	4.58	3.64

4.3.2 Flight Dynamics Model Validation

To further demonstrate the validity of both models, a second analysis was performed. The purpose of this analysis was to verify the accuracy of both models based on the behavior of the aircraft. Both the MLP and the RNN models were implemented into a simulation platform of the Bombardier CRJ-700, designed in Matlab/Simulink (see Figure 4.20). This simulation platform was used to reproduce the 12 validation flight cases and then to compare the flight parameters predicted by the platform with those obtained using the CRJ-700 VRESIM.

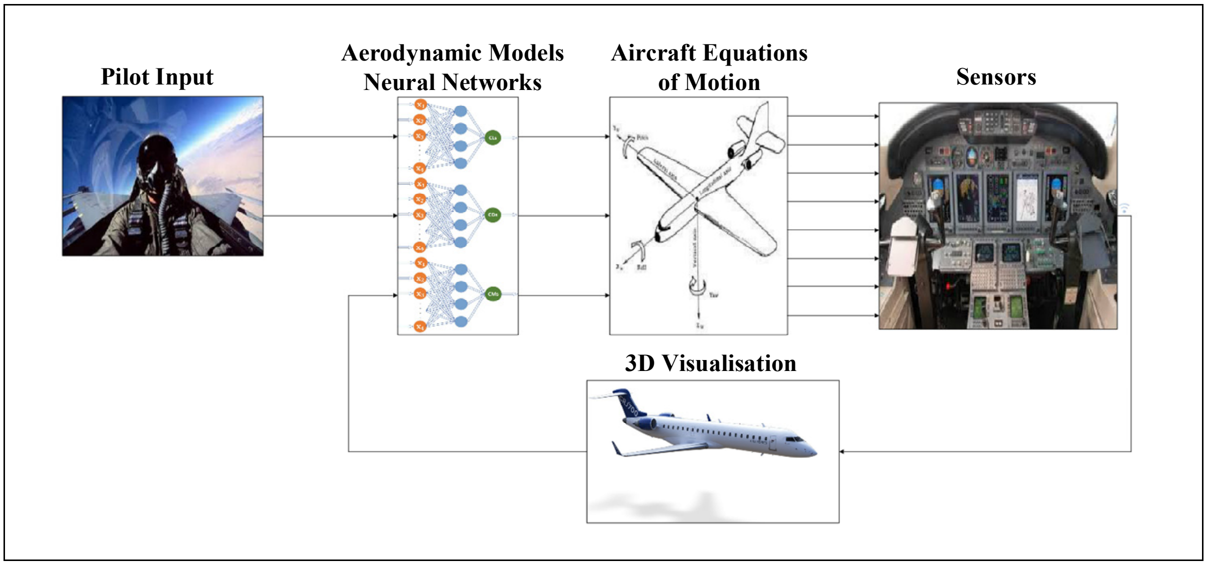


Figure 4.20 Developed Bombardier CRJ-700 flight dynamics platform

The criteria used to validate both models were defined by the FAA, which specifies the tolerances for both approaches to stall and static stall phases. The angle of attack and the pitch angle should be estimated within a tolerance of $\pm 2°$, and the true airspeed within $\pm 5\ ft/s$ ($3\ kts$). There were no specified criteria by the FAA for the altitude estimation or any other variable prediction in dynamic stall phases. For this reason, we established a reasonable tolerance of $\pm 39\ ft$ ($12\ m$) for the altitude estimation. The same criteria were considered to evaluate the accuracy of these models in the dynamic stall regime.

Figure 4.21 to Figure 4.23 show the comparison results obtained for three different flight cases. As shown in these figures, all the flight parameters values (angle of attack α, altitude h, pitch angle θ and true airspeed V_t) are found within their requested tolerances. The aircraft dynamics was very well predicted in both static and dynamic stall phases. Indeed, for all flight parameters, the outputs of the simulation platform are superposed on the outputs collected from the VRESIM, which implies by default that all tolerances are satisfied. Finally, the success rate obtained for the validation flight cases is 100%, which means that for all the flight conditions tested, the models predicted the flight parameters well within the tolerances.

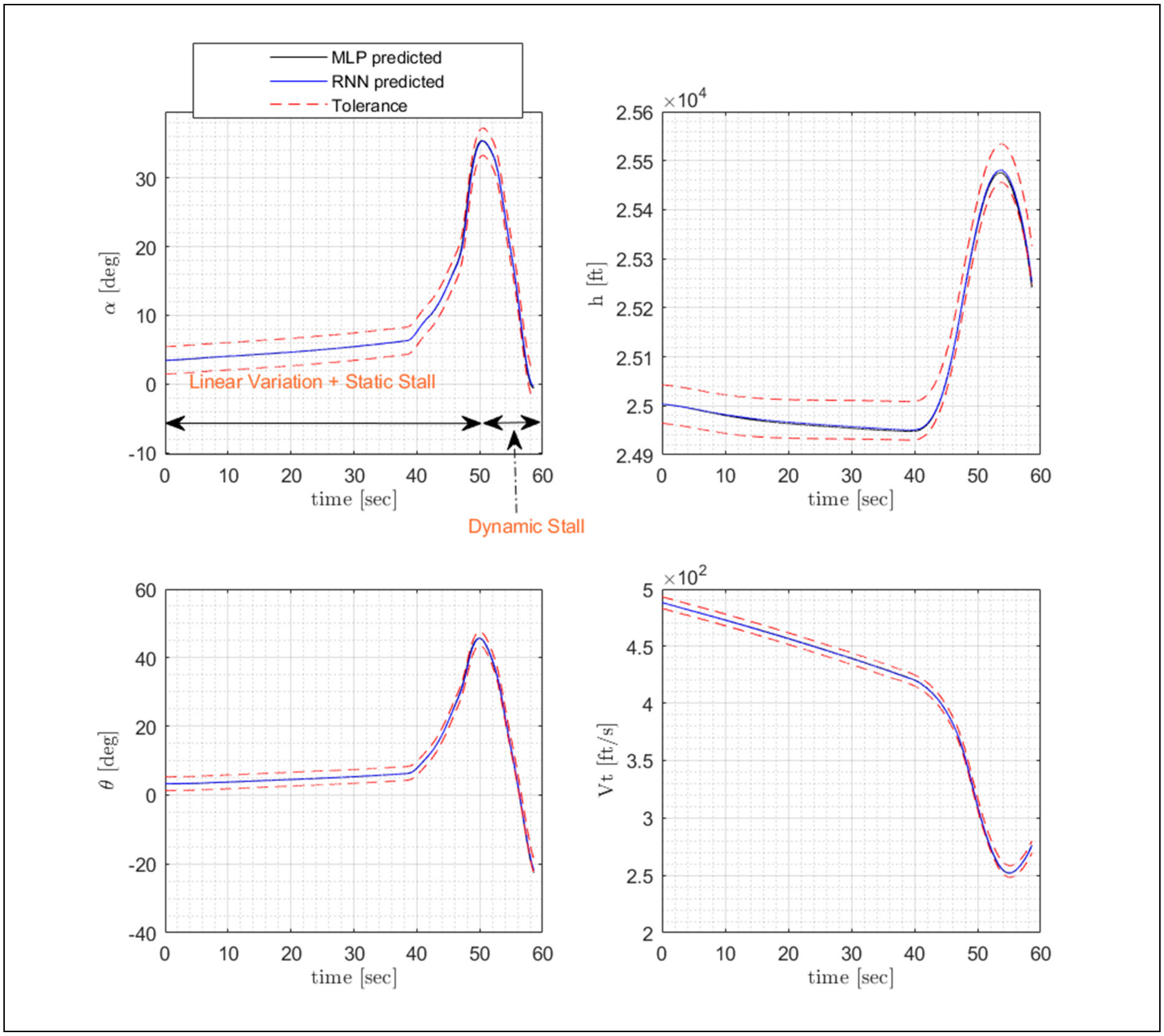


Figure 4.21 Example of predicted flight parameters for a flight test at 25,000 ft with slats retracted

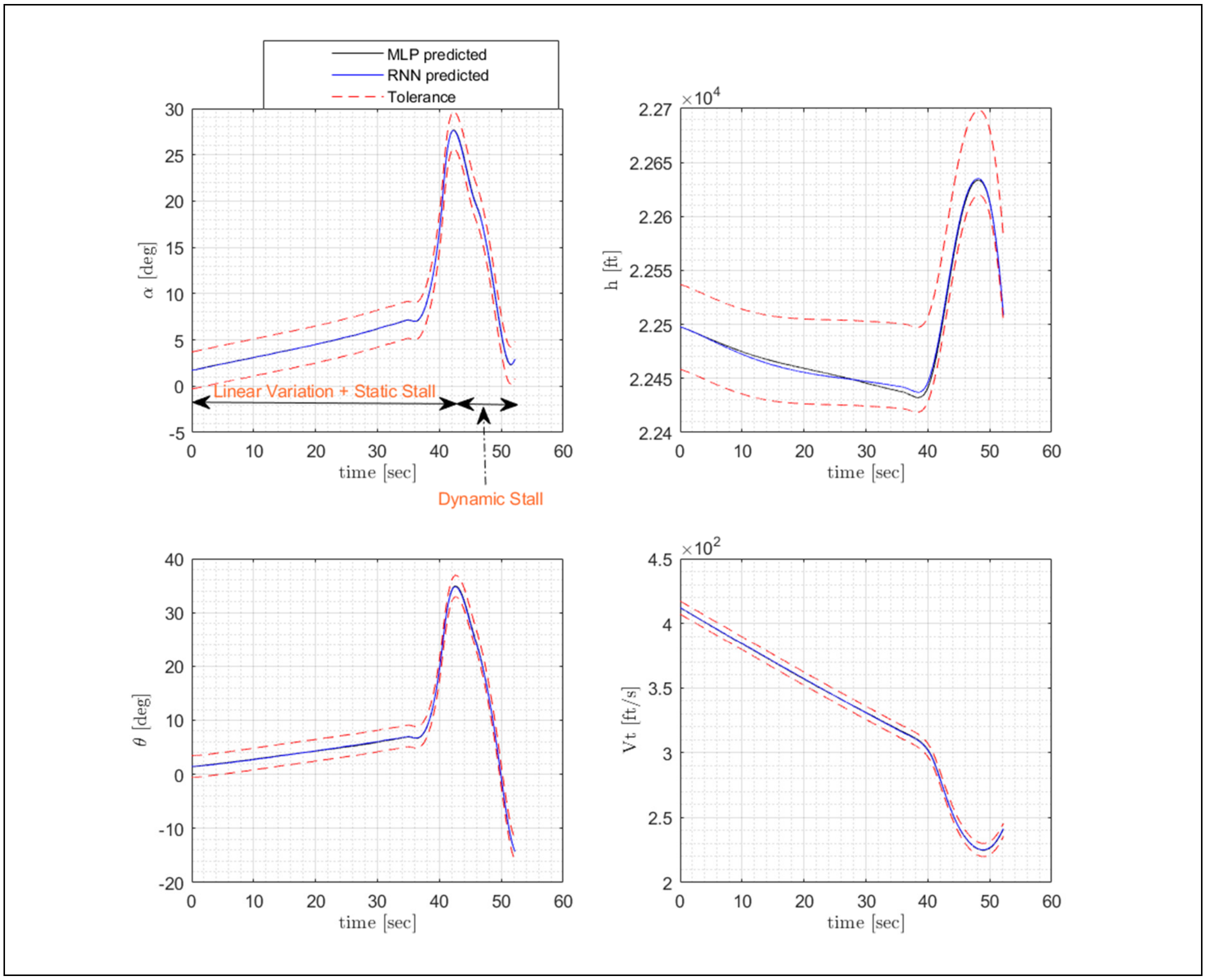


Figure 4.22 Example of predicted flight parameters for a flight test at 22,500 ft with slats at 20°

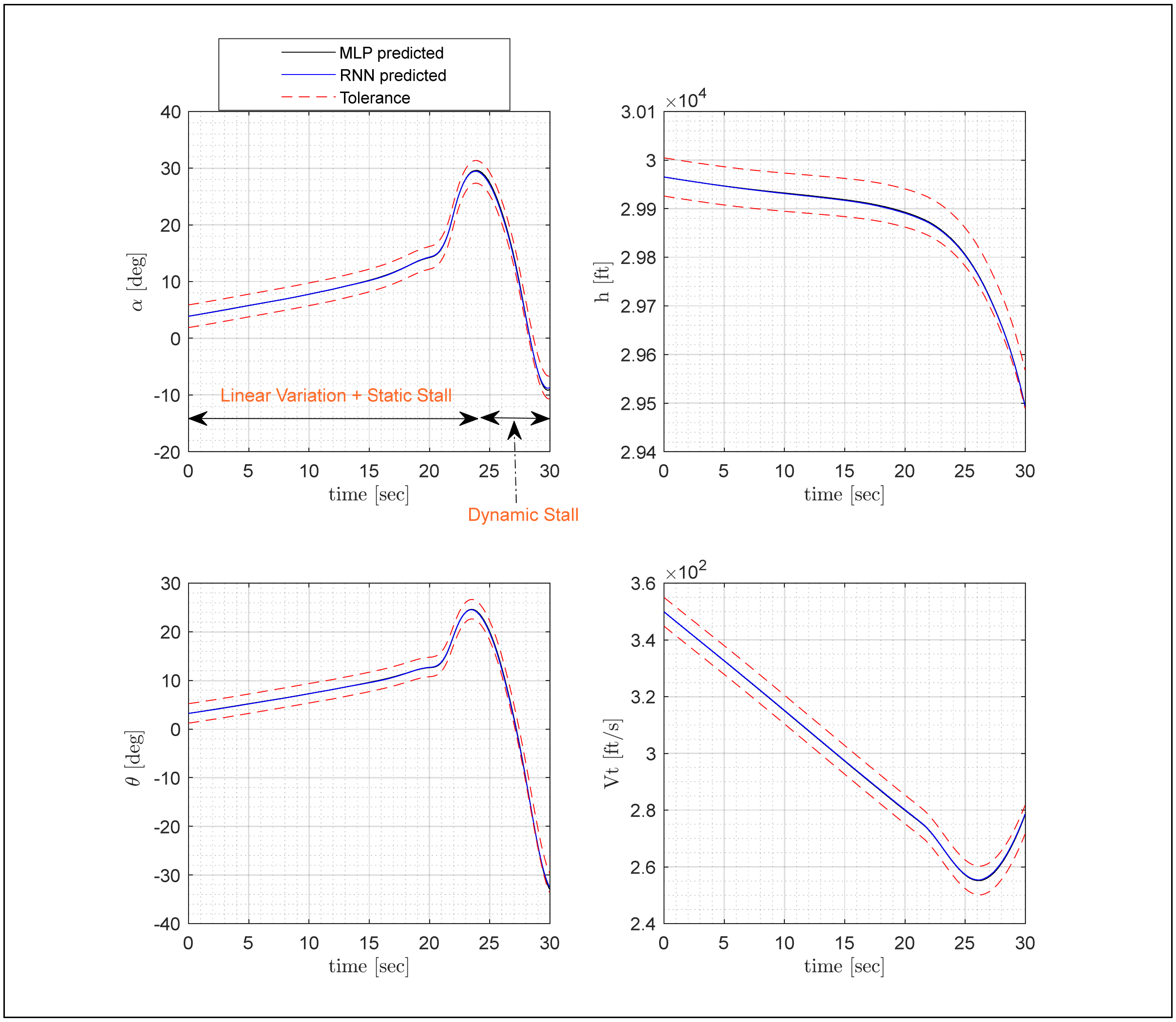


Figure 4.23 Example of predicted flight parameters for a flight test at 33,000 ft with slats at 45°

4.4 Conclusions

This paper presents a methodology to model the longitudinal aerodynamic coefficients and predict an aircraft's longitudinal dynamics during a stall recovery maneuver in static and dynamic stall conditions. This methodology was successfully applied on the CRJ-700 aircraft. The linear and nonlinear variations of lift, drag and pitching moment aerodynamic coefficients were estimated along the stall hysteresis curve. A system identification technique was used to build a mathematical model of aerodynamic coefficients of the aircraft, capable of predicting the coefficients' values as a function of the flight parameters. To achieve it, a large amount of flight data and a powerful tool were needed, such as neural networks, to analyze and learn from these flight data.

Flight test data were obtained from the CRJ-700 VRESIM designed and manufactured by CAE and Bombardier. The CRJ-700 VRESIM has a level D qualification for its flight dynamics, meaning that the data collected with it is similar to the actual data collected during flight tests on real aircraft. A total of 39 flight cases were conducted for 13 different altitudes ranging from 5000 to 35,000 ft and at 3 different slat configurations (0^o, 20^o and 45^o). Data from 27 (70%) flight cases were used to train the following two types of Neural Network models: multilayer perceptrons and recurrent neural networks. The MLP model was chosen for its ability to solve regression problems for which real data were predicted from input parameters, whereas the RNN model was considered for its effectiveness in designing models from timeseries data.

The procedure to select the neural network hyperparameters (training algorithms, activation function) was detailed. Nine training algorithms and ten different activation functions were tested. The results showed that the *BR* training algorithm, associated with the tansig activation function, was the best suited for training the lift and pitching moment coefficient models. In contrast, the LM training algorithm associated with the *logsig* activation function was the best suited for training the drag coefficient model (section 4.3, Table 4.6). In addition, a new algorithm was developed to determine the optimal number of layers and number of neurons to achieve a good compromise between small prediction errors and appropriate learning time.

The developed models were validated by comparing the aerodynamic coefficients estimated by the models with the experimental data from the CRJ-700 VRESIM for the remaining 12 (30 %) flight cases that were not used for model training. Both the MLP and RNN models were able to predict the aerodynamic coefficients with an average mean absolute relative error smaller than 0.1%, 0.15%, and 4.6 % for the lift, drag and pitching moment coefficients, respectively. These small errors allowed the conclusion that the developed models performed very well.

Finally, the developed models were then integrated into a simulated platform of a CRJ-700 airplane developed in MATLAB/Simulink to validate the CRJ-700 flight dynamics. The flight parameters (angle of attack α, altitude h, pitch angle θ and true airspeed V_t) predicted by the platform were compared with those obtained by the CRJ-700 VRESIM for the 12 validation flight cases. The criteria used to validate the models were defined by the FAA, which specifies the tolerances for estimating the angle of attack, the pitch angle, the true airspeed, and the altitude for approaches to both dynamic stall and static stall phases. The outputs of the simulation platform were superposed on the outputs collected from the VRESIM, showing that the FAA tolerances were satisfied. All 12 validation flight tests gave excellent results with a success rate of 100%.

CHAPITRE 5

IDENTIFICATION AND VALIDATION OF THE CESSNA CITATION X LONGITUDINAL AERODYNAMIC COEFFICIENTS IN STALL CONDITIONS USING MULTI-LAYER PERCEPTRONS AND RECURRENT NEURAL NETWORKS

Yvan Tondji [a], Mouhamadou Wade [b], Georges Ghazi [c], Ruxandra Mihaela Botez [d]

[a,b,c,d] Department of System Engineering, École de Technologie Supérieure,
1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3

Paper published in the *INCAS BULLETIN,* Vol. 14, No. 2, June 2022, pp. 103-119.
DOI: http://dx.doi.org/10.13111/2066-8201.2022.14.2.9

Résumé

L'augmentation du nombre d'accidents dans l'aviation générale due à la perte de contrôle des aéronefs nécessite le développement de modèles aérodynamiques d'avions précis. Ces modèles doivent indiquer les variations linéaires des coefficients aérodynamiques en vol stabilisé ainsi que les variations hautement non linéaires des coefficients aérodynamiques dues aux conditions de décrochage et post-décrochage. Cet article présente une méthodologie détaillée pour modéliser les coefficients aérodynamiques de portance, de traînée et de moment de tangage dans le régime de décrochage, en utilisant les réseaux de neurones « Neural Networks (NN) ». Une technique d'identification de système a été utilisée pour développer des modèles de coefficients aérodynamiques à partir de données de vol. Ces données ont été recueillies grâce à un simulateur de vol pour avions de recherche (RAFS) de niveau D qui a été utilisé pour exécuter les manœuvres de décrochage. Des perceptrons multicouches « Multilayer Perceptrons (MLP) » et des réseaux de neurone récurrents « Recurrent Neural Networks (RNN) » ont été utilisés pour apprendre à partir des données de vol et trouver des corrélations entre les coefficients aérodynamiques et les paramètres de vol. Cette méthodologie est employée ici pour optimiser les structures de réseaux neuronaux et trouver les hyperparamètres optimaux : algorithmes d'entraînement et fonctions d'activation utilisées pour apprendre les données. Les modèles aérodynamiques de décrochage développés ont été validés avec succès en comparant les coefficients aérodynamiques de portance, de traînée et de moment de tangage prédits pour des entrées de pilote données avec des données expérimentales obtenues à partir du RAFS Cessna Citation X pour les mêmes entrées de pilote.

Abstract

The increased number of accidents in general aviation due to loss of aircraft control has necessitated the development of accurate aerodynamic airplane models. These models should indicate the linear variations of aerodynamic coefficients in steady flight and the highly nonlinear variations of the aerodynamic coefficients due to stall and post-stall conditions. This paper presents a detailed methodology to model the lift, drag, and pitching moment aerodynamic coefficients in the stall regime, using Neural Networks (NN). A system identification technique was used to develop aerodynamic coefficients models from flight data. These data were gathered from a level-D Research Aircraft Flight Simulator (RAFS) that was used to execute the stall maneuvers. Multilayer Perceptrons (MLP) and Recurrent Neural Networks (RNN) were used to learn from flight data and find correlations between aerodynamic coefficients and flight parameters. This methodology is employed in here to optimize neural network structures and find ideal hyperparameters: training algorithms and activation functions used to learn the data. The developed stall aerodynamic models were successfully validated by comparing the lift, drag, and pitching moment aerodynamic coefficients predicted for given pilot inputs with experimental data obtained from the Cessna Citation X RAFS for the same pilot inputs.

5.1 Introduction

The need to improve aircraft safety has been one of the major concerns in the aviation industry (Cunningham et al., 2004). Over the last few years, the loss of control in-flight, one of the primary causes of flight accidents, has been addressed from different perspectives, one of them is using flight simulators to teach pilots how to execute stall recovering maneuvers to developing high-fidelity stall models (Ananda & Selig, 2016).

A high-fidelity stall model would enable the development of new control strategies (Shao et al., 2019), that would extend the operational envelope and capabilities of the next generation of aircraft. In fact, flying near-stall conditions could help increase an aircraft's lift capacity, reduce landing distances, and enable safe recovery from stall in case of emergency maneuvers.

Today, the most commonly used techniques for modeling aerodynamic coefficients in stall conditions can be classified into the following groups: semi-empirical methods (Botez, 1989), Computational Fluid Dynamics (CFD) methods (Spentzos et al., 2005), experimental techniques (Wernert et al., 1996), and system identification techniques (Tondji et al., 2022). Semi-empirical methods are based on approximations established based on experimental data obtained from flight or wind tunnel tests. They have the advantage of allowing the rapid modeling of a wide range of aerodynamic coefficients variations with angle of attack. However, in the case of stall modelling, the use of semi-empirical methods has some limitations due to the highly non-linear nature of the stall phenomenon and the quality of the existing database, which might not be sufficiently representative of the system under test (Albisser, 2015).

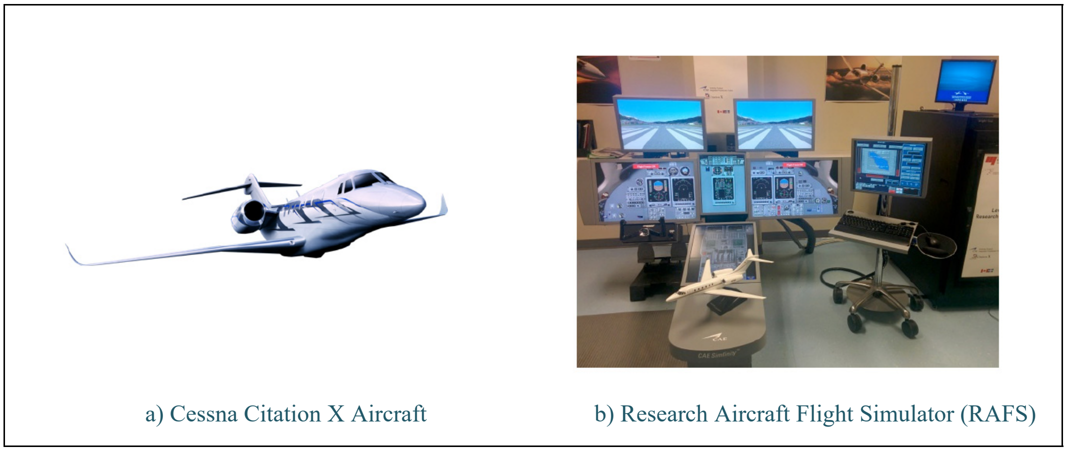
CFD methods are based on solving the fundamental equations of fluid dynamics. Codes such as Ansys CFX (Phillips & Snyder, 2000) and Ansys Fluent (Koreanschi, Oliviu & Botez, 2014), for instance, are based on the resolution of the Navier-Stokes equations and represent practical tools for aerodynamic modeling. However, the accuracy of CFD methods is limited by the assumptions of the mathematical equations used by their algorithms.

Experimental methods based on wind tunnel tests overcome the weaknesses of CFD methods by considering a “real” fluid to reproduce flight conditions over a wide range of the envelope of an aircraft. Many experimental methods such as pressure measurements (Piziali, 1994), Laser Sheet Visualization (LSV) (Wernert et al., 1996), and Time-Resolved Particle Image Velocimetry (TR-PIV) techniques (Mulleners et al., 2012) have been used to develop stall models by visualizing and characterizing the airflow during wind tunnel experiments.

However, the results obtained are highly dependent on the scale of the model, and require costly and time-consuming experiments (Botez, 2018).

The system identification methods we have chosen to use in our study aim at designing a mathematical model of the aerodynamic coefficients from flight test, under stall conditions. System identification techniques combine the advantages of the previously presented methods and overcome their weaknesses. They can provide an accurate model based on real flight test data learning. These types of methods require a large amount of data and a powerful tool to learn the data scheme. At our LARCASE laboratory, we have used neural network techniques for the morphing wing tip modeling in the CRIAQ MDO 505 project (Ben Mosbah et al., 2020), for the wind tunnel calibration (Ben Mosbah et al., 2013), for the modeling of the Bell-427 helicopter (De Jesus Mota & Botez, 2011), F/A-18 (Boely et al., 2011; Boely & Botez, 2010) and Cessna Citation X Engine Model (Zaag & Botez, 2017).

The main objective of this paper is to develop a methodology to predict the aerodynamic coefficients of the Cessna Citation X aircraft in stall conditions using neural networks. The aerodynamic coefficients were estimated from data obtained from flight tests performed on a level-D Cessna Citation X Research Aircraft Flight Simulator (RAFS) designed and manufactured by CAE Inc. According to the Federal Aviation Administration (FAA), level D is the highest qualification level for flight dynamics. Therefore, it was assumed that the RAFS was accurate enough to be considered as a test aircraft. All the collected data was similar to the actual data that would be collected during a flight test on an actual Cessna Citation X Aircraft.



a) Cessna Citation X Aircraft

b) Research Aircraft Flight Simulator (RAFS)

Figure 5.1 Cessna Citation X Business Aircraft and its corresponding Level D Research Aircraft Flight Simulator (RAFS)

The rest of this article is organized as follows: section 5.2 presents the methodology, including the data acquisition procedure from the RAFS and the data preprocessing of flight parameters, so that they can be used to compute aerodynamic coefficients. Next, we present the method used to select the neural network's inputs and outputs, and its hyperparameters, including the training algorithms and the activation functions. Finally, the numerical results and their comparisons with experimental data obtained from the Cessna Citation X RAFS are presented in section 5.3.

5.2 Methodology

This section presents the methodology developed at the LARCASE for modeling the lift, drag, and pitching moment aerodynamic coefficients in stall conditions using system identification techniques. The identification was made through neural network optimization using Cessna Citation X flight simulator data.

5.2.1 Flight Test Procedure and Data Gathering

Several flight tests were conducted with the Cessna Citation X Level D flight simulator to collect data and prepare these data for the identification process. As shown in Figure 5.2 these flight tests were designed to replicate different stall maneuvers, and they included several steps.

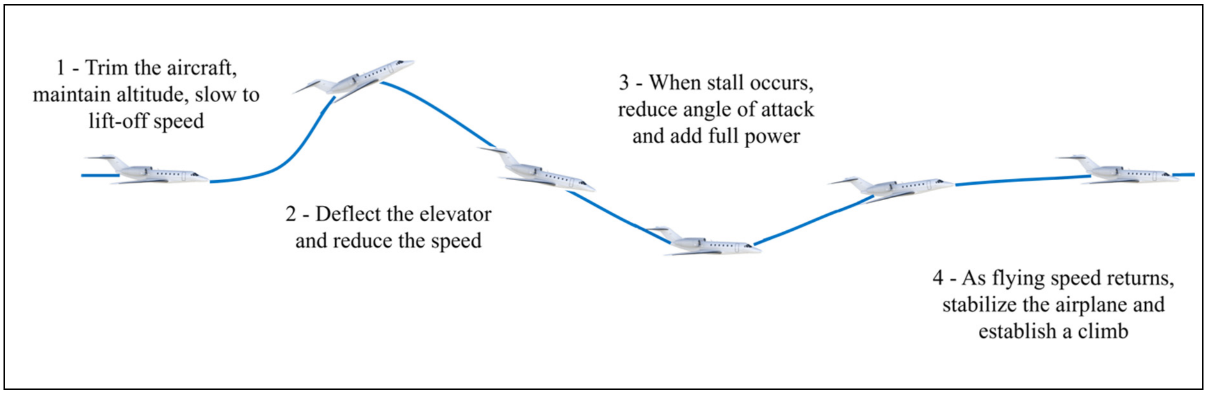

1 - Trim the aircraft, maintain altitude, slow to lift-off speed
2 - Deflect the elevator and reduce the speed
3 - When stall occurs, reduce angle of attack and add full power
4 - As flying speed returns, stabilize the airplane and establish a climb

Figure 5.2 Recovery from stall flight test procedure illustration

First, the aircraft was trimmed for a stable flight at a given altitude and airspeed (1). For this purpose, the altitude was maintained using the autopilot's altitude hold mode, while the airspeed was stabilized manually by adjusting the throttle position. Then, to stall the aircraft, the pilot disengaged the autopilot and pulled back on the control column to deflect the elevators while gradually reducing thrust (2). The aircraft's airspeed decreased, and the angle of attack increased until the stall occurred. The aircraft was maintained in stall conditions as long as possible by controlling the elevators in order to better observe the dynamic stall. Finally, a stall recovering procedure was initiated. The engine thrust was increased to gain airspeed, and the elevators were manually controlled to re-stabilize the aircraft (3), which returned to normal level flight conditions (4).

For each flight test, pilot inputs such as elevator angle deflection δ_e and slats angle deflection δ_s, and flight parameters such as altitude h, true airspeed V_t, angle of attack α and finally aircraft longitudinal and vertical accelerations a_x and a_z were recorded at a sampling rate of 25 Hz.

Figure 5.3 shows an example of recorded data on the RAFS for a flight case carried out at an altitude of 7000 ft, a Mach number of 0.2, and with slats fully retracted (Figure 5.3.g). In this example, the aircraft was trimmed with an elevator angle deflection of $\delta_e = -0.6^o$ (Figure 5.3.d), and at a true airspeed $V_t = 284\, ft/s$ (Figure 5.3.e). At about 3 s, the autopilot was disengaged, the thrust was reduced (Figure 5.3.a) by adjusting the throttle command. Then, the elevator was deflected (Figure 5.3.d) to pitch the aircraft up. Consequently, the angle of attack immediately increased until reaching its stall value α_{stall} (Figure 5.3.h) at about 17 s, leading to a significant reduction of the lift force and, therefore, a drop in vertical acceleration (Figure 5.3.f). We can also observe the drastic change in vertical and horizontal accelerations a_z and a_x (Figure 5.3.f and Figure 5.3.c) at about 17 s, reflecting the significant increase in the drag that happens when stall occurs. Finally, these drops in longitudinal accelerations result in a significant loss in altitude (Figure 5.3.b).

The procedure described in Figure 5.3 was replicated 33 times with the Cessna Citation X RAFS for different flight scenarios by varying initial altitudes and slat configurations. Fifteen flight tests were conducted for cases with a slat-in (i.e., retracted) configuration at altitudes ranging from 5000 to 50,000 ft, while 18 flight tests were conducted at altitudes ranging from 15,000 to 50,000 ft for cases with a slat-out (i.e., extended) configuration.

When slats are out, they increase the wing's camber and, therefore, change the wing shape. Consequently, slats may have the effect of delaying the stall phenomenon (Anderson, 2010).

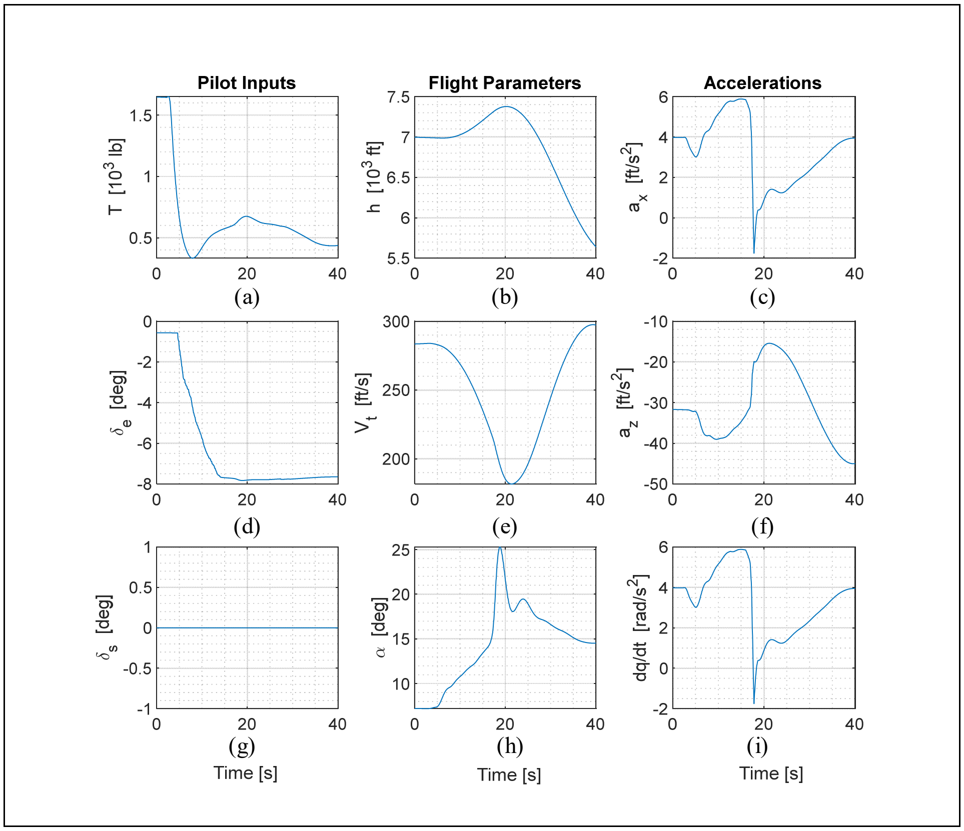

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 5.3 Example of data recorded for a flight test at 7000 ft, Mach 0.20, and slats $\delta_s$retracted

5.2.2 Data Processing and Aerodynamic Coefficients' Determination

The aerodynamic coefficients are not measurable during the flight test. Therefore, once all the flight tests were completed, the next step was to process the measurable flight parameters to estimate the aerodynamic coefficients of the aircraft and then to create a database to prepare for the neural network learning process. The lift, drag and pitching moment aerodynamic coefficients, expressed in the stability axis, are given by the following equations (Ghazi et al., 2017):

$$CL_s = CL_b \cos(\alpha) - CD_b \sin(\alpha) \tag{5.1}$$

$$CD_s = CD_b \cos(\alpha) + CL_b \sin(\alpha) \tag{5.2}$$

$$Cm_s = Cm_b - CD_b z_{cg} - CL_b x_{cg} \tag{5.3}$$

where $\{x_{cg}, z_{cg}\}$ are the longitudinal and vertical distances, respectively, between the aircraft center of gravity and its aerodynamic center. CL_b, CD_b and Cm_b are the lift, drag and pitching moment coefficients, respectively, expressed in the aircraft body axis. The aerodynamic coefficients expressed in the body axis are given by the following expressions:

$$CL_b = \frac{ma_z - T_z}{1/2\rho V_T^2 S_w} \tag{5.4}$$

$$CD_b = \frac{ma_x - T_x}{1/2\rho V_T^2 S_w} \tag{5.5}$$

$$Cm_b = \frac{I_{yy}\dot{q} - T_x z_{eng} - T_z x_{eng}}{1/2\rho V_T^2 S_w c_w} \tag{5.6}$$

where ρ is the air density, $\{T_x\ , T_z\}$ are the components of the engine thrust, I_{yy} is the aircraft moment of inertia about the lateral axis, S_w is the wing reference area, c_w is the mean aerodynamic chord of the wing, and a_x and a_z are the longitudinal and vertical accelerations of the aircraft, respectively.

Before neural network training, a good practice is to normalize the data. The objective is to ensure that all the neural network's input and output parameters have the same scale and are all centered around zero. Consequently, the difference in data magnitude will not affect the training results, as only the correlation between the data will be considered during the training process. For this purpose, the input and output parameters (defined in section 5.2.3.1), were normalized using the following equation:

$$Data_{k\ (normalized)} = \frac{Data_k - \mu}{\sigma} \tag{5.7}$$

where $Data_k$ is the k^{th} value of the considered training data set, μ is the mean of the whole data set, and σ is the standard deviation of the data set. Figure 5.4 and Figure 5.5 show the normalized aerodynamic coefficients estimated from flight test data for the Cessna Citation X RAFS, with slat-in and slat-out configurations, respectively.

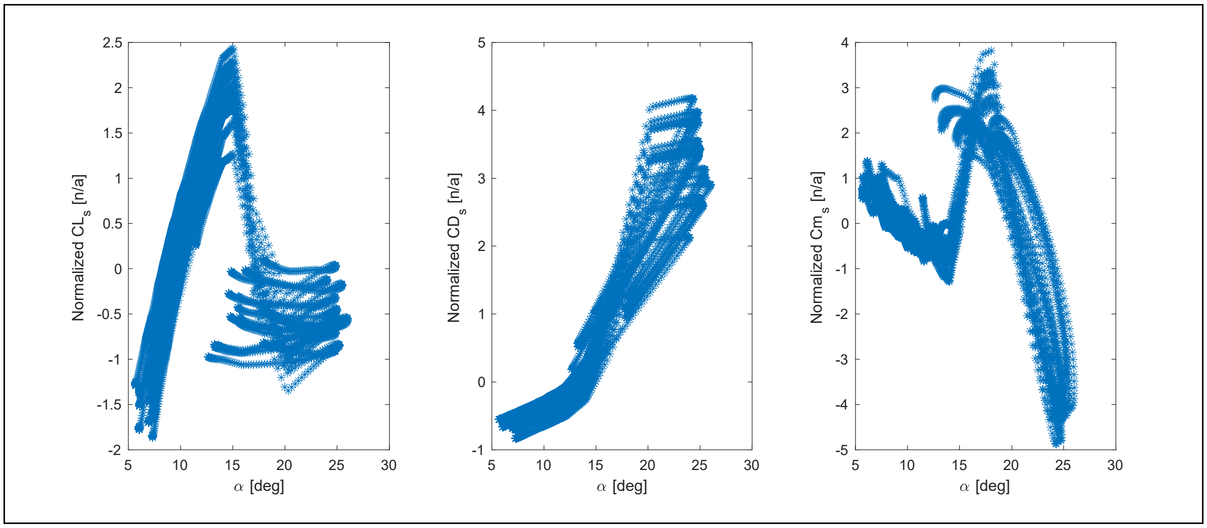


Figure 5.4 Normalized aerodynamic coefficients' estimation from flight test data obtained from the Cessna Citation X RAFS (slat-in)

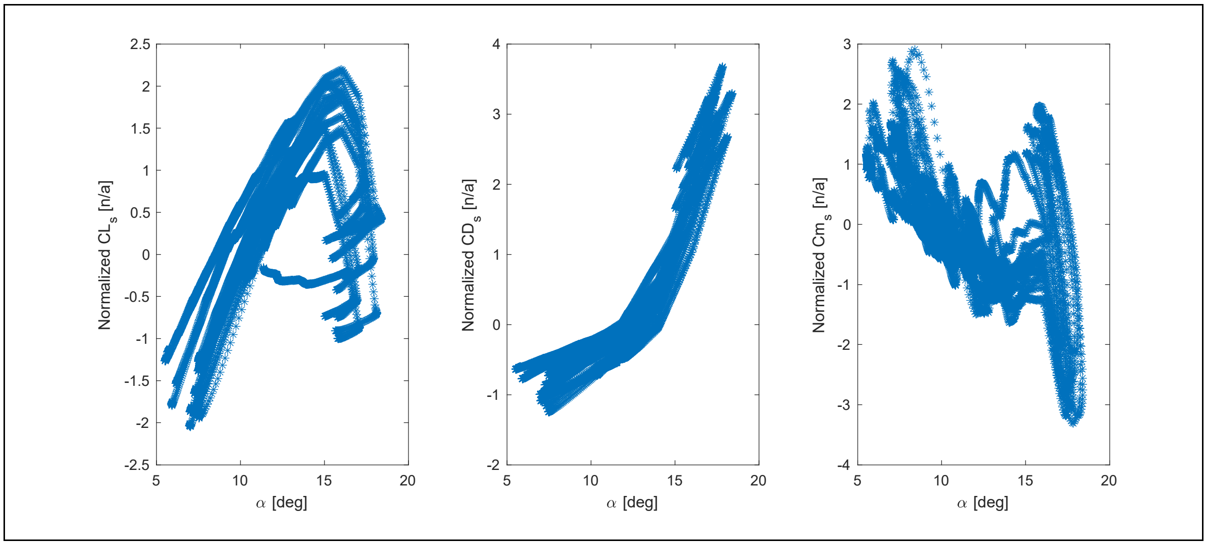


Figure 5.5 Normalized aerodynamic coefficients' estimation from flight test data obtained from the Cessna Citation X RAFS (slat-out)

5.2.3 Neural Network Modeling

5.2.3.1 Choice of Neural Networks Inputs

The first step in defining the neural networks was the determination of the parameters that correlate significantly with the aerodynamic coefficients. These parameters were then used as inputs for the neural network models. Based on the Buckingham Pi's theorem (Anderson, 2010) applied to the dimensional aerodynamic analysis, the aerodynamic coefficients depend on the following variables: the angle attack α, its derivative $\dot{\alpha}$, the Mach M and Reynolds numbers R_e. However, as the Reynolds number is not directly measurable in the flight simulator, it has been approximated using to Equation (5.8):

$$R_e = \frac{\rho V_{TAS} c}{\mu} = \frac{c}{\mu} * \rho V_{TAS} = \frac{2c}{\mu} * \frac{Q}{V_{TAS}} \tag{5.8}$$

where ρ is the density of the air and μ is the dynamic viscosity of air.

Thus, the variables considered as input parameters are the angle of attack α, its derivative with respect to the time $\dot{\alpha}$, the Mach number M, the True Air Speed V_{TAS} and the dynamic pressure Q, resulting from Buckingham Pi's theorem analysis. Additional variables, such as the elevator angle deflection δ_e, the slat angle deflection δ_s and the stabilizer angle δ_{stab} were also considered as inputs, because these surfaces are used for longitudinal dynamic control. The aircraft center of gravity position $\{x_{cg}, z_{cg}\}$, was considered, as the pitching moment aerodynamic coefficient is dependent on it. The pitch rate q may also affect the aerodynamic coefficients in stall conditions (McCroskey, 1981), and finally the total engine thrust T was taken into account because the wing airflow may also be affected by the air coming from the engines.

5.2.3.2 Selection of the Neural Network Type

Two types of Neural Networks (NNs) were evaluated in this research: the Multilayer perceptron (MLP) and the Recurrent neural network (RNN). MLPs have been widely used to solve regression and function approximation problems in the literature. Therefore, this type of neural network is well suited for predicting aerodynamic coefficients from a given data set (Ghazi et al., 2017; Tondji et al., 2022). Nevertheless, RNNs have also been considered in this paper as they have been effective in designing models from time series data by using information regarding previous states during the network learning process (Tondji et al., 2022; Williams & Zipser, 1989).

The fundamental element of a neural network, whatever its type, is the artificial neuron (or “perceptron”). Neural networks are composed of neurons organized in layers and linked together by synaptic weights. Figure 5.6 presents a typical architecture of a perceptron.

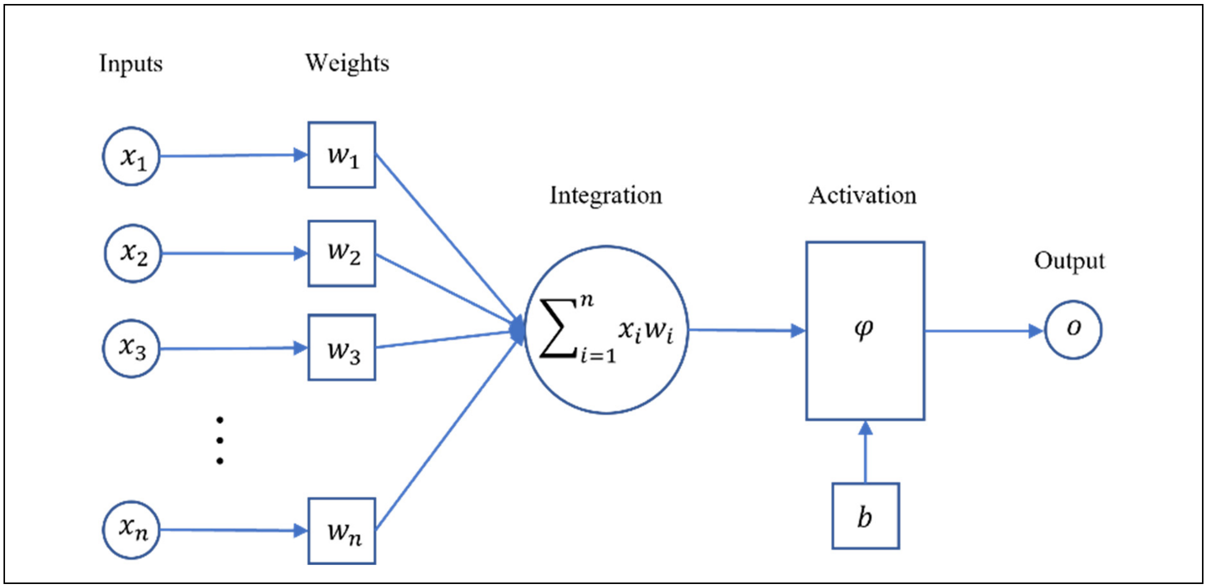


Figure 5.6 Architecture of a perceptron

Therefore, as shown in Figure 5.6, operating principle of a neuron is essentially composed of two calculation steps. First, the input signal of the neuron $X = [x_1, x_2, \dots, x_n]$ is multiplied by its corresponding weights $W = [w_1, w_2, \dots, w_n]$, summed up, and then added to a scalar called the bias b, which is used as a "decision threshold". Secondly, the resulting value is fed into a transfer function called the "activation function", which is used to determine whether the neuron should be activated or not (Haykin, 1998). As a result, the output o of a neuron is given by the following equation:

$$o = \varphi\left(\sum_{i=1}^{n} x_i \, w_i + \ b\right) \tag{5.9}$$

MultiLayer Perceptron

The MLP is one of the simplest forms of neural networks and is composed of several layers of perceptrons, called the hidden layers. A representation of a MLP is given in Figure 5.7, where each hidden layer is associated with a to vector, and the outputs of the neurons in one layer are used as inputs for the next layer. All neurons of the same layer share the same input vector. The predicted output $\hat{o}$ of a MLP can be computed according to Equation (5.10) (Haykin, 1998):

$$\hat{o} = \varphi_m\left(\sum_{k\,=\,1}^{k\,=\,n_m} w_{m,k} \times \dots \times \varphi_2\left[\sum_{i\,=\,1}^{i\,=\,n_2} w_{2,i} \times \varphi_1\big(x_{1,j}w_{1,j} + b_{1,j}\big) + b_{2,i}\right] + b_{m,k}\right) \tag{5.10}$$

where X is the input vector and m is the number of network layers, φ_i and n_i are the activation function and the number of neurons, respectively, in the layer i. $W_{i,j}$ and $b_{i,j}$ are the weights and bias, respectively, of the j^{th} neuron of the layer i.

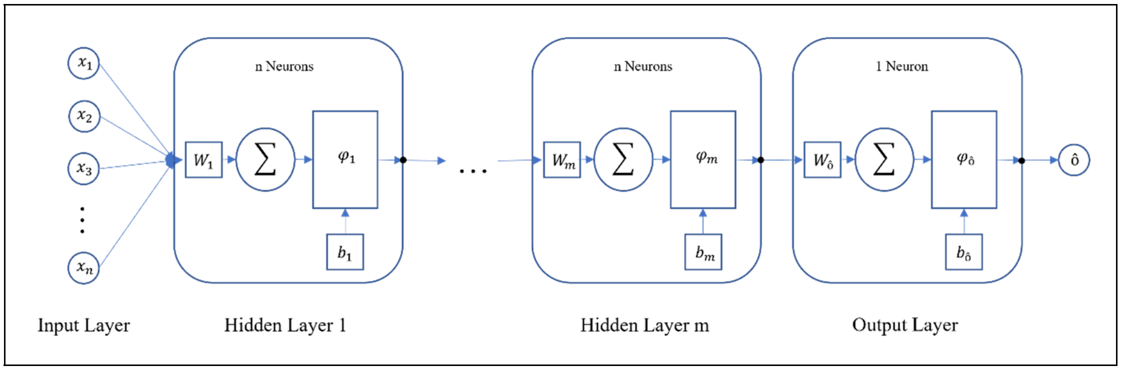


Figure 5.7 Graphical representation of an MLP Neural Network

Recurrent Neural Networks

As their name suggests, RNNs are networks that have recurrent interconnections. The idea behind this architecture is to preserve the neurons' information over time. RNNs are often used for processing time-series signals, such as voice or semantic analysis of videos or sentences. They have also demonstrated their ability to learn the behavior of complex dynamic systems, such as the behavior of an aircraft at high angles of attack (Tondji et al., 2022). Aerodynamic coefficients are in some way presented as a time series signal, where previous states determine the behavior of a future state. A neural network model predicting aerodynamic coefficient variations over time should consider the dynamic nature of the longitudinal behavior of an aircraft. One of the best types of networks that meet this criterion is the Elman Neural Network (ENN) proposed by Elman (1990), and which is defined as a feed-forward network with additional memory neurons (called the "context layer") and local feedback. Figure 5.8 shows a graphical representation of an Elman Neural Network with one-step delay and a hidden layer h.

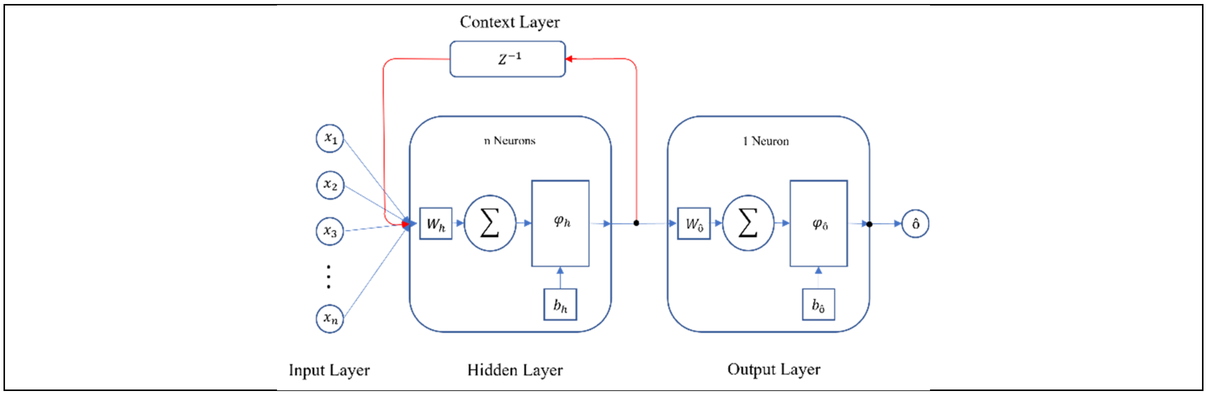


Figure 5.8 Graphical representation of an Elman Neural Network

As shown in Figure 5.8, the hidden layer h receives its information from both the input and the context layers during the training cycle, which are then combined and analyzed simultaneously. Subsequently, the output of each hidden layer is fed back to the context layers at every time step t to provide additional input to the same hidden layer at time $t + 1$. This process is repeated for successive training cycles.

Three different MLPs and three different RNNs were developed for the prediction of the aerodynamic coefficients of the Cessna Citation X (i.e., one MLP and one RNN for each aerodynamic coefficient). This strategy was used because it was found that single output NNs were more accurate than multiple outputs NNs and did not need a complex architecture to learn the correlation between the input data and the target values. Thus, for each neural network model, its output was the coefficient we wanted to predict.

5.2.3.3 Data Management

Building a neural network model necessitates several phases: the training phase, the test phase, and the generalization phase. During the training phase, the neural network weights are updated by a training algorithm to find the inherent relationships among the data in the training data set. Another data set (called the test data set), that was not used for training, is used to calculate the neural network performance. The calculated performance is mainly used to adjust the key model parameters, such as the learning function, the activation function, or the number of hidden layers and neurons. For a given set of n data points and a given set of values of weights $w_{i,j}$, the neural network performance is calculated by the Mean Square Error (MSE), according to the following equation:

$$MSE\ (w) = \frac{1}{n}\sum_{k=1}^{n}\left[\hat{o}_k\left(w_{i,j}\right) - o_k\right]^2 \tag{5.11}$$

Once the hyperparameters of the neural network are optimized and the neural network has been trained, the “generalization” process can be started. It consists of using the final trained model to predict the aerodynamic coefficients of new flight cases that were not used for training (training set) or hyperparameter tuning (test set). The generalization performance is evaluated by means of the Mean Absolute Relative Error ($MARE$), as follows:

$$MARE = \frac{1}{n}\left(\sum_{k=1}^{n}\left|\frac{\hat{o}_k - o_k}{\hat{o}_k}\right|\right) \times 100 \tag{5.12}$$

where o_k is the k^{th} experimental data used for validation, and $\hat{o}_k$ is the k^{th} value predicted by the network.

In this study, from the 33 flight cases conducted with the Cessna Citation X RAFS, only 22 were used for identification (training and test phases), while the remaining 11 cases were used for validation. The training set and tests set were selected from the 22 identification flight cases using cross validation method (Stone, 1974).

5.2.3.4 Fine Tuning

When training neural networks, three parameters are essential: the training algorithm, the activation function, and the neural network structure (the number of hidden layers and the number of neurons per hidden layer). The procedure for selecting these ideal hyperparameters is given in this section below for the determination of the lift coefficient using a MLP.

However, the procedure remains the same for determining other coefficients (CD_s and Cm_s) using both a MLP and a RNN.

Training Algorithm

The first analysis consisted of evaluating the performance of the neural network models using several training algorithms. The activation function (tansig), the number of hidden layers (2 layers), and the number of neurons in the networks (5 neurons per hidden layer) were fixed and chosen randomly. Only the training function varied. Table 5.1 presents the list of the tested algorithms.

Table 5.1 Training algorithms considered to train the network.

Algorithms	Description
LM	Levenberg-Marquardt
BR	Bayesian Regularization
BFG	BFGS[3] Quasi-Newton
RP	Resilient Backpropagation
SCG	Scaled Conjugate Gradient
CGB	Conjugate Gradient Powell/Beale Restarts
CGF	Fletcher-Powell Conjugate Gradient
CGP	Polak-Ribiére Conjugate Gradient
OSS	One Step Secant
GDX	Variable Learning Rate Gradient Descent
GDM	Gradient Descent with Momentum
GD	Gradient Descent

Both MLP and RNN networks were trained with the twelve training algorithms presented in Table 5.1. During the training, the weights and biases were updated to minimize the network performance (MSE). At the end of the training, the performance of each neural network was evaluated using the test data. Figure 5.9 shows the performance in terms of MSE minimization

[3] Broyden-Fletcher-Goldfarb-Shanno

obtained for the prediction of the lift coefficient of the Cessna Citation X using the MLP training process. The performances of the GDX, GDM, and GD were removed for scaling purposes, as their errors were too high compared to that of the results of the other methods.

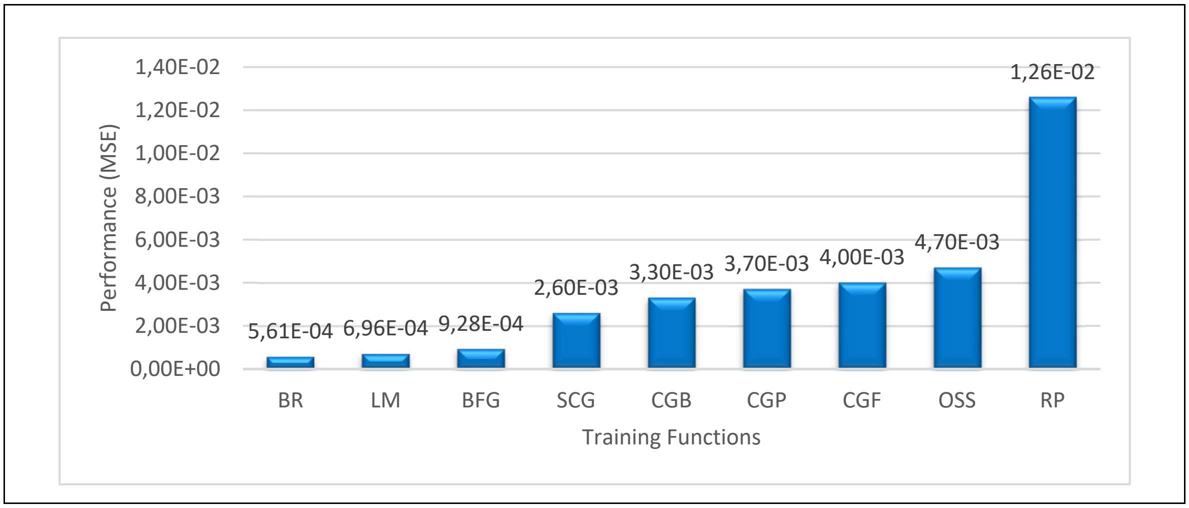


Figure 5.9 MLP performance using different training algorithms for the determination of CL_s

Figure 5.9 shows that the Bayesian Regularization (*BR*) and the Levenberg-Marquardt (*LM*) algorithms provided the lowest MSE. This result was actually expected, as the BR and LM algorithms are well known for their excellent performance in solving nonlinear regression problems. Both algorithms operate using the same procedure, except that in the BR algorithm, a backpropagation is used to compute the Jacobian of the network performance with respect to the weight and bias variables. However, even if they gave similar results, the BR performed slightly better than the LM, with a $MSE = 5.61 \times 10^{-4}$ for the BR, and a $MSE = 6.96 \times 10^{-4}$ for the LM. Based on this analysis, the BR algorithm was considered as the most effective training algorithm for the determination of the lift coefficient using the MLP.

Activation Function

Once the training algorithm was identified, different activation functions were tested to find the function associated with the BR algorithm that would give the best performance when determining the lift coefficient using the MLP. The list of activation functions available in MATLAB, that were tested is presented in Table 5.2.

Table 5.2 Implemented activation function: a is the neuron's activation, y is the neuron's output

Activation Function	**Mathematical Equation**
Log Sigmoid (logsig)	$y(a) = \frac{1}{1 + \exp{(-a)}}$
Hyperbolic Tangent Sigmoid (tansig)	$y(a) = \frac{2}{(1 + \exp(-2 * a))} - 1$
Elliot Symmetric Sigmoid (elliotsig)	$y(a) = \frac{a}{(1 + \lvert a \rvert)}$
Radial basis (radbas)	$y(a) = \exp{(-a^2)}$
Normalized radial basis (radbasn)	$y(a)_i = \frac{\exp(-{a_i}^2)}{\sum_{j=1}^{n} \exp\left(-{a_j}^2\right)}$ where a is the input vector to a Radbasn function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Soft max (softmax)	$y(a)_i = \frac{\exp(a_i)}{\sum_{j=1}^{n} \exp{(a_j)}}$ where a is the input vector to a soft max function that consists of n elements of n classes, and a_i is the i-th element of the input vector.
Saturating linear (Satlin)	$y(a) = \begin{cases} 0, & \text{if } a \leq 0 \\ a, & \text{if } 0 \leq a \leq 1 \\ 1, & \text{if } 1 \leq a \end{cases}$
Symmetric saturating linear (Satlins)	$y(a) = \begin{cases} -1, & \text{if } a \leq -1 \\ a, & \text{if} - 1 \leq a \leq 1 \\ 1, & \text{if } 1 \leq a \end{cases}$
Triangular basis (Tribas)	$y(a) = \begin{cases} 1 - \lvert a \rvert, & \text{if} - 1 \leq a \leq 1 \\ 0, & \text{otherwise} \end{cases}$
Positive linear (Poslin)	$y(a) = \begin{cases} a, & \text{if } a \geq 0 \\ 0, & \text{if } a \leq 0 \end{cases}$

Figure 5.10 shows the MSE error obtained for each activation function when a MLP was trained for predicting the lift coefficient of a Cessna Citation X with the BR algorithm. It can be clearly seen that the Sigmoid-Type activation functions performed better than others. In fact, the Log-Sigmoid, the Hyperbolic-Tangent-Sigmoid, and the Elliot-Symmetric-Sigmoid gave lower errors than the other functions, with a MSE of the order of 10^{-4}. The best performance was achieved with Log-Sigmoid function, for a MSE of 5.14×10^{-4} as seen on Figure 5.10.

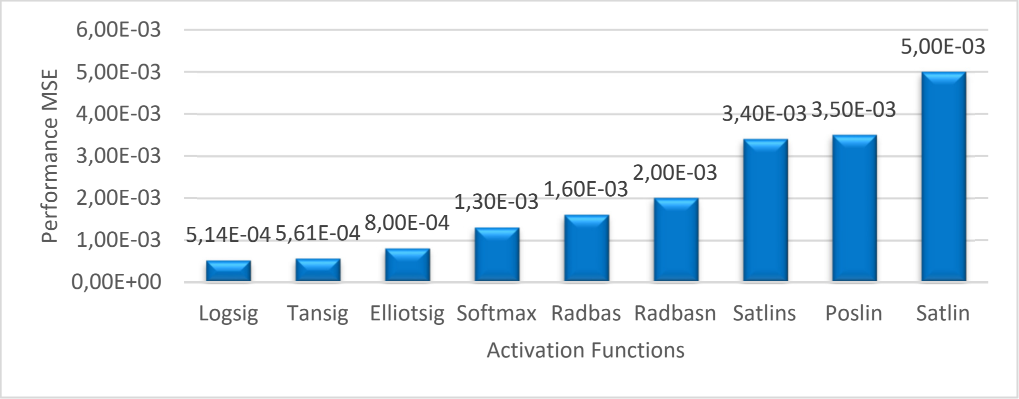


Figure 5.10 MLP performance using different activation function for the determination CL_s using BR algorithm

Neural Network Structure Optimization

The final step in the neural network design process was the selection of the structure that provides the minimum MSE in an appropriate learning time. The structure of the neural network is defined as a combination (m, n), where m is the number of hidden layers of the network, and n is the number of neurons per hidden layer. Similar studies were explored and have demonstrated that the aerodynamic coefficients of a Bombardier CRJ700 in dynamic stall condition can be identified with a MLP model of a maximum of 5 hidden layers and a maximum of 15 neurons per hidden layer (Tondji et al., 2022). Based on this observation, a minimum number of hidden layers $m_{min} = 1$, and a maximum number of hidden layers

$m_{max} = 5$ was assumed. Similarly, the minimum number of neurons per layer was set to $n_{min} = 1$, and the maximum number was set to $n_{max} = 15$.

Once the range of these two parameters was defined, the analysis consisted of training several structures with hidden layers ranging from 1 to 5 and a number of neurons ranging from 1 to 15, and then the resulting MSEs were compared. Thus, the number of structures to be trained would be equivalent to 15^5, which is clearly a very large number. To reduce the number of possible structures, it was assumed that all hidden layers should have the same number of neurons. Such an assumption was considered because it was found that varying the number of neurons from one hidden layer to another did not significantly improve the network performance. This assumption reduces the number of possible structures from 15^5 to 75. The MSEs of different structures trained to predict the lift coefficient of the Cessna Citation X airplane are shown in Figure 5.11.

Figure 5.11 shows the $MSEs$ obtained for each tested structure. Structures with a different number of layers are displayed in different colors. For example, the MSE obtained with a structure with $m = 2$ layers are displayed in "orange", while the MSE obtained with a structure with $m = 3$ layers are displayed in "blue". Performances obtained for the structure with $m = 1$ hidden layer and structures with less than $n = 7$ neurons per hidden layer have been removed for scaling purposes as the MSE was too high. We can clearly see a convergence of the MSE value, which decreases and tends to zero when the number of neurons increases. The convergence is faster for structures with a higher number of layers. For example, structure with 3 layers converge faster than structure with 2 layers). However, it is challenging to determine which is the optimal network structure, as there is no significant improvement on the network's performance after 11 neurons for networks with $m = 2$, $m = 3$, $m = 4$, and $m = 5$. Any of these structures could be used to identify the lift coefficient and would give satisfactory results. However, as a structure with $m = 5$ hidden layers converges even faster than others, a convergence threshold of $MSE = 1.19^{-5}$ is reached even earlier (before $n = 11$) for $n = 9$ neurons. The network structure with $(m = 5, n = 9)$ was therefore considered for the determination of the lift coefficient CL_s of the Cessna Citation X.

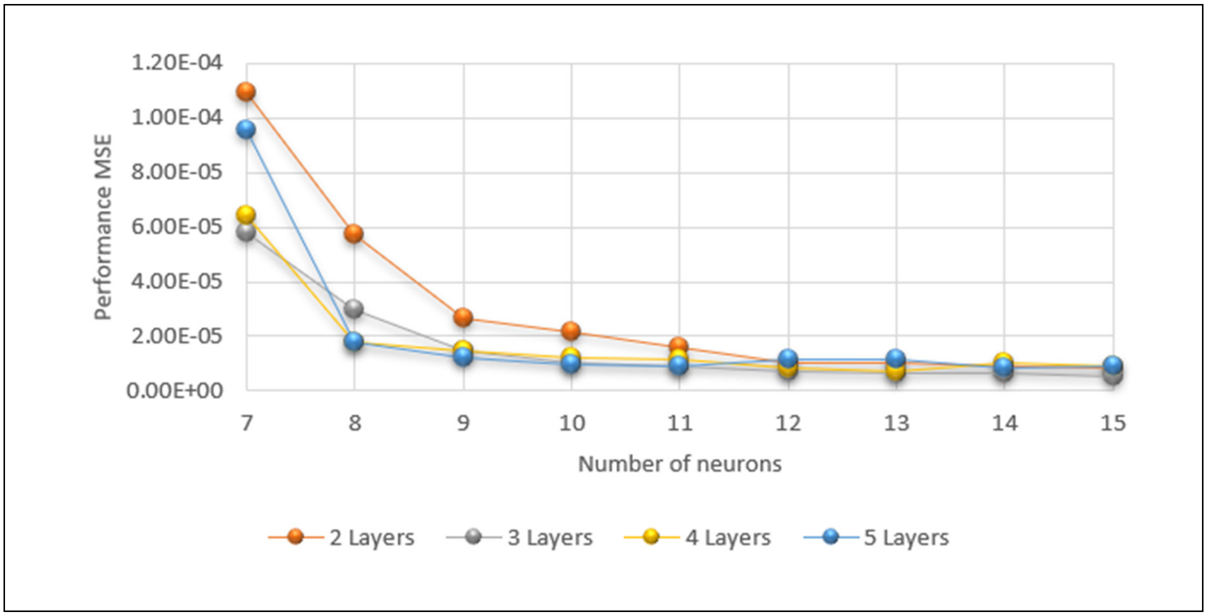


Figure 5.11 Performances for various MLP structures for the estimation of the CL_s of the Cessna Citation X

5.3 Results

This section presents the validation results of the proposed methodology. An evaluation of the accuracy of the two types of neural networks (i.e., MLP and RNN) in terms of the prediction of aerodynamic coefficients is presented. For this purpose, the outputs of both the MLP and the RNN models were compared with experimental aerodynamic coefficients' data obtained from the Cessna Citation X RAFS.

The best performances in terms of MSE obtained for the estimation of the three longitudinal aerodynamic coefficients (CL_s, CD_s and Cm_s) were obtained with the BR algorithms, combined with the *tansig* and *logsig* activation functions. RNN models performed as well as MLP models, but with smaller structures, as the total number of neurons of the hidden layers ($m \times n$) on RNN structures were smaller than with the MLP structures for the identification of the same coefficient. For example, the identification of the lift coefficient using an MLP model required a structure of 45 (9 × 5) hidden neurons, while an RNN required only 30 (3 × 10) hidden neurons. The two models performed similarly ($MSE = 1.2 \times 10^{-5}$ for MLP and

$MSE = 1.1 \times 10^{-5}$ for RNN). This faster convergence of RNN models could be explained by the ability of their algorithm to take into account the dynamic behavior of complex systems during the identification process.

Table 5.3 Optimal parameters obtained for the MLP and the RNN using the BR algorithm

	$\boldsymbol{CL_s}$		$\boldsymbol{CD_s}$		$\boldsymbol{Cm_s}$	
Type of neural network	**MLP**	**RNN**	**MLP**	**RNN**	**MLP**	**RNN**
Activation function	logsig	tansig	logsig	tansig	logsig	tansig
Number of hidden layers	5	3	2	3	4	3
Number of nodes per hidden layer	9	10	14	6	12	12
$\boldsymbol{MSE}$ **value**	1.2× 10^{-5}	1.1× 10^{-5}	4.5× 10^{-6}	5.4× 10^{-6}	1.9× 10^{-4}	1.0× 10^{-5}

Once the hyperparameter were fixed and the neural networks trained, the accuracy of each model (MLP and RNN) was evaluated using validation cases that were not used for training. For this purpose, the outputs of the MLP and RNN models were compared in terms of the $MARE$, with experimental aerodynamic coefficients obtained from the Cessna Citation X RAFS. Figure 5.12 and Figure 5.13 show two examples of the results obtained at respectively 32,500 ft with slats-in and 27,500 ft with slats-out.

In general, the results show good agreement between the experimental data obtained by the RAFS (in “blue”), the data predicted by the MLP (in “red”), and the data predicted by the RNN (in “yellow”). The results showed that the lift and drag coefficients of the two cases were predicted with a Mean Absolute Relative Error ($MARE$) smaller than 1%. For the pitching moment coefficient, the $MARE$ was smaller than 5 %. These very small errors allow us to conclude that both models (RNN and MLP) were able to successfully predict the aerodynamic coefficients for these two flight cases presented on Figure 5.12 and Figure 5.13. The comparisons presented in these examples were repeated for all the other validation flight cases.

Table 5.4 and Table 5.5 presents the $MARE$ and the Mean Absolute Residual errors obtained for the prediction of each coefficient with MLP and RNN models respectively.

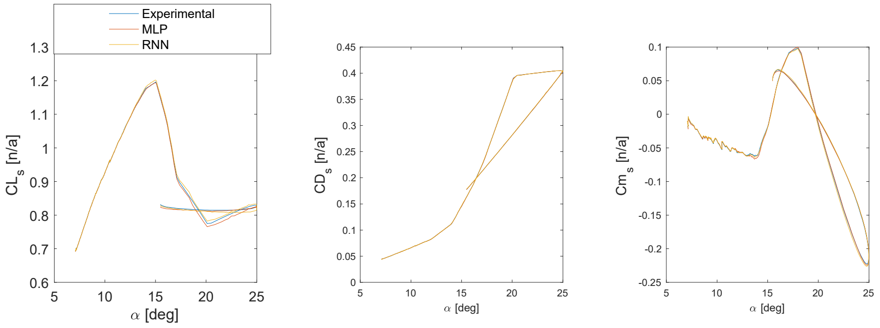


Figure 5.12 Validation of predicted aerodynamic coefficients (with MLP and RNN) for a flight test at 32,500 ft with slats-in

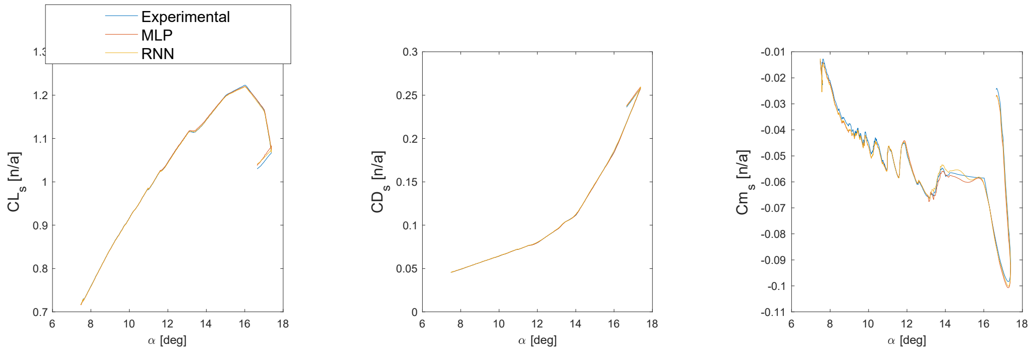


Figure 5.13 Validation of predicted aerodynamic coefficients (with MLP and RNN) for a flight test at 27,500 ft with slats-out

Table 5.4 Mean Absolute Relative Error ($MARE$) and Mean Absolute Residual error obtained between experimental data and values predicted with MLP models

Flight Case		CL_s		CD_s		Cm_s	
Altitude [ft]	**Slat angle**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**
7,000	In	0.998	8.40	0.561	0.68	7.689	2.61
15,000	In	0.325	2.60	0.329	0.60	5.207	1.67
17,500	Out	0.712	6.06	0.535	0.87	5.253	3.80
22,500	In	0.807	7.29	0.390	0.69	14.257	5.49
27,500	Out	0.153	1.52	0.113	0.16	3.011	1.02
32,500	In	0.149	1.27	0.104	0.09	3.625	0.98
37,500	Out	0.386	3.31	0.181	0.16	1.421	1.18
40,000	In	0.540	4.88	0.308	0.63	5.770	2.26
42,500	Out	0.378	4.12	0.263	0.33	8.065	3.42
47,500	In	0.284	2..98	0.203	0.33	3.340	1.51
47,500	Out	0.469	4.79	0.351	0.64	4.529	1.42

Table 5.5 Mean Absolute Relative Error ($MARE$) and Mean Absolute Residual error obtained between experimental data and values predicted with RNN models

Flight Case		CL_s		CD_s		Cm_s	
Altitude [ft]	**Slat angle**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**	$MARE$ **[in %]**	**Mean Residual [× 10^{-3}]**
7,000	In	0.340	3.00	0.702	0.15	11.203	3.82
15,000	In	0.351	2.90	0.457	0.86	5.573	1.55
17,500	Out	0.620	5.85	0.570	0.74	7.482	4.65
22,500	In	0.686	6.74	0.392	0.57	14.842	4.22
27,500	Out	0.109	1.09	0.206	0.19	2.921	9.4
32,500	In	0.168	1.57	0.176	0.16	4.250	8.61
37,500	Out	0.478	4.70	0.228	0.27	0.933	7.49
40,000	In	0.413	3.95	0.273	0.41	7.041	1.39
42,500	Out	0.224	2.38	0.396	0.67	6.593	1.15
47,500	In	0.248	2.63	0.170	0.26	3.714	9.85
47,500	Out	0.593	6.42	0.226	0.28	2.318	7.94

We can see that both MLP and RNN methodologies can globally estimate the aerodynamic coefficients quite well, as the obtained errors are verry small. For all 11 flight cases used for validation, the lift and drag coefficients were estimated with a $MARE$ of less than 1 %. For the pitching moment, 9 flight cases were estimated with a $MARE$ of less than 10 %, and only 2 flight cases were estimated with errors between 10% and 15%. The slats position did not affect the models' precision. Aerodynamic coefficients of slat-in and slat-out flight cases were estimated with the same range of precision. The higher error in percentage obtained for the determination of the pitching moment coefficient could be explained by the fact that for most of the flight cases (in both validation and identification cases), the pitching moment changes its sign when varying (as shown on Figure 5.12 and Figure 5.13). Consequently, at some flight test points, the Cm_s has relatively low values, around zero, which leads to large relative errors ($MARE$). Therefore, even if the $MARE$ errors of the Cm_s seem relatively large (above 10%) for some flight cases, the precision of the estimation remains very good, as the residual errors are very low, even negligible (of the order of 10^{-3}).

In summary, the lift coefficient was estimated with an average $MARE$ of 0.5% for MLP and 0.4 % for RNN. For the estimation of the drag coefficient, the average $MARE$ was 0.3% for MLP and 0.35 % for RNN. Finally, the MLP performed better than the RNN for the prediction of the pitching moment coefficient, with an average $MARE$ of 5.6 % comparatively to 6.1% for the RNN. We can also conclude that the RNN models performed similar to the MLP models.

5.4 Conclusions

This paper presents a detailed methodology for the lift, drag, and pitching moment aerodynamic coefficients modeling in stall regime using Neural Networks. The linear and nonlinear variations of lift and drag aerodynamic coefficients are estimated along the stall hysteresis curve. This presented methodology was applied to the Cessna Citation X airplane developed by CAE Inc and it gave successful results. A stall recovery procedure was developed and executed on the Cessna Citation X RAFS to obtain flight data. A total of 33 flight cases were conducted for different altitudes ranging from 5000 to 50,000 ft and for the “in” and “out” slat configurations. The obtained data were then processed to obtain the lift, drag, and pitching moment aerodynamic coefficients. Data from 22 (67%) flight cases were used to train the following two types of Neural Network models: Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN). The procedure to select the Neural Network parameters (training algorithms, activation function) was detailed, and the process to optimize the models' structures was also developed. Both the MLP and RNN models predicted the aerodynamic coefficients with an average Mean Absolute Relative Error ($MARE$) smaller than 0.5% for the lift and drag coefficients and smaller than 6.2 % for the pitching moment coefficients. These minor errors allowed us to conclude that the developed models performed very well.

CONCLUSIONS AND RECOMMANDATIONS

This PhD thesis investigated the application and integration of AI techniques into aerodynamic and flight dynamics modeling, with a specific focus on the dynamics of the Bombardier CRJ-700 and Cessna Citation X aircraft. The main objective was accomplished through three distinct aspects. The first aspect concentrated on developing and validating AI models for accurate prediction of longitudinal flight dynamics, the second on modeling lateral flight dynamics, and the third on simulating complex scenarios related to aircraft stall dynamics.

Summary of Key Findings and Originality

In CHAPITRE 2, a detailed examination of established aerodynamic modeling techniques was conducted, and the integration of AI methodologies into this domain. The review covered conventional methods, such as CFD, semi-empirical methods and wind tunnel testing, explaining their constraints in capturing complex aerodynamic phenomena. It also explored the growing influence of AI, particularly MLP and SVR, in enhancing the accuracy of predictive flight modeling.

The first sub-objective demonstrated the effectiveness of MLP and SVR in accurately predicting the longitudinal aerodynamic coefficients for the CRJ-700 aircraft. The comparison of the models developed with CRJ-700 VRESIM data, certified Level D for its flight dynamics by the FAA, highlighted the precision of these AI models. Notably, the application of the BO and K-folds cross-validation techniques ensured finely tuned models and robust performance across various flight conditions. These models were validated for 1,992 flight conditions, with mean absolute relative errors below 1% for lift and drag coefficients, and below 5.74% for the pitching moment coefficient. Furthermore, these methods achieved a 100% success rate, satisfying the FAA Level D tolerances. The impressive learning rate of the SVR model, when compared to the ANN, offers a significant advantage in rapid model development. This attribute is advantageous in scenarios that need the prompt implementation of predictive models. Although the initial setup with SVR can be accelerated, it is important to note that both ANN and SVR models operate with their response times equal to fractions of a second once trained. This rapid responsiveness is essential for real-time applications, where fast data processing and immediate output generation play their key roles in ensuring efficient and safe aerospace operations.

Building on the robust findings from the study of longitudinal dynamics, CHAPITRE 3 then shifted its focus on lateral dynamics, a critical aspect of aircraft control and maneuverability. This second objective centered on the Bombardier CRJ-700 model, utilizing the same rigorous AI methodology developed in CHAPITRE 2 and further refined for the accurate prediction of lateral flight dynamics. Data from the Level D CRJ-700 VRESIM provided a solid foundation for model training and optimization. To develop AI models for the accurate prediction of lateral aerodynamic coefficients, the BO combined with cross-validation techniques was employed as a strategic approach in this study. The models were validated using experimental data from the CRJ-700 VRESIM under 2,136 different flight conditions. The MLP and SVR models achieved impressive precision with their Mean Absolute Relative Errors below 5%, demonstrating their predictive power. In addition, the residual errors achieved by the SVR models were found to be on the order of 10^{-5}, indicating a level of accuracy well beyond the initial precision set by the epsilon (ε) hyperparameter (10^{-4}). All developed models met 100% of the FAA tolerance criteria for all validation flight cases, confirming their reliability for practical applications.

In CHAPITRE 4, the complex domain of stall dynamics was investigated, aiming to model and thus to predict the aircraft longitudinal dynamics under both static and dynamic stall conditions. Through a methodical approach, extensive flight data were collected from the Level D-certified CRJ-700 VRESIM, including different altitudes and slats configurations. These data were used to train MLP and RNN models in order to capture both static and dynamic stall non-linearities.

The models went through a rigorous hyperparameters selection process, where various training algorithms and activation functions were tested to identify the most effective combinations.

This meticulous hyperparameters optimization resulted in exceptional model predictive capabilities, as indicated by the Mean Absolute Relative Errors that remained impressively low for all aerodynamic coefficients under study. Subsequently, these models were integrated into a MATLAB/Simulink simulation platform to simulate the CRJ-700 flight dynamics. Validation of the models predictions against experimental VRESIM data for 1,992 different flight cases confirmed their accuracy, while all models have met the FAA validation standards (tolerances).

In CHAPITRE 5, the CRJ-700 methodology was adapted to the Cessna Citation X aircraft, further highlighting the versatility of the approach. Despite the distinct challenges posed by the Cessna Citation X, the use of a similar AI methodology (MLP and RNN) led to equally promising outcomes. The training of AI models used a comparable portion of the dataset collected from the Cessna Citation X RAFS. The subsequent validation process exhibited similar success to the process observed with the CRJ-700, thereby confirming the methodology's robustness for two different aircraft types. This third objective fundamentally validates the thesis’ major contributions to aerodynamic modeling, showcasing AI’s crucial role in precisely predicting and simulating aircraft dynamics during stall.

Discussion of Methodological Strengths and Limitations

The research presented in this thesis reflects a series of methodological strengths that contribute to the field of aerodynamic modeling. The use of high-fidelity simulation data from Level-D certified VRESIM for the CRJ-700 and Cessna Citation X RAFS ensured that the models were trained and validated with data that closely mimics real-world conditions. The application of BO for hyperparameter tuning and the use of cross-validation techniques provided a robust and systematic approach to models’ development, enhancing the accuracy and generalizability of the AI models.

However, the methodology developed in this thesis still represents a beginning in AI and machine learning in application in aerodynamics and stall. As algorithms improve, more sophisticated and computational power increases, the potential exists for even more accurate and comprehensive modeling in reduced computational time. There is an opportunity for future research to expand upon the work presented here.

In summary, the methodological approach of this thesis represents a significant step forward in the application of AI in flight dynamics modeling and stall. It provides a solid platform for future work, which could include the exploration of hybrid models that combine the strengths of different AI techniques, the real-time data acquisition for dynamic model updating, and the application of adaptive control systems models for enhanced flight safety and efficiency.

Future Research Direction

This PhD thesis paves the way for future studies in a variety of directions. One promising avenue involves applying the developed AI methodologies to a broad spectrum of aircraft modeling, such as engine performance, control systems, and structural dynamics. Extending these techniques to different types of aircraft modeling can provide an integrated view of an aircraft operation and behavior.

Further research could also delve into real-time data acquisition and integration of AI models into active flight control systems, allowing for dynamic adjustments to flight parameters based on environmental and aircraft data. This research could significantly enhance the responsiveness of control systems to changing flight dynamics.

Exploring more advanced Machine Learning algorithms beyond the scope of ANN or SVR offers a scope for future research. Techniques such as Long Short-Term Memory (LSTM) networks, known for their efficacy in processing time-series data, could enhance the modeling of dynamic aerodynamic phenomena. Such advancements in AI methodologies could significantly enhance our understanding and prediction of various aircraft aerodynamic performance in different flight scenarios.

Lastly, extending this research to include the modeling of the environmental impact of aircraft, such as noise and emissions, can lead to a more sustainable aviation technologies development, aligning with the global push towards reducing the environmental footprint of air travel.

In summary, this PhD thesis concludes with the integration of AI into aerodynamic modeling, thus representing a substantial advancement in Aerospace Engineering. The successful application of advanced ML techniques to complex aerodynamic phenomena not only validates the potential of AI in this field but also marks the beginning of a new period where AI-enhanced models could become an industrial standard. This research serves as a bridge between theoretical knowledge and practical applications, promising to influence future innovations in aircraft design, safety, and efficiency. The PhD thesis research has been challenging but rewarding, showcasing the endless possibilities when technology and human innovations converge.

LIST OF BIBLIOGRAPHICAL REFERENCES

Abbas, A., De Vicente, J., & Valero, E. (2013). Aerodynamic technologies to improve aircraft performance. *Aerospace Science and Technology*, *28*(1), 100-132. https://doi.org/10.1016/j.ast.2012.10.008

Abdolhosseini, M., Zhang, Y. M., & Rabbath, C. A. (2013). An Efficient Model Predictive Control Scheme for an Unmanned Quadrotor Helicopter. *Journal of Intelligent & Robotic Systems*, *70*(1-4), 27-38. https://doi.org/10.1007/s10846-012-9724-3

Albisser, M. (2015). Identification of aerodynamic coefficients from free flight data. (Doctoral thesis, Université de Lorraine, Lorraine, France). Retrieved from https://theses.hal.science/tel-01213843/

Alexandersson, M., Mao, W., & Ringsberg, J. W. (2021). Analysis of roll damping model scale data. *Ships and Offshore Structures*, *16*(sup1), 85-92. https://doi.org/10.1080/17445302.2021.1907070

Al-Faiz, M. Z., Ibrahim, A. A., & Hadi, S. M. (2019). The effect of Z-Score standardization (normalization) on binary input due the speed of learning in back-propagation neural network. *Iraqi Journal of Information & Communications Technology*, *1*(3), 42-48. https://doi.org/10.31987/ijict.1.3.41

Al-Shareef, A., Mohamed, E., & Al-Judaibi, E. (2008). Next 24-Hours Load Forecasting Using Artificial Neural Network (ANN) for the Western Area of Saudi Arabia. *Journal of King Abdulaziz University-Engineering Sciences*, *19*(2), 25-40. https://doi.org/10.4197/Eng.19-2.2

Ananda, G. K., & Selig, M. S. (2016). *Stall/Post-Stall Modeling of the Longitudinal Characteristics of a General Aviation Aircraft*. Paper presented at AIAA Atmospheric Flight Mechanics Conference., Washington, D.C, https://doi.org/10.2514/6.2016-3541

Anderson, J. (2010). *Fundamentals of Aerodynamics* (5th ed.). New York: McGraw-Hill Education.

Anderson, J. D. (1997). *A History of Aerodynamics: And Its Impact on Flying Machines* (1st ed.). Maryland: Cambridge University Press. https://doi.org/10.1017/CBO9780511607158

Anderson, J. D. (2000). The Evolution of Aerodynamics in the Twentieth Century: Engineering or Science? In P. Galison & A. Roland (Ed.), *Atmospheric Flight in the Twentieth Century* (Vol. 3, pp. 241-256). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-011-4379-0_9

Anderson, J. D. (2006). Airplane Design Methodology: Setting the Gold Standard. *AIAA Journal*, *44*(12), 2817-2819. https://doi.org/10.2514/1.27756

Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2021). *Aircraft Engine Performance Model Identification using Artificial Neural Networks*. Paper presented at AIAA Propulsion and Energy 2021 Forum. Virtual event: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2021-3247

Anton, N., Botez, R. M., & Popescu, D. (2010). New methodology and code for Hawker 800XP aircraft stability derivatives calculation from geometrical data. *The Aeronautical Journal*, *114*(1156), 367-376. https://doi.org/10.1017/S0001924000003821

Anton, N., Botez, R. M., & Popescu, D. (2011). Stability derivatives for a delta-wing X-31 aircraft validated using wind tunnel test data. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *225*(4), 403-416. https://doi.org/10.1243/09544100JAERO799

Appleby, G., Liu, L., & Liu, L.-P. (2020). Kriging Convolutional Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 3187-3194. https://doi.org/10.1609/aaai.v34i04.5716

Asuero, A. G., Sayago, A., & González, A. G. (2006). The Correlation Coefficient: An Overview. *Critical Reviews in Analytical Chemistry*, *36*(1), 41-59. https://doi.org/10.1080/10408340500526766

Bagherzadeh, S. A. (2020). Nonlinear aeroelastic modeling of aircraft using support vector machine method. *Aircraft Engineering and Aerospace Technology*, *92*(3), 502-518. https://doi.org/10.1108/AEAT-06-2019-0129

Baldelli, D. H., Lind, R., & Brenner, M. (2005). Nonlinear Aeroelastic/Aeroservoelastic Modeling by Block-Oriented Identification. *Journal of Guidance, Control, and Dynamics*, *28*(5), 1056-1064. https://doi.org/10.2514/1.11792

Barakos, G. N., & Drikakis, D. (2003). Computational study of unsteady turbulent flows around oscillating and ramping aerofoils. *International Journal for Numerical Methods in Fluids*, *42*(2), 163-186. https://doi.org/10.1002/fld.478

Basappa, & Jategaonkar, R. V. (1995). Aspects of feed forward neural network modeling and its application to lateral-directional flight data. Retrieved from https://elib.dlr.de/30183/

Ben Mosbah, A., Botez, R. M., & Dao, T.-M. (2016). New methodology combining neural network and extended great deluge algorithms for the ATR-42 wing aerodynamics analysis. *The Aeronautical Journal*, *120*(1229), 1049-1080. https://doi.org/10.1017/aer.2016.46

Ben Mosbah, Abdallah, Botez, R. M., Medini, S. M., & Dao, T.-M. (2020). Artificial Neural Networks-Extended Great Deluge Model to predict Actuators Displacements for a

Morphing Wing Tip System. *INCAS BULLETIN*, *12*(4), 13-24. https://doi.org/10.13111/2066-8201.2020.12.4.2

Ben Mosbah, Abdallah, Flores Salinas, M., Botez, R., & Dao, T. (2013). New Methodology for Wind Tunnel Calibration Using Neural Networks - EGD Approach. *SAE International Journal of Aerospace*, *6*(2), 761-766. https://doi.org/10.4271/2013-01-2285

Bierbooms, W. A. A. M. (1992). A comparison between unsteady aerodynamic models. *Journal of Wind Engineering and Industrial Aerodynamics*, *39*(1-3), 23-33. https://doi.org/10.1016/0167-6105(92)90529-J

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Blake, W. (1985). Prediction of fighter aircraft dynamic derivatives using Digital Datcom. In *3rd Applied Aerodynamics Conference*. Colorado Springs, CO, USA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.1985-4070

Boeing. (2023). *2023 Sustainability report*. Retreived from https://www.boeing.com/principles/sustainability/annual-report/index.page

Boely, N, Botez, R. M., & Kouba, G. (2011). Identification of a non-linear F/A-18 model by the use of fuzzy logic and neural network methods. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *225*(5), 559-574. https://doi.org/10.1177/2041302510392871

Boely, Nicolas, & Botez, R. M. (2010). New Approach for the Identification and Validation of a Nonlinear F/A-18 Model by Use of Neural Networks. *IEEE Transactions on Neural Networks*, *21*(11), 1759-1765. https://doi.org/10.1109/TNN.2010.2071398

Boggs, P. T., & Tolle, J. W. (2000). Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, *124*(1-2), 123-137. https://doi.org/10.1016/S0377-0427(00)00429-5

Botez, R. (2018). Morphing Wing, UAV and Aircraft Multidisciplinary Studies at the Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE. *AerospaceLab Journal*, *Issue 14*, September 2018; ISSN: 21076596. https://doi.org/10.12762/2018.AL14-02

Botez, R. M., Kammegne, M. J. T., & Grigorie, L. T. (2015). Design, numerical simulation and experimental testing of a controlled electrical actuation system in a real aircraft morphing wing model. *The Aeronautical Journal*, *119*(1219), 1047-1072. https://doi.org/10.1017/S0001924000011131

Botez, Ruxandra Mihaela. (1989). *Une étude comparative des modèles semi-empiriques pour la prédiction du décrochage dynamique*. (Master thesis, École polytechnique, Montreal, QC, Canada). Retreived from https://espace2.etsmtl.ca/id/eprint/14025/

Braza, M., Berton, E., Allain, C., Favier, D., Maresca, C., Selmin, V., … Climent, H. (2003). Summary of work carried out in the main tasks of the UNSI project. In W. Haase, V. Selmin, & B. Winzell (Ed.), *Progress in Computational Flow-Structure Interaction* (pp. 145-224). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-45489-2_3

Brereton, R. G., & Lloyd, G. R. (2010). Support Vector Machines for classification and regression. *Analyst*, *135*(2), 230-267. https://doi.org/10.1039/B918972F

Chen, D., Kaiser, F., Hu, J., Rival, D. E., Fukami, K., & Taira, K. (2023). Sparse Pressure-Based Machine Learning Approach for Aerodynamic Loads Estimation During Gust Encounters. *AIAA Journal*, 1-16. https://doi.org/10.2514/1.J063263

Chen, H., He, L., Qian, W., & Wang, S. (2020). Multiple Aerodynamic Coefficient Prediction of Airfoils Using a Convolutional Neural Network. *Symmetry*, *12*(4), 544. https://doi.org/10.3390/sym12040544

Chernyshev, S. L., Lyapunov, S. V., & Wolkov, A. V. (2019). Modern problems of aircraft aerodynamics. *Advances in Aerodynamics*, *1*(1), 7. https://doi.org/10.1186/s42774-019-0007-6

Chih-Chung Chang, Chih-Wei Hsu, & Chih-Jen Lin. (2000). The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, *11*(4), 1003-1008. https://doi.org/10.1109/72.857780

Clancy, L. J. (1975). *Aerodynamics*. (illustrated, reprint Ed.). Michigan, USA: Wiley.

Cunningham, K., Foster, J. V., Shah, G. H., Stewart, E. C., Rivers, R. A., Wilborn, J. E., & Gato, W. (2004). *Simulation Study of a Commercial Transport Airplane During Stall and Post-Stall Flight*. Paper presented at World Aviation Congress & Exposition (pp. 2004-01-3100). https://doi.org/10.4271/2004-01-3100

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303-314. https://doi.org/10.1007/BF02551274

Dan Foresee, F., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. In *Proceedings of International Conference on Neural Networks (ICNN'97)* (Vol. 3, pp. 1930-1935 vol.3). https://doi.org/10.1109/ICNN.1997.614194

De Jesus Mota, S., & Botez, R. M. (2011). New helicopter model identification method based on flight test data. *The Aeronautical Journal*, *115*(1167), 295-314. https://doi.org/10.1017/S0001924000005789

De Jesus Mota, Sandrine, & Botez, R. (2009). New Identification Method Based on Neural Network for Helicopters from Flight Test Data. In *AIAA Atmospheric Flight Mechanics Conference*. Chicago, Illinois: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2009-5938

Di Pasquale, D., & Savill, M. (2022). The importance of coupling aerodynamic and cost analysis in aircraft design. *CEAS Aeronautical Journal*, *13*(4), 1085-1100. https://doi.org/10.1007/s13272-022-00600-7

Drikakis, D., Kwak, D., & Kiris, C. C. (2016). Computational aerodynamics: Advances and challenges. *The Aeronautical Journal*, *120*(1223), 13-36. https://doi.org/10.1017/aer.2015.2

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support Vector Regression Machines. In M. C. Mozer, M. Jordan, & T. Petsche (Ed.), *Advances in Neural Information Processing Systems* (Vol. 9). MIT Press. Retreived from https://proceedings.neurips.cc/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf

Ekaterinaris, J. A., & Platzer, M. F. (1998). Computational prediction of airfoil dynamic stall. *Progress in Aerospace Sciences*, *33*(11-12), 759-846. https://doi.org/10.1016/S0376-0421(97)00012-2

Elliott, D. (1998). *A better Activation Function for Artificial Neural Networks*. (Report number T.R. 993-8). Maryland: The National Science Foundation.

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, *14*(2), 179-211. https://doi.org/10.1207/s15516709cog1402_1

FAA. (1991). *Airplane Simulator Qualification* (Report number 120-40B). FAA. Retrieved from https://www.faa.gov/about/initiatives/nsp/ac

FAA. (2021). *2021 United States Aviation Climate Action Plan*. Retreived from https://www.faa.gov/sustainability/aviation-climate-action-plan

Fan, H.-Y., Dulikravich, G. S., & Han, Z.-X. (2005). Aerodynamic data modeling using support vector machines. *Inverse Problems in Science and Engineering*, *13*(3), 261-278. https://doi.org/10.1080/10682760412331330177

Fischenberg, D. (1995). Identification of an unsteady aerodynamic stall model from flight test data. In *20th Atmospheric Flight Mechanics Conference*. Baltimore, MD, USA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.1995-3438

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, *13*(3), 317-322. https://doi.org/10.1093/comjnl/13.3.317

Gabor, O. S., Koreanschi, A., & Botez, R. M. (2012). Low-speed aerodynamic characteristics improvement of ATR 42 airfoil using a morphing wing approach. In *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society* (pp. 5451-5456). Montreal, QC, Canada: IEEE. https://doi.org/10.1109/IECON.2012.6388954

Gabor, O. S., Simon, A., Koreanschi, A., & Botez, R. M. (2016). Improving the UAS-S4 Éhecal airfoil high angles-of-attack performance characteristics using a morphing wing approach. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *230*(1), 118-131. https://doi.org/10.1177/0954410015587725

Garvin, M., Daniela, W., Trevor, H., Robert, T. (2013). *An Introduction to Statistical Learning*. New York: Springer.

Ghazi, G., Bosne, M., Sammartano, Q., & Botez, R. M. (2017). Cessna Citation X Stall Characteristics Identification from Flight Data using Neural Networks. In *AIAA Atmospheric Flight Mechanics Conference* (Vol. 2017-0937). Grapevine, Texas: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2017-0937

Graupe, D. (2013). *Principles of Artificial Neural Networks* (3rd Ed., Vol. 7). (S.l.): WORLD SCIENTIFIC. https://doi.org/10.1142/8868

Hamel, C., Sassi, A., Botez, R., & Dartigues, C. (2013). Cessna Citation X Aircraft Global Model Identification from Flight Tests. *SAE International Journal of Aerospace*, *6*(1), 106-114. https://doi.org/10.4271/2013-01-2094

Hansman, R. J., & Craig, A. P. (1987). Low Reynolds number tests of NACA 64-210, NACA 0012, and Wortmann FX67-K170 airfoils in rain. *Journal of Aircraft*, *24*(8), 559-566. https://doi.org/10.2514/3.45476

Haykin, S. O. (1998). *Neural Networks: A Comprehensive Foundation* (2nd Ed). Upper Saddle River, N.J: Pearson.

Heaton, J. (2008). *Introduction to neural networks with Java* (2nd. Ed., 1. print). St. Louis, Mo: Heaton Research.

Hiliuta, A., Botez, R. M., Brenner, M. (2005). Approximation of Unsteady Aerodynamic Forces Q(k,M) by Use of Fuzzy Techniques, *Journal of Aerospace Information Systems*, *43(10)*, https://doi.org/10.2514/1.13600

Hiliuta, A., Botez, R. M., (2007). Flight dynamics helicopter model validation based on flight test data, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 221(5),* https://doi.org/10.1243/09544100JAERO205

ICAO. (2015). *Manual of Criteria for the Qualification of Flight Simulation Training Devices - Volume I - Aeroplanes* (Report number 9625-1). ICAO. Retreived from https://store.icao.int/en/manual-of-criteria-for-the-qualification-of-flight-simulation-training-devices-volume-i-aeroplanes-9625-1

ICAO. (2016). *2016 Environmental Report*. Retreived from https://www.icao.int/environmental-protection/pages/env2016.aspx

Islas-Narvaez, E. A., Ituna-Yudonago, J. F., Ramos-Velasco, L. E., Vega-Navarrete, M. A., & Garcia-Salazar, O. (2022). Design and Determination of Aerodynamic Coefficients of a Tail-Sitter Aircraft by Means of CFD Numerical Simulation. *Machines*, *11*(1), 17. https://doi.org/10.3390/machines11010017

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, *13*(4), 455-492. https://doi.org/10.1023/A:1008306431147

Kármán, T. V. (2004). *Aerodynamics: Selected Topics in the Light of Their Historical Development*. (S.l.): Courier Corporation. (Google-Books-ID: NcGWXoDw7c8C).

Katz J, Plotkin A. (2001). *Low-Speed Aerodynamics*. (2nd Ed). Cambridge: Cambridge University Press

Kecman, V., Huang, T.-M., & Vogt, M. (2005). Iterative Single Data Algorithm for Training Kernel Machines from Huge Data Sets: Theory and Performance. In L. Wang (Ed.), *Support Vector Machines: Theory and Applications* (Vol. 177, pp. 255-274). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/10984697_12

Klein, V., & Morelli, E. A. (2006). *Aircraft System Identification: Theory and Practice*. (Illustrated Ed, Vol. 213): American Institute of Aeronautics and Astronautics. Retreived from https://books.google.ca/books?id=SC90QgAACAAJ

Koreanschi, A., Sugar Gabor, O., & Botez, R. M. (2016a). Drag optimisation of a wing equipped with a morphing upper surface. *The Aeronautical Journal*, *120*(1225), 473-493. https://doi.org/10.1017/aer.2016.6

Koreanschi, A., Sugar Gabor, O., & Botez, R. M. (2016b). Numerical and experimental validation of a morphed wing geometry using Price-Païdoussis wind-tunnel testing. *The Aeronautical Journal*, *120*(1227), 757-795. https://doi.org/10.1017/aer.2016.30

Koreanschi, Andreea, Gabor, O. S., Acotto, J., Brianchon, G., Portier, G., Botez, R. M., … Mebarki, Y. (2017). Optimization and design of an aircraft's morphing wing-tip demonstrator for drag reduction at low speeds, Part II - Experimental validation using Infra-Red transition measurement from Wind Tunnel tests. *Chinese Journal of Aeronautics*, *30*(1), 164-174. https://doi.org/10.1016/j.cja.2016.12.018

Koreanschi, Andreea, Oliviu, S. G., Ayrault, T., Botez, R. M., Mamou, M., & Mebarki, Y. (2016). Numerical Optimization and Experimental Testing of a Morphing Wing with Aileron System. In *24th AIAA/AHS Adaptive Structures Conference*. San Diego, California, USA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2016-1083

Koreanschi, Andreea, Oliviu, S. G., & Botez, R. M. (2014). New Numerical Study of Boundary Layer Behavior on A Morphing Wing-with-Aileron System. In *32nd AIAA Applied*

Aerodynamics Conference. Atlanta, GA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2014-3170

Koreanschi, Andreea, Oliviu, S. G., & Botez, R. M. (2015). Experimental Validation of an Optimized Wing Geometry Using Small Wind Tunnel Testing. In *33rd AIAA Applied Aerodynamics Conference*. Dallas, TX: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2015-3386

Kuitche, M. A. J., & Botez, R. M. (2019). Modeling novel methodologies for unmanned aerial systems – Applications to the UAS-S4 Ehecatl and the UAS-S45 Bálaam. *Chinese Journal of Aeronautics*, *32*(1), 58-77. https://doi.org/10.1016/j.cja.2018.10.012

Kushner, H. J. (1964). A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, *86*(1), 97-106. https://doi.org/10.1115/1.3653121

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). Efficient BackProp. In G. Montavon, G. B. Orr, & K.-R. Müller (Ed.), *Neural Networks: Tricks of the Trade* (Vol. 7700, pp. 9-48). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_3

Li, L., Das, S., John Hansman, R., Palacios, R., & Srivastava, A. N. (2015). Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations. *Journal of Aerospace Information Systems*, *12*(9), 587-598. https://doi.org/10.2514/1.I010329

Linse, D. J., & Stengel, R. F. (1993). Identification of aerodynamic coefficients using computational neural networks. *Journal of Guidance, Control, and Dynamics*, *16*(6), 1018-1025. https://doi.org/10.2514/3.21122

Maca, P., Pech, P., & Pavlasek, J. (2014). Comparing the Selected Transfer Functions and Local Optimization Methods for Neural Network Flood Runoff Forecast. *Mathematical Problems in Engineering*, *2014*, 1-10. https://doi.org/10.1155/2014/782351

MacKay, D. J. C. (1992). Bayesian Interpolation. *Neural Computation*, *4*(3), 415-447. https://doi.org/10.1162/neco.1992.4.3.415

Mahajan, H. K., Kaur, J., Singh, S., & Banerjee, S. (2021). Recurrent Neural Network for Estimation of Aerodynamic Parameters. In *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)* (pp. 150-155). Noida, India: IEEE. https://doi.org/10.1109/SPIN52536.2021.9566075

Mani, M., & Dorgan, A. J. (2023). A Perspective on the State of Aerospace Computational Fluid Dynamics Technology. *Annual Review of Fluid Mechanics*, *55*(1), 431-457. https://doi.org/10.1146/annurev-fluid-120720-124800

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431-441.

Mccroskey, W. (1981). *The Phenomenon of Dynamic Stall*. (Report number TR 81 A-6). USA: NASA Retrieved from https://apps.dtic.mil/sti/citations/ADA098191

Mei, Y., Zhou, Y., Su, S., Shan, X., & Wang, H. (2023). The aerodynamics of a tailsitter UAV at large angle of attack based on simulation. In *AIAA SCITECH 2023 Forum*. National Harbor, MD & Online: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2023-1561

Moin, H., Khan, H. Z. I., Mobeen, S., & Riaz, J. (2021). *Airfoil's Aerodynamic Coefficients Prediction using Artificial Neural Network. arXiv:2109.12149 [physics]*. Retreived from http://arxiv.org/abs/2109.12149

Moir, S., & Coton, F. N. (1995, décembre). *An Examination of the Dynamic Stalling of Two Wing Planforms*. G.U. Aero Report 9526. [Research Reports or Papers], Department of Aerospace Engineering, University of Glasgow. Retreived from http://eprints.gla.ac.uk/183243/

Moncayo, H., Perhinschi, M., Wilburn, B., Wilburn, J., & Karas, O. (2012). UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity-based Mechanism. In *AIAA Guidance, Navigation, and Control Conference*. Minneapolis, Minnesota: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2012-4678

Mosbah, B., Botez, R. M., & Dao, T. M. (2013). New methodology for calculating flight parameters with neural network - Extended Great Deluge method applied on a reduced scale wind tunnel model of an ATR-42 wing. In *AIAA Modeling and Simulation Technologies (MST) Conference*. Boston, MA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2013-5074

Mozdzanowska, A. L., Weibel, R. E., & Hansman, R. J. (2008). Feedback Model of Air Transportation System Change: Implementation Challenges for Aviation Information Systems. *Proceedings of the IEEE*, *96*(12), 1976-1991. https://doi.org/10.1109/JPROC.2008.2006118

Mulleners, K., Le Pape, A., Heine, B., & Raffel, M. (2012). The Dynamics of Static Stall (pp. 9-12). Paper presented at 16th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal: AIAA.

Nelson, R. C. (1998). *Flight Stability and Automatic Control*. (S.l.): McGraw-Hill Education. Retreived from https://books.google.ca/books?id=Z4lTAAAAMAAJ

Paulete-Perianez, C., Andres-Perez, E., & Lozano, C. (2019). Surrogate modelling for aerodynamic coefficients prediction in aeronautical configurations. *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, (15), 1-4. https://doi.org/10.13009/EUCASS2019-870

Pelikan, M., & Goldberg, D. E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary computation. 1* (pp. 525-532) San Francisco, CA, USA.

Perhinschi, M., Campa, G., Napolitano, M., Lando, M., Massotti, L., & Fravolini, M. (2002). A Simulation Tool for On-Line Real Time Parameter Identification. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*. Monterey, California: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2002-4685

Perhinschi, M. G., Wayne, W. S., Clark, N., & Lyons, D. W. (2007). *Neural Network Modeling of Emissions from Medium-Duty Vehicles Operating on Fisher-Tropsch Synthetic Fuel* (p. 13). Paper presented at SAE World Congress and Exhibition, Soc. of Automotive Engineers, Warrendale, PA, USA: SAE. https://doi.org/10.4271/2007-01-1080

Peyada, N., & Ghosh, A. (2009). Aircraft Parameter Estimation Using Neural Network Based Algorithm. In *AIAA Atmospheric Flight Mechanics Conference*. Chicago, Illinois: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2009-5941

Phillips, W. F., & Snyder, D. O. (2000). Modern Adaptation of Prandtl's Classic Lifting-Line Theory. *Journal of Aircraft*, *37*(4), 662-670. https://doi.org/10.2514/2.2649

Piziali, R. A. (1994). *2-D and 3-D oscillating wing aerodynamics for a range of angles of attack including stall*. (Report number 94-A-011) Moffett Field, CA, UAS: National Aeronautics and Space Administration, Ames Research Center: US Army Aviation and Troop Command; National Technical Information Service, distributor.

Platt, J. (1998). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines* (Rapport No. MSR-TR-98-14). Microsoft. Retreived from https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/

Punjani, A., & Abbeel, P. (2015). Deep learning helicopter dynamics models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3223-3230). Seattle, WA, USA: IEEE. https://doi.org/10.1109/ICRA.2015.7139643

Reisenthel, P., & Bettencourt, M. (1999). Data-based aerodynamic modeling using nonlinear indicial theory. In *37th Aerospace Sciences Meeting and Exhibit*. Reno, NV, USA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.1999-763

Rizzi, A. (2023). Separated and vortical flow in aircraft aerodynamics: a CFD perspective. *The Aeronautical Journal*, *127*(1313), 1065-1103. https://doi.org/10.1017/aer.2023.39

Rosema, C., Doyle, J., & Blake, W. (2014). *MISSILE DATA COMPENDIUM (DATCOM) User Manual 2014 Revision*. Retreived from https://apps.dtic.mil/sti/citations/AD1000581

Sabater, C., Stürmer, P., & Bekemeyer, P. (2022). Fast Predictions of Aircraft Aerodynamics Using Deep-Learning Techniques. *AIAA Journal*, *60*(9), 5249-5261. https://doi.org/10.2514/1.J061234

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, *104*(1), 148-175. https://doi.org/10.1109/JPROC.2015.2494218

Shao, S., Chen, M., & Zhang, Y. (2019). Adaptive Discrete-Time Flight Control Using Disturbance Observer and Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(12), 3708-3721. https://doi.org/10.1109/TNNLS.2019.2893643

Singh, S., & Ghosh, A. K. (2007). Estimation of lateral-directional parameters using neural networks based modified delta method. *The Aeronautical Journal*, *111*(1124), 659-667. https://doi.org/10.1017/S0001924000004838

Spentzos, A., Barakos, G., Badcock, K., Richards, B., Wernert, P., Schreck, S., & Raffel, M. (2005). Investigation of Three-Dimensional Dynamic Stall Using Computational Fluid Dynamics. *AIAA Journal*, *43*(5), 1023-1033. https://doi.org/10.2514/1.8830

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. https://doi.org/10.48550/ARXIV.0912.3995

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(2), 111-133. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x

Sugar Gabor, O., Koreanschi, A., Botez, R. M., Mamou, M., & Mebarki, Y. (2016). Numerical simulation and wind tunnel tests investigation and validation of a morphing wing-tip demonstrator aerodynamic performance. *Aerospace Science and Technology*, *53*, 136-153. https://doi.org/10.1016/j.ast.2016.03.014

Suresh, S., Omkar, S. N., Mani, V., & Guru Prakash, T. N. (2003). Lift coefficient prediction at high angle of attack using recurrent neural network. *Aerospace Science and Technology*, *7*(8), 595-602. https://doi.org/10.1016/S1270-9638(03)00053-1

Tam, C. K. W. (1995). Computational aeroacoustics - Issues and methods. *AIAA Journal*, *33*(10), 1788-1796. https://doi.org/10.2514/3.12728

Tang, D. M., & Dowell, E. H. (1995). Experimental investigation of three-dimensional dynamic stall model oscillating in pitch. *Journal of Aircraft*, *32*(5), 1062-1071. https://doi.org/10.2514/3.46837

Tchatchueng Kammegne, M. J., Tondji, Y., Botez, R. M., Grigorie, L. T., Mamou, M., & Mébarki, Y. (2018). New control methodology for a morphing wing demonstrator. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *232*(8), 1479-1494. https://doi.org/10.1177/0954410017699003

Tinoco, E. (1998). The changing role of computational fluid dynamics in aircraft development. In *16th AIAA Applied Aerodynamics Conference*. Albuquerque, NM, USA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.1998-2512

Tondji, Y., & Botez, R. M. (2017). Semi-empirical estimation and experimental method for determining inertial properties of the Unmanned Aerial System – UAS-S4 of Hydra Technologies. *The Aeronautical Journal*, *121*(1245), 1648-1682. https://doi.org/10.1017/aer.2017.105

Tondji, Yvan, Ghazi, G., & Botez, R. M. (2022). CRJ-700 Aerodynamic Coefficients Identification in Dynamic Stall Conditions using Neural Networks. in *AIAA SCITECH 2022 Forum*. San Diego, CA & Virtual: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2022-2577

Tondji, Yvan, Ghazi, G., & Botez, R. M. (2023). CRJ-700 Longitudinal Aerodynamic Coefficients Identification using Support Vector Machine. In *AIAA AVIATION 2023 Forum*. San Diego, CA and Online: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2023-4073

Tondji, Yvan, Wade, M., Ghazi, G., & Botez, R. M. (2022a). Identification and Validation of the Cessna Citation X Longitudinal Aerodynamic Coefficients in Stall Conditions using Multi-Layer Perceptrons and Recurrent Neural Networks. *INCAS BULLETIN*, *14*(2), 103-119. https://doi.org/10.13111/2066-8201.2022.14.2.9

Tondji, Yvan, Wade, M., Ghazi, G., & Mihaela Botez, R. (2022b). Identification of the Bombardier CRJ-700 Stall Dynamics Model Using Neural Networks. *Journal of Aerospace Information Systems*, 1-18. https://doi.org/10.2514/1.I011104

Vakil, S. S., & John Hansman, R. (2002). Approaches to mitigating complexity-driven issues in commercial autoflight systems. *Reliability Engineering & System Safety*, *75*(2), 133-145. https://doi.org/10.1016/S0951-8320(01)00090-4

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4757-2440-0

Verma, H. O., & Peyada, N. K. (2020). Parameter estimation of aircraft using extreme learning machine and Gauss-Newton algorithm. *The Aeronautical Journal*, *124*(1272), 271-295. https://doi.org/10.1017/aer.2019.123

Verma, Hari Om, & Peyada, N. K. (2020). Aircraft parameter estimation using ELM network. *Aircraft Engineering and Aerospace Technology*, *92*(6), 895-907. https://doi.org/10.1108/AEAT-01-2019-0003

Verma, H.O., & Peyada, N. K. (2021). Estimation of aerodynamic parameters near stall using maximum likelihood and extreme learning machine-based methods. *The Aeronautical Journal*, *125*(1285), 489-509. https://doi.org/10.1017/aer.2020.95

Wang, Ban, & Zhang, Y. (2018). An Adaptive Fault-Tolerant Sliding Mode Control Allocation Scheme for Multirotor Helicopter Subject to Simultaneous Actuator Faults. *IEEE Transactions on Industrial Electronics*, *65*(5), 4227-4236. https://doi.org/10.1109/TIE.2017.2772153

Wang, Bo, & Wang, J. (2021). Application of Artificial Intelligence in Computational Fluid Dynamics. *Industrial & Engineering Chemistry Research*, *60*(7), 2772-2790. https://doi.org/10.1021/acs.iecr.0c05045

Wang, Q., Qian, W., & He, K. (2015). Unsteady aerodynamic modeling at high angles of attack using support vector machines. *Chinese Journal of Aeronautics*, *28*(3), 659-668. https://doi.org/10.1016/j.cja.2015.03.010

Wernert, P., Geissler, W., Raffel, M., & Kompenhans, J. (1996). Experimental and numerical investigations of dynamic stall on a pitching airfoil. *AIAA Journal*, *34*(5), 982-989. https://doi.org/10.2514/3.13177

Williams, R. J., & Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, *1*(2), 270-280. https://doi.org/10.1162/neco.1989.1.2.270

Xie, J., Huang, J., Song, L., Fu, J., & Lu, X. (2022). An effort saving method to establish global aerodynamic model using CFD. *Aircraft Engineering and Aerospace Technology*, *94*(11), 1-19. https://doi.org/10.1108/AEAT-10-2021-0299

Yang, X., & Zhang, W. (2013). A faster optimization method based on support vector regression for aerodynamic problems. *Advances in Space Research*, *52*(6), 1008-1017. https://doi.org/10.1016/j.asr.2013.05.030

Yao, F., Müller, H.-G., & Wang, J.-L. (2005). Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, *33*(6). https://doi.org/10.1214/009053605000000660

Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2018). Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *arXiv:1709.01604 [cs, stat]*. Retreived from http://arxiv.org/abs/1709.01604

Zaag, M., & Botez, R. M. (2017). Cessna Citation X Engine Model Identification and Validation in the Cruise Regime from Flight Tests based on Neural Networks combined with Extended Great Deluge Algorithm. In *AIAA Modeling and Simulation Technologies Conference* (Vol. 2017-1941). Grapevine, Texas: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2017-1941

Zan, B.-W., Han, Z.-H., Xu, C.-Z., Liu, M.-Q., & Wang, W.-Z. (2022). High-dimensional aerodynamic data modeling using a machine learning method based on a convolutional neural network. *Advances in Aerodynamics*, *4*(1), 39. https://doi.org/10.1186/s42774-022-00128-8

Zhang, M., Kong, P., Xia, A., Tuo, W., Lv, Y., & Wang, S. (2023). Aerodynamic System Machine Learning Modeling with Gray Wolf Optimization Support Vector Regression and Instability Identification Strategy of Wavelet Singular Spectrum. *Biomimetics*, *8*(2), 132. https://doi.org/10.3390/biomimetics8020132

Zou, T., & Sun, K. (2021). Application and Prospect of Artificial Intelligence in Aircraft Design. In *2021 International Conference on Networking Systems of AI (INSAI)* (pp. 201-205). Shanghai, China: IEEE. https://doi.org/10.1109/INSAI54028.2021.00045