

Application of Intelligent Robust Adaptive Flight Control Laws for the Cessna Citation X business jet

by

Mahefatiana Rojo Princy ANDRIANANTARA

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

MONTREAL, THE 19TH JUNE 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

© All rights reserved

This license means that it is prohibited to reproduce, record or distribute this document in whole or in part.

Readers who wish to print or store a significant part of this document on another medium must request permission from the author.

BOARD OF EXAMINERS

**THIS THESIS HAS BEEN EVALUATED
BY THE FOLLOWING BOARD OF EXAMINERS**

Mrs. Ruxandra Botez, Thesis Supervisor
System Engineering Department at École de technologie supérieure

Mr. Georges Ghazi, Thesis Co-supervisor
System Engineering Department at École de technologie supérieure

Mr. René Jr Landry, President of the Board of Examiners
Electrical Engineering Department at École de technologie supérieure

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

Mr. Adrian Hiliuta, Independent External Examiner
CMC Electronics

**THIS THESIS WAS PRESENTED AND DEFENDED
IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC**

8TH JUNE 2026

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

This doctoral research was carried out at the École de technologie supérieure within the Research Laboratory in Active Commands, Avionics, and AeroServoelasticity (LARCASE). The scientific, technical, and human environment provided by LARCASE played a central role in the successful completion of this work.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Ruxandra Botez, for her continuous support, funding, trust, and guidance throughout my PhD studies. I am sincerely thankful for the opportunity to pursue my research within her team, for her high scientific standards, and for the freedom she provided to develop this work with rigor and ambition.

I would also like to thank my co-supervisor, Professor Georges Ghazi, for his invaluable technical support, patience, and expertise. His deep knowledge and constant availability were essential throughout this research, and it has been a privilege to work alongside him.

I would like to thank the members of the examination committee for the time they devoted to reviewing this thesis and for their valuable comments and suggestions.

I would like to thank Professor Jeremy Laliberté of Carleton University for funding part of this thesis through the NSERC CREATE Uninhabited Aircraft System Training, Innovation and Leadership Initiative program. I gratefully acknowledge NSERC for awarding to Dr Botez the Canada Research Chair in Aircraft Modeling and Simulation Technology, whose funds supported part of this research. I also thank the funding agencies FCI (*Fondation Canadienne d'Innovation*) and MDEIE (Ministère du Développement Économique, de l'Innovation et de l'Exportation) for enabling the acquisition of two Level D flight simulators at LARCASE for the Cessna Citation X business jet, which was used validating this research. I am also grateful to Mr. Oscar Carranza, research assistant at LARCASE, for his assistance and collaboration during different phases of this work.

My sincere thanks go to the entire LARCASE team, including its researchers, engineers, interns, and staff members, for our stimulating discussions, teamwork, and friendly atmosphere that made these years both productive and enjoyable. Special thanks are extended to my lab mates Momo and Elias for their collaboration, support, and shared moments throughout this journey. I would also like to acknowledge all interns and students who contributed, directly or indirectly, to this research, naming Cyril, Elliot, Daniel, Tahar, Louis, Valentine, Hugo, Yvan, Abdou, Yanis, Seydou, Nicolas, Alan, Mobina and Raphael.

On a personal level, I am deeply grateful to my parents Emma and Amédée for their unconditional love, sacrifices, and unwavering support throughout my academic journey. I would also like to thank my brother Aina and my sister Ravaka, as well as my entire family, for their encouragement and belief in me.

I extend my sincere thanks to Mr. Maher Marwani, my roommate, for all these years, for his daily support, patience, and friendship. I am also thankful to all my friends and family, near and far, for their constant encouragement. Special thanks go to my friends from the STK Montreal community, whose presence and friendship provided balance and strength outside the academic world.

Finally, I would like to thank all those who, in one way or another, contributed to this journey. This PhD has been both a scientific and personal experience, shaped by the people who supported, challenged, and inspired me along the way.

Application de lois de commande de vol intelligentes, robustes et adaptatives sur l'avion d'affaire Cessna Citation X

Mahefatiana Rojo Princy ANDRIANANTARA

RÉSUMÉ

La conception des systèmes de commande de vol pour les avions commerciaux reste limitée par la dépendance des contrôleurs classiques et modernes à un modèle mathématique précis, à un séquençement des gains et à un réglage effectué à la suite d'une expertise approfondie. Si les méthodes de commande adaptative et intelligente offrent des alternatives prometteuses, leur application aux systèmes aéronautiques est limitée par des préoccupations liées à la robustesse, à la stabilité numérique et à la compatibilité avec leurs certifications. Cette thèse aborde ces défis à travers le développement et la validation d'une architecture de commande de vol adaptative hybride appliquée à l'avion d'affaire Cessna Citation X.

La méthode de contrôle proposée combine des contrôleurs PID pour la stabilité avec une inversion dynamique basée sur l'estimation par la méthode des moindres carrés récurrents et une adaptation à l'aide des réseaux de neurones pour compenser les incertitudes de modélisation, les non-linéarités et les dynamiques variables dans le temps. L'inversion dynamique est réalisée à l'aide d'une estimation en ligne de la dynamique locale de l'avion à partir des incréments d'état et de contrôle mesurés, éliminant ainsi le besoin d'obtenir des modèles aérodynamiques explicites ou de séquençement des gains. L'adaptation du réseau de neurones compense en outre les erreurs d'inversion résiduelles et les non-linéarités des actionneurs, permettant une double adaptation en ligne dans une architecture de contrôle structurée et interprétable.

La méthodologie est validée à l'aide d'un modèle de simulation non linéaire d'haute-fidélité du Cessna Citation X dans 64 conditions de vol de croisière couvrant une large gamme d'altitudes et de vitesses. Les contrôleurs en boucle interne pour la stabilisation du tangage, du roulis et du lacet, ainsi que les pilotes automatiques en boucle externe pour le contrôle de la vitesse verticale, de l'altitude et du cap, sont conçus et évalués à l'aide d'une configuration unique du contrôleur. Des analyses de robustesse approfondies sont menées dans des conditions de rafales de vent, de turbulence de Dryden, de bruit des actionneurs et de perte d'efficacité des actionneurs. Des stratégies de stabilisation numérique, notamment la réinitialisation de la covariance et l'adaptation basée sur des seuils, sont introduites afin de garantir une estimation et une inversion en ligne fiables.

L'évaluation des performances démontre un suivi précis avec des erreurs en régime permanent inférieures à 2 %, des temps de stabilisation rapides pour la dynamique en boucle fermée et une réjection robuste des perturbations sur toute l'enveloppe de vol. Les qualités de vol sont évaluées de manière explicite à l'aide des critères MIL-STD 1797A, et les exigences de niveau 1 sont systématiquement satisfaites pour les modes de dynamique période courte, le roulis et le lacet hollandais. L'analyse de stabilité basée sur Lyapunov confirme la convergence de tous

VIII

les éléments adaptatifs. Une validation comparative à l'aide d'un simulateur de vol de recherche de niveau D montre une amélioration des performances de notre contrôleur par rapport au contrôleur de référence.

Cette thèse démontre que le contrôle adaptatif hybride basé sur la stabilisation PID, l'inversion dynamique RLS et l'adaptation des réseaux de neurones offre une solution adaptative, robuste et compatible pour la certification du contrôle de vol sur l'enveloppe complète des avions commerciaux. Les résultats établissent une voie pratique vers le déploiement de systèmes de contrôle de vol adaptatifs dans les applications aéronautiques, assurant la sécurité de vol.

Mots-clés : Contrôle de vol adaptatif, Inversion dynamique non-linéaire, Estimation récursive des moindres carrés, Contrôle par réseau de neurones, Validation des qualités de vol

Application of Intelligent Robust Adaptive Flight Control Laws for the Cessna Citation X business jet

Mahefatiana Rojo Princy ANDRIANANTARA

ABSTRACT

The flight control systems design for commercial aircraft remains constrained by the reliance of classical and modern controllers on accurate modeling, extensive gain scheduling, and expert-driven tuning. While adaptive and intelligent control techniques offer promising alternatives, their application to safety-critical aviation systems is limited by concerns related to robustness, numerical stability, and certification compatibility. This thesis addresses these challenges through the development and validation of a hybrid adaptive flight control architecture applied to the Cessna Citation X business jet.

The proposed control method combines Proportional–Integral–Derivative (PID) controllers for baseline stability with Recursive Least Squares–based Dynamic Inversion and Neural Network adaptation to compensate for modeling uncertainties, nonlinearities, and time-varying dynamics. Dynamic Inversion is achieved using online estimation of local aircraft dynamics from measured state and control increments, thus eliminating the need for explicit aerodynamic models or gain scheduling. Neural network adaptation further compensates for residual inversion errors and actuator nonlinearities, enabling dual online adaptation within a structured and interpretable control architecture.

The methodology is validated using a high-fidelity nonlinear simulation model of the Cessna Citation X across 64 cruise flight conditions covering a wide range of altitudes and airspeeds. Inner-loop controllers for pitch rate, roll rate, and yaw stabilization, as well as outer-loop autopilots for vertical speed, altitude, and heading control, are designed and evaluated using a single controller configuration. Extensive robustness analyses are conducted under wind gusts, Dryden turbulence, actuator noise, and actuator loss-of-effectiveness. Numerical stabilization strategies, including covariance reinitialization and threshold-based adaptation, are introduced to ensure reliable online estimation and dynamic inversion.

Performance evaluation demonstrates accurate signal tracking with steady-state errors below 2%, fast settling times for inner-loop dynamics, and robust disturbance rejection across the full flight envelope. Flying qualities are explicitly assessed using MIL-STD 1797A criteria, and Level 1 requirements are consistently satisfied for short-period, roll, and Dutch-roll dynamics modes. The Lyapunov-based stability analysis confirms boundedness and convergence of all adaptive control elements. Comparative results validation using a Level D research flight simulator data shows improved performance of the controller relative to the baseline onboard controller.

This thesis demonstrates that hybrid adaptive control based on PID stabilization, RLS dynamic inversion, and neural network adaptation provides a scalable, robust, and certification-

compatible solution for envelope-wide flight control of commercial aircraft. The results establish a practical pathway toward the deployment of adaptive flight control systems in safety-critical aviation applications.

Keywords: Adaptive Flight Control, Nonlinear Dynamic Inversion, Recursive Least Squares Estimation, Neural Network Control, Flying Qualities Validation

TABLE OF CONTENTS

	Page
INTRODUCTION	1
CHAPTER 1 LITTERATURE REVIEW	11
1.1 Introduction to Aircraft Flight Dynamics.....	11
1.2 Introduction to Aircraft Flight Control	12
1.3 Linear control.....	13
1.4 Classical control.....	14
1.4.1 PID Controllers	15
1.4.2 Gain Scheduling.....	16
1.5 Modern Control.....	17
1.5.1 Optimal Control	18
1.5.2 C* Control Architecture.....	19
1.6 Robust Control.....	20
1.7 Advanced control.....	21
1.7.1 Model Predictive Controller	21
1.7.2 Dynamic Inversion Controller	22
1.7.3 Online State Estimation	24
1.8 Adaptive control.....	26
1.8.1 MRAC and L1 Controllers.....	27
1.8.2 Nonlinear Dynamic Inversion.....	28
1.9 Intelligent Control.....	29
1.9.1 Fuzzy controller	29
1.9.2 Neural Networks Controller.....	30
1.10 Reinforcement Learning (RL) based Controllers	32
1.10.1 Principles.....	32
1.10.2 Issues and Challenges of RL-based controller.....	33
1.11 Handling Qualities	34
1.12 High Fidelity Nonlinear Model.....	35
1.13 Conclusion	37
CHAPTER 2 NONLINEAR ADAPTIVE LONGITUDINAL CONTROLLER AND FLIGHT QUALITIES VALIDATION FOR A BUSINESS AIRCRAFT	39
2.1 Introduction.....	41
2.1.1 Aircraft model and Simulation Platform.....	45
2.1.1.1 Cessna Citation X Aircraft Specifications	45
2.1.1.2 Nonlinear Simulink Platform.....	46
2.1.2 Nonlinear Dynamic Model	48
2.1.2.1 Cessna Citation X nonlinear dynamic model.....	48
2.1.2.2 Nonlinear Actuator Dynamics	48
2.2 Pitch Rate Controller Design	49

2.2.1	Controller Architecture	49
2.2.2	Controller Elements	49
2.2.2.1	Nonlinear Dynamic Inversion Controller	49
2.2.2.2	Online Parameter Estimation	52
2.2.2.3	Proportional Integral Derivative (PID) Controller.....	54
2.2.2.4	Neural Network Adaptive Controller.....	54
2.2.2.5	Adaptation laws	56
2.2.3	Control Law	56
2.2.4	Reference model	56
2.2.5	Flight Quality Requirements.....	57
2.2.6	Performance metrics	57
2.3	Controller Simulation for Ideal Conditions	58
2.3.1	Controller tuning parameters	58
2.3.2	Simulation for cruise at 35,000 ft and 290 knots	59
2.3.3	Simulation and validation over the flight envelope	61
2.3.3.1	Flight operating conditions	61
2.3.3.2	NN and RLS parameter sensitivity analysis	63
2.3.3.3	Controller tuning under wind gusts and turbulences	64
2.3.4	Comparison of Controller Configurations	65
2.3.5	Flight Quality Validation	65
2.4	Robustness Validation	67
2.4.1	Wind Gust and Turbulence tests	67
2.4.2	Actuator Fault tests	69
2.5	Results Discussion	69
2.5.1	Controller performance	69
2.5.2	Comparative Study.....	71
2.6	Conclusion	73
2.7	Appendix.....	74
2.7.1	Error dynamics	74
2.7.2	Disturbance Boundedness.....	76
2.7.3	Lyapunov Candidate and its Derivative.....	76
2.7.4	RLS Controller Stability	76
2.7.5	Boundedness and Convergence	78
2.7.5.1	Case 1	80
2.7.5.2	Case 2.....	80
CHAPTER 3	ROBUST ADAPTIVE LATERAL CONTROL FOR THE CESSNA CITATION X WITH FLIGHT QUALITY VALIDATION	83
3.1	Introduction.....	84
3.1.1	Research Objectives and Organization	88
3.1.2	Cessna Citation X business jet model.....	89
3.1.2.1	Aircraft Description and Simulink Platform.....	89
3.1.3	Aircraft Nonlinear Model	91
3.1.3.1	Nonlinear dynamics	91
3.1.3.2	Actuator Dynamics	92

3.2	Lateral Controller Design Methodology	92
3.2.1	Roll Rate Controller Architecture	92
3.2.2	Controller Elements	93
3.2.2.1	Nonlinear Inversion Controller	93
3.2.2.2	RLS online state estimation	98
3.2.2.3	Adaptive Neural Network Controller	101
3.2.2.4	Adaptation laws	104
3.2.2.5	PID Controller	104
3.2.2.6	Sideslip Angle Derivative Approximation	105
3.2.3	Final Command Laws	105
3.2.4	Roll Rate Reference model	105
3.2.5	Flight Quality Requirements	106
3.2.6	Performance metrics	106
3.3	Controller Tuning and Simulation under Ideal Conditions	107
3.3.1	Controller tuning parameters	107
3.3.2	Simulation for cruise at 35,000 ft and 290 knots	108
3.3.3	Validation over the flight envelope	112
3.3.3.1	Flight operating conditions	112
3.3.3.2	Learning rate and forgetting factor tuning	113
3.3.3.3	Optimal tuning	114
3.3.4	Controller configurations comparison	116
3.3.5	Flight Qualities Validation	117
3.4	Robustness Validation	119
3.4.1	Wind Gust and Turbulence Simulations	119
3.4.2	Actuator Faults Simulations	120
3.5	Results discussion	121
3.5.1	Controller stability and convergence	122
3.5.2	Tuning process	123
3.5.3	Flight Quality Assessment	123
3.5.4	Controller robustness under disturbances	124
3.5.5	Computational Requirements	124
3.5.6	Future Work	124
3.6	Conclusion	125
3.7	Appendix	126
3.7.1	Roll rate and Yaw rate error dynamics	126
3.7.2	Disturbance dynamics	127
3.7.3	Disturbance boundedness	129
3.7.4	Lyapunov candidate and its derivative	130
3.7.5	RLS controller stability	131
3.7.6	Boundedness and Convergence	133
3.7.6.1	Case 1	136
3.7.6.2	Case 2	136
3.7.6.3	Case 3	137

CHAPTER 4	ADAPTIVE AUTOPILOT DESIGN AND IMPLEMENTATION FOR THE CESSNA CITATION X.....	139
4.1	Introduction.....	140
4.1.1	Research Objectives.....	143
4.1.2	Related Work	143
4.2	Validation methods	144
4.3	Aircraft Description and Simulation platform	145
4.3.1	Cessna Citation X Aircraft Description	145
4.3.2	Cessna Citation X Simulink platform	146
4.4	Design methodology	147
4.4.1	Inner loop controllers and Autopilots design requirements	147
4.4.2	Adaptive Autopilot Architectures	148
4.4.3	Aircraft Nonlinear Dynamics.....	150
4.4.4	Nonlinear Actuator Dynamics	151
4.4.5	Mathematical foundations and control elements	152
	4.4.5.1 Dynamic Inversion Controller	153
	4.4.5.2 Online State Estimation	154
	4.4.5.3 NN Controller	156
	4.4.5.4 NN Adaptation laws.....	158
	4.4.5.5 PID Controller.....	158
4.4.6	Longitudinal and lateral CSAS inner loops	159
4.4.7	Pitch rate CSAS	159
4.4.8	Vertical Speed autopilot.....	160
4.4.9	Altitude Autopilot	161
4.4.10	Roll rate CSAS and yaw rate stabilization.....	162
4.4.11	Roll angle CSAS	164
4.4.12	Heading autopilot.....	164
4.5	Results.....	165
4.5.1	Control parameters.....	165
4.5.2	Vertical speed Autopilot Tuning.....	167
4.5.3	Altitude and Heading Command at 35,000 ft and 290 knots.....	169
4.5.4	Validation for the Cessna Citation X cruise flight envelope	176
4.5.5	Performance results.....	180
4.6	Robustness Tests	181
4.6.1	Wind and Turbulence Tests	181
4.6.2	Flight Tests Comparison on the RAFS	182
4.7	Discussion of Results.....	183
4.7.1	CSAS and Autopilot architectures	183
4.7.2	Tuning and Adaptation	183
4.7.3	CSAS and Autopilot Performance	184
4.7.4	Robustness validation	185
4.7.5	Stability assessment	185
4.7.6	General remarks	186
4.7.7	Future works	186
4.8	Conclusions.....	187

4.8.1	Research Summary	187
4.8.2	Conclusions	188
CHAPTER 5	DISCUSSIONS AND RECOMMANDATIONS	189
5.1	Performance assesment	189
5.2	Thesis Discussions	190
5.3	Recommendations	193
CONCLUSION		195
ANNEX I	MODEL PREDICTIVE CONTROLLER WITH ADAPTIVE NEURAL NETWORKS AND ONLINE STATE ESTIMATION FOR PITCH RATE CONTROL OF THE CESSNA CITATION X.....	199
ANNEX II	ROBUST PITCH CONTROL OF THE CESSNA CITATION X USING C-STAR ARCHITECTURE AND ADAPTIVE NEURAL NETWORKS	221
ANNEX III	FLYING QUALITIES ASSESSMENT FOR NONLINEAR ADAPTIVE CONTROL VALIDATION ON THE CESSNA CITATION X LONGITUDINAL AND LATERAL DYNAMICS	249
LIST OF BIBLIOGRAPHICAL REFERENCES.....		265

LIST OF TABLES

	Page
Table 2.1	Cessna Citation X flight specifications 46
Table 2.2	Actuator dynamics parameter 48
Table 2.3	Level 1 Longitudinal Flight Quality requirements for category B cruise flights 57
Table 2.4	Design Parameters for the Pitch Rate Controller..... 58
Table 2.5	Pitch Rate Reference model dynamic characteristics for Level 1 flight quality 66
Table 2.6	Reference signal and pitch rate dynamic performance results 66
Table 3.1	Actuators' Dynamic Characteristics 92
Table 3.2	Level 1 Flight quality requirements for lateral-directional dynamics and category B flight phase 106
Table 3.3	Lateral Controller Design Parameters in 107
Table 3.4	Reference model signals: Level 1 dynamic characteristics 117
Table 3.5	Reference signal, ideal roll rate controller and reconfigured controller performances on unit signal tracking 119
Table 4.1	Cessna Citation X performance characteristics 146
Table 4.2	CSAS and Autopilots design specifications 148
Table 4.3	Second Order Nonlinear Actuator dynamics 151
Table 4.4	RLS estimation algorithm and covariance matrix update 156
Table 4.5	Pitch rate CSAS controller design equations..... 160
Table 4.6	Vertical Speed Autopilot design equations 160
Table 4.7	Altitude Autopilot design equations 162
Table 4.8	Roll rate CSAS controller design equations 163
Table 4.9	Roll angle CSAS controller design equations 164

Table 4.10	Heading Autopilot design equations.....	165
Table 4.11	Longitudinal CSAS and Autopilot Design Parameters	166
Table 4.12	Lateral CSAS and Autopilot Design Parameters.....	167
Table 4.13	CSAS and Autopilot transient performance results.....	180
Table A I-1	Design Parameters for the Controller	213
Table A II-1	Design Parameters for the Pitch Controller	238
Table A III-1	Level 1 Flying Qualities requirements for longitudinal and lateral- directional dynamics and category B flight phase.....	252
Table A III-2	Reference signal and pitch rate controller performances	257
Table A III-3	Reference signal, ideal roll rate controller and reconfigured controller performances on unit signal tracking	261

LIST OF FIGURES

	Page
Figure 2.1	Cessna Citation X Research Aircraft Flight Simulator (RAFS) manufactured by CAE 47
Figure 2.2	Pitch Rate Model Reference Adaptive Controller using PID, RLS Dynamic Inversion and adaptive NN 49
Figure 2.3	Representation of a Feedforward Neural Network..... 55
Figure 2.4	Pitch rate and elevator responses for cruises at 35,000 ft and 290 kt..... 59
Figure 2.5	Hyperparameter variation during cruise flight 60
Figure 2.6	(a) Covariance matrix $\mathbf{Lk}$ update and (b) prediction error during cruise..... 60
Figure 2.7	Longitudinal states estimation using the RLS algorithm for cruise flight at 10 668 m AGL and 537 km/h 61
Figure 2.8	Selected cruise conditions for $m = 15\,000$ kg and CoG $x_{cg} =$ 25% MAC 62
Figure 2.9	Pitch Rate Response for Different Learning Rates Γ and Forgetting Factors κ at all cruise conditions 63
Figure 2.10	Pitch rate response for three different neural networks gains $\mathbf{KNN}$ values for all cruise conditions 63
Figure 2.11	Sum Squared Error contour plot for 10 flight conditions for (a) ideal conditions, (b) wind gust and (c) moderate turbulence 64
Figure 2.12	Pitch rate response for different configurations for all cruise conditions : (a) PID controller , (b) PID – RLS Dynamic Inversion controller and (c) PID – RLS Dynamic Inversion – NN controller..... 65
Figure 2.13	Short period flight qualities for (a) damping and frequency requirements and (b) CAP requirements 67
Figure 2.14	Pitch rate and elevator responses under tailwind gust..... 68
Figure 2.15	Turbulence profiles at different altitudes, pitch rate and elevator responses..... 68

Figure 2.16	Pitch rate, elevator and adaptive element responses for (a) 50% LOE and (b) actuator noise.....	69
Figure 3.1	Cessna Citation X Level D Research Aircraft Flight Simulator (RAFS) manufactured by CAE.Inc	91
Figure 3.2	Neural Network-based Control Architecture for Roll Rate Control with Recursive Least Square Inversion	93
Figure 3.3	SHL Feedforward Neural Network	103
Figure 3.4	Lateral States and Control Responses during cruise	109
Figure 3.5	Controller Parameter Responses during cruise.....	110
Figure 3.6	RLS parameters update : (a) covariance matrix $\mathbf{Lk}$ (b) roll rate prediction error and (c) yaw rate prediction error during cruise at 35,000 ft and 290 knots	111
Figure 3.7	RLS Lateral states estimation during cruise	112
Figure 3.8	Selected cruise conditions for the CCX with mass $m = 33000$ lb and a center-of-gravity $x_{cg} = 25\%$	113
Figure 3.9	Roll Rate Response for varying Learning Rates and Forgetting Factor for 64 cruise conditions	114
Figure 3.10	Neural Network Learning rate tuning under disturbances for 64 cruise conditions : (a) ideal condition (b) light turbulence (c) moderate turbulence (d) nose wind gust (e) tail wind gust.....	115
Figure 3.11	Controller configurations comparison for 64 cruise conditions : (a) fixed PID controller, (b) PID – RLS Inversion controller, (c) PID – RLS Inversion – NN controller on ailerons, (d) PID – RLS Inversion – NN controller on ailerons and rudders	116
Figure 3.12	Roll rate response for three different roll rate reference signals in Table 5.5 for 64 cruise conditions	117
Figure 3.13	Dutch-roll flight quality assessment for three roll rate reference signals in Table 5.5 for 64 cruise conditions	118
Figure 3.14	(a) Turbulence and (b) Wind Gust profiles and controller responses on roll rate for model reference signal tracking for 64 cruise conditions.....	120

Figure 3.15	Lateral controller responses on roll rate for model reference signal tracking under (a) actuator Loss-of-Effectiveness and (b) actuator noise for 64 cruise conditions	121
Figure 4.1	Research Aircraft Flight Simulator of the Cessna Citation X available at LARCASE laboratory	146
Figure 4.2	Adaptive Pitch rate CSAS and Adaptive Vertical Speed Autopilot architectures	149
Figure 4.3	Adaptive Altitude autopilot architecture	149
Figure 4.4	Adaptive Yaw rate stabilization and adaptive roll rate CSAS architectures	150
Figure 4.5	Adaptive Roll angle CSAS Controller and adaptive heading autopilot architectures	150
Figure 4.6	Single Layer Feedforward Neural Network architecture	157
Figure 4.7	Vertical Speed responses for varying learning rates and NN gain for 64 representative cruise operating conditions	168
Figure 4.8	Overall Sum Squared Error SSE for 64 flight conditions vs learning rates ΓV_z and $KNNV_z$ in the presence of (a) wind gust and (b) turbulence for different learning rate and NN gain variations	169
Figure 4.9	Altitude and heading command and control surface responses for coordinated turn and climb at 35, 000 ft and 290 knots	172
Figure 4.10	Adaptive hyperparameter variations for (a) longitudinal controllers and (b) lateral controllers	173
Figure 4.11	RLS hyperparameter variations during Altitude and Heading command for (a) longitudinal and (b) lateral states	174
Figure 4.12	RLS estimation during Altitude and Heading command for (a) longitudinal and (b) lateral states	175
Figure 4.13	Cessna Citation X flight envelope expressed in terms of 64 representative cruise conditions considered for validation	176
Figure 4.14	Pitch rate q variation with time responses for 64 cruise conditions for: (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations	177

Figure 4.15	Vertical speed variations with time responses for 64 cruise conditions using: (a) PID, (b) PID – NN controller configurations 177
Figure 4.16	Altitude variations with time responses for 64 cruise conditions using: (a) PID, (b) PID – NN controller configurations 178
Figure 4.17	Roll rate p variations with time responses for 64 cruise conditions using : (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations 178
Figure 4.18	Roll ϕ variation with time responses for 64 representative cruise conditions: (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations..... 179
Figure 4.19	Heading control for 64 cruise conditions for : (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations 179
Figure 4.20	(a) Dryden turbulence profile and (b) Wind gust profile..... 181
Figure 4.21	Dual altitude and heading command for 64 representative cruise conditions in the presence of (a) turbulence and (b) wind gusts 182
Figure 4.22	Altitude and heading time variations comparison between (a) flight tests on the RAFS and (b) Simulink simulations..... 182
Figure A I-1	Architecture for the MPC Controller using Recursive Least Square Estimation and Adaptive Neural Network Controller 203
Figure A I-2	Representation of an Artificial Neuron (a) and a SHL Feedforward Neural Network (b)..... 211
Figure A I-3	Pitch Rate and Elevator Responses for Simulation at 35,000 ft and 290 knots for the cruise phase. 214
Figure A I-4	Parameter Responses for Simulation at 35,000 ft and 290 knots for the cruise phase..... 214
Figure A I-5	Selection of the parameter $r\omega$ 215
Figure A I-6	Comparison of 3 values of $r\omega$ for 63 flight conditions..... 216
Figure A I-7	Comparison of 2 different combinations of N_c and N_p for 63 flight conditions..... 216
Figure A I-8	Pitch Rate Response for 3 different Learning Rates Γ and 63 flight conditions..... 218

Figure A I-9	Comparison of the proposed MPC - NN controller with different Configurations for One Flight Condition at 35,000 ft and 290 knots for the cruise phase.	218
Figure A I-10	Zoom on Figure A I-9. Steady state error.....	219
Figure A II-1	Model architecture for pitch control.	226
Figure A II-2	Representation of an Artificial Neuron (a) and a SHL Feedforward Neural Network (b).....	234
Figure A II-3	Pitch Reference Model Configuration.....	236
Figure A II-4	State responses on model validation for one flight condition at 35,000 ft and 290 knots	239
Figure A II-5	Neural Networks hyperparameters and estimated state and control matrices for one simulation at 35,000 ft and 290 knots.....	240
Figure A II-6	Comparison of different controller configurations for one flight condition at 35,000 ft and 290 knots	241
Figure A II-7	Pitch and elevator responses to a step command signal for all flight conditions.....	242
Figure A II-8	Validation for ideal conditions over the flight envelope	242
Figure A II-9	Robustness validation for wind gust.....	244
Figure A II-10	Robustness validation for wind turbulence and wind shear for all cruise conditions	245
Figure A III-1	Designed pitch rate controller including PID, RLS Dynamic Inversion and NN controller	252
Figure A III-2	Designed roll rate controller and yaw damper including PID, RLS Dynamic Inversion and NN controller	254
Figure A III-3	Selected 64 cruise flight conditions.....	256
Figure A III-4	Pitch rate, relative error and elevator deflection responses for all cruise conditions.	258
Figure A III-5	Short Period Flying Quality assessment for pitch rate controller (black dot), reference model performance (red dot), and FQ limits (black line).....	259

Figure A III-6	CAP assessment for pitch rate controller	260
Figure A III-7	Roll rate, relative error and aileron deflection responses for all cruise conditions, for ideal controller (a), for reconfigured controller (b).....	262
Figure A III-8	Yaw rate and sideslip derivative responses for all cruise conditions, for ideal controller (a), for reconfigured controller (b)	262
Figure A III-9	Dutch roll Flying Quality Assessment for ideal controller (a), for reconfigured controller (b).....	263

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial intelligence
ARI	Aileron Rudder Interconnect
CAP	Control Anticipation Parameter
CAS	Calibrated Airspeed
CCX	Cessna Citation X
CoG	Center of Gravity
CSAS	Control Stability Augmented System
DI	Dynamic Inversion
ÉTS	École de technologie supérieure
FQ	Flight Quality
ICAO	International Civil Aviation Organization
LARCASE	Research Laboratory of Active Controls, Avionics, and AeroSErvoelasticity
LOC-II	Loss-of-Control In Flight
LOE	Loss-of-Effectiveness
MPC	Model Predictive Controller
NDI	Nonlinear Dynamic Inversion
NN	Neural Networks
PID	Proportional Integral Derivative
RAFS	Research Aircraft Flight Simulator
RLS	Recursive Least Square

LIST OF SYMBOLS AND UNITS OF MEASUREMENT

c_i	constant
$\mathbf{f}, \mathbf{g}$	unknown nonlinear functions
FQ	Flight Quality
$\mathbf{F}, \hat{\mathbf{F}}$	local state matrix and its estimate, respectively
$\mathbf{G}, \hat{\mathbf{G}}$	local control matrix and its estimates, respectively
h	altitude
$\mathbf{I}_{N \times N}$	unit matrix of size $N \times N$
$\mathbf{L}$	Covariance matrix
m	mass
$\mathbf{M}_k$	computed RLS parameter
p, q, r	roll, pitch, and yaw rates
T_{θ_2}	short period time constant
u, v, w	longitudinal, lateral, and vertical velocity components
$\mathbf{V}, \mathbf{W}$	Neural network weight matrices
V_z	vertical speed
$\mathbf{x}, \mathbf{x}_{lon}, \mathbf{x}_{lat}$	full state, longitudinal state and lateral state vectors
x_{cg}	center of gravity
Γ_i	learning rate matrix corresponding to output i
$\delta_e, \delta_a, \delta_r$	elevator, ailerons and rudder deflections
ξ_{dr}, ω_{dr}	Dutch-roll damping and frequency
ξ_{sp}, ω_{sp}	short period damping and frequency

ϵ_i	RLS approximation error respectively to i -th state
Δt	time step
$\Theta, \hat{\Theta}$	matrix of unknown parameters and its estimate
$\varphi, \aleph$	unknown nonlinear functions
ψ	heading angle
$\beta, \dot{\beta}$	sideslip angle and its derivative
κ	RLS forgetting factor
λ	NN scalar parameter
τ	roll time constant
ϕ, θ, ψ	roll, pitch and yaw angles
σ, σ'	tangent sigmoid function and its gradient

INTRODUCTION

One major component of an aircraft consists of its flight system, which allows them to accomplish a given mission and realize proper maneuvers. Since the beginning of aircraft design and manufacturing, flight controllers have constantly improved, and even later demonstrated their efficiency. Earlier flight control methodologies consist of classical controllers such as Proportional – Integral – Derivative (PID) controllers using phase and gain margin techniques (Nelson, 1998 ; Stevens, Lewis, & Johnson, 2015). These techniques are still in use on aircraft and in modern robotic fields. However, they rely on expert design. Modern controllers such as LQR and robust controllers including Linear Quadratic Gaussian regulators, and H_∞ controllers appeared in the late 1960s with the new interest in spacecraft and supersonic flights. Advanced controllers such as Dynamic Inversion (DI) controllers and Model Predictive Controllers (MPC) were developed in the early 1990s, offering adaptive features, and compensating for nonlinearities. Nowadays, intelligent controllers have emerged, among which we could cite Deep Learning based, Fuzzy and Neural Network controllers. The application of intelligent controllers became possible with recent advancement in hardware, as computers were able to handle high volume of computation and memory.

Following the growing interest in the area of automated control, it will be a question of time for intelligent controllers to be applied to the next aircraft generations. As a matter of fact, the common goal consists of improving flight performance and enhancing flight safety. Since the development of supersonic aircraft such as actual fighter jets, flight controllers have become more complex and sophisticated as they deal with higher order dynamics and nonlinearities. Indeed, they are developed to perform high accuracy maneuvers and often rely on sensory data.

Flight controllers are developed according to the targeted control objective, from low level inner loop control, such as attitude control, to high level outer loop control such as guidance and navigation. Besides, intelligent flight controllers and autopilots have demonstrated their application on quadcopters, which are actually available to the great public for recreational purposes.

0.1 Background and Research Motivation

The aviation industry has been granted as the safest means of transport. However, aerospace companies constantly seek to improve aircraft design as new challenges occur, and new safety measures are necessary. Indeed, with the increasing number of flights and in order to ensure flight safety, extensive research is in progress in the field of flight controls and is led by industrials and academics.

In 2024, the International Civil Aviation Organization (ICAO) reported the rate of fatal accidents that was due to Loss of Control In-Flight (LOC-I)(ICAO, 2024), which represents the second cause of accident after runways during takeoff and landing. According to the Federal Aviation Administration (Federal Aviation Administration, 2009), nearly 60 to 80% of the accidents were due to human factor faults, including those of the pilot and the crew.

A loss of control generally occurs due to uncontrollable events. This instability is often caused by the presence and accumulation of uncertainties, for which the controller was not designed to mitigate. Therefore, it is necessary to develop adaptive controllers that are able to properly function in a changing environment, and thus, to compensate for the lack of robustness of conventional controllers.

Classical and modern control methods are widely applied to commercial aircraft. They usually consist of linear controllers; they often result from expert design and are designed only for specific flight conditions. We can cite gain scheduling controllers (Y. Boughari, Ghazi, Botez, & Theel, 2017a, 2017b). Even though they showed good performance, they lack adaptation in the presence of nonlinearities and unmodelled uncertainties. In fact, linear controllers are mainly designed using a linearized aircraft model and its surrounding environment. Thus, alternative solutions consist of integrating nonlinear controllers, among which we can cite Artificial Intelligence (AI) controllers based on Reinforcement Learning, Neural Networks and Fuzzy logic.

The choice of Machine Learning (ML) is mainly motivated by the recent advancement of computers. Indeed, nowadays computers are more powerful in terms of computation capacity and memory, which are particularly useful for ML applications, which require a high amount of data and computing time. Additionally, several working groups are actually evaluating the application of ML in Aerospace systems. The FAA, the European Union Aviation Safety Agency (EASA) and NASA have addressed AI certification issues in aerospace systems (Agogino et al., 2024 ; European Union Aviation Safety Agency, 2024 ; Federal Aviation Administration, 2024)

0.2 Problem Statement

The literature review in CHAPTER 1 gives an overview of several control algorithms that are applied in Aerospace Systems. Each controller is presented by the date of its design. As a matter of fact, each control algorithm tries to fill the gap of its precedent. For instance, optimal state feedback control is able to reduce the extensive design of a classical controller, by solving an optimization problem, and thus by reducing the controller's dimension. For example, robust control is able to deal with uncertainties for which optimal controllers were not able to mitigate. Then, adaptive control can solve the issues related to changing environment and dynamics, by improving controller robustness, and covering wider range of nonlinearities. Finally, in order to solve linearization issues, intelligent control strategies have been designed to deal with nonlinear systems.

As an example, Neural Networks (NN) are powerful algorithms, that are able to capture highly nonlinear dynamics. Therefore, a new challenge has emerged, which consists of assessing the performance of intelligent controllers, specifically their stability and robustness. In fact, intelligent control has been so far applied in practice on UAVs and simulation platforms and still expecting certification for manned aircraft. However, the practical application of AI control on manned aircraft seems not to be far from now.

Therefore, this research aims to solve these gaps by using AI based adaptive controls applied on Cessna Citation X flight dynamics. The research problem could be summarized with the following question:

How can the performance of aircraft flight controllers be improved using AI while guaranteeing flight safety?

The main reason and attractive characteristics of nonlinear controllers consist of dealing with nonlinearities, such as nonlinear flight dynamics, actuator dynamics or even turbulences. NNs and adaptive filters have been presented as solutions in this thesis. The main objective of control design consists of stabilizing the flight dynamics model, by totally or partially controlling its desired states. However, the design process should meet certain requirements in order to ensure aircraft stability and maneuverability, by including flight qualities in its design. These have been the main guidelines in this research, to which we will refer and assess several times in this thesis.

0.3 Research Objectives

The main objective of this research, as mentioned above, is to design, implement, and validate robust and adaptive flight control systems based on neural networks and adaptive filtering techniques, with the aim of improving aircraft flight performance while ensuring its stability, robustness, and operational safety.

To achieve this objective, the study addresses multiple control tasks corresponding to different aircraft dynamic modes and control loops. An aircraft system can be decomposed into two main dynamic subsystems: longitudinal dynamics and lateral-directional dynamics. In addition, flight control tasks can be classified into low-level control tasks, such as attitude and rate stabilization, and high-level control tasks, such as guidance and navigation. Based on this classification, the research is structured around three (3) specific and complementary

objectives, each of them corresponding to a distinct control study. These objectives are detailed in the next paragraphs :

1. The first objective focuses on the longitudinal control study of the Cessna Citation X, with particular emphasis on pitch-rate regulation. This study addresses low-level longitudinal control through the design of a Stability Augmentation System (SAS) and a Controllability Augmentation System (CAS), aiming to enhance pitch-axis stability, response dynamics, and robustness across the aircraft operating conditions.
2. The second objective addresses lateral-directional control study, including roll-rate control and yaw-rate stabilization. This study also involves the development of SAS and CAS control loops, with the goal of improving roll and yaw stability, coordination, and disturbance rejection.
3. The third objective focuses on high-level control for guidance and navigation, namely on the design and implementation of autopilot functions. These functions are built based on the validated low-level controllers, and they accomplish the high-level control objectives, such as attitude control, trajectory tracking, and overall flight-path regulation.

In addition, the robustness of the proposed control architectures is evaluated through simulation-based disturbance scenarios, including wind gusts and turbulence. In addition, the stability and convergence properties of the adaptive controllers are analytically assessed using Lyapunov stability theory for nonlinear adaptive systems.

Each of the study cases described above corresponds to a peer-reviewed journal article and is presented in CHAPTER 2, CHAPTER 3, and CHAPTER 4, respectively. Accordingly, this thesis is written under a manuscript-based format and is composed of three journal articles that together form the core scientific research contributions corresponding to each objective.

In addition to these main studies, supplementary research works are included in the ANNEX section. These annexed papers present complementary control methodologies and analyses that support and extend the main contributions of the thesis and were published and presented at AIAA conference. Specifically, they include: (ANNEX I) a hybrid Model Predictive Control–Neural Network (MPC–NN) approach for longitudinal flight control, (ANNEX II) a C* control strategy based on Dynamic Inversion and neural networks, and (ANNEX III) a comprehensive flying qualities analysis of nonlinear flight control systems for longitudinal and lateral control algorithms.

0.4 Novelty and Originality

The novelty of this research lies in the design and implementation of a unified adaptive nonlinear flight control algorithm for the Cessna Citation X, based on neural networks and adaptive filtering techniques, such as Recursive Least Squares (RLS) based Dynamic Inversion controller. The proposed controllers are developed for the cruise phase of the aircraft and validated across the entire flight envelope. Unlike conventional approaches that address single aircraft dynamics models, the proposed algorithms simultaneously consider multiple aircraft dynamic modes and control layers, providing very good adaptive control strategies for both longitudinal and lateral-directional dynamics modes.

The resulting control architecture consists of a hybrid nonlinear adaptive control scheme combining intelligent control techniques and adaptive Dynamic Inversion. It provides dual adaptation capabilities through the combined use of RLS estimation and NN-based compensation, enabling online adjustment to modeling uncertainties and dynamic variations in both longitudinal and lateral modes. The designed hybrid controllers accomplished low-level stabilization objectives, and they include Stability Augmentation Systems (SAS) and Controllability Augmentation Systems (CAS), and they provide high-level guidance with adaptive autopilot design.

This research advances the state of the art through the implementation of a hybrid RLS–NN adaptive control architecture for longitudinal pitch rate control as well as lateral roll rate control and yaw rate stabilization of a business jet. The controllers are designed and validated on a realistic, high-fidelity nonlinear aircraft model of the Cessna Citation X, including actuator dynamics and disturbance effects. The adaptive nature of the proposed approach removes evidently the need for classical gain-scheduling techniques, allowing to obtain very good performance, and to maintain it across varying operating conditions using a single controller structure.

Another novelty lies in the controller architecture itself, which combines a baseline PID controller with fixed gains to ensure nominal stability, while allowing online adaptation to the nonlinear components of the control law. Specifically, the Dynamic Inversion controller, the Recursive Least Squares (RLS) estimator, and the neural network (NN) residual compensator are adapted in real time. This structured separation between fixed and adaptive components enhances controller robustness, while preserving stability margins. Additional adaptive control architectures are presented in the ANNEX, including a hybrid Model Predictive Control – Neural Network approach and a C* control architecture based on Dynamic Inversion and neural networks for longitudinal control.

The proposed designs enable the automatic handling of changes in aircraft dynamics due to uncertainties and disturbances. In addition, this work provides a systematic analysis of RLS adaptation mechanisms, including their initialization strategies, forgetting factor selection, and covariance matrix resetting, and it evaluates their impact on the estimation accuracy and closed-loop stability. Neural networks are subsequently integrated as direct adaptive control elements, providing robust compensation even in cases of degraded or biased RLS estimates.

Finally, this research adopts a robustness- and certification-oriented validation methodology. The controller performance is evaluated under wind gust and turbulence conditions, and longitudinal and lateral flight qualities are explicitly assessed, demonstrating Level 1 compliance accordingly to established criteria (Department of Defense, 1997 ; Hakim &

Choukri, 2024 ; Hodgkinson, 1999). Moreover, the nonlinear stability and convergence properties of the proposed adaptive control laws are guaranteed using Lyapunov stability theory for nonlinear adaptive systems (Lewis, Yeşildirek, & Liu, 1995 ; Narendra & Annaswamy, 1987), thereby reinforcing the application of the proposed approach to safety-critical flight control systems.

0.5 Contributions and Thesis Organization

This research has been published in three peer-reviewed journal articles and five conference papers, which have been presented at the AIAA SciTech and the AIAA Aviation Forums. These publications collectively document the theoretical developments, controller designs, and validation studies conducted throughout this thesis.

CHAPTER 1 presents a comprehensive literature review of aircraft flight-control methodologies and state-of-the-art techniques. This chapter establishes the scientific foundation of the research by critically analyzing conventional, nonlinear, and adaptive control approaches, with a particular emphasis on neural-network-based controllers and adaptive dynamic inversion methods. The advantages and limitations of existing techniques are discussed in order to justify the proposed control architectures in this thesis.

CHAPTER 2 presents the first journal article, entitled “Nonlinear Adaptive Longitudinal Controller and Flight-Qualities Validation for a Business Aircraft.” This chapter focuses on the design of an adaptive longitudinal flight-control system for pitch-rate regulation using neural networks combined with Recursive Least Squares (RLS)–based Dynamic Inversion. The robustness of the controller is evaluated under disturbance conditions, and its stability properties are assessed using Lyapunov stability theory for nonlinear adaptive systems.

CHAPTER 3 presents the second journal article, entitled “Robust Adaptive Lateral Control for the Cessna Citation X with Flight Quality Validation and Lyapunov Stability Assessment.” This chapter addresses the design of adaptive lateral-directional controllers, including roll-rate

control and yaw-rate stabilization. The chapter presents robustness performance analyses in the presence of uncertainties and disturbances, along with Lyapunov-based stability verification and flight qualities assessment.

CHAPTER 4 presents the third journal article, entitled “Adaptive Autopilot Design and Implementation for the Cessna Citation X.” This chapter explains several adaptive autopilot architectures for high-level guidance and navigation functions, including vertical-speed, altitude, and heading control. The proposed autopilots are developed upon the validated low-level adaptive controllers and demonstrate consistent performance across the cruise flight envelope.

CHAPTER 5 provides a comprehensive results and discussion section, in which the performance of the proposed adaptive controllers is analyzed across longitudinal, lateral-directional, and autopilot control tasks. This chapter synthesizes the results obtained in the preceding chapters, highlighting controller improvements in terms of tracking performance, robustness to disturbances, and compliance with flight quality requirements. The influence of adaptive control elements, including RLS estimation and neural network compensation, is discussed in detail, and design insights and tuning recommendations are provided to the proposed control methodologies.

The three annex sections present supplementary research contributions and supporting materials.

ANNEX I presents a conference paper on the design of a hybrid Model Predictive Control–Neural Network (MPC–NN) controller applied to the Cessna Citation X. This paper was published at the 2024 AIAA SciTech Forum.

ANNEX II presents a conference paper on robust pitch-rate control using a C^* control architecture combined with Dynamic Inversion and neural-network-based adaptation. This paper was published at the 2024 AIAA SciTech Forum.

ANNEX III presents a conference paper dedicated to the flight-qualities assessment of the proposed longitudinal and lateral controllers, demonstrating their compliance with Level 1 flight quality criteria. This paper was published at the 2025 AIAA SciTech Forum.

CHAPTER 1

LITTERATURE REVIEW

This chapter presents a comprehensive review of control methodologies applied to aerospace systems. The main objective is to identify the current state of the art in aircraft flight control, while critically assessing the strengths and limitations of existing control strategies. It then surveys a range of classical, nonlinear, and adaptive control techniques, highlighting their practical implementations in aircraft control systems. Finally, the key limitations identified in existing literature are summarized, leading to the formulation of the research gap that motivates and guides this thesis.

1.1 Introduction to Aircraft Flight Dynamics

An overview of aircraft flight dynamics is described in this section, along with its primary flight equations, and state space representation. The linear representation of the aircraft dynamics may be described using the following linear state space equation :

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{0.1}$$

where $\mathbf{x}$ is the states vector, $\dot{\mathbf{x}}$ is the state variation, u is the control input vector, $\mathbf{A}$ is the state matrix and $\mathbf{B}$ is the control matrix. $\mathbf{y}$ is the output equation, while $\mathbf{C}$ and $\mathbf{D}$ are the output matrices corresponding to the states and the control inputs.

The state matrix $\mathbf{A}$, is obtained through linearization of the aircraft dynamics, and contains the aircraft aerodynamic stability derivatives. On the other hand, the control matrix $\mathbf{B}$ contains the control effectiveness derivatives. Their approximation values are given in any aeronautical textbook, citing (Nelson, 1998 ; Stevens et al., 2015).

The full longitudinal linear state-space equation can be expressed as follows :

$$\begin{aligned}
\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} &= \begin{bmatrix} X_u & X_w & 0 & -g_0 \\ Z_u & Z_w & u_0 & 0 \\ M_u + M_{\dot{w}}Z_u & M_w + M_{\dot{w}}Z_w & M_q + M_{\dot{w}}u_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix} \\
&+ \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} + M_{\dot{w}}Z_{\delta_e} & M_{\delta_t} + M_{\dot{w}}Z_{\delta_t} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix}
\end{aligned} \tag{0.2}$$

The full lateral-directional linear state-space equation can be expressed as follows :

$$\begin{bmatrix} \Delta \dot{v} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \\ \Delta \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & (Y_r - u_0) & g_0 \cos(\theta_0) & 0 \\ L_v & L_p & L_r & 0 & 0 \\ N_v & N_p & N_r & 0 & 0 \\ 0 & 1 & \tan(\theta_0) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta \phi \\ \Delta \psi \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \tag{0.3}$$

1.2 Introduction to Aircraft Flight Control

Aircraft flight control systems are designed to govern flight stability and dynamics of an aircraft by regulating its dynamic response to pilot commands and external disturbances. In aerospace engineering, the controlled systems include aircraft dynamics, propulsion system dynamics, actuator dynamics, and the surrounding flight environment. The main objective of a flight control system is to ensure that the aircraft exhibits its desired dynamic behavior, in terms of stability, damping, and acceptable transient response, while satisfying performance, safety, and handling-quality requirements across the operational flight envelope.

Historically, aircraft control systems design has evolved from classical linear control techniques toward more advanced nonlinear and adaptive methodologies. Early flight controllers were primarily based on linearized models of the aircraft dynamics, and they relied on classical control methodologies such as Proportional–Integral–Derivative (PID) control, root-locus, and frequency-domain methods. These approaches typically use mathematical representations of an aircraft, commonly referred to as the plant, derived using linearization through transfer functions. Control laws are then designed using output or state feedback to

regulate tracking errors, and to ensure stable response to reference commands. While effective around specific operating points, such linear controllers often require gain scheduling to maintain acceptable performance over a wide range of flight conditions.

As aircraft dynamics are inherently nonlinear and strongly dependent on flight condition, modern flight control design has increasingly incorporated nonlinear and adaptive control techniques. These methods aim to explicitly account for nonlinearities, model uncertainties, and variations in aerodynamic characteristics, thereby they improve model robustness and performance throughout the flight envelope. Adaptive control strategies, in particular, have been studied to reduce reliance on precise modeling and to compensate for unknown or time-varying dynamics in real time.

Regardless of the applied control methodology, aircraft flight control systems must satisfy certification and handling quality requirements. Standards such as MIL-STD-1797A define quantitative criteria for stability, damping, bandwidth, and transient response to ensure acceptable flying qualities and pilot workload. Consequently, the design of flight control laws is not solely a theoretical control problem but is also constrained by certification guidelines and flight quality metrics, which play a central role in evaluating and validating both classical and advanced control approaches.

1.3 Linear control

Classical flight controllers are commonly designed using linearized representations of aircraft dynamics. Linearization is typically performed about a specific operating point, referred to as a *trim* condition, and is obtained by retaining the first order terms of the Taylor series expansion of the aircraft nonlinear equations of motion (Ogata & Yang, 2002). The trimming process consists in determining an equilibrium state in which the aircraft maintains steady flight, defined by flight conditions expressed in terms of altitude, Mach number or airspeed, aircraft mass, center-of-gravity location, and temperature deviations.

Because of the fact that aircraft dynamics vary significantly across the flight envelope, linearized models are only valid around their corresponding trim conditions. Consequently, for large operational flight envelopes, the flight domain must be partitioned into multiple operating regions, each of them corresponding to a distinct trim point (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, Ghazi, & Theel, 2017 ; Yamina Boughari, Ghazi, & Botez, 2017). A dedicated controller is then designed for each operating region, typically using gain scheduling techniques to interpolate between control configurations. Moreover, the consideration of abnormal or off-nominal flight conditions further increases the required number of controllers, adding to the complexity of the overall control architecture.

The linearization and controller design processes are often demanding, as they rely on accurate aircraft models and assume that the linearized states or outputs are measurable or can be reliably estimated during real flight. Despite these challenges, linear control techniques remain widely used in practice due to their simplicity and well-established theoretical foundations. Many linear control strategies have been reported in the literature and applied to operational aircraft systems, including PID control, linear quadratic regulation (LQR), Model Predictive Control (MPC), and H_∞ control.

1.4 Classical control

Classical control is often referred to as linear control theory, and it works very well in various applications. Indeed, the science of linear control has been well established, and its efficiency has been proven in practice. Many control books would assess the synthesis of linear controllers, their stability requirements in terms of phase and frequency margins, pole locations (Dorf & Bishop, 2008 ; Franklin, Powell, Emami-Naeini, & Powell, 2002 ; Goodwin, Graebe, Salgado, & others, 2001 ; Ogata & Yang, 2002). The objective of classical control consists in obtaining good performance in terms of time delay, overshoot, setting time and steady-state error.

As examples of classical controllers, we can enumerate transfer functions and PID controllers. Despite the simplicity of their design and their ease of application, they have shown a lack of performance and adaptation when changes occur in the environment dynamics. In aerospace systems, and especially in flight controllers, classical controllers are usually based on successive single-loop designs, where the inner loop, known as Stability Augmented System (SAS), insures stability, while the outer loop called Controllability Augmented System (CAS) (Nelson, 1998) performs control task. The SAS consists of a feedback gain on the angular velocities and allows to damp the system faster dynamics, for instance the Short Period and the Dutch Roll modes. On the other hand, the CAS consists of a compensator that allows you to track a reference signal by controlling the error signal. The compensator may be a PID, a lead, lag or a notch filter, depending on design specifications. The design of a classical controller often results from successive tuning that mainly relies on trial and error until the desired closed-loop performance is obtained. This process is referred to as *expert design* and may be an extensive task for the control designer. Hence, automating the design process remains a challenge for classical control.

1.4.1 PID Controllers

Proportional Integral Derivative (PID) controllers may be so far the best-known control technique. It is preferred as an initial approach to any control problem before upgrading it to more advanced control. Besides, it is able to handle complex dynamics, and its stability is guaranteed (Nelson, 1998). The main advantage of PID controllers lies in the fact that they do not require the plant mathematical model for their design. The tuning of the controller gains may be obtained directly from dynamic system. The “proportional” gain is computed from the current error and increases the output signal; the “integral” gain is computed from the previous error and removes the steady-state error, but the response is slow; while the “derivative” gain is computed from future errors and the system response is fast.

There are simple techniques for the PID design such as the time-frequency response, root locus or the Ziegler-Nichols method (Nelson, 1998). Despite its good features, PID control remains

a time-consuming process if it is designed for the overall flight envelope of the aircraft. For example, for only attitude control in the pitch, roll and yaw axes, the PID controller requires at least 9 gains for only one operating condition. Moreover, the PID gains should not be excessively tuned in order to be applicable to actuators. However, even though PID controllers take time to design, most control designers refer to PID control to validate their flight controllers, especially when conducting experiments. For instance, in the case of active controls which involve real experimentation with actuators, experimental tests should be conducted safely as loss of materials may occur.

To this end, PID controllers have been extensively used at the Laboratory of Applied Research in Active Controls Avionics and AeroServoelasticity (LARCASE). One can cite the works of (Grigorie, Botez, Popov, Mamou, & Mébarki, 2012b, 2012a ; Grigorie, Popov, Botez, Mamou, & Mébarki, 2012a, 2012b ; Khan, Grigorie, Botez, Mamou, & Mébarki, 2020 ; Mosbah, Botez, Dao, Guezguez, & Zaag, 2016 ; Andrei V Popov, Grigorie, Botez, Mamou, & Mebarki, 2010 ; Andrei Vladimir Popov, Labib, Fays, & Botez, 2008) which used PI controllers on Smart Memory Alloy (SMA) and actuators to control the position of the transition point from laminar to turbulent on a morphing wing. The designer must rely on classical controller approach to deal with highly nonlinear aeroservoelastic dynamics and real experimentations. Once the classical control approach was validated, more advanced controllers were used for morphing wing such as fuzzy-based (Grigorie, Botez, et al., 2012b) or neural network-based controllers (Mosbah et al., 2016). These works were performed within a major research project in collaboration with the well-known companies Bombardier and Thales. Another example is brought by (Koch III, 2019) for UAV applications, for which a neural network controller has been first compared and validated with a PID controller before being deployed on a real UAV.

1.4.2 Gain Scheduling

Gain scheduling is widely applied in both classical and modern flight control systems to ensure acceptable performance over a wide aircraft flight envelope. This approach is effective in addressing aircraft dynamics variation across different operating conditions. However, gain

scheduling remains a time-consuming process. In classical control design, techniques such as root-locus analysis and single-loop modeling must be carried out separately for each flight condition.

To reduce the design effort, several studies have proposed automated gain-tuning approaches by combining conventional control laws with optimization algorithms, including Differential Evolution (Y. Boughari, Ghazi, et al., 2017a, 2017b), Genetic Algorithms, and Particle Swarm Optimization (Ghazi & Botez, 2015b). The resulting controller parameters are typically stored in lookup tables and interpolated online as functions of the operating conditions. Nevertheless, the gain scheduling process often requires the intervention of experienced flight control engineers to define appropriate operating points and validate controller configurations.

Despite their ability to cover large flight envelopes, gain-scheduled controllers are generally for systems with rapidly changing or strongly nonlinear dynamics. As a result, adaptive control techniques have been increasingly investigated as an alternative control, offering the capability to adjust controller parameters online in response to dynamic variations and uncertainties (Koch III, 2019).

1.5 Modern Control

The modern control aims to reduce the time and design cost of the flight controllers. Its architecture consists of multivariable control, where the control inputs are generated from multiple state feedback. In contrast to classical control, state feedback control is a low order control and is particularly efficient as it does not require successive single-loop design. The stability and convergence of the technique have been demonstrated by Pontryagin in 1957 and Bellman in 1962 when setting up the optimal control theory. As of today, feedback control has been successfully applied on Boeing 767 aircraft in order to improve its lateral dynamics (Tischler, 1996). Following the apparition of Fly-By-Wire technologies, issues have emerged such as Pilot Induced Oscillation (PIO), due to the latency between the pilot command and the computer signals.

1.5.1 Optimal Control

Several state optimal feedback techniques exist, such as Pole Placement, Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian regulator (LQG), Model Predictive Controller (MPC) and H_∞ controllers. Pole Placement allows us to allocate the eigenvalues of the system at specific positions on the complex plane for given desired performances. On the other hand, LQR, LQG and MPC controllers are simple low order controllers, which use optimization algorithms over a given cost function computation.

The LQR and LQG methods aim to optimize a quadratic cost function, which imposes a weighting matrix Q on the states and another weighting matrix R on the control inputs. Thus, the main task of the designer consists in finding suitable configurations of the Q and R matrices in order to minimize the cost function, and hence to reduce undesired behaviors (Y. Boughari, Ghazi, et al., 2017a, 2017b). If Q is predominant on R , then the system will struggle to follow the command signal. Contrarily, if R is predominant on Q , the system would be sensitive to small commands such as small noises on the stick. In other words, the designer must find a balance between the control efforts and the aircraft responses. A method was successfully applied by (Nelson, 1998). (Y. Boughari, Ghazi, et al., 2017a, 2017b) successfully automated the weighing process of Q and R using optimization algorithms. They have validated an LQR based flight controller for stability, while they have used a PI controller for tracking. Specifically, they combined the LQR method with a Differential Evolution algorithm, while on the other hand (Ghazi & Botez, 2015b) combined LQR with a Genetic Algorithm in order to tune the optimal gains of the controller, resulting in a hybrid algorithm. They were able to satisfy all controller requirements and assess its robustness against uncertainties for the entire flight envelope, before validating their controllers on the nonlinear model of the Cessna Citation X.

The optimization algorithms are useful to compute the optimal parameters of the controller. Among them, we can cite the Differential Evolution (DE) (Y. Boughari, Ghazi, et al., 2017a,

2017b ; Yamina Boughari, Botez, et al., 2017), the Genetic (GA) (Ghazi & Botez, 2015b), and the Particle Swarm Optimization (PSO) algorithms (S. M. Hosseini, Bematol, Ghazi, & Botez, 2024a ; Khan et al., 2020). Basically, the DE algorithm has the following 5 steps : an initialization, a mutation, a crossover, a selection and then an iteration phase. The GA algorithm is based on generic principle for stochastic search of optimal solutions, while the PSO algorithm searches for an optimal solution among a population of candidate solutions. (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, et al., 2017) and (Ghazi & Botez, 2015b) have successfully applied the optimization algorithm for LQR flight controllers, while (Khan et al., 2020) applied the PSO algorithm to tune the parameters of a PI controller for morphing wing actuator. The Extended Great Deluge algorithm to find the optimal parameters of a Neural Networks controller for morphing wing actuator was applied (Mosbah et al., 2016).

1.5.2 C* Control Architecture

Modern flight control designs emerged in the early 1960s as they were driven largely by the advancements in Aerospace Engineering. In this context, the C* control architecture was implemented in the flight control systems of the Space Shuttle, and later it was applied in civil aviation, notably on the Concorde within the French and British aerospace programs (Tischler, 1996). The C* architecture features a fixed control structure mainly dedicated to longitudinal flight control. It was introduced in response to pilot-induced oscillation (PIO) phenomena observed with the emergence of fly-by-wire technologies, which, in some cases, degraded aircraft flying qualities (Saussié, Akhrif, Bérard, & Saydy, 2009).

The C* control law combines the feedback of the pitch rate q and the normal acceleration a_z to achieve stabilization and command augmentation, typically within Stability Augmentation Systems (SAS) and Control Augmentation Systems (CAS). The C* architecture enables the aircraft to exhibit neutral static stability while ensuring strong damping of the phugoid mode, often resulting in a non-oscillatory response. To ensure safe and smooth operation, the C* architecture incorporates several protection and limiting filters, including constraints on the

angle of attack α , Mach number, and commanded pitch angle θ_{cmd} , thereby minimizing excessive accelerations and decelerations (Tischler, 1996). In this work, the C* control architecture is implemented and evaluated for the longitudinal control of the Cessna Citation X, as detailed in ANNEX II.

1.6 Robust Control

Robustness in flight control refers to the ability of a controller to maintain acceptable performance and stability in the presence of modeling uncertainties and external disturbances. Assessing robustness is a critical step throughout the flight control design process and is a fundamental certification requirement (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, et al., 2017). A major challenge in aircraft control system synthesis resides in accurately representing the inherently nonlinear aircraft dynamics while accounting for the interaction between the aircraft and its surrounding environment.

As discussed previously, classical and many modern control approaches rely on linearized models for controller design. This assumption implies that the aircraft dynamics is sufficiently well known and that an accurate linear approximation can be derived around specific operating envelope points (Y. Boughari, Ghazi, et al., 2017a, 2017b). However, the linearization process often fails to fully capture the aircraft nonlinear dynamics, particularly over wide flight envelopes or under off-nominal conditions. As a result, linearized models may omit critical dynamic features, thus leading to degraded performance or reduced robustness when subjected to uncertainties.

Uncertainties affecting aircraft dynamics may arise from a wide range of sources, including unmodeled nonlinearities, external disturbances, and variations in system parameters. Typical examples include atmospheric effects such as wind gusts, wind shear, turbulence, aerodynamic instabilities, sensor noise and delays, pilot-induced oscillations, as well as mass and inertia variations caused by fuel consumption or payload redistribution. Consequently, flight control

systems must be designed to guarantee safety and stability while demonstrating robust performance in the presence of these unmodeled or poorly modeled phenomena.

Among modern robust control techniques, the H_∞ control method has been widely used in aerospace applications. Designed by Zames and Francis in the early 1980s, the H_∞ method provides guaranteed controller robustness by minimizing the norm of the generalized closed-loop transfer function of the system. The design formulation typically balances two competing objectives: 1) limiting the control effort applied to the actuators and 2) while ensuring accurate tracking of reference commands and disturbance attenuation (Morari & Zafiriou, 1989 ; I. S. Khalil, Doyle, & Glover, 1996). However, a well-known limitation of H_∞ controllers is that their order is at least equal to that of the system, often resulting in high-order controllers that are difficult to implement and require significant computational resources.

To address these challenges, the integration of H_∞ control with optimization techniques has been investigated to reduce controller complexity and computational burden. In particular, (Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017) combined H_∞ synthesis with Differential Evolution and Genetic Algorithms to optimize controller parameters and reduce the order of the generalized transfer function while preserving stability and performance of the controller. The resulting controllers were validated over the entire flight envelope of the Cessna Citation X, and they were subsequently evaluated on different nonlinear aircraft models, demonstrating satisfactory robustness and performance under realistic flight conditions.

1.7 Advanced control

1.7.1 Model Predictive Controller

The Model Predictive Control (MPC) methodology is an advanced controller that can be viewed as an extension of the Linear Quadratic Regulator (LQR) method. In the LQR formulation, the quadratic cost function is optimized over an infinite time horizon, whereas in the MPC the optimization is performed over a finite prediction horizon and is repeatedly

updated at each sampling instant in a receding-horizon fashion. A key advantage of MPC is its explicit ability to incorporate input and output constraints directly into the optimization problem, allowing the handling of actuator saturations and system limitations in a systematic manner.

A comprehensive assessment of the state of the art of MPC applications in aerospace systems was provided (Eren et al., 2017), highlighting both its strengths and its practical challenges. As an online optimization-based approach, MPC is well suited to consider changes in system dynamics by continuously updating the cost function, and constraints at each time step. Also, (T. Zhang, Kahn, Levine, & Abbeel, 2016) successfully combined the MPC with Reinforcement Learning for quadcopter control, where the MPC was used to generate training data for the actor–critic policy. In the context of fixed-wing aircraft, a hybrid MPC–neural network control architecture has been applied to the Cessna Citation X (R. P. Andrianantara, Ghazi, & Botez, 2024) and is presented in ANNEX I of this thesis.

1.7.2 Dynamic Inversion Controller

Dynamic Inversion (DI) emerged as a flight control methodology in the 1980s (Lane & Stengel, 1988), driven by the increasing complexity of aircraft dynamics and the need to design more advanced linear control techniques limited to small perturbations around trim conditions. Early developments of the DI were closely associated with the feedback linearization theory, where the nonlinear system dynamics is algebraically transformed into an equivalent linear form through the use of inverse system models (Enns, Bugajski, Hendrick, & Stein, 1994). In Aerospace applications, this concept gained popularity with the advance of fly-by-wire systems and increased onboard computational capability, enabling real-time implementation of model-based nonlinear control laws. Dynamic Inversion became particularly attractive for high-performance aircraft (Snell & Stout, 1996), where strong nonlinearities and wide flight envelopes made classical linear approaches difficult to apply.

The classical Nonlinear Dynamic Inversion (NDI) (Horn, 2019 ; Moncayo, Perhinschi, Wilburn, Wilburn, & Karas, 2012) relies on an explicit mathematical representation of the aircraft dynamic model to compute the inverse mapping between desired accelerations and control surface deflections. When accurate models are available, the NDI allows the designer to directly impose desired closed-loop dynamics, leading to excellent tracking performance and intuitive control law design. However, early NDI applications revealed a strong sensitivity to modeling errors, actuator dynamics, and unmodeled disturbances, which could significantly degrade performance or compromise stability. These limitations motivated the development of more robust formulations, notably Incremental Dynamic Inversion (Steffensen, Steinert, & Smeur, 2023 ; Zhou, Ho, & Chu, 2021), which emerged in the 1990s and 2000s as a means of reducing model dependence by using incremental measurements of states and control inputs. By operating on local variations rather than global models, INDI significantly improved model robustness to uncertainties and facilitated its practical implementation across larger flight envelopes.

Based on these chronological controller developments, modern DI-based controllers increasingly integrate online estimation and adaptive elements to further mitigate modeling limitations. In this thesis, the DI is implemented using online Recursive Least Squares (RLS) estimation to identify local system dynamics in real time, effectively extending the incremental inversion philosophy. This adaptive formulation enables the inversion to evolve continuously with changing flight conditions, mass properties, and environmental effects, without relying on extensive offline modeling or gain scheduling. Neural Network adaptation is subsequently incorporated in the DI controller to compensate for residual inversion errors and actuator nonlinearities. In this way, the present thesis research situates dynamic inversion within its chronological evolution, from model-based feedback linearization to incrementally adaptive and learning-augmented control, while advancing it toward a robust, scalable, and certification-oriented flight control methodology.

1.7.3 Online State Estimation

State estimation plays a central role in modern adaptive flight control systems, particularly when control laws rely on an online identification of local system dynamics rather than on fixed mathematical models. Recent advances in estimation theory have enabled the development of real-time adaptive filters capable of estimating system states, parameters, and control effectiveness under uncertainties and time-varying conditions. Among the most widely used approaches are Recursive Least Squares (RLS) estimation, as mentioned also above (Haykin, 2013 ; Mahadi, Ballal, Moinuddin, & Al-Saggaf, 2022 ; Mohseni & Bernstein, 2022), Kalman Filtering (KF) and its nonlinear variants (Xiaoqian, Feicheng, Zhengbing, & Hongying, 2022), as well as intelligent estimation techniques based on fuzzy logic (Grigorie, Botez, et al., 2012a ; Hashemi, Botez, & Grigorie, 2020 ; S. M. Hosseini, Ghazi, & Botez, 2024a) and neural networks (Emami, Castaldi, & Banazadeh, 2022 ; Ge, Hang, Lee, & Zhang, 2013 ; Liu, Lyu, Zhang, Yang, & Whidborne, 2025).

On one hand, classical Kalman filtering techniques, including the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), are widely used for state estimation in aerospace systems due to their optimality properties under Gaussian noise assumptions. These methods rely on a priori knowledge of the system dynamics and noise statistics and require repeated linearization or sigma-point propagation for nonlinear systems. While effective for state estimation, their reliance on accurate process and measurement models limits their suitability for adaptive control architectures in which model uncertainty and parameters variation are significant.

On the other hand, Recursive Least Squares estimation (R. P. Andrianantara, Ghazi, & Botez, 2023a ; R. P. Andrianantara, Ghazi, et al., 2024 ; Mahadi et al., 2022 ; Mohseni & Bernstein, 2022) offers a computationally efficient and model-free alternative for online identification of local system dynamics. The RLS estimates unknown parameters by minimizing a weighted prediction error using measured state and control increments, without requiring explicit knowledge of aerodynamic coefficients or global nonlinear models. The RLS forgetting factor

allows the estimator to balance sensitivity to new data while retaining past information, making RLS particularly well suited for tracking slowly time-varying dynamics across the flight envelope. For these reasons, the RLS has been increasingly used in adaptive flight control and dynamic inversion methods, where accurate estimation of control effectiveness and state transition matrices is critical.

In this thesis, the RLS-based state estimation is at the core of the proposed adaptive control architecture. The estimator provides real-time identification of the local linearized dynamics required for DI, effectively allowing the controller to locally linearize online the nonlinear aircraft dynamics. This approach removes the need for extensive trimming and gain scheduling, as the estimated dynamics is continuously adapting to changes in flight conditions, mass properties, and environmental disturbances. The RLS estimator operates solely on measured state and control increments, which enhances control robustness to modeling errors and makes the approach compatible with high-fidelity simulation and its real-time implementation.

However, online estimation in safety-critical systems introduces numerical and stability challenges. Estimation, accuracy and convergence are highly sensitive to the choice of initialization parameters, such as the forgetting factor and covariance matrix in the case of RLS estimation. Small value of the forgetting factors κ may lead to rapid adaptation but can amplify noise and induce oscillations, while large values may improve stability at the expense of responsiveness. To address these issues, parameter reinitialization and threshold-based update strategies are employed in this thesis to prevent divergence and maintain numerical stability. These mechanisms ensure reliable estimation even under disturbances, actuator faults, and abrupt changes in dynamics, which is essential for ensuring the system stability of the subsequent DI step.

Intelligent estimation methods, including fuzzy logic (S. M. Hosseini, Bematol, Ghazi, & Botez, 2024b ; S. M. Hosseini, Ghazi, et al., 2024a) systems and neural networks, offer complementary capabilities by approximating input-output mappings directly from data. Such

methods have been successfully applied to system identification and state estimation in aerospace applications. Nevertheless, intelligent methods typically require large datasets, careful training, and additional validation effort, which can complicate certification processes. For this reason, purely data-driven estimation is not adopted as the primary estimation mechanism in this thesis. Instead, neural networks are used in a structured manner to compensate for residual errors from the RLS-based inversion, preserving interpretability and stability while benefiting from learning-based mechanism.

Hybrid model estimation and control architecture combining classical control, online estimation, and learning-based adaptation have shown promising results in both aircraft and spacecraft applications (Pedro & Meyer, 2007). However, challenges remain in ensuring numerical robustness, convergence guarantees, and certification compatibility. By including the adaptive control architecture in both RLS state estimation and DI, and by augmenting it with neural network compensation, this thesis adopts a balanced approach that leverages the strengths of both model-based and data-driven methods.

Therefore, state estimation is not treated as an auxiliary component but as a core element of the control design philosophy chosen in this thesis. The ability to reliably estimate local dynamics online enables envelope-wide adaptive control, reduces reliance on offline modeling and gain scheduling, and provides a structured pathway toward certifiable adaptive flight control systems.

1.8 Adaptive control

By “adaptation”, we understand the ability of the flight controller to adjust its control parameters in the presence of substantial changes occurring in the system dynamics (Lee & Van Kampen, 2021). As an extension of classical controls, adaptive control can deal with changing dynamics. While the control parameters are fixed for classical controllers, they are variable for adaptive controllers and can be tuned with an adjusting mechanism (Slotine & Li, 1991).

1.8.1 MRAC and L1 Controllers

There are two types of adaptive control: Model Reference Adaptive Controllers (MRAC), and Self-Tuning Controllers (STC). The MRAC is a model-based technique for which the control parameters are tuned in order to track a desired reference with specific performances in terms of rising time, settling time or steady state error. The elements of MRAC models are composed of 1) the plant, which contains unknown parameters with a defined structure, 2) the reference model, the flight controller, which contains the adjustable parameters, and 3) the adaptation mechanism (Slotine & Li, 1991). Self-tuning controllers do not have reference models. The adjusting mechanism computes the optimal control parameters from the output estimates. This former technique requires an online estimator, or in case of state feedback control, it requires state observer. In both configurations, the controller gains are tuned dynamically, based on the signal outputs obtained from the plant.

As stated in Section 1.4.2, several controllers are required to cover a large operating envelope. Therefore, it is necessary to optimize the number of controllers for the entire flight envelope (Y. Boughari, Ghazi, et al., 2017a, 2017b). One can cite the L1 controller (Harris, Elliott, & Tallant, 2022), a Lyapunov function-based controller, which uses a lowpass filter for fast adaptation without losing its robustness. In fact, for this specific controller, the adaptation loop is separated from the control loop, thus increasing the adaptation rate of the controller.

The online tuning principle of gain scheduling may be considered as an adaptive technique as the controller updates its gains when the operating conditions change. On the other hand, as stated in Section 1.4.2, Gain Scheduling relies on a defined mathematical model of the system. Therefore, the controller parameter is not designed for any other model and thus will not adapt if any change occurs in the system dynamics.

1.8.2 Nonlinear Dynamic Inversion

Adaptive flight control strategies often rely on system identification techniques to estimate control parameters online. However, online system identification can be computationally demanding and may converge slowly, particularly when the controller architecture is fixed and the estimated parameters alone are not sufficient to achieve the desired closed-loop performance. The Nonlinear Dynamic Inversion (NDI) offers an alternative adaptive control method that is strongly model-based and aims to explicitly cancel the nonlinearities of the system by inverting the nonlinear relationship between control inputs and system outputs.

In classical NDI, the aircraft nonlinear dynamics are assumed to be known with sufficient accuracy, allowing the control law to directly impose desired closed-loop dynamics. While this approach can provide excellent performance, it is inherently sensitive to modeling errors and uncertainties. To mitigate this limitation, incremental Nonlinear Dynamic Inversion (INDI) has been introduced, where control updates are computed based on incremental changes in measured states rather than on the full nonlinear model. (Grondman, Looye, Kuchar, Chu, & Van Kampen, 2018) successfully applied INDI to a nonlinear model of the Cessna Citation II, demonstrating improved robustness and less dependence on accurate flight dynamics modeling.

The incremental formulation mentioned above can be interpreted as a disturbance observer that estimates future system behavior using current and previous measurements. By relying on measured state increments, INDI significantly reduces sensitivity to modeling uncertainties, aerodynamic parameter variations, and external disturbances. However, this advantage comes at the cost of increased reliance on sensor accuracy and measurement quality, as noisy or delayed measurements can directly affect controller performance.

More recently, several studies have explored the combination of incremental NDI with learning-based techniques, such as RL, to further enhance model adaptability and robustness. In these hybrid approaches, learning algorithms are used to compensate for residual

uncertainties, unmodeled dynamics, or actuator limitations that are not fully addressed by the inversion process (Zhou, van Kampen, & Chu, 2018 ; Feith, 2020 ; Konatala, Van Kampen, & Looye, 2021 ; Lee & Van Kampen, 2021). Such combinations have shown promising results in enabling their online adaptation while maintaining the fast dynamic response and stability properties characteristic of NDI-based controllers.

1.9 Intelligent Control

There are 3 categories of intelligent controls (Koch III, 2019). The first category is knowledge-based control, namely supervised control, such as fuzzy controllers. The second category consists of learning-based adaptive control such as Neural Network adaptive controllers. The third category consists of Reinforcement Learning-based controllers. Knowledge-based control often lacks robustness, as it cannot adapt to unforeseen events, while Reinforcement Learning algorithms take very much time for control purposes.

1.9.1 Fuzzy controller

Fuzzy control is an artificial intelligence–based approach that relies on linguistic variables, membership functions, and rule-based reasoning to address nonlinear and uncertain system dynamics. Fuzzy methods can be applied both to control design and system identification, particularly when accurate mathematical models are difficult to obtain. A typical fuzzy controller consists of three main components (Hashemi et al., 2020 ; S. M. Hosseini, Bematol, et al., 2024b ; S. M. Hosseini, Ghazi, et al., 2024a) : 1) a fuzzifier, which converts numerical input variables into linguistic variables using predefined membership functions; 2) a knowledge base, which contains the fuzzy membership functions and a set of IF–THEN inference rules; and 3) a defuzzifier, which maps the inferred linguistic output back into a numerical control signal.

Conventional linear controllers are generally effective only in the vicinity of specific operating points, and they often fail to adequately handle the nonlinear dynamics of real aerospace systems. As system dynamics become increasingly complex and difficult to describe using

classical analytical models, alternative control approaches based on artificial intelligence have been developed. In recent years, very much research has focused on nonlinear control strategies, including fuzzy control, to overcome the limitations of linearization-based methods. For example, (Grigorie, Botez, et al., 2012b, 2012a) have proposed fuzzy-based approach for morphing wings control.

Fuzzy logic methods have also been successfully applied to the identification of highly nonlinear aeroelastic phenomena, such as capturing flutter dynamics using F-18 aircraft flight-test data of control surface deflections (Boely, Botez, & Kouba, 2011 ; Kouba, Botez, & Boely, 2009, 2010). In the context of control applications, (Grigorie, Botez, et al., 2012b, 2012a), followed by (Kammegne, Botez, Grigorie, Mamou, & Mébarki, 2017), employed fuzzy controllers combined with proportional–integral (PI) control to delay the laminar-to-turbulent transition on a morphing wing. More recently, (Hashemi et al., 2020) designed a model-reference-based fuzzy controller for nano air vehicles, demonstrating sufficient robustness to external disturbances. Finally, (S. M. Hosseini, Bematol, et al., 2024b ; S. M. Hosseini, Ghazi, et al., 2024a) has designed fuzzy controller for the control of the Cessna Citation X business jet aircraft and obtained very good results, including low and high level control.

1.9.2 Neural Networks Controller

Neural Networks (NNs) are universal function approximators and have been widely applied in aerospace engineering for both system identification and control tasks, particularly when dealing with highly nonlinear dynamics. In the context of system identification, (Boely et al., 2011 ; Boely & Botez, 2010) successfully identified strongly nonlinear aeroservoelastic models from flutter test data using NN, achieving fit coefficients exceeding 92%. Similarly (R. P. Andrianantara, Ghazi, & Botez, 2021 ; Rojo Princy Andrianantara, Ghazi, & Botez, 2021 ; Zaag & Botez, 2017 ; Zaag, Botez, & Wong, 2019), employed neural networks to identify nonlinear turbofan engine models, demonstrating the effectiveness of NNs in capturing complex propulsion dynamics.

The application of NN to flight control systems for commercial aircraft remains an active area of research, particularly with respect to its certification. One of the NN main challenges lies in the lack of established stability and performance criteria for nonlinear and learning-based controllers, as the current certification standards are largely based on linear control theory. Nevertheless, stability analysis of nonlinear controllers can be conducted using well-known theoretical tools, such as the circle criterion introduced by (Zames, 1966) and Lyapunov-based methods for assessing asymptotic stability of nonlinear systems (Franklin et al., 2002).

Neural Networks (NNs) can be classified according to their structural characteristics, including feedforward networks, recurrent neural networks, nonlinear autoregressive models with exogenous inputs (NARX), convolutional neural networks, and graph neural networks. Deep Neural Networks are defined by the presence of multiple hidden layers, enabling them to represent complex nonlinear mappings. In the field of UAV, (Koch, Mancuso, & Bestavros, 2019 ; Koch, Mancuso, West, & Bestavros, 2019), successfully applied Deep Neural Networks within a reinforcement learning method for the attitude control of a quadcopter, demonstrating strong performance in highly nonlinear flight regimes.

One of the earliest practical demonstrations of neural-network-based flight control on a fixed-wing aircraft was conducted by NASA's Dryden Flight Research Center using the McDonnell Douglas F-15B. In this Intelligent Flight Control System program, Sigma-Pi neural networks were employed for two objectives. In the first phase, NNs were used for online identification of changes in aircraft dynamics, followed by DI to generate control signals that accounted for these variations. In the second phase, NNs were directly integrated into the control loop, operating in parallel with conventional controllers to provide adaptive control augmentation. These flight tests demonstrated the feasibility of neural-network-based adaptive control, but they also highlighted the necessity of validating advanced flying-quality criteria for learning systems, which differ fundamentally from those used for linear controllers (Schumann & Liu, 2010 ; Smith, Barhorst, & Urnes, 2010 ; Williams, 2004 ; Williams-Hayes, 2005).

1.10 Reinforcement Learning (RL) based Controllers

1.10.1 Principles

The Reinforcement Learning (RL) is a promising approach for flight control and has been proposed as an alternative to supervised and classical control methods (Koch, Mancuso, West, et al., 2019), (Koch, Mancuso, & Bestavros, 2019), (Sun & van Kampen, 2021). Unlike model-based control techniques, the RL does not require prior knowledge of aircraft dynamics and can directly synthesize control laws using Deep Neural Networks. RL-based controllers have primarily been applied to Unmanned Aerial Vehicles (UAVs). Despite their relatively small size, UAVs exhibit strongly nonlinear dynamics, particularly during aggressive or acrobatic maneuvers. The adoption of RL in UAV flight control has been facilitated by fly-by-wire technologies, advances in onboard computational hardware (CPU/GPU), and the availability of high-fidelity simulations (Koch III, 2019), enabling extensive and cost-effective testing

As explained earlier, linear controllers do not consider nonlinearities which may lead to uncontrolled states, and thus to degradations and damages. Traditional control does not have the ability to learn nor plan (Koch III, 2019). They are also expensive in terms of time design, due to the large range of operating conditions. These issues can be tackled by using RL-based controllers, which reduces significantly the number of controllers to one single generalized controller, and would remove the necessity for extensive gain scheduling and system identification for adaptive control. Furthermore, the designer would avoid the trimming of initial conditions.

The RL-based controller is a state-of-the-art flight controller that belongs to Machine Learning algorithms. It is model-free and self-adaptive, which makes it attractive nowadays. It is suitable for nonlinear control and has online learning capability. The RL based controller learns an optimal control policy based on real-time observations through measurements (Lee & Van Kampen, 2021). As an example, the RL based controller has been proposed to control the transition phase of a VTOL UAV from vertical to horizontal flight (Al-Gabalawy, 2019). This choice stands for the fact that the transition phase is highly nonlinear and cannot be linearized

with common methods. RL based controllers have been tested so far on quadcopters (Koch III, 2019) and simulations of fixed wing aircraft (Lee & Van Kampen, 2021).

The basic formulation of the RL method is known as a Markov Decision. It was introduced by Richard Bellman in later 1950 (Sutton & Barto, 2018) and is related to as *reward engineering*. It is described as an optimal way to move from one position to another. The RL elements are the agent, the environment and the policy. The agent is the decision maker, in our case it is the flight controller, and learns a policy from past observations, by receiving a scalar reward and position signal from the aircraft. The observations are then collected from sensor measurements, and as a result, the agent outputs action commands on aircraft control surfaces. On the other hand, the aircraft receives action from the agent, and it outputs new states and new scalar reward.

The RL research methodologies have identified the best structure for control algorithm, which is the actor-critic architecture, where the actor network provides control signals, while the critic network provides evaluation with reward function (Konatala et al., 2021). It is also possible to add state estimator between the actor and the critic, as applied on an F-16 aircraft by (van Kampen, Chu, & Mulder, 2006), and on a Cessna Citation II aircraft by (Zhou et al., 2018 ; Feith, 2020 ; Konatala et al., 2021 ; Lee & Van Kampen, 2021). They demonstrated good results, making them suitable examples of RL based controllers on fixed wing aircraft. The actor-critic configuration was further improved by adding target-critic component to improve the learning stability (Heyer, Kroezen, & Van Kampen, 2020), while a globalized configuration in which the critic approximated both the cost function and its derivatives with respect to the state was used by (Zhou et al., 2018).

1.10.2 Issues and Challenges of RL-based controller

The RL based controller demands a high computational cost as it needs to solve an optimization problem for a large state space. The main risk of RL based controller is its tendency to explore the environment (in terms of ML) as the agent may diverge in the wrong direction or unstable

states. This phenomenon is named as *Curse of Dimensionality*. Some researchers have addressed this issue by adding a safety layer to the RL controller, that increases its safety and ensure convergence (Feith, 2020). The additional layer defines the boundaries of the controller during training. It consists of a negative reward, which penalizes unwanted states. As a result, it limits the exploration ability of the RL algorithm to specific directions, so the agent will learn to “avoid bad cases”. As in the case with most of the AI algorithms, there is still an unsolved gap regarding its application and solution stability.

(Koch, Mancuso, West, et al., 2019) have addressed several challenges for the RL application on quadcopter, such as on the control oscillations minimization. To this end, they included “the control outputs” into “the reward function”. As a result, they were able to fly a quadcopter with high precision in the real world. (Al-Gabalawy, 2019) noted that the agent had the particularity to slow down its training as it approached its goal, and the use of Long Short-Term Memory (LSTM) network was proposed to mitigate this problem.

1.11 Handling Qualities

Before deployment of flight control systems on operational aircraft, they must satisfy a rigorous certification process to ensure its adequate stability, robustness, and handling qualities. These requirements are formally defined in standards such as MIL-STD-1797A 1797A (Department of Defense, 1997), which establishes quantitative criteria for acceptable aircraft response characteristics. While individual manufacturers may adopt additional internal performance policies, compliance with handling-quality standards remains a fundamental requirement for both manual and automatic flight control systems.

Handling qualities, first defined by Cooper and Harper, classify aircraft performance according to its class, flight phase, and pilot satisfaction level. These criteria aim to ensure safe and predictable aircraft behavior in both its longitudinal and lateral–directional motions. For linear systems, handling-quality requirements are typically expressed in terms of time- and frequency-domain metrics, including pitch acceleration peak time, pitch and bank attitude

frequency responses, average phase rate and amplitude, short-period mode frequency and damping, Dutch-roll mode frequency and damping, closed-loop bandwidth, and phase and gain margins (Y. Boughari, Ghazi, et al., 2017a, 2017b).

In addition to these HQ criteria, Saussié et al. (2009) and Ghazi and Botez (2015b) investigated the drop back criterion as a handling-quality metric. These authors introduced the concept of *guardian maps*, in which handling-quality requirements are translated into admissible regions of the complex plane. Control gains are automatically tuned such that the closed-loop system eigenvalues remain within these predefined regions. As a result, the guardian-map technique ensures generalized stability and robustness consistent with handling-quality constraints. Despite its effectiveness, this approach is computationally intensive and may require significant design effort.

It should be noted that the handling-quality criteria defined in MIL-STD-1797A were initially developed for linear systems, and their control laws. For nonlinear and adaptive controllers, no standardized flying-quality requirements currently exist to directly assess their stability and robustness. Nevertheless, the boundedness and stability of nonlinear systems can be evaluated using established theoretical tools, including the circle criterion (Zames, 1966) and Lyapunov-based stability theory (Franklin et al., 2002). Therefore, in this research, these nonlinear stability criteria are employed in accordance with conventional handling-quality metrics to validate the proposed control algorithms in terms of both stability and robustness.

1.12 High Fidelity Nonlinear Model

The design and validation of any flight control system rely mainly on the availability of an accurate mathematical representation of the aircraft dynamics. This requirement implies that the underlying model must capture the essential nonlinear behavior of the aircraft with sufficient fidelity to allow controller development and subsequent evaluation prior to its deployment. High-fidelity nonlinear aircraft models are typically developed by integrating multiple detailed subsystems (Ghazi & Botez, 2015a), including aerodynamic coefficient

databases, atmospheric and weather models, wind and turbulence models, propulsion systems, sensor and actuator dynamics, equations of motion, as well as embedded flight control and autopilot modules.

Previous studies have demonstrated in our LARCASE laboratory that accurate modeling of the Cessna Citation X business aircraft enables the successful design and validation of advanced Linear Quadratic Regulator (LQR) and H_∞ controllers, over the entire flight envelope (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017 ; Ghazi & Botez, 2015b). In this context, a high-fidelity nonlinear model of the Cessna Citation X has been developed and implemented in MATLAB/Simulink at the LARCASE laboratory for research purposes. The resulting flight simulator provides realistic aircraft behavior, and it allows linearization of the nonlinear dynamics around selected trim conditions, thereby supporting both linear and nonlinear control design methodologies.

The importance of high-fidelity flight dynamic models was further emphasized in the development of learning-based flight controllers. Several studies conducted at Delft University of Technology have employed a detailed flight simulator of the Cessna Citation II to investigate RL-based control strategies (Zhou et al., 2018 ; Feith, 2020 ; Konatalla et al., 2021 ; Lee & Van Kampen, 2021). In these RL-based applications, a digital twin of the aircraft was required to safely train control policies in simulation before transferring them to real systems. Similarly, (Koch, Mancuso, West, et al., 2019) successfully designed and validated an RL-based controller using a simulated environment prior to its implementation on a physical quadcopter platform.

A major challenge associated with simulation-based controller development is the so-called reality gap, which refers to performance degradation when it is transitioning from simulation to hardware. In reinforcement learning applications, this issue is further compounded by reward functions, that may include environment-specific components, that are not fully captured in simulation (Koch, Mancuso, West, et al., 2019). To mitigate this limitation, the realism of the simulated environment must be enhanced by introducing variability in model

parameters, such as sensor and actuator dynamics, aerodynamic uncertainties, external disturbances, and wind effects. This approach, known as domain randomization, improves controller robustness by exposing the RL algorithm to a wide range of operating conditions. Consequently, continuous updating of the simulator using real-world data is essential to progressively improve simulation fidelity and enable effective real-time adaptation.

1.13 Conclusion

This literature review has presented a comprehensive overview of flight control methodologies, ranging from classical linear approaches to modern nonlinear and intelligent control strategies. Through the critical analysis of existing techniques, several limitations have been identified that continue to challenge the design and deployment of advanced flight control systems. In particular, classical and modern controllers remain highly dependent on accurate modeling, extensive gain scheduling, and expert-driven tuning procedures, which limit their adaptability and scalability across large flight envelopes. Moreover, while intelligent control methods have shown strong potential, the lack of unified design methods and certification-oriented validation strategies remains a significant gap that could be filled in future.

This review highlights that modern state-feedback controllers can achieve good performance, but they still rely heavily on manual design expertise and offline tuning. Previous efforts to automate controller synthesis, such as combining optimal control with evolutionary algorithms, have reduced design effort, but they have not eliminated the need for gain scheduling, resulting in increased computational and memory requirements. These limitations motivate the investigation of alternative adaptive control architectures capable of operating without prior knowledge to the model and while maintaining robustness and stability.

In this context, NN-based adaptive control combined with DI emerges as a promising solution. Rather than relying on purely data-driven or model-free approaches, this method combines the strengths of nonlinear DI to shape desired closed-loop dynamic models with Recursive Least Squares (RLS) state estimation and neural network controllers to compensate for modeling

uncertainties and time-varying dynamics. Such an approach enables online model adaptation without the need for extensive gain scheduling or multiple controllers, while remaining compatible with Lyapunov-based stability analysis and certification-oriented validation practices.

Accordingly, this thesis focuses on the development and validation of an adaptive flight control architecture based on RLS-based dynamic inversion and NN controllers. The proposed methodology is applied to a high-fidelity nonlinear simulator of the Cessna Citation X, allowing full evaluation across a wide range of operating conditions. The next chapters present the journal papers on the design, development and validation of adaptive flight controllers for the Cessna Citation X.

CHAPTER 2

NONLINEAR ADAPTIVE LONGITUDINAL CONTROLLER AND FLIGHT QUALITIES VALIDATION FOR A BUSINESS AIRCRAFT

Rojo Princy Andrianantara ^a, Georges Ghazi^a, and Ruxandra Botez ^a

^a Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE), System Engineering Department, École de technologie supérieure, 1100 Rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3

Article published in *The Aeronautical Journal*, February 2026
<https://doi.org/10.1017/aer.2026.10132>

Résumé

Cet article présente la conception d'un système de commande de vol adaptatif non linéaire pour la dynamique longitudinale du Cessna Citation X. Le taux de tangage de l'avion est contrôlé à l'aide d'une combinaison d'inversion dynamique non linéaire basée sur les moindres carrés récursifs et d'un contrôleur adaptatif à réseau de neurones. L'algorithme des moindres carrés récursifs fournit des estimations de paramètres en ligne, tandis que le réseau de neurones compense les erreurs de modélisation résiduelles grâce à l'adaptation des poids en ligne. Afin d'améliorer la robustesse et d'assurer la stabilité, un contrôleur proportionnel-intégral-dérivé à gain fixe est intégré à la structure de contrôle. Contrairement aux contrôleurs à séquence de gains classiques, où les gains PID varient en fonction des conditions de vol, le contrôleur adaptatif proposé utilise un seul ensemble de gains fixes. Le composant adaptatif met à jour l'action de contrôle en ligne, ce qui permet à la même configuration de contrôleur de fonctionner efficacement dans les 64 conditions de croisière sans aucune programmation de gain. Une méthodologie de réglage systématique est introduite pour initialiser les moindres carrés récursifs, sélectionner les facteurs d'oubli et appliquer des réinitialisations de la matrice de covariance afin de garantir une adaptation précise. Le contrôleur est capable de suivre un modèle de référence de tangage qui satisfait aux exigences de qualité de vol longitudinal. La robustesse est évaluée dans des conditions de perturbations réalistes, notamment des rafales de vent, des turbulences de Dryden, une perte d'efficacité des actionneurs et le bruit des actionneurs. Les résultats de la simulation démontrent que le contrôleur permet un suivi précis

de la référence tout en maintenant des qualités de vol de niveau 1. La stabilité est formellement garantie à l'aide d'une analyse basée sur Lyapunov, ce qui assure une adaptation non linéaire fiable. Les résultats soulignent la capacité du contrôleur adaptatif hybride conçu à surmonter les limites de la linéarisation, du séquençement de gains et de la sensibilité de l'estimateur, laissant entrevoir une méthode pratique et certifiable pour l'intégration de systèmes de contrôle de vol adaptatifs intelligents dans des avions commerciaux tels que le Cessna Citation X.

Abstract

This paper presents the design of a nonlinear adaptive flight control system for the longitudinal dynamics of the Cessna Citation X. The aircraft pitch rate is controlled using a combination of Recursive Least Squares -based nonlinear dynamic inversion and an adaptive neural network controller. The recursive least squares algorithm provides online parameter estimates to support the inversion, while the NN compensates for residual modeling errors through online weight adaptation. To enhance robustness and ensure stability, a fixed-gain Proportional–Integral–Derivative controller is integrated into the control structure. Unlike conventional gain-scheduled controllers, where PID gains vary with flight condition, the proposed adaptive controller uses a single baseline set of fixed gains. The adaptive component updates the control action online, enabling the same controller configuration to operate effectively across all 64 cruise conditions without any gain scheduling. A systematic tuning methodology is introduced for initializing the recursive least squares, selecting forgetting factors, and applying covariance resets to ensure accurate adaptation. The controller is able to track a pitch-rate reference model that satisfies longitudinal flight quality requirements. Robustness is assessed under realistic disturbances, including wind gusts, Dryden turbulence, actuator loss-of-effectiveness, and actuator noise. Simulation results demonstrate that the controller achieves precise reference tracking while maintaining Level 1 flight quality. Stability is formally guaranteed using Lyapunov-based analysis, ensuring reliable nonlinear adaptation. The findings highlight the ability of the designed hybrid adaptive controller to overcome limitations of linearization, gain scheduling, and estimator sensitivity, forecasting a practical and certifiable method for the integration of intelligent adaptive flight control systems into commercial aircraft such as the Cessna Citation X.

2.1 Introduction

Flight control systems are critical for maintaining stability and safety in aircraft operations. According to the Federal Administration Aviation (FAA), human errors contribute to nearly 60% to 80% of aircraft accidents (Federal Aviation Administration, 2009), which are often due to unmitigated environmental hazards or system failures. Advanced flight control laws therefore play a central role in reducing pilot workload, enhancing autonomy, and improving operational safety. Compared to costly structural modifications, software-based enhancements at the control level, such as flutter suppression or maneuver load alleviation, provide more flexible and efficient solutions (Wunderlich, Dähne, Reimer, & Schuster, 2022). This motivates the development of adaptive and intelligent controllers that can reliably handle uncertainties in real time.

Traditional flight control architectures still primarily rely on linear controllers such as Proportional–Integral–Derivative (PID) and Linear Quadratic Regulator (LQR) designs (Stevens et al., 2015). These methods are straightforward and well-established, offering proven stability with guaranteed gain and phase margins. However, they require precise gain scheduling (Reynolds, Pachter, & Houppis, 1996) to ensure adequate performance across the aircraft’s full envelope and under varying operating conditions. The tuning of these controllers is based on linearized models, which often neglect higher-order nonlinearities (Stevens et al., 2015). Consequently, performance degrades and stability margins may deteriorate when the aircraft operates outside the scheduled regime. Robust control techniques such as H_∞ (Efremov, Mbikayi, & Efremov, 2021 ; A. Khalil & Fezans, 2021) and Linear Quadratic Gaussian (LQG) controllers (Ferrier et al., 2018) mitigate certain uncertainties, but they are still strongly dependent on accurate mathematical models.

Advanced nonlinear control methods, such as Nonlinear Dynamic Inversion (NDI) (Horn, 2019 ; Moncayo et al., 2012 ; Steffensen et al., 2023) and Model Predictive Control (MPC) (Doff-Sotta, Cannon, & Bacic, 2025), address several of the limitations of traditional linear

techniques. However, both approaches depend heavily on precise state-space models (Eren et al., 2017 ; Harris et al., 2022 ; Kouvaritakis & Cannon, 2016), making them sensitive to parameter variations, unmodeled dynamics, or environmental disturbances. This reliance on precise linearization remains one of the main obstacles to their widespread application in commercial aircraft.

To mitigate this dependence, online estimation techniques such as Recursive Least Squares (RLS) (Haykin, 2013 ; Mahadi et al., 2022 ; Mohseni & Bernstein, 2022) and Extended Kalman Filtering (EKF) (Xiaoqian et al., 2022) have been developed, enabling incremental model updates (Konatala, Milz, Weiser, Looye, & van Kampen, 2024) and adaptive compensation of the dynamic inversion (R. P. Andrianantara et al., 2023a ; N. Nguyen, 2011 ; Quintin, Andrianantara, Ghazi, & Botez, 2024) or MPC controllers (R. P. Andrianantara, Ghazi, et al., 2024). Although effective in principle, their performance is often sensitive to initialization conditions, forgetting factor selection, and numerical instabilities, which can compromise real-time implementation.

In parallel, intelligent control approaches have demonstrated the ability of neural networks (Emami et al., 2022 ; Ge et al., 2013 ; Liu et al., 2025) fuzzy logic (Grigorie, Botez, et al., 2012a ; Hashemi et al., 2020 ; S. M. Hosseini, Ghazi, et al., 2024a), and hybrid adaptive controllers (Pedro & Meyer, 2007 ; M. Lungu & Lungu, 2017 ; R. Lungu, Lungu, & Efrim, 2021) to capture nonlinearity and compensate for unmodeled dynamics. Robust adaptive control, in particular, has raised interests due to its effectiveness in compensating for time-varying dynamics and in controlling highly nonlinear systems, including actuator dynamics (Steffensen et al., 2023).

Foundational works by (N. Nguyen, 2011) introduced hybrid adaptive control architectures that combines baseline inversion controllers, indirect parameter estimation using RLS and adaptive neural networks adaptive controllers for compensation (N. Nguyen, 2011 ; Park, Shin, Lee, & Antonios, 2021). These studies validated the hybrid concept for damaged aircraft and launch vehicles (N. Nguyen & Krishnakumar, 2007). More recent applications on the Cessna

Citation X (R. P. Andrianantara et al., 2023a ; R. P. Andrianantara, Ghazi, et al., 2024 ; S. Hosseini, Inga, Ghazi, & Botez, 2023) have demonstrated the potential of dynamic inversion combined with neural network controllers. However, while these studies established the foundation for hybrid adaptive dynamic inversion, their scope was limited in terms of flight conditions and did not address certification-level validation. This study extends their findings by systematically analyzing RLS estimation sensitivity, integrating neural networks adaptive learning, and demonstrating Level 1 flight qualities under FAA standards.

This paper advances the state of the art by implementing a hybrid RLS–NN adaptive controller for the longitudinal pitch-rate control of the Cessna Citation X. Unlike prior studies, the controller is designed and validated across for 64 flight conditions covering the aircraft cruise envelope, thereby removing the need to gain scheduling. In this architecture, the baseline PID controller operates with fixed gains to provide stability, while only the RLS estimator and NN residual compensator adapt online. This design eliminates the need for repeated across conditions, as adaptation is handled automatically by the adaptive controllers. Furthermore, this study provides a systematic analysis of RLS adaptation mechanisms, including initialization, forgetting factors, and covariance resets, and evaluates their impact on both stability and accuracy. Neural networks are then integrated for direct control, providing robust adaptation even in case of degraded RLS estimates. Importantly, flight qualities are explicitly validated, showing Level 1 compliance (Department of Defense, 1997 ; Hakim & Choukri, 2024 ; Hodgkinson, 1999), while nonlinear stability is formally guaranteed through Lyapunov-based analysis (Lewis et al., 1995 ; Narendra & Annaswamy, 1987).

Related controller structures with different compensation strategies have been assessed by other researchers. (Steffensen et al., 2023) examined nonlinear DI with incremental control perspectives, including first-order actuator dynamics, showing improved robustness but without NN augmentation nor analysis of wide-range uncertainties. Harris et al. (Harris et al., 2022) applied L1 adaptive DI ensuring fast adaptation but with a strong dependence on accurate modelling and without formal flight quality assessment. More recently, (R. P. Andrianantara, Ghazi, et al., 2024) extended their earlier study by combining MPC with RLS

and NN augmentation for the Cessna Citation X, demonstrating performance gains but without systematic RLS estimator tuning or robustness validation. In contrast, the present study incorporates higher-order actuator dynamics, introduces a formal tuning methodology, and conducts comprehensive robustness testing. Taken together, these contributions help bridge the gap between theoretical adaptive controller designs and certifiable implementations suitable for commercial aircraft.

Several researchers have focused their efforts on unmanned aerial vehicle (UAV) applications, where agility and adaptability are prioritized over certifiability. This is made possible due to the fact that UAVs present less risk to human life, as being remotely controlled. O’Connell et al. (O’Connell et al., 2022) demonstrated rapid NN-based adaptation for UAV flight in strong winds, but relied primarily on reinforcement learning without model-based stability guarantees. Zhou et al. (Zhou et al., 2021) proposed extended incremental DI with optical-flow feedback for micro air vehicles, confirming feasibility but only in small-scale systems. Earlier, Pedro and Meyer (Pedro & Meyer, 2007) applied NN-based DI to fighter aircraft, showing improved nonlinear tracking in simulations but without robustness validation. Similarly, Lungu & Lungu (M. Lungu & Lungu, 2017) applied NN–DI to landing autopilots, enhancing robustness during approach and flare, while Lungu et al. (R. Lungu et al., 2021) later extended NN–DI concepts to spacecraft control. These studies confirm the adaptability of NN-augmented DI across various domains; however, they are limited to UAV-scale or highly specific applications. This study, instead, focuses on manned commercial aircraft, for which a high-fidelity research aircraft flight simulator is available, thereby extending NN–DI hybrid approaches into a certifiable, envelope-wide method.

In summary, the benefits of the proposed controller are twofold. First, the architecture is simplified, since gain-scheduled controllers are replaced by a single adaptive configuration capable of covering the entire cruise envelope. Second, the controller demonstrates robustness against a wide range of uncertainties, including turbulence, actuator loss-of-effectiveness, and actuator noise, while maintaining compliance with flight quality requirements.

By addressing long-standing limitations related to gain scheduling, model dependence, and estimator sensitivity, this work provides a practical and certifiable method for integrating intelligent adaptive controllers into commercial aircraft. While previous studies (R. P. Andrianantara, Ghazi, et al., 2024 ; Campbell, Nguyen, Kaneshige, & Krishnakumar, 2009 ; Harris et al., 2022 ; R. Lungu et al., 2021 ; N. Nguyen, 2011 ; O’Connell et al., 2022 ; Pedro & Meyer, 2007 ; Steffensen et al., 2023 ; Zhou et al., 2021) have validated adaptive NN – DI concepts in specific operating conditions, UAVs, or specialized mission phases, this study systematically assesses 64 cruise conditions of the Cessna Citation X, integrates NN residual learning with RLS sensitivity analysis, incorporates higher-order actuator dynamics, and validates flight qualities against FAA Level 1 standards. This represents a significant step beyond feasibility demonstrations and aligns with ongoing certification initiatives from the FAA, EASA, and NASA (Agogino et al., 2024 ; European Union Aviation Safety Agency, 2024 ; Federal Aviation Administration, 2024), thereby reinforcing both the relevance and applicability of the proposed approach.

The remainder of this paper is organized as follows: Section 2 presents the methodology and controller design; Section 3 details the tuning strategy and results for ideal conditions; Section 4 provides robustness evaluation; Section 5 discusses findings; Section 6 concludes with final remarks. The Lyapunov Stability assessment is provided in the Appendix.

2.1.1 Aircraft model and Simulation Platform

2.1.1.1 Cessna Citation X Aircraft Specifications

The Cessna Citation X is a long-range business jet that has been widely used as a platform for advanced control design (R. P. Andrianantara, Ghazi, et al., 2024 ; Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017 ; Ghazi & Botez, 2015b ; S. M. Hosseini, Ghazi, et al., 2024a). Since its introduction in 1996, the business jet aircraft has been recognized as one of the fastest business jets ever produced, with a maximum cruise speed of 350 knots and a maximum Mach number of 0.92. The aircraft is powered by two Rolls-Royce

AE3007C-1 turbofan engines, each providing a maximum thrust of 28 655 N. Its cabin accommodates up to 12 passengers and 3 crew members.

The Cessna Citation X has a maximum range of 5,724 km and a service ceiling of 15 544 m. Its main performance characteristics, summarized in Table 2.1, make the Cessna Citation X an excellent candidate for adaptive control studies. In fact, its wide and available operational envelope, high performance, and high-fidelity simulation models (Ghazi & Botez, 2015a) provide a realistic platform for evaluating adaptive flight control architectures.

Table 2.1 Cessna Citation X flight specifications

Characteristics	Value
Maximum cruise speed	648 km/h [350 knots]
Maximum Mach number	0.92
Maximum range	5,724 km [3,091 nm]
Maximum altitude	15 544 m [51,000 ft]
Maximum takeoff weight	16 193 kg [35,700 lb]
Maximum zero fuel weight	11 067 kg [24,400 lb]
Maximum thrust	28 655 N [6442 lbf]

2.1.1.2 Nonlinear Simulink Platform

A high fidelity nonlinear model of the Cessna Citation X was developed in MATLAB/Simulink (Ghazi & Botez, 2015a) and serves as a research platform for the controller design. This Simulink model integrates several modules such as aerodynamics, engine dynamics, actuator and sensor dynamics, atmospheric disturbance effects and other relevant aircraft systems.

The Simulink model was cross-validated using simulated flight data from a Research Aircraft Flight Simulator (RAFS) of the Cessna Citation X, designed and manufactured by CAE (Ghazi & Botez, 2015a) and shown in Figure 2.1. The RAFS has been validated by CAE engineers, based on Qualification Test Guide (QTG) procedures and real flight-test data from the Cessna Citation X. In addition, the RAFS satisfies the flight simulator validation criteria imposed by the Federal Aviation Administration and can be considered comparable to Level D fidelity

with respect to flight dynamics. The flight dynamics implemented within the RAFS are based on CAE and Cessna Aircraft (Textron) proprietary data and are therefore not publicly accessible. However, the RAFS provides access to high-fidelity simulated flight data and supports the execution of a wide range of flight scenarios, maneuvers, and full-mission simulations. The MATLAB/Simulink environment was developed to reproduce the RAFS flight dynamics with sufficient accuracy to support advanced control design, thereby ensuring that controller development and evaluation are conducted on high-fidelity flight dynamics.

In addition, the Cessna Citation X simulation platforms, both the RAFS and the derived Simulink models, have been widely used in advanced control research and peer-reviewed research literature. Examples of application include neural-network-based adaptive controllers with approximate dynamic inversion (R. P. Andrianantara et al., 2023a ; Quintin et al., 2024) and model predictive controllers (Eren et al., 2017). Similarly, fuzzy logic and recurrent neural network controllers have been demonstrated for longitudinal and lateral control (S. M. Hosseini, Bematol, et al., 2024b ; S. M. Hosseini, Ghazi, et al., 2024a), while control clearance and lateral flight qualities were investigated in (Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017).



Figure 2.1 Cessna Citation X Research Aircraft Flight Simulator (RAFS) manufactured by CAE

2.1.2 Nonlinear Dynamic Model

2.1.2.1 Cessna Citation X nonlinear dynamic model

The aircraft's longitudinal dynamics were considered in this study. This assumption allows us to write the following nonlinear representation of the aircraft's longitudinal dynamics:

$$\begin{aligned}\dot{\mathbf{x}} &= \boldsymbol{\varphi}(\mathbf{x}, \eta) \\ y &= h(\mathbf{x})\end{aligned}\tag{2.1}$$

where $\mathbf{x} = [u \quad w \quad q \quad \theta]^T$ is the state vector, u and w are the longitudinal and lateral velocity components, respectively, q is the pitch rate, and θ is the pitch angle, h is a nonlinear function, $\eta = \delta_e$ is the elevator input, and φ is an unknown nonlinear function.

2.1.2.2 Nonlinear Actuator Dynamics

The actuators are modelled in Simulink using a second-order nonlinear transfer function. This allows us to realistically reproduce the higher-order nonlinearity that strongly affect aircraft stability and control performance, particularly in adaptive flight control studies (Steffensen et al., 2023). The actuator parameters are summarized in Table 2.2.

Table 2.2 Actuator dynamics
parameter

Parameters	Values
Natural frequency	20 rad/s
Maximum deflection	14°
Minimum deflection	-19°
Damping ratio	0.8
Rate limit	20°/sec

2.2 Pitch Rate Controller Design

2.2.1 Controller Architecture

Figure 2.2 presents the controller architecture for pitch rate control. The control elements include a baseline PID controller, an RLS-based Dynamic Inversion controller and a NN adaptive controller. The stability analysis is detailed in the Appendix.

Controlling the pitch rate q for longitudinal control aims to track the desired short period performances (Stevens et al., 2015). Therefore, the proposed control algorithm is designed to track a reference signal q_{ref} .

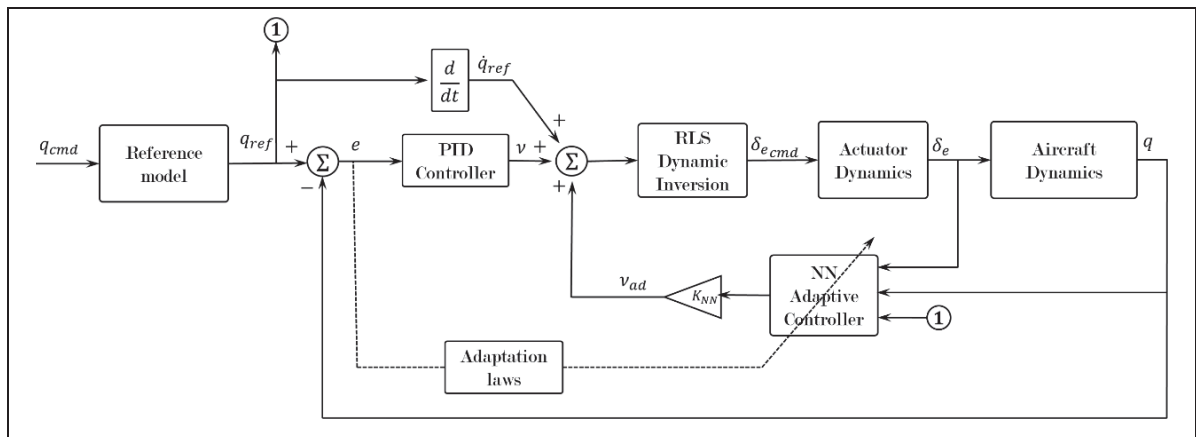


Figure 2.2 Pitch Rate Model Reference Adaptive Controller using PID, RLS Dynamic Inversion and adaptive NN

2.2.2 Controller Elements

2.2.2.1 Nonlinear Dynamic Inversion Controller

For linear systems, DI is straightforward to execute (Stevens et al., 2015). For nonlinear systems, the inversion is more difficult, as nonlinear systems may include unmodeled dynamics. According to (Stevens et al., 2015), most aircraft dynamics can be represented in affine form, with respect to the control input. Therefore, Eq. (2.1) can be rewritten as below:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\eta \\ y &= q = \mathbf{C}\mathbf{x}\end{aligned}\tag{2.2}$$

where $\mathbf{f}$ and $\mathbf{g}$ are nonlinear functions, and y is the output. Hence, $\mathbf{C} = [0 \quad 0 \quad 1 \quad 0]$ for pitch rate output.

Inverting Eq.(2.1), the control input may be written as:

$$\eta = \aleph(\dot{\mathbf{x}}, \mathbf{x})\tag{2.3}$$

where $\aleph = \boldsymbol{\varphi}^{-1}$ is the inverse function of $\boldsymbol{\varphi}$. Next, let's express $\eta = \delta_e$ according to the control algorithm in Figure 2.2. Considering Eqs.(2.1) and (2.2):

$$\dot{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}, \eta) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\eta\tag{2.4}$$

Developing Eq. (2.4) using the first-order linear approximation around $(\mathbf{x}_0, \eta_0)$, we obtain :

$$\boldsymbol{\varphi}(\mathbf{x}, \eta) = \boldsymbol{\varphi}(\mathbf{x}_0, \eta_0) + \boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}(\eta - \eta_0)\tag{2.5}$$

where $\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0)$ is the value at $(\mathbf{x}_0, \eta_0)$, $\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}$ and $\boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}$ are the partial derivatives of $\boldsymbol{\varphi}$ with respect to $\mathbf{x}$ and η , respectively. We may formulate the terms in Eq. (2.5), as such:

$$\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta_0\tag{2.6.a}$$

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\eta\tag{2.6.b}$$

$$\boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \eta} + \frac{\partial [\mathbf{g}(\mathbf{x})\eta]}{\partial \eta} = \mathbf{g}(\mathbf{x})\tag{2.6.c}$$

Considering the following changes, so that $\Delta \mathbf{x} = (\mathbf{x} - \mathbf{x}_0)$ and $\Delta \eta = (\eta - \eta_0)$, and replacing Eq. (2.6) into Eq.(2.5), the following expression is obtained:

$$\boldsymbol{\varphi}(\mathbf{x}, \eta) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta_0 + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\eta \right)_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \Delta \eta\tag{2.7}$$

Let us express $\dot{\mathbf{x}}$ in Eq. (2.4) for small perturbations around $(\mathbf{x}_0, \eta_0)$:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \Delta\dot{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}, \eta) \quad (2.8)$$

From Eq. (2.7) and Eq. (2.8), and considering $\dot{\mathbf{x}}_0 = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta$, $\Delta\dot{\mathbf{x}}$ can be written as such:

$$\Delta\dot{\mathbf{x}} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \eta \right)_{\mathbf{x}_0, \eta_0} \Delta\mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \Delta\eta \quad (2.9)$$

Making the following changes :

$$\mathbf{F}_{\mathbf{x}_0, \eta_0} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \eta \right)_{\mathbf{x}_0, \eta_0} ; \mathbf{G}_{\mathbf{x}_0, \eta_0} = \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \quad (2.10)$$

Eq. (2.9) can be expressed in a simplified form:

$$\Delta\dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta\mathbf{x} + \mathbf{G}_{\mathbf{x}_0, \eta_0} \Delta\eta \quad (2.11)$$

Recalling from Eq. (2.2), $\Delta\dot{\mathbf{y}}$ may be computed as follows:

$$\Delta\dot{\mathbf{y}} = \mathbf{C} \Delta\dot{\mathbf{x}} \quad (2.12)$$

Replacing $\Delta\dot{\mathbf{x}}$ from Eq. (2.11) into Eq. (2.12), we obtain:

$$\Delta\dot{\mathbf{y}} = \mathbf{C} \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta\mathbf{x} + \mathbf{C} \mathbf{G}_{\mathbf{x}_0, \eta_0} \Delta\eta \quad (2.13)$$

From Eq. (2.13), $\Delta\eta$ may be obtained by inversion as follows:

$$\Delta\eta = (\mathbf{C} \mathbf{G}_{\mathbf{x}_0, \eta_0})^{-1} (\Delta\dot{\mathbf{y}} - \mathbf{C} \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta\mathbf{x}) \quad (2.14)$$

The control task consists of tracking a reference signal r . Consequently, the error dynamics e may be determined as follows:

$$e = r - y \quad (2.15.a)$$

$$\Rightarrow \dot{e} = \dot{r} - \dot{y} \quad (2.15.b)$$

where y is the output variable.

Taking $\dot{y} \approx \Delta\dot{y}$ for small discrete time steps and recalling from Eq. (2.15.b), for which $\dot{y} = \dot{r} - \dot{e}$, then $\Delta\eta$ may be written as such:

$$\Delta\eta = (\mathbf{C}\mathbf{G}_{\mathbf{x}_0, \eta_0})^{-1}(\dot{r} + v - \mathbf{C}\mathbf{F}_{\mathbf{x}_0, \eta_0}\Delta\mathbf{x}) \quad (2.16)$$

where v is chosen so that the following relationship is fulfilled (Stevens et al., 2015):

$$\dot{e} = -v \quad (2.17)$$

Thus, the error can be stabilized by choosing a suitable form of v .

2.2.2.2 Online Parameter Estimation

Finding analytical solutions of $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ and $\mathbf{G}_{\mathbf{x}_0, \eta_0}$ given in Eq. (2.10) can be tasking. Using real-time estimations, such as the Recursive Least Square (RLS) method, is an alternative. The goal is to approximate $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ and $\mathbf{G}_{\mathbf{x}_0, \eta_0}$ based on local state observations. The RLS formulation can be found in (Haykin, 2013). This technique allows us to estimate the states and control variations based on their increments (Mahadi et al., 2022 ; Mohseni & Bernstein, 2022).

Transforming Eq. (2.13) into discrete times yields the following representation:

$$\Delta y_{k+1} = \mathbf{C}\tilde{\mathbf{F}}_k\Delta\mathbf{x}_k + \mathbf{C}\tilde{\mathbf{G}}_k\Delta\eta_k \quad (2.18)$$

where $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{G}}_k$ are discrete form of $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ and $\mathbf{G}_{\mathbf{x}_0, \eta_0}$.

Δy_{k+1} corresponds to the output variation and depends on the current states increment $\Delta\mathbf{x}_k$ and input increment $\Delta\eta_k$. $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{G}}_k$ may then be replaced by their estimates $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ so that:

$$\Delta\hat{y}_{k+1} = \mathbf{C}[\Delta\mathbf{x}_k \quad \Delta\eta_k] \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix} \quad (2.19)$$

Let us consider the vector of the estimated parameter $\hat{\boldsymbol{\Theta}}_k = \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix}$ and the vector of past increments $\mathbf{X}_k = [\Delta\mathbf{x}_k \quad \Delta\eta_k]^T$. Then, from Eq. (2.19), $\Delta\hat{y}_{k+1}$ can be compacted as follows:

$$\Delta \hat{\mathbf{y}}_{k+1} = \mathbf{C} \mathbf{X}_k^T \hat{\boldsymbol{\Theta}}_k \quad (2.20)$$

The following defines the error matrix $\boldsymbol{\epsilon}_k$ as:

$$\boldsymbol{\epsilon}_k = \Delta \mathbf{x}_k - \Delta \hat{\mathbf{x}}_k \quad (2.21)$$

From (Haykin, 2013), $\hat{\boldsymbol{\Theta}}_k$ can be numerically computed as such:

$$\hat{\boldsymbol{\Theta}}_k = \hat{\boldsymbol{\Theta}}_{k-1} + \frac{\mathbf{L}_{k-1} \mathbf{X}_k}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \boldsymbol{\epsilon}_k \quad (2.22)$$

where the covariance matrix $\mathbf{L}_k$ can be expressed as follows :

$$\mathbf{L}_k = \frac{1}{\kappa} \left(\mathbf{L}_{k-1} - \frac{\mathbf{L}_{k-1} \mathbf{X}_k \mathbf{X}_k^T \mathbf{L}_{k-1}}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \right) \quad (2.23)$$

$\hat{\boldsymbol{\Theta}}_k$ and $\mathbf{L}_k$ are obtained using the matrix inversion lemma (Haykin, 2013). $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ entirely depend on the state variations and control inputs. The initial parameters $\boldsymbol{\Theta}_0$, $\mathbf{L}_0$ and forgetting factor κ must be set beforehand.

Algorithm 2.1 presents the update of the covariance matrix $\mathbf{L}_k$. The covariance matrix is updated when the estimation error $\boldsymbol{\epsilon}_k$ reaches a threshold value $\boldsymbol{\epsilon}_{threshold}$.

Algorithm 2.1 RLS algorithm and covariance matrix
update

```

Initialize  $\boldsymbol{\Theta}_0$ ,  $\mathbf{L}_0$ 
Set  $\kappa$  and  $\boldsymbol{\epsilon}_{threshold}$ 
Compute  $\hat{\boldsymbol{\Theta}}_k$  using Eq.(2.22)
Compute  $\Delta \hat{\mathbf{y}}_{k+1}$  using Eq. (2.20)
Compute  $\boldsymbol{\epsilon}_k$  using Eq. (2.21)
Update  $\mathbf{L}_k$ 
if abs ( $\boldsymbol{\epsilon}_k$ )  $\leq \boldsymbol{\epsilon}_{threshold}$ 
then  $\mathbf{L}_k = 10^3 \mathbf{I}$ 
else compute  $\mathbf{L}_k$  using Eq.(2.23)

```

2.2.2.3 Proportional Integral Derivative (PID) Controller

From Eq. (2.14), the DI requires a linear feedback controller v (Stevens et al., 2015). Its value should be chosen so that stability is ensured according to Lyapunov criteria detailed in Appendix. For this reason, a basic Proportional-Integral-Derivative (PID) controller is sufficient and may be expressed in Laplace form as such (Stevens et al., 2015):

$$v = \left(K_p + \frac{K_I}{s} + K_D s \right) e \quad (2.24)$$

where K_p , K_I and K_D are the PID gains, respectively.

2.2.2.4 Neural Network Adaptive Controller

Using the estimations in Eq.(2.20) frequently results in prediction errors in real-world applications (Emami et al., 2022 ; Ge et al., 2013). Including prediction errors ε_1 into (2.14), the input $\Delta\eta_k$ can be written as:

$$\Delta\eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} [\Delta y_{k+1} - \mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x} - \varepsilon_1] \quad (2.25)$$

From Eqs. (2.16) and (2.17), $\Delta\eta_k$ in Eq. (2.25) can be written:

$$\Delta\eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} [\dot{r} + v - \mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x} - \varepsilon_1] \quad (2.26)$$

The variables $\mathbf{C}\hat{\mathbf{G}}_k$ and $\mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x}$ in Eq. (2.26) represent the aircraft's local dynamics. To rectify the uncertainties and estimation errors for $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$, the estimation error can be compensated with an adaptive term v_{ad} so that:

$$\varepsilon_1 = v_{ad} \quad (2.27)$$

An FFNN was used to train v_{ad} as seen in Figure 2.3.

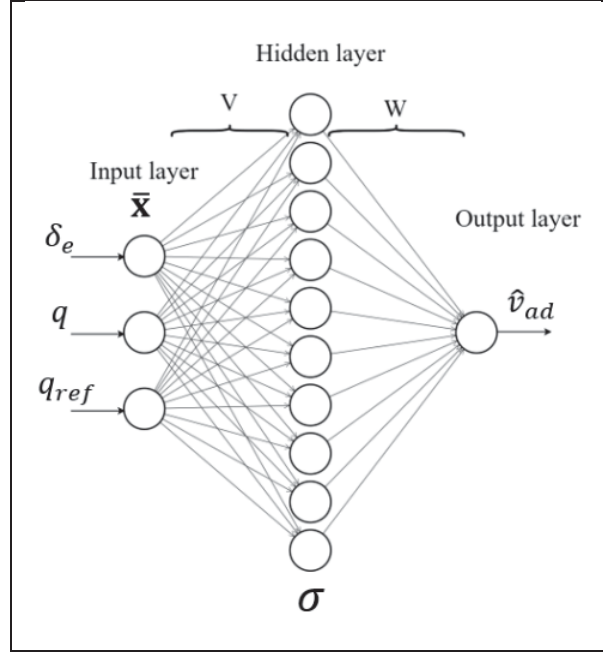


Figure 2.3 Representation of a Feedforward Neural Network

The effectiveness of the FFNN in Figure 2.3 is determined by its hyperparameters, including the number of neurons and hidden layers, the activation function σ , and the weights $\mathbf{W}$ and $\mathbf{V}$. The output from the NN may be written as such (Stevens et al., 2015):

$$\hat{v}_{ad} = \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}) \quad (2.28)$$

where $\sigma \in \mathbb{R}^{N \times N}$, with N – the number of neurons in the hidden layer, $\bar{\mathbf{x}} = [\delta_e \ q_{ref} \ q]^T$, with $\bar{\mathbf{x}} \in \mathbb{R}^{m \times 1}$ is the input vector, and $\mathbf{W} \in \mathbb{R}^{N \times n}$ and $\mathbf{V} \in \mathbb{R}^{m \times N}$, with n representing the number of network outputs. The activation function in the middle layer corresponds to a tangent sigmoid function as such :

$$\sigma_i(z) = \frac{1}{1 + e^{-z}} \quad (2.29)$$

where i denotes the neuron's rank. An affine function was used in the input and output layers. The adaptive signal $\hat{v}_{ad}$ is multiplied by a gain K_{NN} , which allows us to control the amplitude of the adaptive term $\hat{v}_{ad}$. Consequently :

$$v_{ad} = K_{NN}\hat{v}_{ad} = K_{NN}\mathbf{W}^T\boldsymbol{\sigma}(\mathbf{V}^T\bar{\mathbf{x}}) \quad (2.30)$$

2.2.2.5 Adaptation laws

The following adaptation laws taken from the backpropagation principle (Kim & Calise, 1997 ; Narendra & Annaswamy, 1987) are used to update the NN hyperparameters :

$$\begin{aligned} \dot{\mathbf{W}} &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}'\mathbf{V}^T\bar{\mathbf{x}})e^T + \lambda\|e\|\mathbf{W}]\boldsymbol{\Gamma}_W \\ \dot{\mathbf{V}} &= -\boldsymbol{\Gamma}_V[\bar{\mathbf{x}}e^T\mathbf{W}^T\boldsymbol{\sigma}' + \lambda\|e\|\mathbf{V}] \end{aligned} \quad (2.31)$$

where e is the tracking error, $\boldsymbol{\Gamma}_W$ and $\boldsymbol{\Gamma}_V$ are learning rate matrices, $\boldsymbol{\sigma}'$ is the derivative of $\boldsymbol{\sigma}$, and λ is a scalar that is equal to 1. In (Narendra & Annaswamy, 1987), the additional terms $-\lambda\|e\|\mathbf{W}\boldsymbol{\Gamma}_W$ and $-\lambda\|e\|\mathbf{V}\boldsymbol{\Gamma}_V$ The adaptation law shown in Eq. (2.31) ensures robustness and stability. A fixed learning rate $\boldsymbol{\Gamma}$ was considered for diagonal matrices $\boldsymbol{\Gamma}_W$ and $\boldsymbol{\Gamma}_V$ for the input and output weights, respectively. The backpropagation algorithm is considered as the steepest descent approximation when $\boldsymbol{\Gamma}$ is set to constant.

2.2.3 Control Law

The overall control law may be obtained by replacing Eqs. (2.24), (2.27) and (2.30) into Eq.(2.26), such that:

$$\Delta\eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} \left[\dot{r} + \left(K_P + \frac{K_I}{S} + K_D S \right) e - \mathbf{C}\hat{\mathbf{F}}_k\Delta\mathbf{x}_k - K_{NN}\mathbf{W}^T\boldsymbol{\sigma}(\mathbf{V}^T\bar{\mathbf{x}}) \right] \quad (2.32)$$

2.2.4 Reference model

The proposed algorithm consists of a Model reference Adaptive Controller, and controls the pitch rate q , which is a primary state variable for short period motion. The short period and can be approximated by a second-order transfer function:

$$\frac{q_{ref}}{q_{cmd}} = \frac{\omega_{sp}^2 (1 + T_{\theta_2} s)}{s^2 + 2\xi_{sp}\omega_{sp}s + \omega_{sp}^2} \quad (2.33)$$

where s is the Laplace variable. The q_{ref} signal consist of the desired second order pitch rate signal, while q_{cmd} is a step input signal.

The design requirements for the damping ratio ξ_{sp} and natural frequency ω_{sp} are given by the flight qualities in MIL-STD-1797A (Department of Defense, 1997). Equation (2.33) is used to model the pitch rate reference model as depicted in Figure 2.2. The zero of Eq. (2.33) is defined as $-1/T_{\theta_2}$, where T_{θ_2} is the short period time constant.

2.2.5 Flight Quality Requirements

According to the MIL-STD 1797A (Department of Defense, 1997), the Cessna Citation X can be categorized as a Class II aircraft. The proposed longitudinal controller was designed for Category B cruise operations. Table 2.3 establishes the Level 1 Flight Quality (FQ) requirements for short period dynamics.

Table 2.3 Level 1 Longitudinal Flight Quality requirements for category B cruise flights

Parameters	Range
Short period frequency, rad/s	$0.3 \leq \omega_{sp} \leq 2$
Control Anticipation Parameter	$0.085 \leq CAP \leq 3.6$
Settling time at $\pm 5\%$, sec	$T_s < 3$
Dropback	$-0.2 \leq \frac{D_{rb}}{q_{ss}} \leq 0.5$
Short-period time constant, sec	$0.1 \leq T_{\theta_2} \leq 1.5$
Delay allowed, sec	0.1

2.2.6 Performance metrics

The overall sum squared error denoted overall SSE was used for optimal tuning of the controller for a given number of flight conditions, so that :

$$\text{overall SSE} = \sum_{i=1}^{i=N} SSE_i \quad (2.34)$$

where N is the number of flight conditions.

The SSE for a single flight condition is computed as follows:

$$SSE = \sum_{t=0}^{t=T} (q_t - q_{ref_t})^2 \quad (2.35)$$

where q_t and q_{ref_t} correspond to the pitch rate and the pitch rate reference at a given time t , respectively, and T is the simulation time with a sample time $\Delta t = 0.02$ sec.

2.3 Controller Simulation for Ideal Conditions

2.3.1 Controller tuning parameters

The controller in Figure 2.2 was tuned through sensitivity-based tuning. First, the aircraft is trimmed for all cruise conditions over the operating envelope, for given altitude h , calibrated airspeed CAS, mass m and centre of gravity x_{cg} . Second, the PID gains K_P , K_I and K_D are tuned so that convergence is fulfilled. Third, the RLS inversion controller is tuned, for which the covariance matrix $\mathbf{L}_0$, the parameter vector $\mathbf{\Theta}_0$ and the forgetting factor κ are initialized. $\mathbf{L}_k$ is then iteratively updated following Algorithm 2.1. Then, the neural network learning rate Γ and gain K_{NN} are set. The NN weights are initialized from 0. Finally, robustness tests are performed. The control parameters are then adjusted throughout the process. The overall control parameter values are summarized in Table 2.4.

Table 2.4 Design Parameters for the Pitch Rate Controller

Control Element	Parameters	Value
PID	K_P, K_I, K_D	$-1, -1, -0.1$
RLS Inversion controller	$\kappa, \mathbf{L}_0, \hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$	$1.015, 10^7 \mathbf{I}_{5,5}, \mathbf{I}_{4,4},$ $-\mathbf{0.1}_{4,1}$
NN Controller	K_{NN} , learning rate Γ hidden layers, neurons	$20, 1$ $1, 30$
Reference Model	ξ_{sp}, ω_{sp} T_{θ_2}	$0.69, 4 \text{ rad/s}$ 0.35 sec

2.3.2 Simulation for cruise at 35,000 ft and 290 knots

Figure 2.4 shows the pitch rate control at 10 668 m [35,000 ft] and 537 km/h [290 knots]. The hyperparameter variations are shown in Figure 2.5. Figure 2.6 shows the RLS parameters, including the covariance matrix $\mathbf{L}_k$ in Eq. (2.23) and the prediction error ϵ_k in Eq. (2.21). Figure 2.7 shows the RLS state estimations.

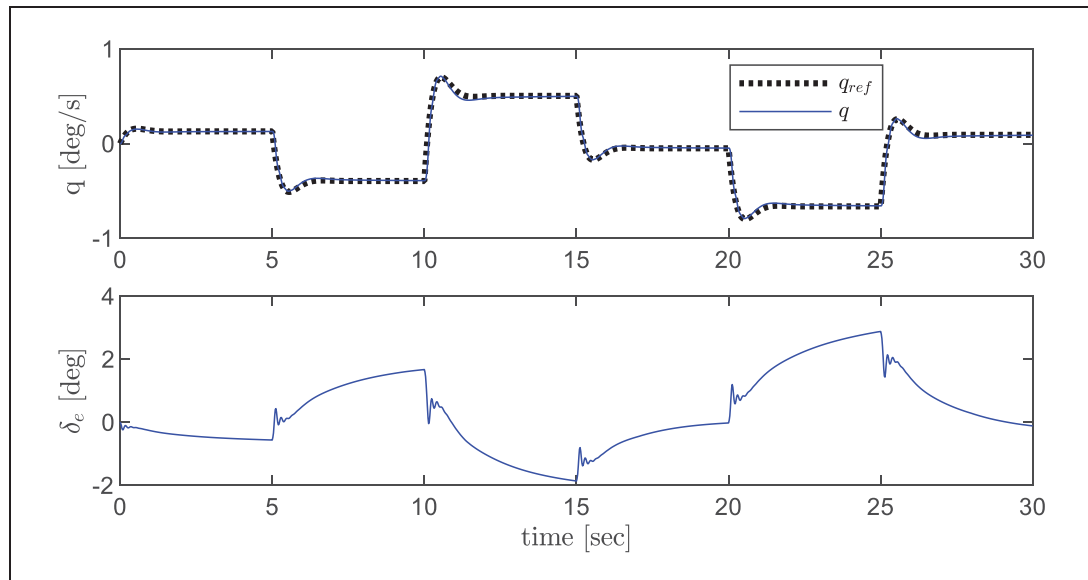


Figure 2.4 Pitch rate and elevator responses for cruises at 35,000 ft and 290 kt

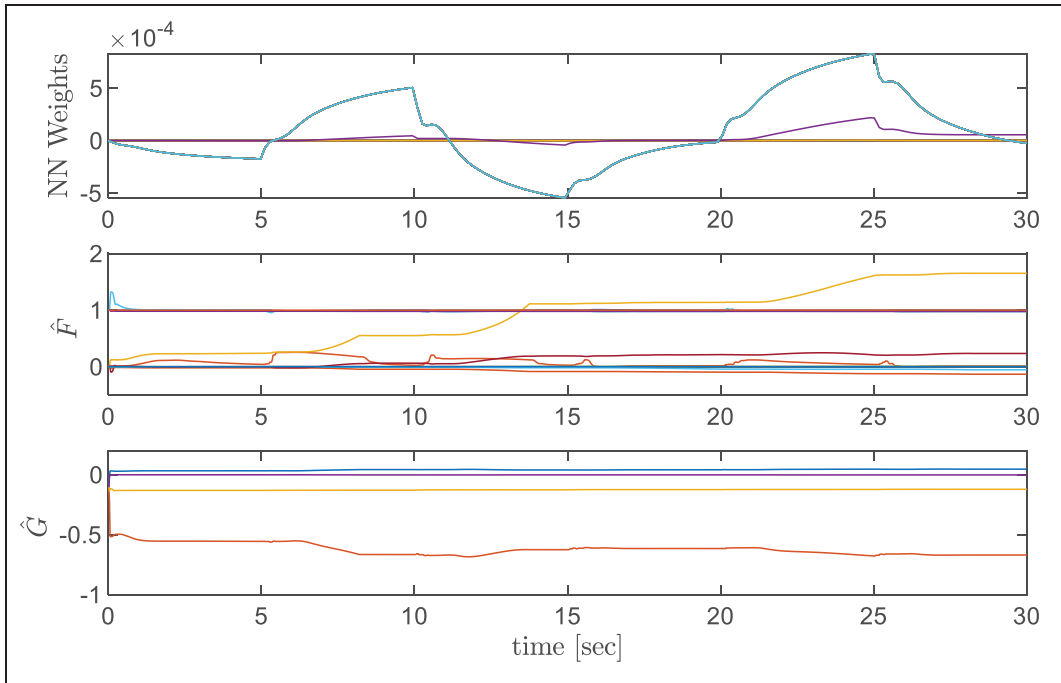


Figure 2.5 Hyperparameter variation during cruise flight

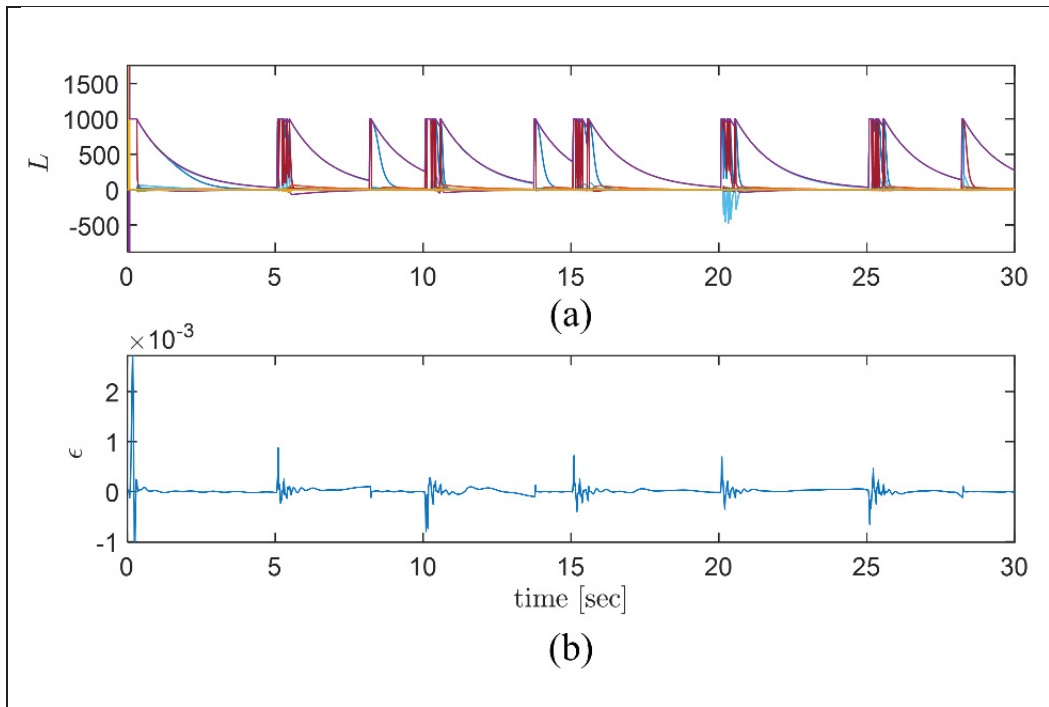


Figure 2.6 (a) Covariance matrix $\mathbf{L}_k$ update and (b) prediction error during cruise

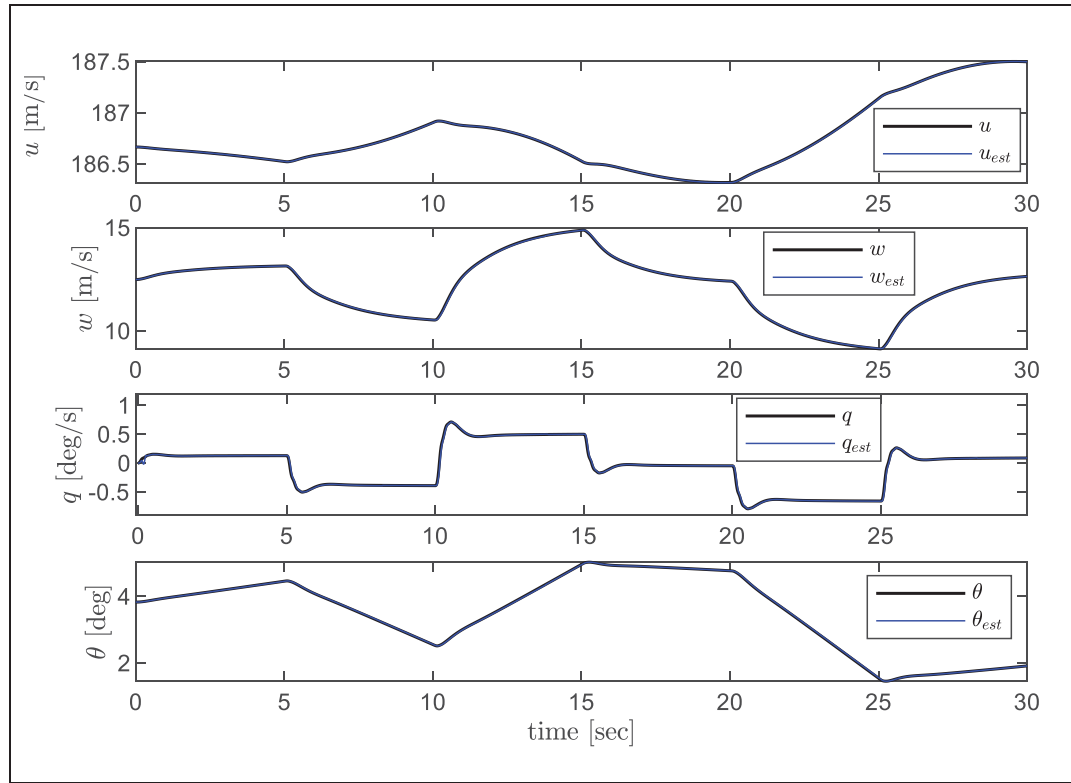


Figure 2.7 Longitudinal states estimation using the RLS algorithm for cruise flight at 10 668 m AGL and 537 km/h

2.3.3 Simulation and validation over the flight envelope

2.3.3.1 Flight operating conditions

64 different cruise conditions were assessed to tune the controller. The aircraft was trimmed at given calibrated airspeeds CAS and altitudes. The flight conditions and flight envelope are shown in Figure 2.8. A CoG of 25% and a weight of 15 000 kg [33,000 lb] were set for all simulations.

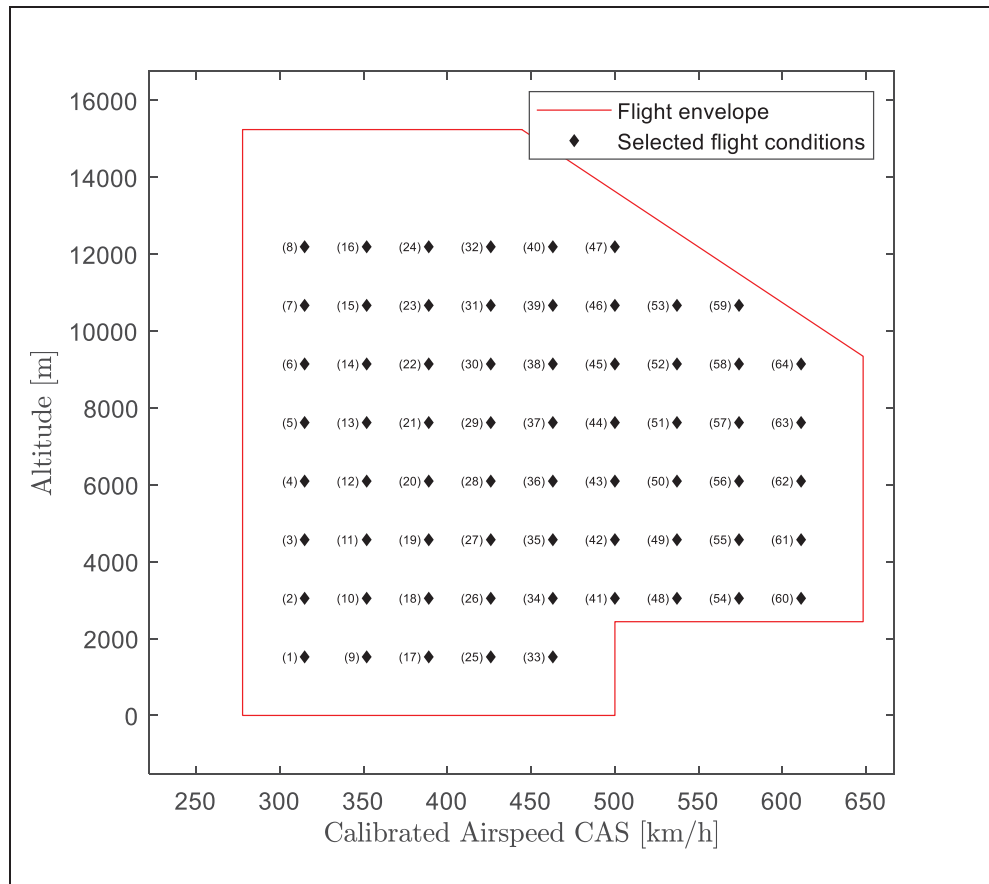


Figure 2.8 Selected cruise conditions for $m = 15\,000\text{ kg}$ and $\text{CoG } x_{cg} = 25\% \text{ MAC}$

2.3.3.2 NN and RLS parameter sensitivity analysis

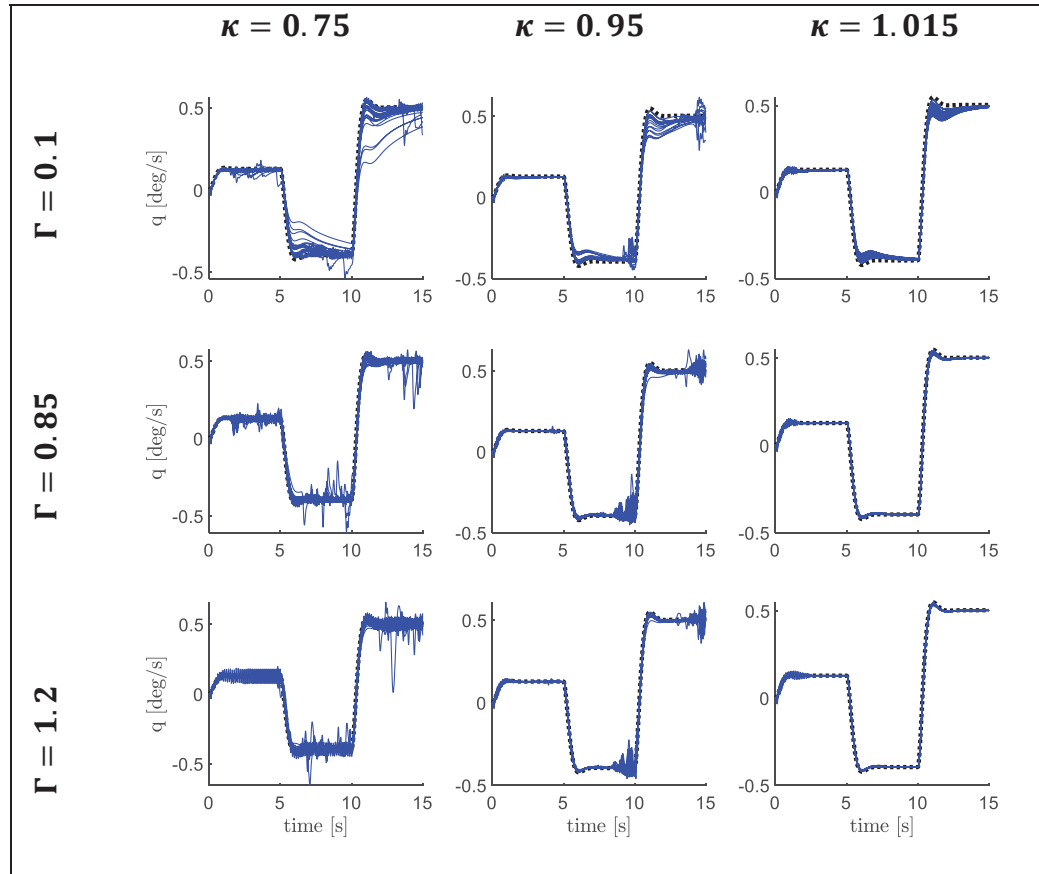


Figure 2.9 Pitch Rate Response for Different Learning Rates Γ and Forgetting Factors κ at all cruise conditions

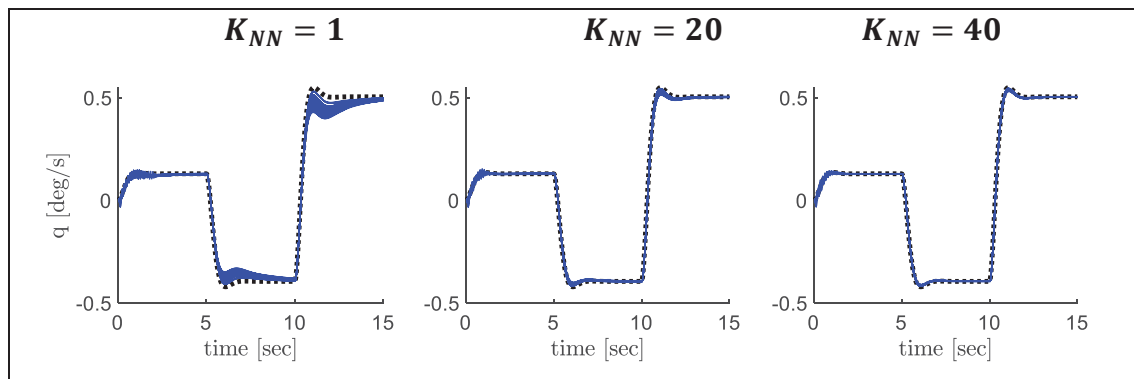


Figure 2.10 Pitch rate response for three different neural networks gains K_{NN} values for all cruise conditions

2.3.3.3 Controller tuning under wind gusts and turbulences

Figure 2.11 shows the overall SSE in Eq. (2.34) for $N = 10$ flight conditions. The simulations in Figure 2.11b include wind gusts with $u_{wind} = 40$ m/s [78 knots] and $w_{wind} = 5$ m/s [9.7 knots]. The cruise flights in Figure 2.11c were simulated using the Dryden turbulence model (Stevens et al., 2015) with moderate intensity. From Figure 2.11b and Figure 2.11c, an optimal compromise of Γ and K_{NN} were found and summed up in Table 2.4.

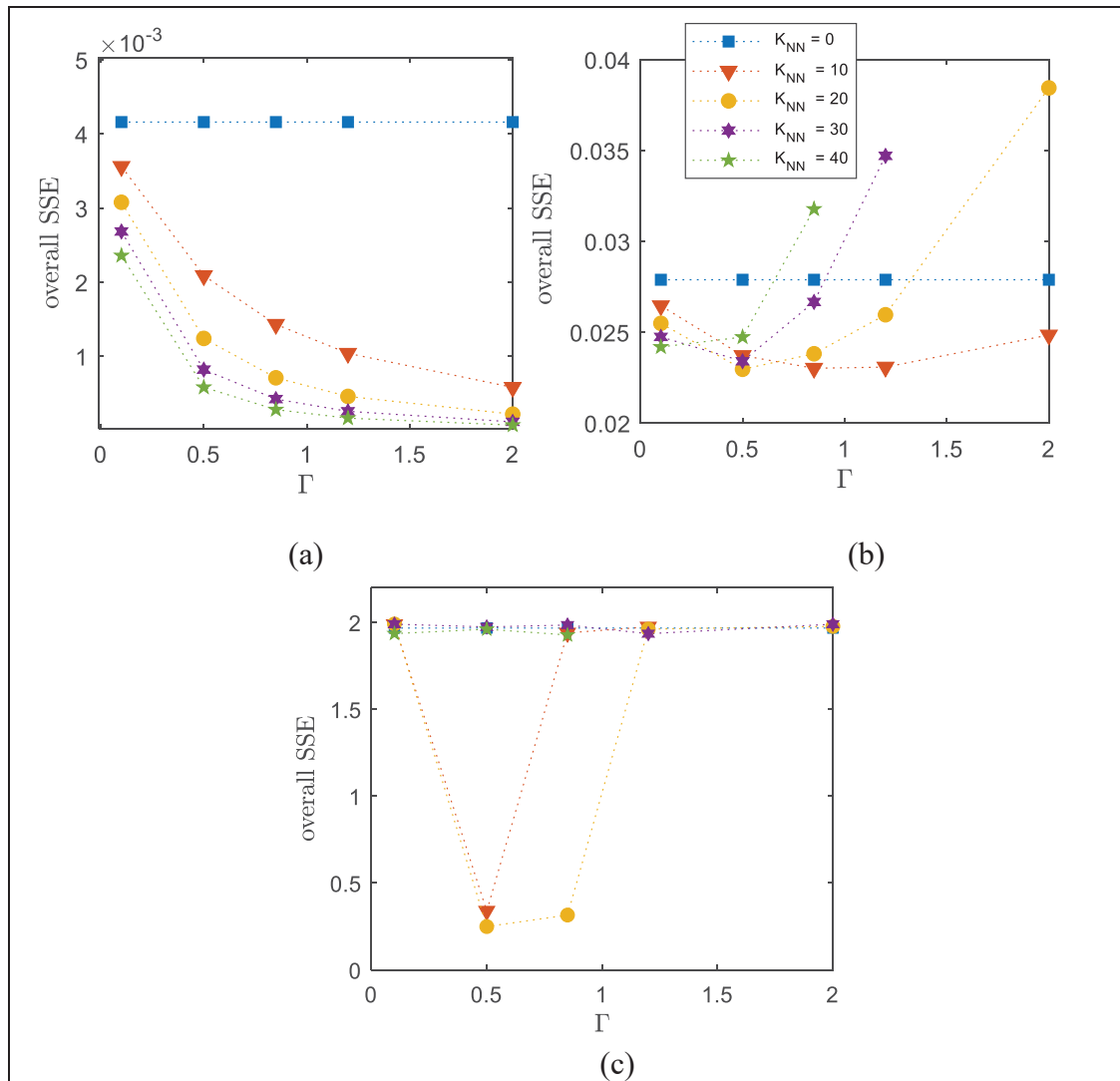


Figure 2.11 Sum Squared Error contour plot for 10 flight conditions for (a) ideal conditions, (b) wind gust and (c) moderate turbulence

2.3.4 Comparison of Controller Configurations

In Figure 2.12, the tracking error is smaller when the adaptive controllers are successively added. This shows the contribution of each control element and the controller's ability to adapt across the flight envelope.

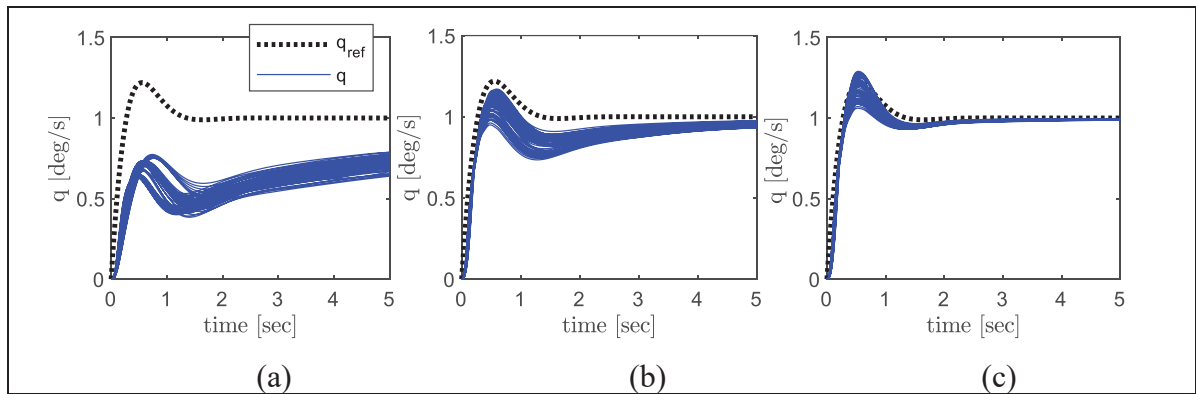


Figure 2.12 Pitch rate response for different configurations for all cruise conditions : (a) PID controller, (b) PID – RLS Dynamic Inversion controller and (c) PID – RLS Dynamic Inversion – NN controller

2.3.5 Flight Quality Validation

The pitch-rate controller was designed to satisfy the Flying Quality (FQ) requirements specified in MIL-STD-1797A for Class II, Category B aircraft. In this study, the reference model parameters (in Table 2.5) were selected to introduce a small overshoot of approximately 20% and steady state relative error below 1.2% in the pitch-rate response. This design choice was made to ensure compliance with the dropback requirement specified in Table 2.3 and defined in MIL-STD-1797A (Department of Defense, 1997), rather than representing a formal Level 1 requirement. The FQ validation was carried out by assessing the short-period damping ratio ξ_{sp} , natural frequency ω_{sp} , and Control Anticipation Parameter (CAP) across the flight envelope. As shown in Figure 2.13, all identified short-period parameters and CAP values remained within Level 1 boundaries, confirming that the aircraft meets the required flying qualities. The controller's transient performances for all flight conditions are summarized in Table 2.6.

Table 2.5 Pitch Rate Reference model dynamic characteristics for Level 1
flight quality

Frequency, rad/s	Damping	Time constant, sec	Parameter	Dropback
ω_{sp}	ξ_{sp}	T_{θ_2}	$\omega_{sp} T_{\theta_2}$	$\frac{D_{rb}}{q_{ss}}$
4	0.69	0.5	2	0.155

Table 2.6 Reference signal and pitch rate dynamic performance results

Signal	Damping	Frequency, rad/s	Rising time at 90%, sec	Settling time at $\pm 5\% q_{ss}$, sec	Overshoot, %
	ξ_{sp}	ω_{sp}	T_r	T_s	—
Reference model q_{ref}	0.69	4	0.21	1.22	21
Pitch rate response	0.59 – 0.77	4.31 – 5.27	0.25 – 0.29	1 – 1.85	6 – 28

The aircraft flight qualities were assessed in Figure 2.13a and Figure 2.13b. The dynamic parameters ξ_{sp} , ω_{sp} and T_{θ_2} in Eq. (2.33) have been identified for each flight condition using system identification. For each flight condition, optimization errors less than 2% were obtained, small enough to validate the identification. The Control Anticipation Parameter (CAP) in Figure 2.13b was also determined (Department of Defense, 1997) and computed as follows :

$$CAP \approx \frac{\omega_{sp}^2}{\frac{V_0}{g} \frac{1}{T_{\theta_2}}} \quad (2.36)$$

where V_0 is the airspeed, g is the gravitational constant, equal to 9.81 m/s² [32.18 ft/s²], ω_{sp} is the short period frequency and T_{θ_2} is the short period time constant.

The Level 1 Flight Quality boundaries given by MIL-STD-1797A (Department of Defense, 1997) are depicted in Figure 2.13a and 2.13b. The FQ boundaries show the FAA requirements

for optimal dynamic parameters ξ_{sp} , ω_{sp} , T_{θ_2} and CAP parameters. The pitch rate reference model dynamic parameters according to Table 6 is also represented. As seen on Figure 2.13a, the controller's dynamic parameters are close to the reference model and lying within the Level 1 region for all 64 flight conditions. This shows that the adaptive controller's dynamic performance is close to the targeted reference model and respects the FAA Level 1 criteria for the overall flight envelope. Furthermore, the CAP parameters for all flight conditions also lie within the Level 1 region as seen in Figure 2.13b. The CAP parameters are gathered far from the Level 1 boundary, showing dynamic stability margin.

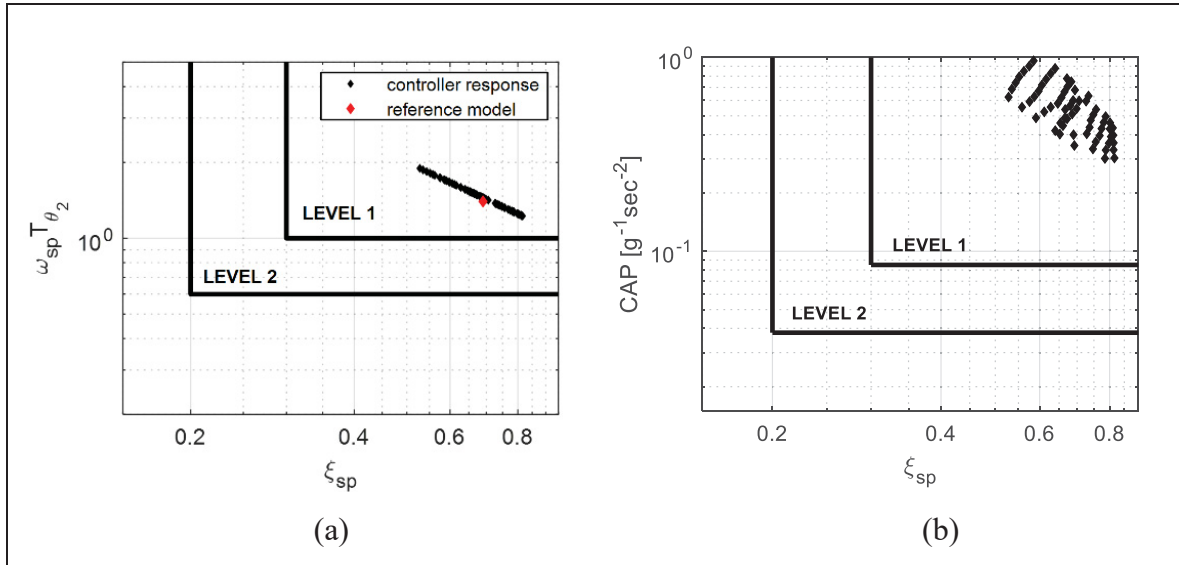


Figure 2.13 Short period flight qualities for (a) damping and frequency requirements and (b) CAP requirements

2.4 Robustness Validation

2.4.1 Wind Gust and Turbulence tests

The controller was tested against wind gusts in Figure 2.14 and turbulence in Figure 2.15 (Gao & Liu, 2017). The tailwind gust was simulated for $u_{wind} = 40\text{m/s}$ and $w_{wind} = 5\text{m/s}$ at $t = 5$ sec. The lateral wind component v_{wind} was set to 0 m/s. Figure 2.15 illustrates the controller's

behavior in the presence of turbulence. Dryden turbulence model was simulated for moderate turbulence in Simulink.

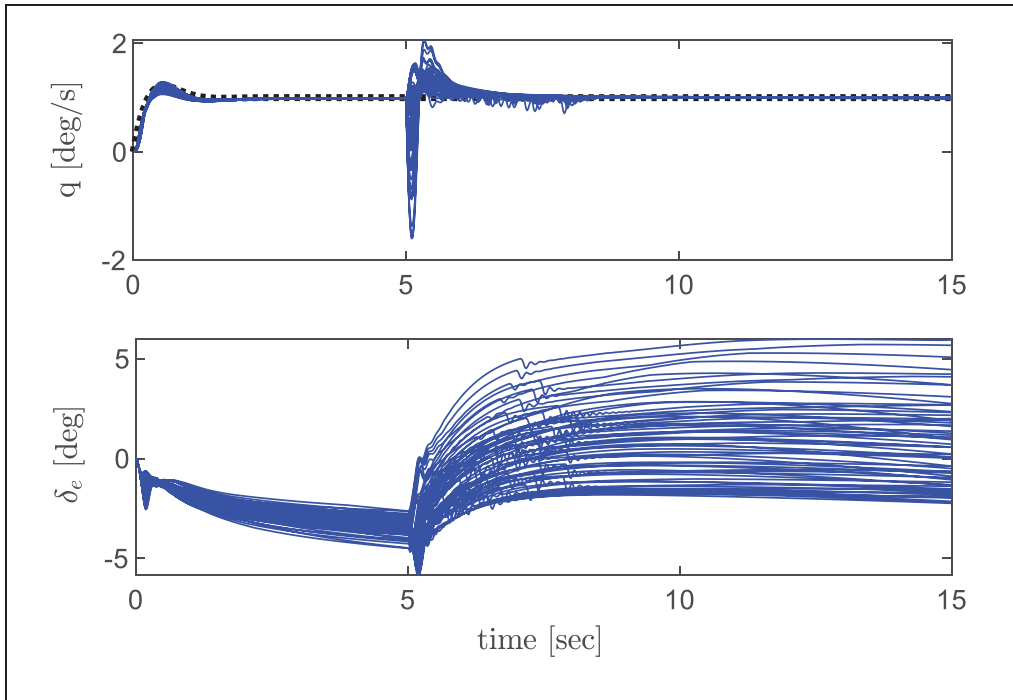


Figure 2.14 Pitch rate and elevator responses under tailwind gust

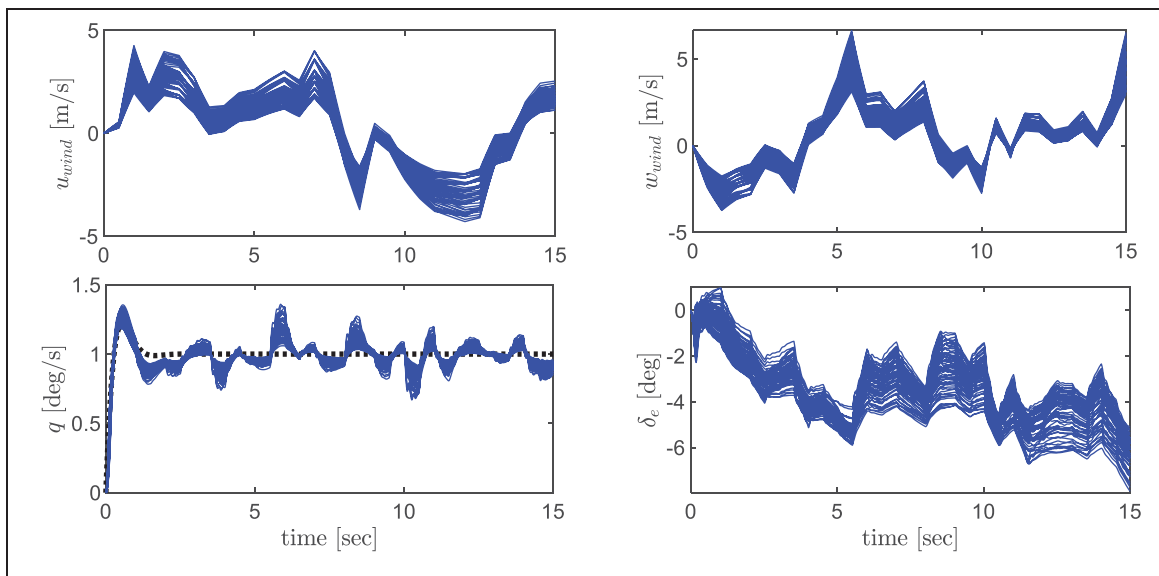


Figure 2.15 Turbulence profiles at different altitudes, pitch rate and elevator responses

2.4.2 Actuator Fault tests

Finally, the flight controller was validated for actuator faults (Hussein, Zhang, & Liu, 2024). 50% Loss-of-Effectiveness (LOE) on the elevators was simulated in Figure 2.16a, while white noises on the actuators are shown in Figure 2.16b. The actuator faults were added to the elevator's effectiveness $C_l^{\delta_e}$.

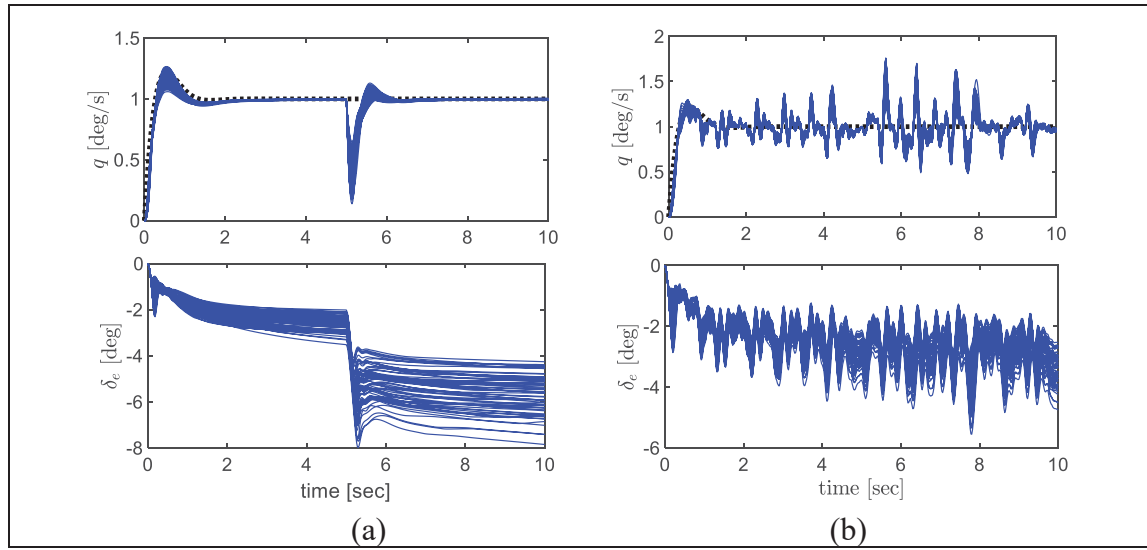


Figure 2.16 Pitch rate, elevator and adaptive element responses for (a) 50% LOE and (b) actuator noise

2.5 Results Discussion

2.5.1 Controller performance

The proposed controller illustrated in Figure 2.1 was first evaluated at a representative cruise condition of 10 668 m [35,000 ft] and CAS 537 km/h [290 knots]. As seen in Figure 2.4, the pitch rate closely follows the reference command, with elevator deflections remain close to their trim values. Figure 2.5 shows the neural network weights and the RLS estimated matrices converged according to the adaptation laws in eq. (2.31) confirming the correct implementation of the online adaptation.

Recursive Least Squares (RLS) updates were triggered when the prediction error exceeded a specified threshold, as shown in Figure 2.6, preventing divergence and numerical instabilities. The estimated longitudinal states closely matched the measured states in Figure 2.7, validating the RLS algorithm for online state identification.

A detailed parameter sensitivity study highlighted the complementary roles of the RLS-based dynamic inversion and the neural network. The RLS estimator was found to be highly sensitive to the forgetting factor κ : small values led to poor estimation due to rapid forgetting, while high values led to slower adaptation but more precise without oscillations. To mitigate these effects, the covariance matrix $\mathbf{L}_k$ was reinitialized when the prediction error exceeded a threshold, ensuring better estimation. As shown in Figure 2.6a, reinitializing $\mathbf{L}_k$ at $10^3 \mathbf{I}_{5,5}$ enabled fast convergence and precise estimation.

The neural network compensates for the RLS estimation errors. As demonstrated in Figure 2.9 to Figure 2.11, increasing the gain K_{NN} and the learning rate Γ reduced the overall SSE and improved adaptation under nominal conditions. However, over tuning these parameters may cause divergence in the presence of disturbances, as shown in Figure 2.11b and Figure 2.11c. For instance, $K_{NN} = 10$ improved disturbance rejection but produced high SSE under ideal conditions, as seen in Figure 2.11a. This highlights the necessity of careful parameter tuning to achieve a balanced trade-off. Additional tests (Figure 2.10) showed that higher NN gains improved short-term adaptation but also increased the risk of overtraining, making the controller overly sensitive to small variations.

The controller was then validated across 64 cruise conditions, for altitudes varying from 1 500 m [5,000 ft] to 12 200 [40,000 ft] and CAS from 314 km/h [170 knots] to 611 km/h [330 knots]. Sensitivity studies in Figure 2.9 to Figure 2.11 showed that higher values of the forgetting factor κ , NN learning rate Γ , and NN gain K_{NN} improved adaptation but increased steady-state error under disturbances. Optimal parameters in Table 2.4 were selected in order to ensure compromise between tracking accuracy, robustness, stability and Level 1 flight qualities.

In Figure 2.12, three configurations were compared: a baseline PID, a PID–RLS, and a PID–RLS –NN configuration. The full hybrid adaptive configuration (PID–RLS–NN) achieved the lowest tracking error and the strongest disturbance rejection, confirming the advantages of combining classical PID controllers with learning-based adaptation.

Flight quality assessment was conducted using the reference model in Table 2.5, corresponding to Level 1 FQ. The reference model tracking produced smooth responses, with relative steady-state errors remaining below 0.1% within 1.9 seconds. As shown in Figure 2.13, Level 1 flight qualities were consistently achieved for all flight conditions, as the short-period parameters and the Control Anticipation Parameter (CAP) values remained within the required boundaries. Additional tests were performed in which the reference model was modified or removed, and Level 1 flight qualities were still maintained, demonstrating the controller’s adaptability to different reference models. It is important to note that this study focuses on Flying Qualities (FQ) validation using quantitative metrics defined in MIL-STD-1797A. A complete Handling Qualities (HQ) assessment requires pilot-in-the-loop evaluations with objective pilot rating criteria such as Gibson, Neal–Smith, or Bandwidth criteria. These Handling Quality metrics are not included in the present work but may be addressed in future studies once pilot-in-the-loop simulation campaigns are conducted using the flight simulator shown in Figure 2.1.

Finally, the robustness of the proposed controller was demonstrated under multiple uncertainties: wind gusts and Dryden turbulence of moderate intensity in Figure 2.14, and actuator loss-of-effectiveness and actuator noise in Figure 2.16. In all cases, the controller was able to maintain stability and to adapt to the pitch rate within 1.9 sec settling time, while maintaining Level 1 flight qualities. The elevator deflections remained effective and unsaturated, even in the presence of uncertainties.

2.5.2 Comparative Study

Across recent adaptive and intelligent flight control studies, different approaches have demonstrated notable improvements in tracking performance. Harris et al. (Harris et al., 2022) showed that L1 adaptive augmentation reduced maneuver tracking errors by 30–80%. This is

also supported by the controller configuration comparative study, for which the PID – RLS – NN combination shows improved performance.

Steffensen et al. (Steffensen et al., 2023) showed that their actuator-inclusive NDI achieved exact reference-model tracking with a 0.3 s settling time and 0% overshoot, while classical INDI exhibited residual error unless actuator bandwidths were effectively infinite. While their actuator model was limited to first-order dynamics, this paper focused on second order nonlinear actuator dynamics.

(R. P. Andrianantara, Ghazi, et al., 2024) demonstrated that the Model Predictive Controller – Neural Network controller for the Cessna Citation X allowed to reduce the steady-state error across its flight envelope, maintaining elevator deflections within ± 2 deg. In addition, this paper highlights the robustness of the adaptive controller with a single parameter set in Table 2.4 and analyzes the control parameters sensitivity.

In the spacecraft control domain, Lungu et al. (R. Lungu et al., 2021) showed that their adaptive NN–Dynamic Inversion controller achieved convergence in less than 3 sec, with negligible steady-state error, and very low overshoot, outperforming conventional PID designs. In fighter aircraft applications, Pedro and Meyer (Pedro & Meyer, 2007) showed that their NN–DI controller predicted control deflections with high precision less than 0.05 deg for aileron, 1.5 deg for elevator, and 0.5 deg for rudders, allowing safe landings with deviations limited to 1.5 m laterally and 4 m vertically, and thus in the presence of storm. Similarly, Lungu and Lungu (M. Lungu & Lungu, 2017) demonstrated that their NN–DI auto-landing controllers achieved altitude errors less than 0.5 m, velocity errors less than 0.3 m/s, and reduced overshoot in comparison with conventional landing control systems. Also, at the micro air vehicle scale, Zhou et al. (Zhou et al., 2021) introduced Extended INDI, which enabled oscillation-free optical-flow landings, reducing flow-divergence tracking RMSE to 0.10 s^{-1} versus 0.58 s^{-1} under conventional feedback, while adapting robustly across different initial heights and noisy environmental conditions. Nguyen et al. (N. Nguyen, 2011) demonstrated their hybrid NN – DI controller had a tracking error by over an order of magnitude compared to baseline control,

achieving errors between 0.05 – 0.1 deg/s for pitch rate control and overshoot below 10%. Settling times were approximately 2–3 s under damaged aircraft scenarios such as tail loss or engine failure. While Nguyen’s controller was validated under structural damage scenarios, it did not provide certification-relevant flight quality metrics.

By contrast, the hybrid PID-RLS–NN controller developed in this paper demonstrated its effectiveness in full-envelope longitudinal control of fixed-wing commercial aircraft such as the Cessna Citation X. It achieved steady-state pitch-rate errors below 0.1%, settling times under 1.9 secs, and overshoot between 6 and 28% across the flight envelope, all while maintaining Level 1 flight qualities. Unlike previous studies, robustness was explicitly validated under 40 m/s gusts, moderate Dryden turbulence, actuator loss-of-effectiveness, and actuator noise, thereby confirming robustness performance under both environmental disturbances and system faults. While Harris et al. (Harris et al., 2022) and Lungu (R. Lungu et al., 2021) controllers improved tracking error for specific cases, the proposed hybrid PID – RLS – NN controller achieved less than 0.1% steady-state errors and Level 1 FQ dynamics across the full operating envelope, with explicit robustness tests extending beyond ideal conditions or isolated scenarios.

2.6 Conclusion

An adaptive pitch-rate controller for the Cessna Citation X was developed and validated using a hybrid RLS Dynamic Inversion and a neural network adaptive controller. The design covered 64 cruise conditions, removing the need for gain scheduling, and achieved tracking with relative steady-state errors less than 0.1% in 1.9 sec.

Compared to classical PID-based methods, the proposed controller relies on a fixed-gain PID for baseline stability, while adaptive features ensure robustness against uncertainties and low tracking errors. Sensitivity analysis highlighted the critical influence of RLS initialization and forgetting factor, as well as the importance of finding compromise between the neural network gain and learning rate. A covariance reinitialization strategy was introduced to prevent

numerical instabilities in the RLS, while the neural network compensated for estimation errors, enabling dual robust adaptation even under disturbances.

Flight quality evaluation confirmed Level 1 compliance across all tested scenarios, including turbulence, actuator faults, and actuator noise. Lyapunov stability analysis further verified the convergence of all adaptive elements.

Overall, the results demonstrate that the proposed controller design provides a certifiable method to validate adaptive flight control in commercial aircraft by combining classical PID controllers with adaptive robust controllers, and validated flight qualities. Future work can be foreseen by extending the proposed hybrid adaptive controller to more general non-affine models to further enhance robustness and applicability to more complex nonlinearities.

2.7 Appendix

The controller boundedness and convergence were assessed using Lyapunov criteria : for a given positive definite candidate $\mathcal{L}$, its derivative $\dot{\mathcal{L}}$ should be negative definite, such that (Lavretsky & Wise, 2024):

$$\mathcal{L} \geq 0 \quad \text{and} \quad \dot{\mathcal{L}} \leq 0 \quad (2.37)$$

2.7.1 Error dynamics

Let us consider the designed control law in Eq.(2.32). Replacing $\dot{y}$ in Eq. (2.15.b) by its expression in Eq.(2.13), and applying the final control in Eq.(2.32), it is easy to demonstrate that the error dynamic $\dot{e}$ may be defined as follows:

$$\dot{e} = - \left(K_P + \frac{K_I}{s} + K_D s \right) e + K_{NN} \mathbf{W}^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \quad (2.38)$$

Let us consider the optimal weights $\mathbf{V}$ as the sum of the current weights $\hat{\mathbf{V}}$ and optimization error $\tilde{\mathbf{V}}$. The same logic applies to the output weight matrix $\mathbf{W}$, so that:

$$\mathbf{V} = \hat{\mathbf{V}} + \tilde{\mathbf{V}} \quad \text{and} \quad \mathbf{W} = \hat{\mathbf{W}} + \tilde{\mathbf{W}} \quad (2.39)$$

where $\hat{\mathbf{W}}$ and $\hat{\mathbf{V}}$ are known values, and $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{V}}$ are unknown errors. The optimal weights $\mathbf{W}$ and $\mathbf{V}$ are therefore unknown. We can express the Taylor series approximation of the tangent sigmoid function $\sigma(\mathbf{V}^T \bar{\mathbf{x}})$ around the current argument $\hat{\mathbf{V}}^T \bar{\mathbf{x}}$ as follows:

$$\sigma(\mathbf{V}^T \bar{\mathbf{x}}) = \sigma(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) + \sigma'(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) \tilde{\mathbf{V}}^T \bar{\mathbf{x}} + \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}) \quad (2.40)$$

where $\sigma'(\hat{\mathbf{V}}^T \bar{\mathbf{x}})$ is the gradient of $\sigma(\mathbf{V}^T \bar{\mathbf{x}})$, and $\mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}})$ is the unknown higher-order term. For further calculations, let us consider the following notations:

$$\sigma(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) = \hat{\sigma} \quad \text{and} \quad \sigma'(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) = \hat{\sigma}' \quad (2.41)$$

where $\hat{\sigma}$ and $\hat{\sigma}'$ are known values.

By replacing Eqs. (2.39) and (2.40) into Eq. (2.38), it is easy to obtain:

$$\begin{aligned} \dot{e} = & -\left(K_P + \frac{K_I}{S} + K_D S\right) e + K_{NN} \hat{\mathbf{W}}^T \hat{\sigma} + K_{NN} \tilde{\mathbf{W}}^T \hat{\sigma} + K_{NN} \mathbf{W}^T \hat{\sigma}' \tilde{\mathbf{V}}^T \bar{\mathbf{x}} \\ & + K_{NN} \mathbf{W}^T \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}) \end{aligned} \quad (2.42)$$

Linear components with respect to $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{V}}$ can be easily approximated with suitable tuning of Γ (Lewis et al., 1995 ; Narendra & Annaswamy, 1987), while the terms containing two or more unknown parameters and higher-order terms constitute disturbances, so that Eq. (2.42) can be written as:

$$\dot{e} = -\left(K_P + \frac{K_I}{S} + K_D S\right) e + K_{NN} \hat{\mathbf{W}}^T \hat{\sigma} + K_{NN} \tilde{\mathbf{W}}^T \hat{\sigma} + d \quad (2.43)$$

where d is the disturbance signal defined as

$$d = K_{NN} \mathbf{W}^T \hat{\sigma}' \tilde{\mathbf{V}}^T \bar{\mathbf{x}} + K_{NN} \mathbf{W}^T \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}) \quad (2.44)$$

where $\mathbf{W}$, $\tilde{\mathbf{V}}$ and $\mathcal{O}^2$ are unknown.

2.7.2 Disturbance Boundedness

Through assumptions, Lewis and al. in (Lewis et al., 1995) assessed the boundedness of d and demonstrated that d may be bounded as follows :

$$\|d\| \leq C_0 + C_1 \|\tilde{\mathbf{V}}\| + C_2 \|\tilde{\mathbf{V}}\| \|e\| \quad (2.45)$$

where C_0 , C_1 and C_2 are computable positive constants.

2.7.3 Lyapunov Candidate and its Derivative

Let us assess the stability of the overall system. To achieve this, let us choose a positive definite Lyapunov candidate function $\mathcal{L}$ as follows :

$$\mathcal{L} = \frac{1}{2} \mathbf{e} \mathbf{e} + \frac{1}{2} \tilde{\mathbf{\Theta}} \mathbf{L}^{-1} \tilde{\mathbf{\Theta}} + \frac{1}{2} \tilde{\mathbf{W}} \mathbf{\Gamma}^{-1} \tilde{\mathbf{W}} + \frac{1}{2} \tilde{\mathbf{V}} \mathbf{\Gamma}^{-1} \tilde{\mathbf{V}} \quad (2.46)$$

where $\tilde{\mathbf{\Theta}}$ is the difference between the optimal RLS parameter vector $\mathbf{\Theta}^*$ and its estimate $\hat{\mathbf{\Theta}}$, such that:

$$\tilde{\mathbf{\Theta}} = \mathbf{\Theta}^* - \hat{\mathbf{\Theta}} \quad (2.47)$$

The Lyapunov candidate function in Eq.(2.46) includes all the controller adaptive parameters. Its derivative may be written as follows :

$$\dot{\mathcal{L}} = \dot{\mathbf{e}} \mathbf{e} + \dot{\tilde{\mathbf{\Theta}}} \mathbf{L}^{-1} \tilde{\mathbf{\Theta}} + \dot{\tilde{\mathbf{W}}} \mathbf{\Gamma}^{-1} \tilde{\mathbf{W}} + \dot{\tilde{\mathbf{V}}} \mathbf{\Gamma}^{-1} \tilde{\mathbf{V}} \quad (2.48)$$

2.7.4 RLS Controller Stability

Here we assess the second term of Eq.(2.46) and demonstrate its negative definiteness.

$$\mathcal{L}_2 = \frac{1}{2} \tilde{\mathbf{\Theta}} \mathbf{L}^{-1} \tilde{\mathbf{\Theta}} \quad (2.49)$$

Its derivative may be computed as follows:

$$\dot{\mathcal{L}}_2 = \tilde{\Theta} \mathbf{L}^{-1} \dot{\tilde{\Theta}} \quad (2.50)$$

where $\mathbf{L}^{-1}$ is the inverse covariance matrix.

Let us express $\tilde{\Theta}$ in discrete time at $k - 1$ and k , so that:

$$\begin{aligned} \tilde{\Theta}_{k-1} &= \Theta^* - \hat{\Theta}_{k-1} \\ \tilde{\Theta}_k &= \Theta^* - \hat{\Theta}_k \end{aligned} \quad (2.51)$$

From Eq.(2.22), $\hat{\Theta}_k$ can be written as:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + \mathbf{K}_k \epsilon_k \quad (2.52)$$

where $\mathbf{K}_k$ is the Kalman gain vector defined as:

$$\mathbf{K}_k = \frac{\mathbf{L}_{k-1} \mathbf{X}_k}{\mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k + 1} \quad (2.53)$$

Taking into account Eq.(2.52), $\dot{\tilde{\Theta}}$ can be expressed as:

$$\dot{\tilde{\Theta}} = \tilde{\Theta}_k - \tilde{\Theta}_{k-1} = -\mathbf{K}_k \epsilon_k \quad (2.54)$$

Therefore, replacing Eq. (2.54) into (2.50), the derivative $\dot{\mathcal{L}}_2$ can be expressed as:

$$\dot{\mathcal{L}}_2 = -\tilde{\Theta} \mathbf{L}^{-1} \mathbf{K}_k \epsilon_k \quad (2.55)$$

Let us assess the definiteness of eq.(2.55). The elements of $\tilde{\Theta}$ can be positive or negative, depending on whether $\hat{\Theta}_k$ are underestimating or overestimating Θ^* . The magnitude of $\tilde{\Theta}$ is expected to decrease over time, tending to zero as $\hat{\Theta}_k$ converges to Θ^* . Consequently, the magnitude of the estimation error ϵ_k decreases. The sign of ϵ_k may still vary; however, these variations become smaller and less frequent, since the estimates $\hat{\Theta}_k$ are stabilizing.

The sign of the Kalman gain vector $\mathbf{K}_k$ in the context of the Recursive Least Squares algorithm is generally positive. However, it is more accurate to say that $\mathbf{K}_k$ is a vector that scales the correction applied to the parameter estimates rather than focusing solely on its sign, as shown in Eq.(2.52).

The inverse covariance matrix $\mathbf{L}^{-1}$ is positive definite; therefore, the definiteness of Eq. (2.55) relies on $\mathbf{K}_k$, $\mathbf{e}_k$ and $\tilde{\mathbf{\Theta}}$. The Kalman gain vector $\mathbf{K}_k$ is chosen to minimize the estimation error $\mathbf{e}_k$ and the optimization error $\tilde{\mathbf{\Theta}}$. Therefore, in order to guarantee that $\mathcal{L}_2$ decreases over time, we typically aim for $\mathbf{K}_k$ to be chosen such that the product $\mathbf{K}_k \mathbf{e}_k$ aligns with the direction of $\tilde{\mathbf{\Theta}}$, leading to a positive value. This positive dot product indicates that $\dot{\mathcal{L}}_2$ is negative definite, and that $\mathcal{L}_2$ in Eq. (2.49) is monotonically decreasing. In that case, the stability and convergence of the RLS algorithm are verified.

2.7.5 Boundedness and Convergence

Let us consider the derivative of the Lyapunov candidate in Eq.(2.48), and demonstrate its boundedness.

$$\dot{\mathcal{L}} = \dot{\mathbf{e}}\mathbf{e} + \dot{\tilde{\mathbf{\Theta}}}\mathbf{L}^{-1}\tilde{\mathbf{\Theta}} + \dot{\tilde{\mathbf{W}}}\mathbf{\Gamma}^{-1}\tilde{\mathbf{W}} + \dot{\tilde{\mathbf{V}}}\mathbf{\Gamma}^{-1}\tilde{\mathbf{V}} \quad (2.56)$$

where $\mathbf{V}$ and $\mathbf{W}$ are the neural network input and output weight matrices, respectively, $\dot{\mathbf{V}}$ and $\dot{\mathbf{W}}$ are their respective time derivatives and, $\mathbf{\Gamma}^{-1}$ is the inverse learning rate diagonal matrix.

Replacing $\dot{\mathbf{e}}$, $\dot{\mathbf{W}}$ and $\dot{\mathbf{V}}$ from their expressions in eqs. (2.42) and (2.31) into Eq. (2.56), we obtain:

Considering the assumptions in Eq. (2.39) and making all simplifications, Eq. (2.56) can be simplified as:

$$\begin{aligned} \dot{\mathcal{L}} = & -e \left(K_p + \frac{K_I}{s} + K_D s \right) e + \hat{\mathbf{W}}^T \hat{\boldsymbol{\sigma}} e + \tilde{\mathbf{W}}^T \hat{\boldsymbol{\sigma}}' \mathbf{V}^T \bar{\mathbf{x}} e + \mathbf{W}^T \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}) e + \dot{\tilde{\mathbf{\Theta}}}\mathbf{L}^{-1}\tilde{\mathbf{\Theta}} \\ & - \tilde{\mathbf{W}}\lambda\|e\|\hat{\mathbf{W}} - \tilde{\mathbf{V}}\lambda\|e\|\hat{\mathbf{V}} \end{aligned} \quad (2.57)$$

where the gradient estimate $\hat{\boldsymbol{\sigma}}$ is assumed to be known.

Next, we assess the boundedness of each term of Eq. (2.57). Following the reasoning in (Lewis et al., 1995) and taking into account the assumption in Eq. (2.39), the term $\hat{\mathbf{W}}^T \hat{\boldsymbol{\sigma}} e$ may be bounded as:

$$\|\hat{\mathbf{W}}^T \hat{\boldsymbol{\sigma}} e\| \leq W_m c_{11} \|e\| - \|\tilde{\mathbf{W}}\| c_{11} \|e\| \quad (2.58)$$

Following the reasoning in (Lewis et al., 1995), it can be demonstrated that $\mathbf{W}^T \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}})e$ may be bounded as follows:

$$\|\mathbf{W}^T \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}})e\| \leq W_m c_3 \|e\| + W_m c_{11} c_1 Q_d \|\tilde{\mathbf{V}}\| \|e\| + W_m c_{11} c_2 \|\tilde{\mathbf{V}}\| \|e\|^2 \quad (2.59)$$

Next, the term $\tilde{\mathbf{W}}^T \hat{\boldsymbol{\sigma}}' \mathbf{V} \bar{\mathbf{x}} e^T$ in Eq. (2.57) may be bounded as follows:

$$\|\tilde{\mathbf{W}}^T \hat{\boldsymbol{\sigma}}' \mathbf{V} \bar{\mathbf{x}} e^T\| \leq V_m c_1 c_{11} Q_d \|\tilde{\mathbf{W}}\| \|e\| + V_m c_2 c_{11} \|\tilde{\mathbf{W}}\| \|e\|^2 \quad (2.60)$$

The term $(\tilde{\mathbf{W}}(\mathbf{W} - \tilde{\mathbf{W}}) + \tilde{\mathbf{V}}(\mathbf{V} - \tilde{\mathbf{V}}))$ in Eq. (2.57) may also be bounded as :

$$\begin{aligned} & \left\| \lambda \|e\| (\tilde{\mathbf{W}}(\mathbf{W} - \tilde{\mathbf{W}}) + \tilde{\mathbf{V}}(\mathbf{V} - \tilde{\mathbf{V}})) \right\| \\ & \leq -\lambda \|e\| (-\|\tilde{\mathbf{W}}\|^2 + W_m \|\tilde{\mathbf{W}}\| - \|\tilde{\mathbf{V}}\|^2 + V_m \|\tilde{\mathbf{V}}\|) \end{aligned} \quad (2.61)$$

It has been demonstrated that $\dot{\mathcal{L}}_2$ in Eq. (2.55) is negative definite if the Kalman gain matrix $\mathbf{K}_k$ is chosen properly. However, even in the case where $\dot{\mathcal{L}}_2$ in Eq. (2.55) is not negative definite, if the estimation is performed correctly, Eq. (2.55) is ensured to converge to 0, as the errors $\boldsymbol{\epsilon}_k$ and $\tilde{\boldsymbol{\Theta}}$ tend to zero over time. Therefore, the norm of $\dot{\mathcal{L}}_2$ in Eq. (2.55) is monotonically decreasing to 0 and can therefore be neglected when accounting $\dot{\mathcal{L}}$ for boundedness.

The bound on $\dot{\mathcal{L}}$ can thus be stated as follows:

$$\begin{aligned} \dot{\mathcal{L}} \leq & -\left(K_P + \frac{K_I}{s} + K_D s\right) \|e\|^2 + W_m c_{11} \|e\| - \|\tilde{\mathbf{W}}\| c_{11} \|e\| + V_m c_1 c_{11} Q_d \|\tilde{\mathbf{W}}\| \|e\| \\ & + V_m c_2 c_{11} \|\tilde{\mathbf{W}}\| \|e\|^2 + W_m c_3 \|e\| + W_m c_{11} c_1 Q_d \|\tilde{\mathbf{V}}\| \|e\| \\ & + W_m c_{11} c_2 \|\tilde{\mathbf{V}}\| \|e\|^2 - \lambda \|e\| (-\|\tilde{\mathbf{W}}\|^2 + W_m \|\tilde{\mathbf{W}}\| - \|\tilde{\mathbf{V}}\|^2 \\ & + V_m \|\tilde{\mathbf{V}}\|) \end{aligned} \quad (2.62)$$

For stability and convergence, the Lyapunov theory states that $\dot{\mathcal{L}}$ must be negative definite, as stated in Eq. (2.37). On one hand, $\dot{\mathcal{L}} = 0$ when $\|e\| = 0$, while on the other hand, Eq. (2.37) is true when the following equation is true.

$$\begin{aligned}
& \left(- \left(K_P + \frac{K_I}{s} + K_D s \right) + V_m c_2 c_{11} \|\tilde{\mathbf{W}}\| + W_m c_{11} c_2 \|\tilde{\mathbf{V}}\| \right) \|e\| + W_m c_{11} - \|\tilde{\mathbf{W}}\| c_{11} \\
& + V_m c_1 c_{11} Q_d \|\tilde{\mathbf{W}}\| + W_m c_3 + W_m c_{11} c_1 Q_d \|\tilde{\mathbf{V}}\| \\
& - \lambda \left(-\|\tilde{\mathbf{W}}\|^2 + W_m \|\tilde{\mathbf{W}}\| - \|\tilde{\mathbf{V}}\|^2 + V_m \|\tilde{\mathbf{V}}\| \right) \leq 0
\end{aligned} \tag{2.63}$$

2.7.5.1 Case 1

The NN approximation is performed properly, so $\|\tilde{\mathbf{W}}\|$ and $\|\tilde{\mathbf{V}}\|$ are small and tend to 0. The term $-\lambda \left(-\|\tilde{\mathbf{W}}\|^2 + W_m \|\tilde{\mathbf{W}}\| - \|\tilde{\mathbf{V}}\|^2 + V_m \|\tilde{\mathbf{V}}\| \right)$ still remains negative definite, as W_m and V_m are higher values. Eq. (2.63) can then be simplified as follows:

$$- \left(K_P + \frac{K_I}{s} + K_D s \right) \|e\| + (W_m c_{11} + W_m c_3) \leq 0 \tag{2.64}$$

and an upper bound on the tracking error $\|e\|$ may be found, such that:

$$\|e\| \leq \frac{W_m c_{11} + W_m c_3}{K_P + \frac{K_I}{s} + K_D s} \tag{2.65}$$

Eq. (2.65) clearly shows that the upper bound on the tracking error depends on the PID gains K_P , K_I and K_D . For the norm $\|e\|$ to remain positive and small, and thus, the negative definiteness of Eq. (2.62) is ensured if the magnitudes of K_P , K_I and K_D are chosen high enough.

2.7.5.2 Case 2

In the case where the neural network approximation is performed poorly, then $\|\tilde{\mathbf{W}}\|$ and $\|\tilde{\mathbf{V}}\|$ are large. The term $-\lambda \left(-\|\tilde{\mathbf{W}}\|^2 + W_m \|\tilde{\mathbf{W}}\| - \|\tilde{\mathbf{V}}\|^2 + V_m \|\tilde{\mathbf{V}}\| \right)$ will still remain negative definite, as W_m and V_m are higher positive values. If $\|e\|$ is negligible, then Eq. (2.37) remains true. On the contrary, if $\|e\|$ is large, then a proper choice of the linear PID controller gains would still ensure that Eq. (2.37) remains true.

In sum, the PID gains ensure ultimate stability, while the neural network ensures negative definiteness. Furthermore, the term $\ddot{\mathbf{\Theta}} \mathbf{L}^{-1} \tilde{\mathbf{\Theta}}$ in Eq. (2.57) is small and converges to 0. In fact,

if the amplitudes of K_P , K_I and K_D are chosen to be large enough, then the negative definiteness in Eq. (2.37) is respected. In sum, in both cases, the overall system stability and convergence of the proposed controller are guaranteed.

CHAPTER 3

ROBUST ADAPTIVE LATERAL CONTROL FOR THE CESSNA CITATION X WITH FLIGHT QUALITY VALIDATION

Rojo Princy Andrianantara ^a, Georges Ghazi^a, and Ruxandra Botez ^a, Elliot Quintin,
Valentine Duvelleroy

^a Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity
(LARCASE), System Engineering Department, École de technologie supérieure,
1100 Rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3

Article submitted for publication in *The Aeronautical Journal* , October 2025

Résumé

Afin de réaliser le contrôle latéral de l'avion d'affaire Cessna Citation X, cet article présente une stratégie de contrôle non linéaire avec une double adaptation pour le contrôle du taux de roulis et la stabilisation du taux de lacet, conçue pour suivre un modèle de référence. L'algorithme de contrôle proposé intègre trois éléments principaux : un contrôleur proportionnel-intégral-dérivé (PID), un contrôleur d'inversion et un contrôleur à réseau de neurones (NN). Le contrôleur d'inversion est réalisé à l'aide de la méthode des moindres carrés récurrents (RLS), qui fournit une estimation en temps réel des matrices de contrôle et d'état. Les résultats de la simulation démontrent que le contrôleur global suit efficacement le signal de référence du taux de roulis tout en maintenant la stabilité du taux de lacet. Les performances du contrôleur ont été évaluées dans 64 conditions de croisière différentes couvrant l'enveloppe opérationnelle du Cessna Citation X. Bien que les gains PID soient restés fixes dans tous les scénarios, le contrôleur a montré une forte adaptabilité, principalement grâce aux contrôleurs d'inversion dynamique RLS et au réseau de neurone NN. De plus, le contrôleur proposé a été réglé de manière optimale en cas de perturbations et a montré des performances robustes en présence de conditions de vent variables et de défaillances des actionneurs.

Abstract

To evaluate the lateral control of the Cessna Citation X business jet, this paper introduces a dual adaptive nonlinear control strategy for roll rate control and yaw rate stabilization, designed to track a reference model. The proposed control algorithm integrates three main elements: a Proportional-Integral-Derivative (PID) controller, an inversion controller, and a neural network (NN) controller. The inversion controller is implemented using the Recursive Least Square (RLS) method, which provides real-time estimation of the control and state matrices. Simulation results demonstrate that the overall controller effectively tracks the roll rate reference signal while maintaining the yaw rate stability. The performance of the controller was evaluated under 64 different cruise conditions covering the Cessna Citation X operating envelope. Although the PID gains remained fixed in all scenarios, the controller showed strong adaptability, primarily due to the RLS inversion and NN controllers. Moreover, the proposed controller was optimally tuned under disturbances and showed robust performance in the presence of varying wind conditions and actuator faults.

3.1 Introduction

Autonomous flight has become a major interest for aircraft manufacturers seeking to develop highly efficient flight controllers. These controllers must not only ensure aircraft stability, they must also be resilient to uncertainties and capable of adapting to variations in both aircraft dynamics and environmental conditions. The International Civil Aviation Organization (ICAO) has highlighted that Loss of Control In-Flight (LOC-II) remains one of the critical issues in aviation, with human error causing about 80% of flight accidents (ICAO, 2024). Consequently, improving autonomous flights could significantly enhance flight safety and performance.

Onboard controllers have primarily been designed using classical methods, such as Proportional-Integral-Derivative (PID) and modern methods, such as Linear Quadratic Regulators (LQR) (Stevens et al., 2015). These controllers are designed by linearizing the aircraft dynamics at given flight conditions (Yamina Boughari, Botez, et al., 2017). However,

this approach limits a controller's adaptability, as each controller's configuration remains valid only for a specific set of operating conditions. Covering all flight conditions would require extensive linearization and gain scheduling (Reynolds et al., 1996). Furthermore, even though they are widely used in commercial aircraft, their dependence on linearized models restricts their ability to handle nonlinearities.

To address these limitations, robust controls were designed in the late 1970's. Robust controllers, such as H_∞ (Efremov et al., 2021 ; A. Khalil & Fezans, 2021) and Linear Quadratic Gaussian Regulator (LQG) (Ferrier et al., 2018), were developed to account for disturbances. However, these methods highly depend on mathematical models, and thus harder to implement when the dynamics are unknown. Other research has focused on automating gain scheduling using optimization algorithms, such as Genetic and Differential evolution algorithms (Y. Boughari, Ghazi, et al., 2017b, 2017a).

Linearization often neglects higher-order nonlinear effects, thereby reducing the model's accuracy when changes occur, such as turbulent wind or aggressive maneuvers. The need to overcome linearization led researchers to explore alternative control methods capable of flight nonlinear dynamics. Advanced controllers such as Dynamic Inversion controllers (Horn, 2019 ; Moncayo et al., 2012 ; Steffensen et al., 2023) and Model Predictive Controllers (MPCs) (Doff-Sotta et al., 2025) were developed to address information loss from linearization. These methods are particularly powerful, as they adapt control inputs to respond to ongoing changes in the environment and aircraft. However, Dynamic Inversion controllers and MPCs also depend on mathematical models, requiring linearization (Ducard & Geering, 2008 ; Horn, 2019).

Nonlinear control such as Nonlinear Dynamic Inversion (NDI) (Grondman et al., 2018) and Nonlinear Model Predictive Control (NMPC) have been developed in order to capture model nonlinearities . Some of these controllers combined MPC with online state estimation (R. P. Andrianantara, Ghazi, et al., 2024). Adaptive filters, such as Recursive Least Squares (RLS) (Mahadi et al., 2022 ; Mohseni & Bernstein, 2022) and the Extended Kalman Filter (EKF)

(Puskorius & Feldkamp, 1991), allow real-time model estimation, avoiding the dependence on linear models. Hybrid control, combining classical control with Artificial Intelligence-driven control, offers improved performance (N. Nguyen & Krishnakumar, 2007). To overcome the modelling errors limitation, NDI has been successfully combined with reinforcement learning (Sun & van Kampen, 2020) and neural networks (Feith, 2020). Compared to most of the controllers described above, which rely on fixed parameters, neural network controllers are inherently adaptive.

Thanks to the significant advancements of computers in terms of memory and power, it is possible to use hybrid controllers (McLaughlin & Perhinschi, 2022 ; Xu et al., 2019) and data-driven controllers such as reinforcement learning and neural networks for control purposes. Reinforcement Learning (RL) offers promising alternatives by relying entirely on data. RL-based controllers optimize control policies by training agents to take actions, thereby improving system performance (Feith, 2020 ; Sutton & Barto, 2018). RL has the ability to adapt to complex nonlinear systems without knowledge of its dynamic model, making it powerful for real-world applications. However, RL requires extensive training time and data (Lewis & Vrabie, 2009). The training can be slow and involves the exploration of all possible scenarios in order to cover all of an aircraft maneuvering capability, which may not always be feasible in critical systems. Moreover, assessing RL-based controllers for stability is still challenging.

Several intelligent controllers have been designed, such as fuzzy and neural network controllers, which are also adaptive. They have been applied for low-level (R. P. Andrianantara, Ghazi, et al., 2024 ; S. M. Hosseini, Bematol, et al., 2024b) and high-level control (Q. Zhang & Liu, 2018). Adaptive control is particularly attractive because it does not require offline training (X. Li et al., 2015). During adaptation, the NN hyperparameters are updated online using adaptation laws (Slotine & Li, 1991). NNs have been successfully used for system identification in order to model nonlinear aeroservoelastic behaviors such as flutter dynamics (Boely & Botez, 2010), aircraft dynamics (Szymanski, Ghazi, & Botez, 2025 ; Tondji, Wade, Ghazi, & Mihaela Botez, 2022 ; Zohreh Nejad, Ghazi, & Botez, 2023), and

engine dynamics from flight tests (R. P. Andrianantara, Ghazi, & Botez, 2023b). By learning from data, NNs continuously improve their performance, and are able to adapt to unforeseen events, making them very robust against noise.

Despite advancements in adaptive and intelligent control, there is no established regulatory framework for their certification. Only linear controllers can still be validated, using flight qualities (Department of Defense, 1997). Neural network controllers have been implemented on UAVs (Hsu & Liu, 2021 ; Santoso, Garratt, & Anavatti, 2017 ; Zhou et al., 2021), due to their low risk for human life. However, aerospace regulatory organizations such as the EASA¹ and the FAA² (Agogino et al., 2024 ; European Union Aviation Safety Agency, 2024 ; Federal Aviation Administration, 2024) have studied the application of adaptive elements in aircraft systems. This lack of regulation for adaptive systems adds complexity to their certification and real-world applications, particularly in safety-critical systems. Therefore, this research proposes to validate the nonlinear lateral controller using existing flight qualities.

Several related works on adaptive control have been completed. In particular, flight tests were conducted to test adaptive controllers (N. Nguyen & Krishnakumar, 2007). Previous research (R. P. Andrianantara, Mancera Coyotl, Ghazi, & Botez, 2024 ; R. P. Andrianantara et al., 2023a) has focused on longitudinal control using adaptive elements, such as Recursive Least Squares (RLS) and nonlinear inversion techniques. Lateral control presents more challenges than longitudinal control due to complex roll-yaw coupling (Ghazi & Botez, 2015b), especially in high-angle-of-attack scenarios. To address this issue, this research introduces an adaptive Aileron-Rudder Interconnect (ARI), which has demonstrated its ability to improve yaw damping. The proposed adaptation laws were derived from (Kim & Calise, 1997 ; Rysdyk & Calise, 2005) and tailored to the CCX aircraft.

¹ European Union Aviation Safety Agency

² Federal Aviation Administration

3.1.1 Research Objectives and Organization

The main objective of this research is to design, implement and validate a robust adaptive lateral control architecture for the Cessna Citation X flight dynamics. The proposed control algorithm integrates a Recursive Least Square-based dynamic inversion, combined with an adaptive Neural Network compensation, thereby ensuring its “dual adaptation” to nonlinear dynamics and disturbance without gain scheduling, while ensuring compliance with flight-quality requirements. To this aim, a Stability-Augmented System (SAS) was developed for low-level yaw rate stabilization, while a Control-Augmented System (CAS) was designed for roll rate control. Previous attempts on lateral controllers have involved automatic tuning using Genetic Algorithms (Yamina Boughari, Botez, et al., 2017); however, those approaches were limited to linear models.

The main novelty and contributions of this research are as follows:

- The development of a dual-adaptive lateral control architecture combining RLS-based dynamic inversion with neural network compensation for nonlinear control.
- The implementation of the proposed architecture on a high-fidelity nonlinear model of the Cessna Citation X, capturing realistic lateral-directional dynamic modes and disturbances.
- A full-envelope validation across 64 cruise flight conditions, demonstrating robust performance without gain scheduling.
- The extension of flight-quality criteria stated in MIL-STD-1797A to nonlinear adaptive control validation, addressing a gap in the certification-oriented evaluation.
- An enhanced RLS estimation strategy with covariance reset logic with the aim to ensure numerical stability and convergence in closed-loop operation.
- A combined theoretical, Lyapunov-based and simulation-based validation under realistic disturbances and actuator faults.

The design methodology is validated using high-fidelity simulations in MATLAB/Simulink, where the controller performance is assessed under 64 cruise flight conditions. The flight

envelope was sampled to enable controller design across the entire cruise operational range of the CCX aircraft. Then, robustness is enhanced by incorporating wind and turbulence effects into the control design, as well as by adding custom logic to the RLS algorithm to prevent numerical instabilities.

The use of high-level flight dynamics further ensures the generalizability and applicability of the proposed controller. This research also addresses certification challenges by including flight quality constraints in the controller design. Finally, a stability analysis using Lyapunov criteria confirms the convergence and boundedness of the proposed RLS–NN combination. The results demonstrate precise tracking of reference signals, effective control surface deflections, and improved stability under varying flight conditions.

The remainder of this paper is structured as follows: the Cessna Citation X business jet and simulation platform are described in Section 3.1.2, and the control design methodology is explained in Section 3.2.0. The simulation results under ideal conditions are provided in Section 3.0, while the controller’s robustness is assessed in Section 3.4.0. Section 3.5.0 then elaborates the result discussions. Finally, the controller’s stability is assessed in the Appendix.

3.1.2 Cessna Citation X business jet model

3.1.2.1 Aircraft Description and Simulink Platform

The studied aircraft in this paper consists of the Cessna Citation X (CCX), which is one of the fastest business jet aircraft with a maximum cruise speed of 350 knots and a maximum Mach number of 0.92. The CCX was first introduced in 1996 and is powered by two turbofan Rolls-Royce AEC3007C-1 engines, delivering a maximum thrust of 6,442 lbf. Its maximum ceiling altitude is 51,000 ft and its maximum range reaches about 3,091 nm. The aircraft can carry 12 passengers and 3 crew members.

A high-fidelity MATLAB/Simulink model of the CCX is available at the LARCASE laboratory for research purposes (Ghazi & Botez, 2015a). The simulation platform gathers

several nonlinear components, such as wind and turbulence models, actuator dynamics, aerodynamic and propulsion models. Each component has been modelled and validated using simulation flight data from the Cessna Citation X Research Aircraft Flight Simulator (RAFS), shown in Figure 3.1.

The RAFS is a full-scale flight simulator, manufactured by CAE, and is qualified as Level-D (the highest certification level) for its flight dynamics. The RAFS (shown in Figure 3.1) reproduces a typical aircraft cockpit configuration. It includes a touchscreen desk displaying the avionics systems, autopilot functions, engine parameters, and pilot command interfaces such as the control wheel, thrust levers, and pedals. The simulator is also equipped with two visual display screens representing the external environment, a master computer, and the corresponding control panels. It provides two flight stations for the pilot and co-pilot. The RAFS can simulate several flight conditions, including takeoff, climb, cruise, descend and landing and taxi phases, and it can also simulate malfunctions and critical flight regime such as stall conditions.

The RAFS has been validated by CAE engineers using the Qualification Test Guide procedure. Therefore, the simulation platform can perfectly simulate the overall dynamics of the Cessna Citation X, and is thus suitable for designing the proposed lateral controller. The Simulink model includes linearization features at given airspeed and altitudes and has been used in earlier works, such as the design of LQR (Y. Boughari, Ghazi, et al., 2017a), H_∞ (Yamina Boughari, Botez, et al., 2017) and fuzzy controllers (S. M. Hosseini, Ghazi, & Botez, 2024b).



Figure 3.1 Cessna Citation X Level D Research Aircraft Flight Simulator (RAFS) manufactured by CAE.Inc

3.1.3 Aircraft Nonlinear Model

3.1.3.1 Nonlinear dynamics

Throughout this paper, only the lateral dynamics of the CCX are considered. While the aircraft linear state space model is relatively simple (Stevens et al., 2015), its nonlinear representation is considerably more complex. The main objective of this paper is to design a roll rate controller and achieve yaw rate stabilization using the ailerons and rudders. The general form of the lateral dynamics can be written as follows:

$$\begin{aligned}\dot{\mathbf{x}} &= \boldsymbol{\varphi}(v, p, r, \phi, \psi, \delta_a, \delta_r) \\ y &= \mathbf{h}(v, p, r, \phi, \psi)\end{aligned}\tag{3.1}$$

where the state vector $\mathbf{x} = [v \ p \ r \ \phi \ \psi]^T$ is composed of the lateral speed v , roll rate p , yaw rate r , roll angle ϕ , and yaw angle ψ . The control vector $\boldsymbol{\eta} = [\delta_a \ \delta_r]^T$ consists of the

aileron and rudder deflections. Here, $\boldsymbol{\varphi}$ is a nonlinear function, y is the output and $\mathbf{h}$ is an output nonlinear function. Eq.(3.1) is the nonlinear state space form of the lateral dynamics.

3.1.3.2 Actuator Dynamics

The ailerons and rudders actuators have been modeled using the Nonlinear Second-Order Integral block available in Simulink, which allows the nonlinearity of the aircraft dynamics to be increased, and thus makes the control algorithm suitable for complex systems. The characteristics of the actuators are given in Table 3.1.

Table 3.1 Actuators' Dynamic Characteristics

Characteristics	Value
Actuator frequency	20 rad/s
Rudder range	$\pm 30^\circ$
Aileron range	$\pm 32^\circ$
Damping	0.8
Rate limit	20°/sec

3.2 Lateral Controller Design Methodology

3.2.1 Roll Rate Controller Architecture

The suggested adaptive controller for roll rate control is illustrated in Figure 3.2. The control architecture is based on a Model Reference Adaptive Controller (MRAC) that integrates a Recursive Least Squares (RLS) based Dynamic Inversion and an Adaptive Neural Network (NN) controller. The hyperparameters of the NN are dynamically updated using custom-designed adaptation laws, derived from backpropagation techniques (Kim & Calise, 1997 ; Narendra & Annaswamy, 1987). These adaptation laws use the roll rate error and the derivative of the sideslip angle to stabilize the yaw rate. In addition, the control algorithm also includes a baseline PID controller and a feedforward gain K_{FF} , both of which help to improve the transient behavior and overall tracking performance. In this way, the control scheme achieves a structured hierarchy in which the dominant dynamics are compensated through RLS-based

$$\begin{aligned}\dot{\mathbf{x}} &= \boldsymbol{\Phi}(v, p, r, \phi, \psi, \delta_a, \delta_r) = \mathbf{f}(v, p, r, \phi, \psi) + \mathbf{g}(v, p, r, \phi, \psi)\boldsymbol{\eta} \\ \mathbf{y} &= \begin{bmatrix} p \\ r \end{bmatrix} = \mathbf{C}\mathbf{x}\end{aligned}\quad (3.2)$$

where $\mathbf{f}$ and $\mathbf{g}$ are nonlinear functions. The output matrix $\mathbf{C}$ is equal to $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ corresponding to the roll and yaw rates. Let us further consider the matrices $\mathbf{C}_1 = [0 \ 1 \ 0 \ 0 \ 0]$ and $\mathbf{C}_2 = [0 \ 0 \ 1 \ 0 \ 0]$, so that:

$$\begin{aligned}p &= \mathbf{C}_1 \cdot \mathbf{x} \\ r &= \mathbf{C}_2 \cdot \mathbf{x}\end{aligned}\quad (3.3)$$

The roll and yaw dynamics may be developed in the full nonlinear form as follows :

$$\begin{aligned}\dot{p} &= f_p(v, p, r, \phi, \psi) + g_{pa}(v, p, r, \phi, \psi)\delta_a + g_{pr}(v, p, r, \phi, \psi)\delta_r \\ \dot{r} &= f_r(v, p, r, \phi, \psi) + g_{ra}(v, p, r, \phi, \psi)\delta_a + g_{rr}(v, p, r, \phi, \psi)\delta_r\end{aligned}\quad (3.4)$$

In this formulation, the nonlinear term f_p and f_r in Eq. (3.4) represents the aerodynamic contributions to roll and yaw dynamics, which depend on flight states $\mathbf{x} = [v \ p \ r \ \phi \ \psi]^T$. g_{pa} , g_{pr} , g_{ra} and g_{rr} represents the effect of the aileron and rudder deflections on the lateral-directional motion. In particular, roll rate dynamics are primarily influenced by aileron deflections δ_a and rolling moment coefficients, while yaw rate dynamics are driven by rudder inputs δ_r and yawing moment coefficients. In addition, coupling terms between roll and yaw dynamics arise due to aerodynamic interactions, especially through sideslip effects. These nonlinearities play a critical role in lateral-directional control and must be properly accounted for in the controller design. In the proposed approach, the dynamic inversion aims to compensate for the dominant nonlinear dynamics, while the neural network is used to capture residual nonlinearities and uncertainties.

Inverting Eq.(3.1), $\boldsymbol{\eta}$ can be expressed as a function of the states and their derivatives, as given below:

$$\boldsymbol{\eta} = \boldsymbol{\varphi}^{-1}(\dot{\mathbf{x}}, \mathbf{x}) \quad (3.5)$$

By applying the first-order Taylor series expansion in Eq.(3.2) around $(\mathbf{x}_0, \boldsymbol{\eta}_0)$, $\boldsymbol{\varphi}$ can be expressed as:

$$\boldsymbol{\varphi}(\mathbf{x}, \boldsymbol{\eta}) = \boldsymbol{\varphi}(\mathbf{x}_0, \boldsymbol{\eta}_0) + \boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\eta})_{\mathbf{x}_0, \boldsymbol{\eta}_0}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\varphi}'_{\boldsymbol{\eta}}(\mathbf{x}, \boldsymbol{\eta})_{\mathbf{x}_0, \boldsymbol{\eta}_0}(\boldsymbol{\eta} - \boldsymbol{\eta}_0) \quad (3.6)$$

where $\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\eta})_{\mathbf{x}_0, \boldsymbol{\eta}_0} \in \mathbb{R}^{5 \times 5}$ and $\boldsymbol{\varphi}'_{\boldsymbol{\eta}}(\mathbf{x}, \boldsymbol{\eta})_{\mathbf{x}_0, \boldsymbol{\eta}_0} \in \mathbb{R}^{5 \times 2}$ are partial gradients with respect to the states $\mathbf{x}$ and controls $\boldsymbol{\eta}$, respectively. The terms in Eq.(3.6) can be defined as follows:

$$\boldsymbol{\varphi}(\mathbf{x}_0, \boldsymbol{\eta}_0) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\boldsymbol{\eta} \quad (3.7.a)$$

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\eta}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\boldsymbol{\eta} \quad (3.7.b)$$

$$\boldsymbol{\varphi}'_{\boldsymbol{\eta}}(\mathbf{x}, \boldsymbol{\eta}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \boldsymbol{\eta}} + \frac{\partial [\mathbf{g}(\mathbf{x})\boldsymbol{\eta}]}{\partial \boldsymbol{\eta}} = \mathbf{g}(\mathbf{x}) \quad (3.7.c)$$

Thus, Eq.(3.6) can be rewritten as:

$$\boldsymbol{\varphi}(\mathbf{x}, \boldsymbol{\eta}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\boldsymbol{\eta} + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\boldsymbol{\eta} \right)_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \boldsymbol{\eta} \quad (3.8)$$

where $\Delta \mathbf{x} = (\mathbf{x} - \mathbf{x}_0) \in \mathbb{R}^{5 \times 1}$ represents the state variations, and $\Delta \boldsymbol{\eta} = (\boldsymbol{\eta} - \boldsymbol{\eta}_0) \in \mathbb{R}^{2 \times 1}$ represents the input variations.

Expressing Eq.(3.2) around an equilibrium point, so that $\boldsymbol{\varphi}(\mathbf{x}, \boldsymbol{\eta}) = \dot{\mathbf{x}}_0 + \Delta \dot{\mathbf{x}}$, and considering $\dot{\mathbf{x}}_0 = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\boldsymbol{\eta}$, the state variation $\Delta \dot{\mathbf{x}}$ in Eq.(3.8) can be identified:

$$\Delta \dot{\mathbf{x}} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\boldsymbol{\eta} \right)_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \boldsymbol{\eta} \quad (3.9)$$

In a more compact form:

$$\Delta \dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \mathbf{x} + \mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \boldsymbol{\eta} \quad (3.10)$$

here $\mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \boldsymbol{\eta} \right)_{\mathbf{x}_0, \boldsymbol{\eta}_0} \in \mathbb{R}^{5 \times 5}$ and $\mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0} = \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \boldsymbol{\eta}_0} \in \mathbb{R}^{5 \times 2}$.

In full matrix form, Eq. (3.10) can be written as follows :

$$\begin{pmatrix} \Delta \dot{v} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \\ \Delta \dot{\psi} \end{pmatrix} = \begin{pmatrix} F_{11} & F_{12} & F_{13} & F_{14} & F_{15} \\ F_{21} & F_{22} & F_{23} & F_{24} & F_{25} \\ F_{31} & F_{32} & F_{33} & F_{34} & F_{35} \\ F_{41} & F_{42} & F_{43} & F_{44} & F_{45} \\ F_{51} & F_{52} & F_{53} & F_{54} & F_{55} \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta \phi \\ \Delta \psi \end{pmatrix} + \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \\ G_{31} & G_{32} \\ G_{41} & G_{42} \\ G_{51} & G_{52} \end{pmatrix} \begin{pmatrix} \Delta \delta_a \\ \Delta \delta_r \end{pmatrix} \quad (3.11)$$

From Eq.(3.10), $\mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \in \mathbb{R}^{5 \times 5}$ is the local state matrix at $(\mathbf{x}_0, \boldsymbol{\eta}_0)$, while $\mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \in \mathbb{R}^{5 \times 2}$ is the local control matrix. Let us consider $\mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0} = [\mathbf{G}_{1_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \quad \mathbf{G}_{2_{\mathbf{x}_0, \boldsymbol{\eta}_0}}]$, where $\mathbf{G}_{1_{\mathbf{x}_0, \boldsymbol{\eta}_0}}$ is the aileron control matrix, and $\mathbf{G}_{2_{\mathbf{x}_0, \boldsymbol{\eta}_0}}$ is the rudder control matrix. Then, Eq.(3.10) can be rewritten linearly as:

$$\Delta \dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \mathbf{x} + \mathbf{G}_{1_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \Delta \delta_a + \mathbf{G}_{2_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \Delta \delta_r \quad (3.12)$$

Taking into account Eq.(3.3), the roll rate and yaw rate dynamics can be derived from Eq.(3.12) as follows:

$$\begin{aligned} \Delta \dot{p} &= \mathbf{C}_1 \Delta \dot{\mathbf{x}} = \mathbf{C}_1 \mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \mathbf{x} + \mathbf{C}_1 \mathbf{G}_{1_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \Delta \delta_a + \mathbf{C}_1 \mathbf{G}_{2_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \Delta \delta_r \\ \Delta \dot{r} &= \mathbf{C}_2 \Delta \dot{\mathbf{x}} = \mathbf{C}_2 \mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta \mathbf{x} + \mathbf{C}_2 \mathbf{G}_{1_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \Delta \delta_a + \mathbf{C}_2 \mathbf{G}_{2_{\mathbf{x}_0, \boldsymbol{\eta}_0}} \Delta \delta_r \end{aligned} \quad (3.13)$$

and then the control allocation for the ailerons and rudders can be expressed by inverting Eq.(3.13):

$$\Delta\delta_a = (\mathbf{C}_1 \mathbf{G}_{1_{x_0, \eta_0}})^{-1} (\Delta\dot{p} - \mathbf{C}_1 \mathbf{F}_{x_0, \eta_0} \Delta\mathbf{x} - \mathbf{C}_1 \mathbf{G}_{2_{x_0, \eta_0}} \Delta\delta_r) \quad (3.14.a)$$

$$\Delta\delta_r = (\mathbf{C}_2 \mathbf{G}_{2_{x_0, \eta_0}})^{-1} (\Delta\dot{r} - \mathbf{C}_2 \mathbf{F}_{x_0, \eta_0} \Delta\mathbf{x} - \mathbf{C}_2 \mathbf{G}_{1_{x_0, \eta_0}} \Delta\delta_a) \quad (3.14.b)$$

The roll rate tracking error e_p can be defined as:

$$\begin{aligned} e_p &= p_{ref} - p \\ \Rightarrow \dot{e}_p &= \dot{p}_{ref} - \dot{p} \end{aligned} \quad (3.15)$$

where p_{ref} is the roll rate reference signal. The following linear controller (Stevens et al., 2015) is chosen to control the roll rate dynamics $\Delta\dot{p}$:

$$\Delta\dot{p} = K_{FF} \dot{p}_{ref} + v \quad (3.16)$$

where v is selected to reduce the roll rate error such that $v = -\dot{e}_p$, and K_{FF} is a feedforward gain. This linear controller allows the overall transient behavior of the controller to be shaped, handling the linear part of the control allocation. The linear controller ensures the overall stability of the control algorithm, as shown by the Lyapunov analysis in the Appendix.

Applying Eq.(3.16), the aileron control in Eq.(3.14.1) can be expressed as follows:

$$\Delta\delta_a = (\mathbf{C}_1 \mathbf{G}_{1_{x_0, \eta_0}})^{-1} (K_{FF} \dot{p}_{ref} + v - \mathbf{C}_1 \mathbf{F}_{x_0, \eta_0} \Delta\mathbf{x} - \mathbf{C}_1 \mathbf{G}_{2_{x_0, \eta_0}} \Delta\delta_r) \quad (3.17)$$

where the yaw rate r is stabilized to reduce the Dutch-roll effect induced by the roll-yaw coupling. As seen in Figure 3.2, a yaw damper K_{WF} and a washout filter are added. The washout works as a low-pass filter (Stevens et al., 2015). Thus, the feedback on the yaw rate may be summed by the following Laplace transfer function:

$$F(s) = \frac{K_{WF} \cdot \tau s}{\tau s + 1} \quad (3.18)$$

where τ is the washout time constant and K_{WF} is a tunable gain. Therefore, the rudder control in Eq.(3.14.2) can be rewritten as follows (Stevens et al., 2015):

$$\Delta\delta_r = (\mathbf{C}_2 \mathbf{G}_{2_{x_0, \eta_0}})^{-1} \left(\frac{K_{WF} \cdot \tau s}{\tau s + 1} r - \mathbf{C}_2 \mathbf{F}_{x_0, \eta_0} \Delta \mathbf{x} - \mathbf{C}_2 \mathbf{G}_{1_{x_0, \eta_0}} \Delta\delta_a \right) \quad (3.19)$$

In Eqs. (3.17) and (3.19), the inversion controller compensates for the estimated local dynamics affecting the roll-rate and yaw-rate responses. More specifically, the terms $\mathbf{C}_1 \mathbf{F}_{x_0, \eta_0} \Delta \mathbf{x}$ and $\mathbf{C}_2 \mathbf{F}_{x_0, \eta_0} \Delta \mathbf{x}$ represent the state-dependent contributions of the local aircraft dynamics to the roll and yaw motions, respectively. The terms $\mathbf{C}_1 \mathbf{G}_{2_{x_0, \eta_0}} \Delta\delta_r$ and $\mathbf{C}_2 \mathbf{G}_{1_{x_0, \eta_0}} \Delta\delta_a$ represent the control cross-coupling effects between rudder and roll dynamics, and between aileron and yaw dynamics. By compensating these estimated terms through dynamic inversion, the controller isolates the desired roll and yaw rate closed-loop dynamics. The remaining estimation errors and unmodeled nonlinearities are addressed by the adaptive neural network compensation.

3.2.2.2 RLS online state estimation

The lateral-directional dynamics of an aircraft are governed by nonlinear aerodynamic forces and moments that depend on flight states, such as lateral velocity, roll rate, yaw rate, and orientation angles, as well as control surface deflections. In particular, roll and yaw responses are influenced by the rolling and yawing moments generated by the ailerons and rudder, along with aerodynamic coupling effects associated with sideslip angles. Due to the nonlinear and flight-condition-dependent nature of these dynamics, an exact analytical representation is difficult to obtain in real time. Therefore, a local approximation of the nonlinear dynamics is necessary using a linear parameterization, which is updated online using the Recursive Least Squares (RLS) algorithm.

The RLS algorithm estimates the local coefficients of the linearized lateral-directional dynamics model, representing the influence of the aircraft states and control inputs on the roll-rate and yaw-rate responses. These coefficients can be interpreted as time-varying approximations of the aerodynamic coefficients and control effectiveness, which vary across the flight envelope. The estimated model is then used within the dynamic inversion controller to compensate for the dominant state-dependent dynamics and coupling effects. In this architecture, the PID controller provides baseline model tracking performance and ensures its stability, while the RLS-based dynamic inversion enhances control performance by adapting to aircraft dynamics variations.

In practice, the analytical expressions of $\mathbf{F}_{\mathbf{x}_0, \eta_0}$, $\mathbf{G}_{1_{\mathbf{x}_0, \eta_0}}$ and $\mathbf{G}_{2_{\mathbf{x}_0, \eta_0}}$ can be difficult to determine. To address this issue, the Recursive Least Squares (RLS) method (Mahadi et al., 2022 ; Mohseni & Bernstein, 2022) is proposed. RLS allows the local dynamics to be approximated at each time step, which avoids the calculation of complex second-order derivatives. From Eq.(3.10), $\Delta \dot{\mathbf{y}}$ can be written as:

$$\Delta \dot{\mathbf{y}} = \mathbf{C} \Delta \dot{\mathbf{x}} = \mathbf{C} \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{C} \mathbf{G}_{\mathbf{x}_0, \eta_0} \Delta \boldsymbol{\eta} \quad (3.20)$$

In discrete form, Eq.(3.20) becomes:

$$\Delta \mathbf{y}_{k+1} = \mathbf{C} \tilde{\mathbf{F}}_k \Delta \mathbf{x}_k + \mathbf{C} \tilde{\mathbf{G}}_k \Delta \boldsymbol{\eta}_k \quad (3.21)$$

and in matrix form:

$$\Delta \mathbf{y}_{k+1} = \mathbf{C} [\Delta \mathbf{x}_k \quad \Delta \boldsymbol{\eta}_k] \begin{bmatrix} \tilde{\mathbf{F}}_k^T \\ \tilde{\mathbf{G}}_k^T \end{bmatrix} \quad (3.22)$$

where $\tilde{\mathbf{F}}_k \in \mathbb{R}^{5 \times 5}$ and $\tilde{\mathbf{G}}_k \in \mathbb{R}^{5 \times 2}$ correspond to the discrete forms of $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ and $\mathbf{G}_{\mathbf{x}_0, \eta_0}$, respectively. Replacing $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{G}}_k$ by their estimates $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$, and considering the notations

$\hat{\boldsymbol{\Theta}}_k = \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix}$ and $\mathbf{X}_k = \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \boldsymbol{\eta}_k \end{bmatrix}$, Eq.(3.22) can be written in a compact form:

$$\Delta \hat{\mathbf{y}}_{k+1} = \mathbf{C} \mathbf{X}_k^T \hat{\boldsymbol{\Theta}}_k \quad (3.23)$$

where $\hat{\boldsymbol{\Theta}}_k \in \mathbb{R}^{7 \times 5}$ is the vector of unknown parameters, $\Delta \hat{\mathbf{y}}_{k+1} \in \mathbb{R}^{2 \times 1}$ corresponds to the estimate of $\Delta \mathbf{y}_{k+1}$, and $\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ so that $\Delta \hat{\mathbf{y}} = \begin{bmatrix} \Delta \hat{p} \\ \Delta \hat{r} \end{bmatrix}$. The parameter vector $\hat{\boldsymbol{\Theta}}_k$ contains the unknown coefficients $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ of the local linearized model estimated online by the RLS algorithm. These coefficients represent the local influence of the lateral states and control inputs on the aircraft's lateral-directional dynamics.

The RLS algorithm is derived from the Matrix Inversion Lemma and involves three main parameters: the estimation error matrix $\boldsymbol{\epsilon}_k$, the covariance matrix $\mathbf{L}_k$, and the forgetting factor κ . These parameters are updated numerically using the following equations (Haykin, 2013):

$$\boldsymbol{\epsilon}_k = \Delta \mathbf{x}_k - \Delta \hat{\mathbf{x}}_k \quad (3.24)$$

and

$$\mathbf{L}_k = \frac{1}{\kappa} \left(\mathbf{L}_{k-1} - \frac{\mathbf{L}_{k-1} \mathbf{X}_k \mathbf{X}_k^T \mathbf{L}_{k-1}}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \right) \quad (3.25)$$

$$\hat{\boldsymbol{\Theta}}_{k+1} = \hat{\boldsymbol{\Theta}}_k + \frac{\mathbf{L}_k \mathbf{X}_{k+1}}{\kappa + \mathbf{X}_{k+1}^T \mathbf{L}_k \mathbf{X}_{k+1}} \boldsymbol{\epsilon}_{k+1} \quad (3.26)$$

where $\hat{\boldsymbol{\Theta}}_{k+1}$ is the updated parameter. The RLS filter uses the previous state variations $\Delta \mathbf{x}_k$ and previous control inputs $\Delta \boldsymbol{\eta}_k$ in order to estimate $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ at time step $k + 1$, making it ideal for online approximations. The initial values $\mathbf{L}_0$, and $\boldsymbol{\Theta}_0$ are pre-defined. An update procedure for the covariance matrix $\mathbf{L}_k$ is suggested in Algorithm 3.1. Two distinct estimation errors are considered: the roll rate error $\boldsymbol{\epsilon}_{p_k}$ and the yaw rate error $\boldsymbol{\epsilon}_{r_k}$. The covariance matrix $\mathbf{L}_k$ is updated whenever these errors exceed their respective thresholds $\boldsymbol{\epsilon}_{p_{threshold}}$ and $\boldsymbol{\epsilon}_{r_{threshold}}$.

Algorithm 3.1 RLS algorithm and covariance matrix
update

```

Initialize  $\Theta_0, \mathbf{L}_0$ 
Set  $\kappa, \epsilon_{p\_threshold}, \epsilon_{r\_threshold}$ 
Algorithm
Compute  $\hat{\Theta}_k$  using Eq.(3.26)
Compute  $\Delta\hat{y}_{k+1}$  using Eq.(3.23)
Compute  $\epsilon_{p_k}$  and  $\epsilon_{r_k}$  using Eq.(3.24)
Update  $\mathbf{L}_k$  using Eq.(3.25)
if  $|\epsilon_{p_k}| \leq \epsilon_{p\_threshold}$  or  $|\epsilon_{r_k}| \leq \epsilon_{r\_threshold}$ 
then  $\mathbf{L}_k = \mathbf{10}^3 \mathbf{I}$ 
else compute  $\mathbf{L}_k$  using Eq.(3.25)

```

3.2.2.3 Adaptive Neural Network Controller

The neural network is used to approximate the residual nonlinearities and modelling uncertainties not captured by the RLS-based local linear model, including higher-order aerodynamic effects and approximation errors arising from the linearization process. Its output is integrated as an adaptive compensation term within the control law, enabling correction of the remaining dynamics not addressed by the dynamic inversion. In this architecture, the RLS-based estimation captures the dominant state-dependent behaviour, while the neural network enhances overall performance by compensating residual nonlinear effects.

In practice, using estimates inevitably introduces prediction errors. The estimated output $\Delta\hat{y}_{k+1}$ can be expressed with a prediction error ϵ as follows:

$$\Delta\hat{y}_{k+1} = \Delta y_{k+1} - \epsilon \quad (3.27)$$

Rewriting Eq.(3.17) and Eq.(3.19) in discrete form, the aileron and rudder control laws can be summed as follows :

$$\begin{aligned}\Delta\delta_{a_k} &= (\mathbf{C}_1\hat{\mathbf{G}}_{1k})^{-1}[-K_{FF}\dot{p}_{ref} + v - \mathbf{C}_1\hat{\mathbf{F}}_k\Delta\mathbf{x}_k - \mathbf{C}_1\hat{\mathbf{G}}_{2k}\Delta\delta_{r_{k-1}}\varepsilon_a] \\ \Delta\delta_{r_k} &= (\mathbf{C}_2\hat{\mathbf{G}}_{2k})^{-1}\left[\frac{K_{WF}\cdot\tau_S}{\tau_S + 1}r - \mathbf{C}_2\hat{\mathbf{F}}_k\Delta\mathbf{x}_k - \mathbf{C}_2\hat{\mathbf{G}}_{1k}\Delta\delta_{a_{k-1}} - \varepsilon_r\right]\end{aligned}\quad (3.28)$$

where ε_a and ε_r are the control compensations applied to the ailerons and rudders due to RLS modelling errors. The term $\mathbf{C}_1\hat{\mathbf{G}}_{2k}\Delta\delta_{r_k}$ corresponds to the rudder effect on the roll rate, while the term $\mathbf{C}_2\hat{\mathbf{G}}_{1k}\Delta\delta_{a_k}$ corresponds to the ailerons' effect on the yaw rate. The cross-coupling effects between roll and yaw dynamics are considered negligible under the considered cruise flight conditions. Although these effects may become significant at high angles of attack or during aggressive maneuvers, these regimes are outside the scope of the present study. Thus, the cross-coupling terms in Eq. (3.28) were considered negligible in comparison with the contributions of the other terms, and they were compensated by the neural network controller. It should also be noted that, to avoid numerical errors during simulations, the values of $\Delta\delta_a$ and $\Delta\delta_r$ in Eq.(3.28) are taken at the previous time step ($k - 1$).

The terms $\mathbf{C}_1\hat{\mathbf{G}}_{1k}$, $\mathbf{C}_1\hat{\mathbf{F}}_k\Delta\mathbf{x}_k$, $\mathbf{C}_2\hat{\mathbf{G}}_{2k}$ and $\mathbf{C}_2\hat{\mathbf{F}}_k\Delta\mathbf{x}_k$ correspond to the estimated local lateral dynamics. In order to correct the RLS estimation errors when estimating $\hat{\mathbf{G}}_{1k}$, $\hat{\mathbf{G}}_{2k}$ and $\hat{\mathbf{F}}_k$, adaptive compensation terms v_{ad_a} and v_{ad_r} are introduced:

$$\begin{aligned}\varepsilon_a &= v_{ad_a} \\ \varepsilon_r &= v_{ad_r}\end{aligned}\quad (3.29)$$

A single layer Feedforward Neural Network (FFNN) is used to model v_{ad_a} and v_{ad_r} from the RLS estimation errors. FFNNs are widely used for regression tasks, as they are well-suited to approximate nonlinear mathematical functions from a given dataset.

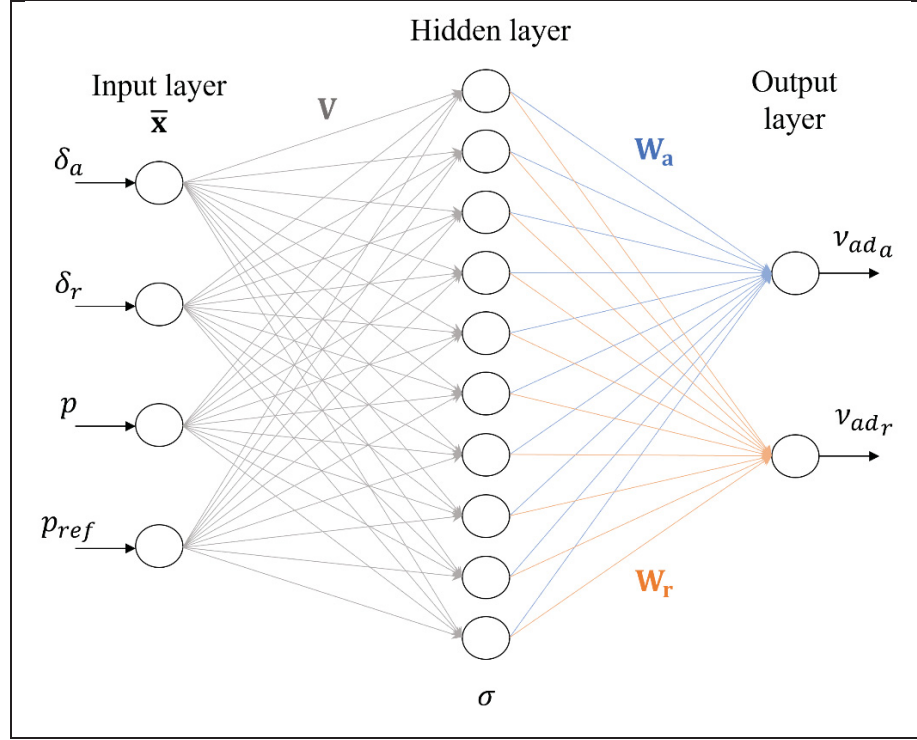


Figure 3.3 SHL Feedforward Neural Network

An FFNN is shown in Figure 3.3, and is composed of an input layer $\bar{\mathbf{x}}$, a hidden layer, an output layer and weight matrices. The information flows in a forward direction. The NN performance depends on its hyperparameters, including the number of neurons and hidden layers, the activation function σ , and the weight matrices $\mathbf{W} = [\mathbf{W}_a \ \mathbf{W}_r]^T$ and $\mathbf{V}$. The network has two adaptive outputs on the ailerons v_{ad_a} and the rudders v_{ad_r} , respectively. The weight matrices $\mathbf{W}_a$ and $\mathbf{W}_r$ are updated with their respective adaptation laws. The neural network adaptive scalar outputs as they are shown in Figure 3.3 can be written as (Stevens et al., 2015):

$$\begin{aligned} v_{ad_a} &= \mathbf{W}_a^T \sigma(\mathbf{V}\bar{\mathbf{x}}) \\ v_{ad_r} &= \mathbf{W}_r^T \sigma(\mathbf{V}\bar{\mathbf{x}}) \end{aligned} \quad (3.30)$$

where $\mathbf{W}_a \in \mathbb{R}^{N \times n}$, $\mathbf{W}_r \in \mathbb{R}^{N \times n}$, $\mathbf{V} \in \mathbb{R}^{m \times N}$, $\sigma \in \mathbb{R}^{N \times N}$, and $\bar{\mathbf{x}} \in \mathbb{R}^{m \times 1}$, defined as $\bar{\mathbf{x}} = [\delta_a \ \delta_r \ p_{ref} \ p]^T$, m is the number of inputs, n is equal to 1 and N is the number of neurons in the hidden layer. The input and output layer use linear activation functions, while the hidden layer uses tangent sigmoid functions, defined as:

$$\sigma_i(z) = \frac{1}{1 + \exp(-z)} \quad (3.31)$$

where i is the position of the neuron in the hidden layer, and z is the neuron input information.

3.2.2.4 Adaptation laws

The neural network weights are updated using backpropagation. The update laws have been derived from (Ge et al., 2013 ; Smith et al., 2010 ; Stevens et al., 2015) and modified to fit the proposed lateral control, so that:

$$\begin{aligned} \dot{\mathbf{W}}_a &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}'\mathbf{V}^T\bar{\mathbf{x}})e_p + \lambda\|e_p\|\mathbf{W}]\boldsymbol{\Gamma}_a \\ \dot{\mathbf{W}}_r &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}'\mathbf{V}^T\bar{\mathbf{x}})e_r + \lambda\|e_r\|\mathbf{W}]\boldsymbol{\Gamma}_r \\ \dot{\mathbf{V}} &= -\boldsymbol{\Gamma}_V[\bar{\mathbf{x}}e_p\mathbf{W}_a^T\boldsymbol{\sigma}' + \bar{\mathbf{x}}e_r\mathbf{W}_r^T\boldsymbol{\sigma}' + \lambda\|e_p + e_r\|\mathbf{V}] \end{aligned} \quad (3.32)$$

where $\dot{\mathbf{W}}_a$, $\dot{\mathbf{W}}_r$, and $\dot{\mathbf{V}}$ are the weight updates, $\boldsymbol{\Gamma}_a$, $\boldsymbol{\Gamma}_r$ and $\boldsymbol{\Gamma}_V$ are the learning rates, $\boldsymbol{\sigma}'$ is the gradient of $\boldsymbol{\sigma}$, λ is a scalar value used in backpropagation (Johnson & Calise, 2003 ; Stevens et al., 2015), e_p is the roll rate tracking error, and e_r is set to damp the sideslip angle derivative, so that $e_r = -\dot{\beta}$.

3.2.2.5 PID Controller

Eq.(3.17) shows that the aileron command is driven by the error controller v . The specific role of this controller is limited to ensuring convergence and stability. Therefore, a simple linear PID controller is enough, formulated as (Stevens et al., 2015):

$$v = \left(K_P + \frac{K_I}{s} + K_D s\right)e \quad (3.33)$$

where K_P , K_I and K_D are the proportional, integral and derivative gains, respectively.

3.2.2.6 Sideslip Angle Derivative Approximation

As shown in Figure 3.2, the control algorithm includes a yaw rate stabilization. A neural network is used to damp the sideslip angle, as discussed in Stevens and Lewis (Stevens et al., 2015). This method is designed to mitigate Dutch-roll effects, reduce yaw-rate oscillations, and improve coordinated-turn performance. Accordingly, the sideslip angle derivative $\dot{\beta}$ is regulated to 0.

Since $\dot{\beta}$ is not a directly measurable state, its feedback is less reliable due to sensor limitations. Nevertheless, it can be approximated using measurable quantities such as the lateral acceleration $\dot{v}$ and the true airspeed TAS (Stevens et al., 2015), as follows:

$$\dot{\beta} \approx \frac{\dot{v}}{TAS} \quad (3.34)$$

3.2.3 Final Command Laws

By replacing Eq.(3.30) and Eq.(3.33) into Eq.(3.28), the command laws on the ailerons and rudders can be written as follows:

$$\begin{aligned} \Delta\delta_{a_k} &= (\mathbf{C}_1 \hat{\mathbf{G}}_{1k})^{-1} \left[\mathbf{K}_{FF} \dot{p}_{ref} + \left(K_P + \frac{K_I}{s} + K_D s \right) e_p - \mathbf{C}_1 \hat{\mathbf{F}}_k \Delta \mathbf{x}_k \right. \\ &\quad \left. - \mathbf{C}_1 \hat{\mathbf{G}}_{2k} \Delta\delta_{r_{k-1}} - \mathbf{W}_a^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \right] \\ \Delta\delta_{r_k} &= (\mathbf{C}_2 \hat{\mathbf{G}}_{2k})^{-1} \left[\frac{K_{WF} \cdot \tau_S}{\tau_S + 1} r - \mathbf{C}_2 \hat{\mathbf{F}}_k \Delta \mathbf{x}_k - \mathbf{C}_2 \hat{\mathbf{G}}_{1k} \Delta\delta_{a_{k-1}} \right. \\ &\quad \left. - \mathbf{W}_r^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \right] \end{aligned} \quad (3.35)$$

3.2.4 Roll Rate Reference model

The lateral controller must satisfy specific performance criteria according to the flight qualities given in MIL-STD-1797A (Department of Defense, 1997). Since the control problem is nonlinear, a MRAC is designed, with a roll-rate reference model specified. The following transfer function is used to approximate the roll and Dutch-roll dynamics (Stevens et al., 2015):

$$\frac{p_{ref}}{p_{cmd}} = \frac{\omega_{dr}^2}{s^2 + 2 \xi_{dr} \omega_{dr} s + \omega_{dr}^2} \frac{1}{\tau s + 1} \quad (3.36)$$

where τ is the roll time constant, and ξ_{dr} and ω_{dr} are the Dutch-roll damping ratio and the frequency, respectively. In Eq.(3.36), the first-order transfer function $1/(\tau s + 1)$ is used to approximate the roll mode, while the second-order transfer function approximates the Dutch-roll mode.

3.2.5 Flight Quality Requirements

As stated above, the actual flight qualities (FQ) in MIL-STD-1797A (Department of Defense, 1997) were originally developed for linear systems, for which time-frequency analyses are possible. The Level 1 lateral flight qualities for roll and Dutch-roll modes are given in Table 3.2 for Category B cruise flights and Class II mid-sized aircraft. In addition, Table 3.3 summarized the required numerical values for ξ_{dr} , ω_{dr} and τ .

Table 3.2 Level 1 Flight quality requirements for lateral-directional dynamics and category B flight phase

Characteristics	Value
Roll time constant	$\tau_p < 1.4 \text{ sec}$
Dutch-roll minimum damping,	$\min \xi_{dr} = 0.08$
Dutch-roll minimum frequency, rad/s	$\min \omega_{dr} = 0.4 \text{ rad/s}$
Minimum cross product $\xi_{dr} \omega_{dr}$	$\min \xi_{dr} \omega_{dr} = 0.15$
Delay allowed, sec	0.1

3.2.6 Performance metrics

The performance of the controller was evaluated using *overall SSE* , expressed as:

$$\text{overall SSE} = \sum_{i=1}^{i=N} SSE_i ; \quad SSE = \Delta t \sum_{t=0}^{t=T} (p_t - p_{ref_t})^2 \quad (3.37)$$

where N is the number of flights. The SSE is computed for one flight condition, p_t and p_{ref_t} are the roll rate and the roll rate reference values at time t , respectively, T is the simulation time, and $\Delta t = 0.02$ sec is the sample time.

3.3 Controller Tuning and Simulation under Ideal Conditions

3.3.1 Controller tuning parameters

All control parameters used in the previous equations and in Figure 3.2 were tuned and are summarized in Table 3.3. The first cruise simulation shown in Figure 3.4 was realized using a 10 second period roll rate reference signal randomly set between $\pm 1^\circ/\text{sec}$, over 50 seconds.

Table 3.3 Lateral Controller Design Parameters in **Figure 3.2**

Controller Element	Parameters	Value
PID Controller	K_P, K_I, K_D	-1, -1, 0
Feedforward controller	K_{FF}	-0.1
Yaw damper, Washout Filter	K_{WF} , time constant τ	3, 2 sec
RLS Inversion Controller	Forgetting factor κ , Initial covariance matrix $\mathbf{L}_0$, Initial parameter estimates $\hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$ Error thresholds $\epsilon_{p_{threshold}},$ $\epsilon_{r_{threshold}}$	1.015, $10^4 \mathbf{I}_{7,7}$, $\mathbf{I}_{5,5}, \mathbf{0.1}_{5,2}$ 0.0001, 0.001
NN Controller	Learning rates $\Gamma_a, \Gamma_r, \Gamma_v, \lambda$ Number of layers Number of neurons	0.1, 0.3, 2, 1 1 30
Reference Model	$\xi_{dr}, \omega_{dr}, \tau$	0.5, 3 rad/s, 0.5

In Table 3.3, the covariance matrix is a diagonal where $\mathbf{L}_0 \in \mathbb{R}^{7 \times 7}$, the parameter vector $\hat{\mathbf{F}}_0 \in \mathbb{R}^{5 \times 5}$ is a diagonal unit matrix, while $\hat{\mathbf{G}}_0 \in \mathbb{R}^{5 \times 2}$ is a two-column vector, where all the matrix elements are set to 0.1.

Furthermore, for roll rate tracking, it is common practice to use a PI controller. In such applications, the derivative action is generally not required, as the controlled variable is already

a rate, and additional differentiation may amplify noise measurement without providing significant performance benefits. In the proposed architecture, the derivative gain was therefore set to zero, resulting in PI behaviour. The proportional and integral gains are selected with a negative sign to ensure proper feedback direction based on the adopted control law formulation.

3.3.2 Simulation for cruise at 35,000 ft and 290 knots

Figure 3.4 presents the simulation results for a cruise at an altitude of 35,000 ft and a calibrated airspeed CAS of 290 knots. The relative center-of-gravity is fixed at 25% and the aircraft weight m is set at 33,000 lb. The results demonstrate that the roll rate response effectively tracks the reference model defined by Eq.(3.36). The sideslip angle derivative is successfully damped to 0. Furthermore, the rudder and ailerons are operating simultaneously around their original positions, showing that the adaptive Aileron-Rudder Interconnect (ARI) has been well implemented.

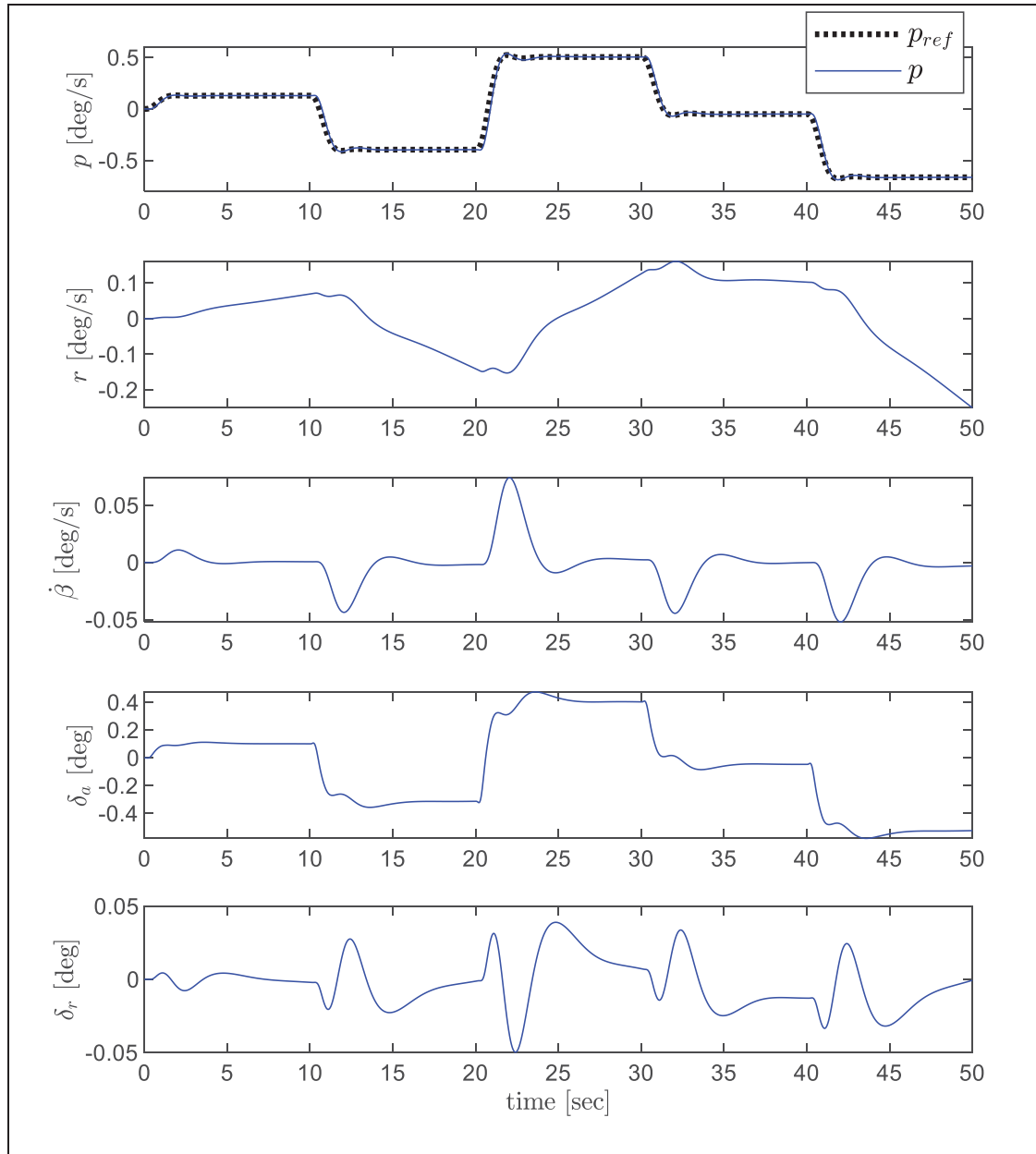


Figure 3.4 Lateral States and Control Responses during cruise

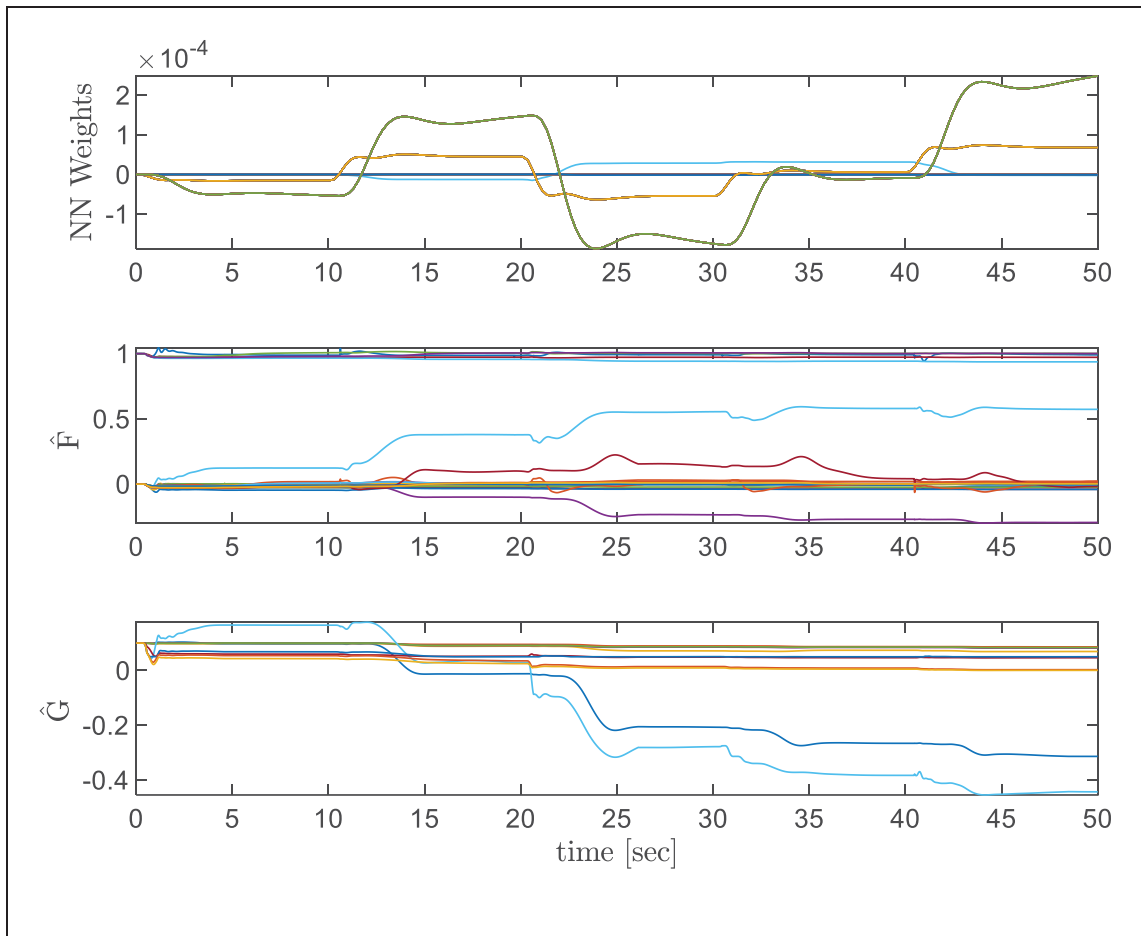


Figure 3.5 Controller Parameter Responses during cruise

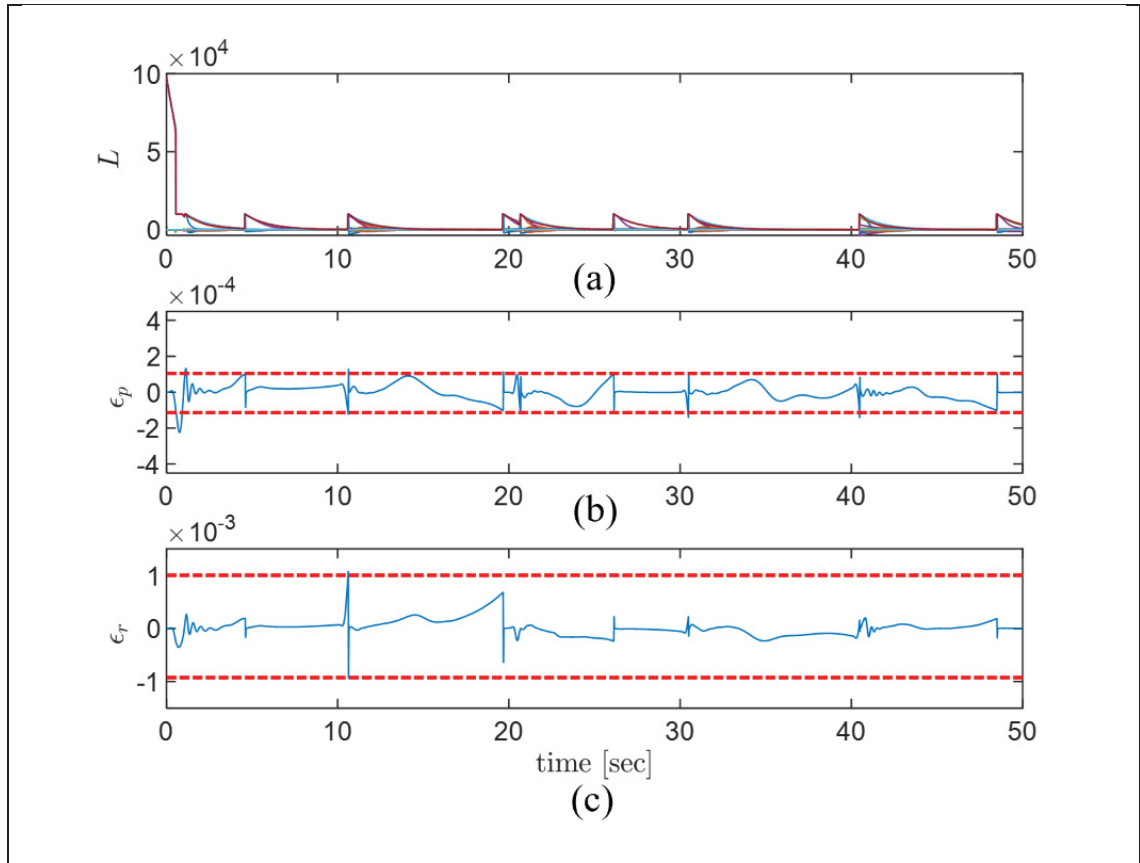


Figure 3.6 RLS parameters update : (a) covariance matrix $\mathbf{L}_k$ (b) roll rate prediction error and (c) yaw rate prediction error during cruise at 35,000 ft and 290 knots

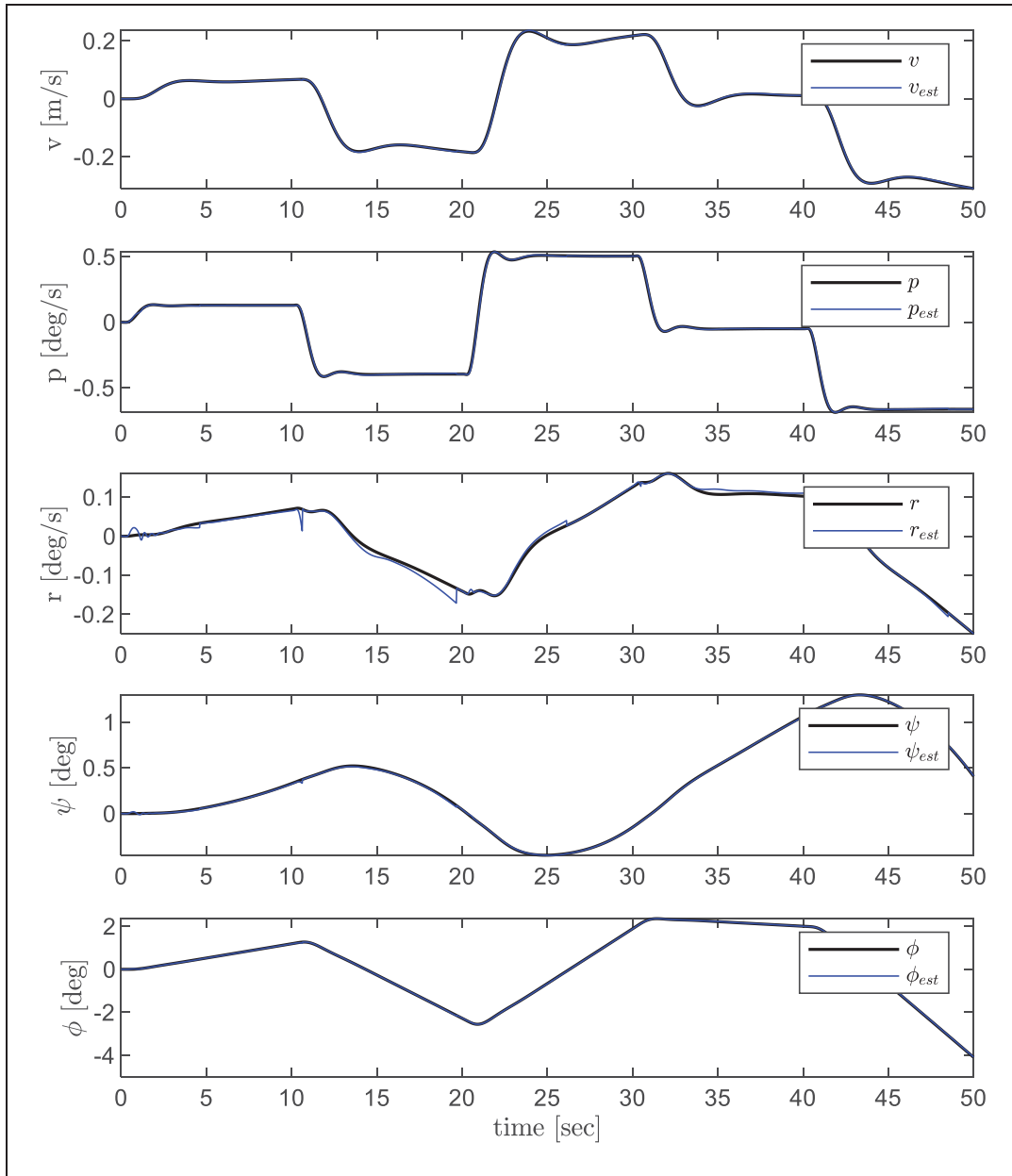


Figure 3.7 RLS Lateral states estimation during cruise

3.3.3 Validation over the flight envelope

3.3.3.1 Flight operating conditions

The control architecture was tested for 64 cruise flight scenarios covering the overall flight envelope of the Cessna Citation X. The altitude ranged from 5,000 ft to 40,000 ft with an

increment of 5,000 ft, and the calibrated airspeed CAS varied from 170 knots to 330 knots with an increment of 20 kts.

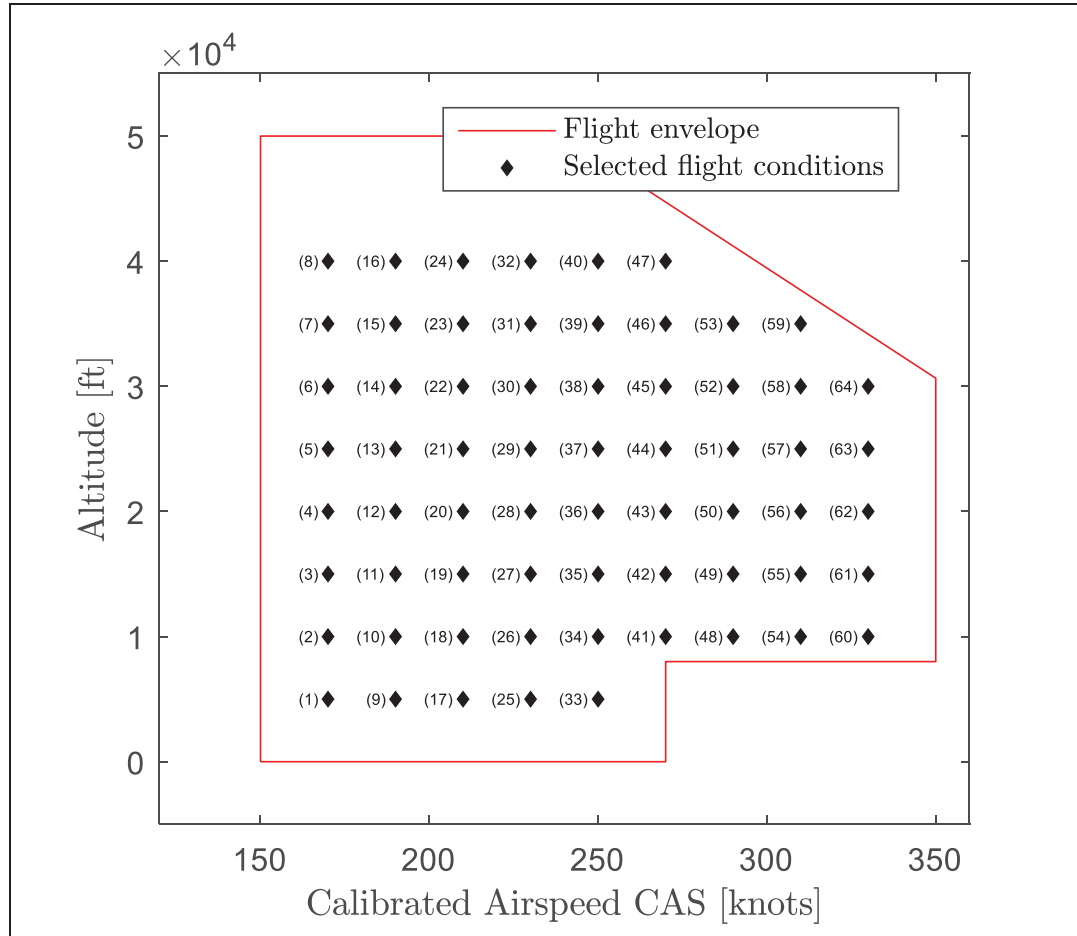


Figure 3.8 Selected cruise conditions for the CCX with mass $m = 33000$ lb and a center-of-gravity $x_{cg} = 25\%$

3.3.3.2 Learning rate and forgetting factor tuning

The neural network learning rates Γ_a and Γ_r and the RLS forgetting factor κ were tuned for the overall flight envelope. Figure 3.9 shows the controller response for 64 cruise conditions, with varying forgetting factor and learning rates. The input layer learning rate Γ_v was set to 2.

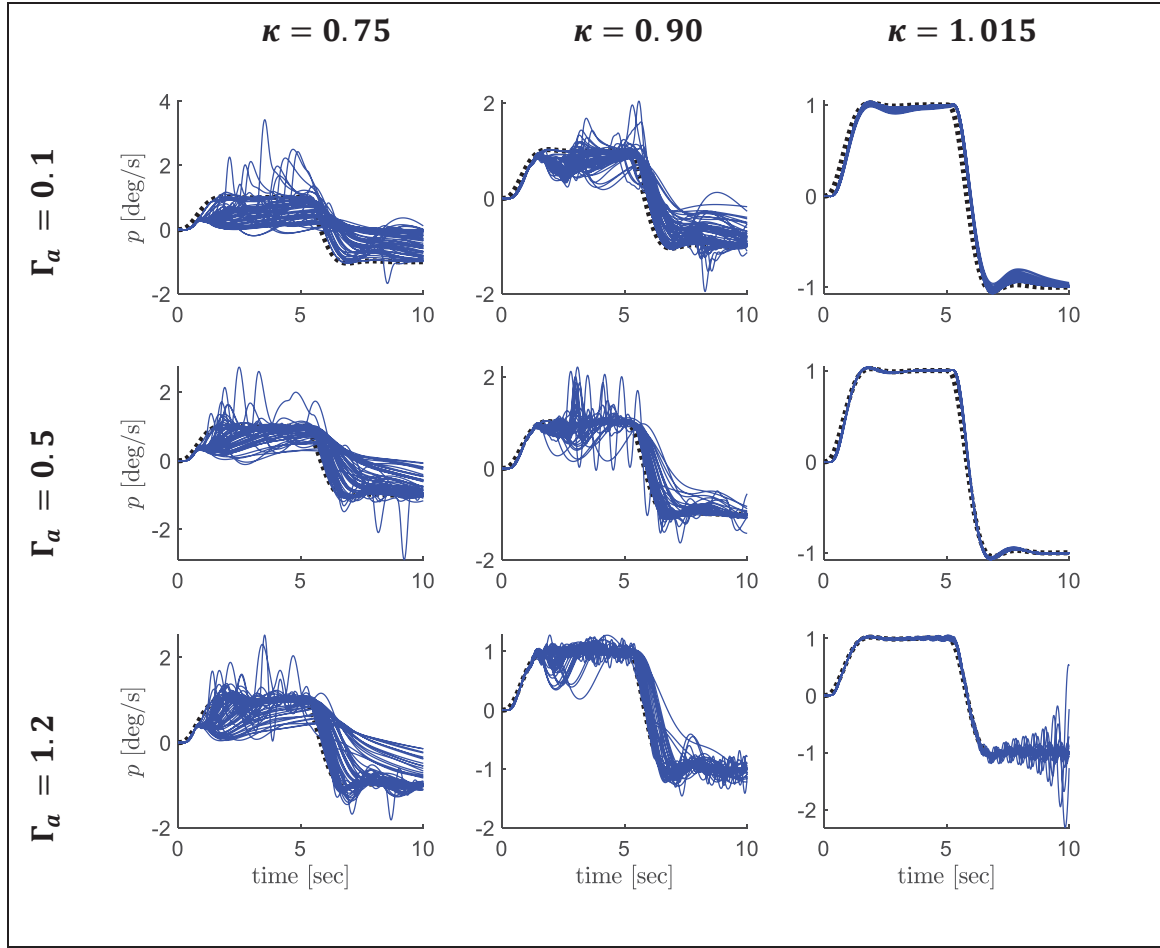


Figure 3.9 Roll Rate Response for varying Learning Rates and Forgetting Factor for 64 cruise conditions

3.3.3.3 Optimal tuning

Additional tuning was fulfilled in order to determine the optimal learning rates Γ_a and Γ_r . Disturbances were added to evaluate for robustness. Turbulences were modeled using the Dryden turbulent model available in Simulink for both light and moderate conditions. Wind gusts were modeled for both tailwind and headwind, while the vertical wind component w_{wind} was set to 0 m/s in all cases.

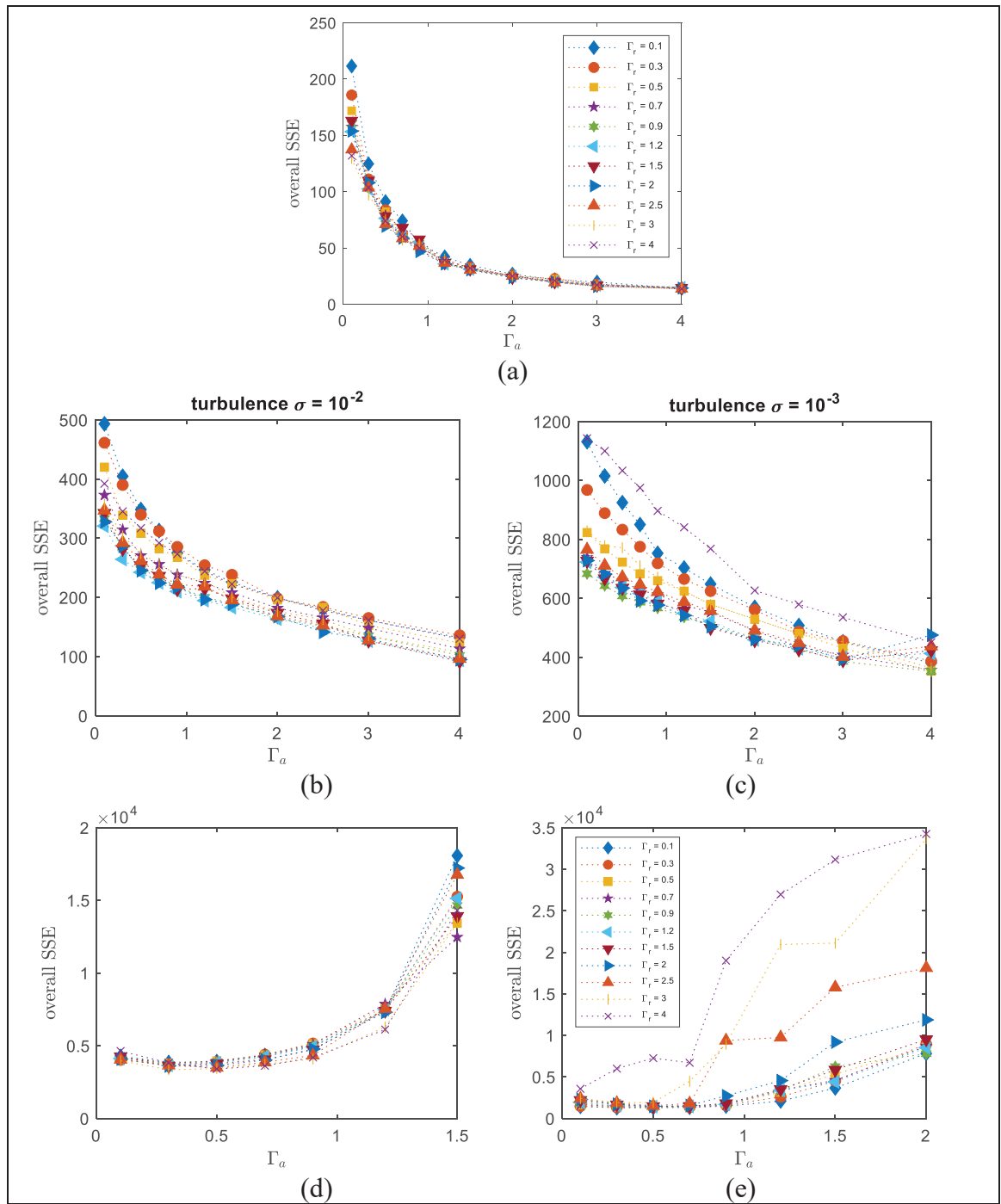


Figure 3.10 Neural Network Learning rate tuning under disturbances for 64 cruise conditions : (a) ideal condition (b) light turbulence (c) moderate turbulence (d) nose wind gust (e) tail wind gust

3.3.4 Controller configurations comparison

A comparison between the baseline PID controller and the proposed adaptive control architecture is presented in Figure 3.11. Different controller configurations are simulated in Figure 3.11, including a fixed PID controller, a PID – RLS dynamic inversion controller, a PID – RLS dynamic inversion – NN controller on the ailerons only, and a PID – RLS Inversion – NN controller on both ailerons and rudders, thus modelling the adaptive ARI. The results highlight the respective contributions of the RLS-based dynamic inversion and the neural network compensation. The inclusion of the RLS estimation improves tracking accuracy and robustness by adapting to variations in the aircraft dynamics across the flight envelope. Furthermore, the addition of the neural network provides an extra level of compensation for residual nonlinearities and modelling uncertainties, further enhancing tracking performance, particularly under nonlinear operating conditions and disturbances.

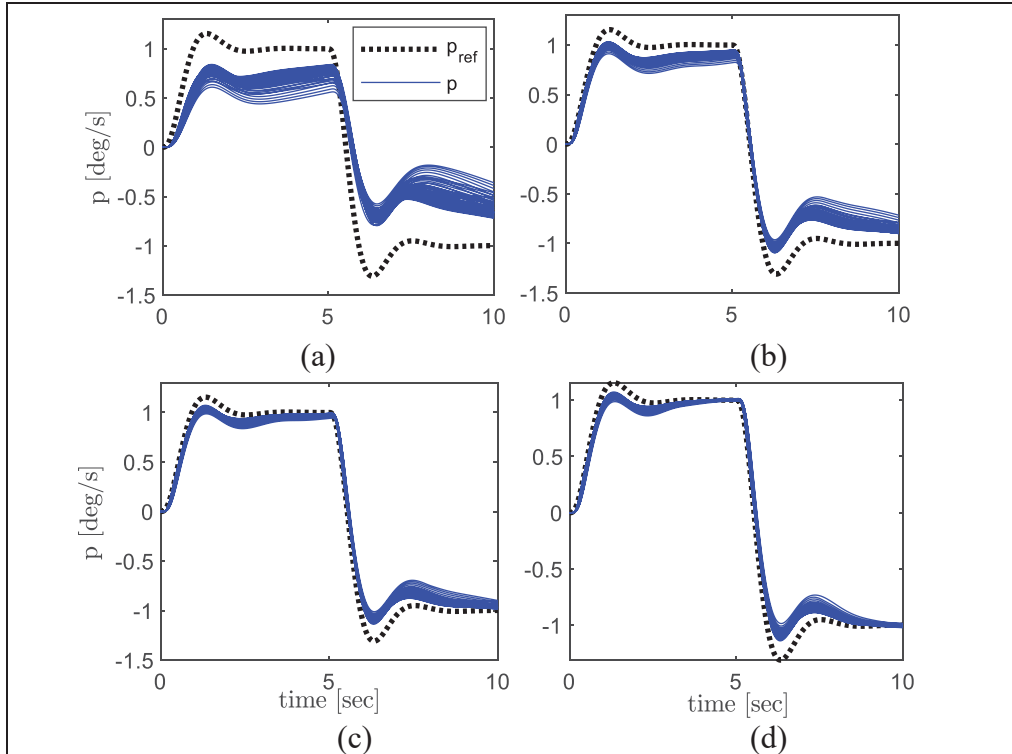


Figure 3.11 Controller configurations comparison for 64 cruise conditions :
 (a) fixed PID controller, (b) PID – RLS Inversion controller, (c) PID – RLS
 Inversion – NN controller on ailerons, (d) PID – RLS Inversion – NN
 controller on ailerons and rudders

3.3.5 Flight Qualities Validation

As stated above, the performances of linear systems and controllers are well established through standard flight-quality criteria (Department of Defense, 1997). However, no equivalent set of criteria currently exists for evaluating nonlinear systems. Nevertheless, the reference signal shown in Eq.(3.36) can be designed with precise requirements. System identification can then be applied to evaluate the controller transient response for all flight conditions.

Three reference signals were studied to demonstrate the adaptability of the lateral controller. The characteristics of these reference signals were chosen to be within Level 1 flight quality requirements. Table 3.4 summarizes the selected reference signals, for which the Dutch-roll damping ratio ξ_{dr} , the Dutch-roll frequency ω_{dr} and the roll time constant τ are given.

Table 3.4 Reference model signals: Level 1 dynamic characteristics

Reference model	Damping	Frequency, rad/s	Roll time constant, sec
	ξ_{dr}	ω_{dr}	τ
(a)	0.5	3	0.1
(b)	0.4	8	0.1
(c)	0.8	10	0.65

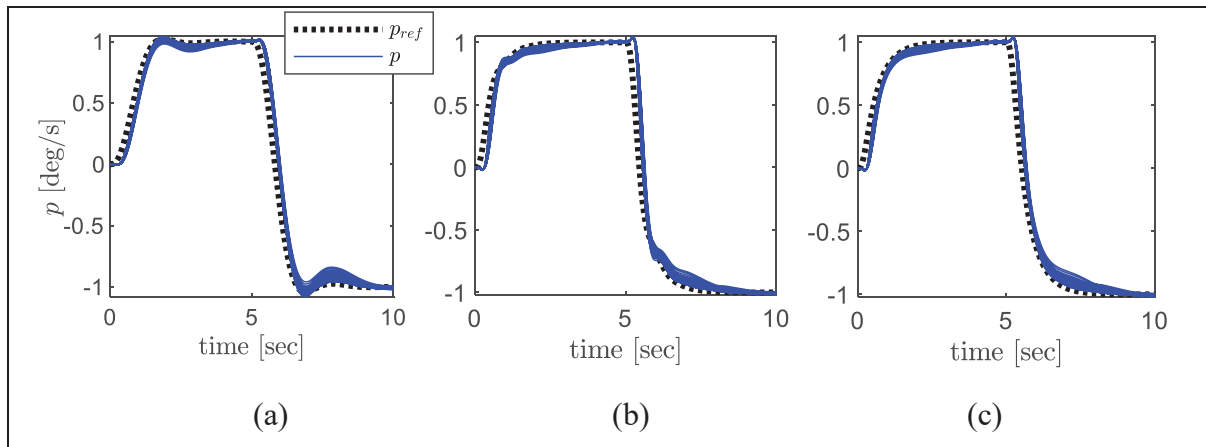


Figure 3.12 Roll rate response for three different roll rate reference signals in Table A III-3 for 64 cruise conditions

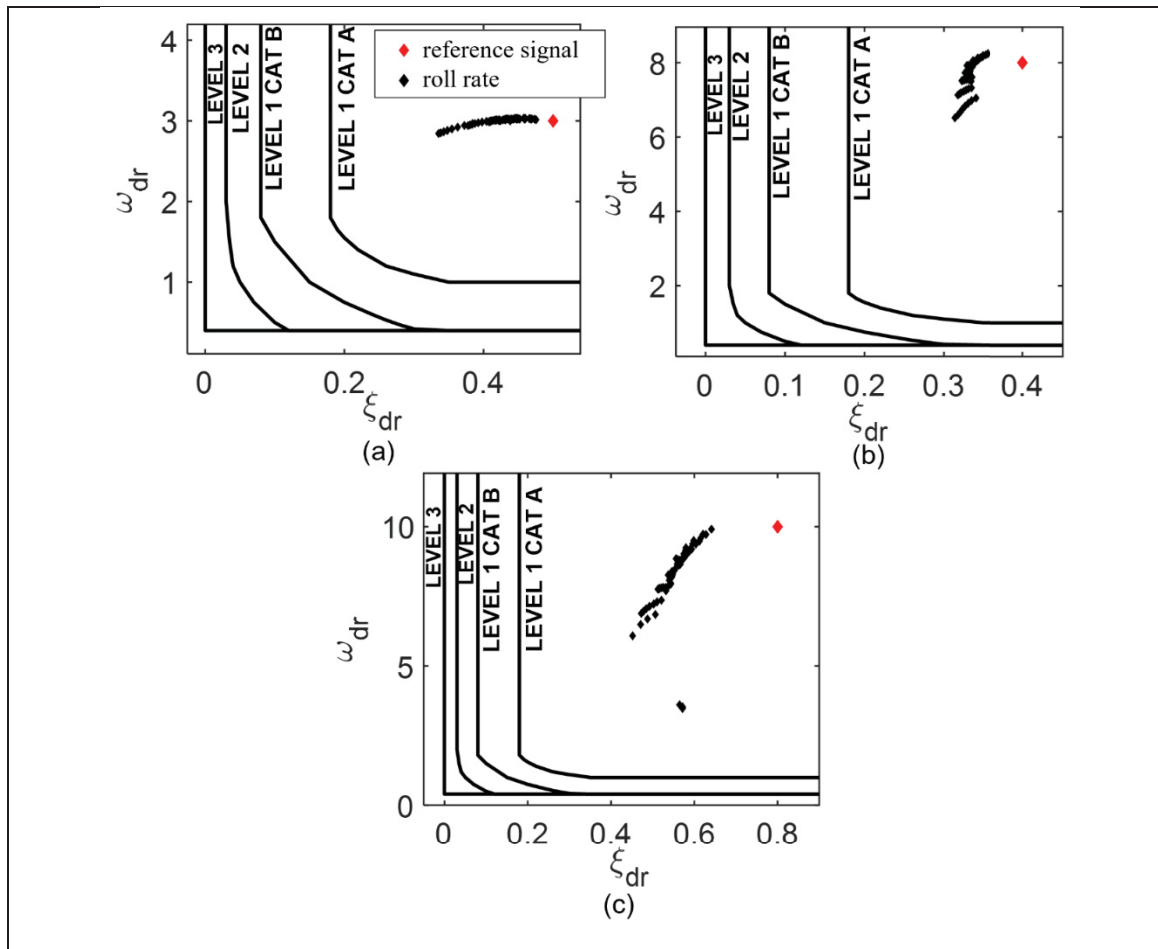


Figure 3.13 Dutch-roll flight quality assessment for three roll rate reference signals in Table A III-3 for 64 cruise conditions

The controller's roll rate transient response characteristics were extracted using curve fitting, with the performance metrics summarized in Table 3.5. An overall fitting error of less than 2% was registered for all flight conditions. The settling time is less than 3.5 sec for all flight conditions, and the rising time is less than 1.9 sec.

Table 3.5 Reference signal and ideal roll rate controller performances on unit signal tracking

	Damping	Frequency, rad/s	Roll time constant, sec	Rising time at 90%, sec	Settling time at $\pm 5\% p_{ss}$, sec	Overshoot, %
	ξ_{dr}	ω_{dr}	τ	T_r	T_s	—
Reference model p_{ref}	0.5	3	0.1	1.31	1.42	2.14
Reference signal tracking (a)	0.33–0.47	2.84–3.03	0.50–0.75	1.37–1.49	1.46–3.53	0–4.27

3.4 Robustness Validation

The robustness of the proposed control architecture is evaluated through simulation by analyzing its performance under disturbances, modelling uncertainties, and varying flight conditions across the cruise envelope. The quantitative assessment is based on standard performance metrics, including tracking errors, rising time, settling time, overshoot, and control effort.

3.4.1 Wind Gust and Turbulence Simulations

Wind gusts and turbulence were considered for linear velocity components, rather than the angular velocities. In the aircraft body frame, the wind linear velocity vector $\bar{V}_{wind}$ is defined as:

$$\bar{V}_{wind} = [u_{wind} \quad v_{wind} \quad w_{wind}]^T \quad (3.38)$$

where u_{wind} , v_{wind} , and w_{wind} represent the longitudinal, lateral, and vertical components, respectively. For lateral-control analysis, only u_{wind} and v_{wind} were considered. The vertical wind component w_{wind} was set to 0 m/s. Figure 3.14d shows the controller performance under

wind gust profile, while Figure 3.14c shows the controller performance under moderate turbulences.

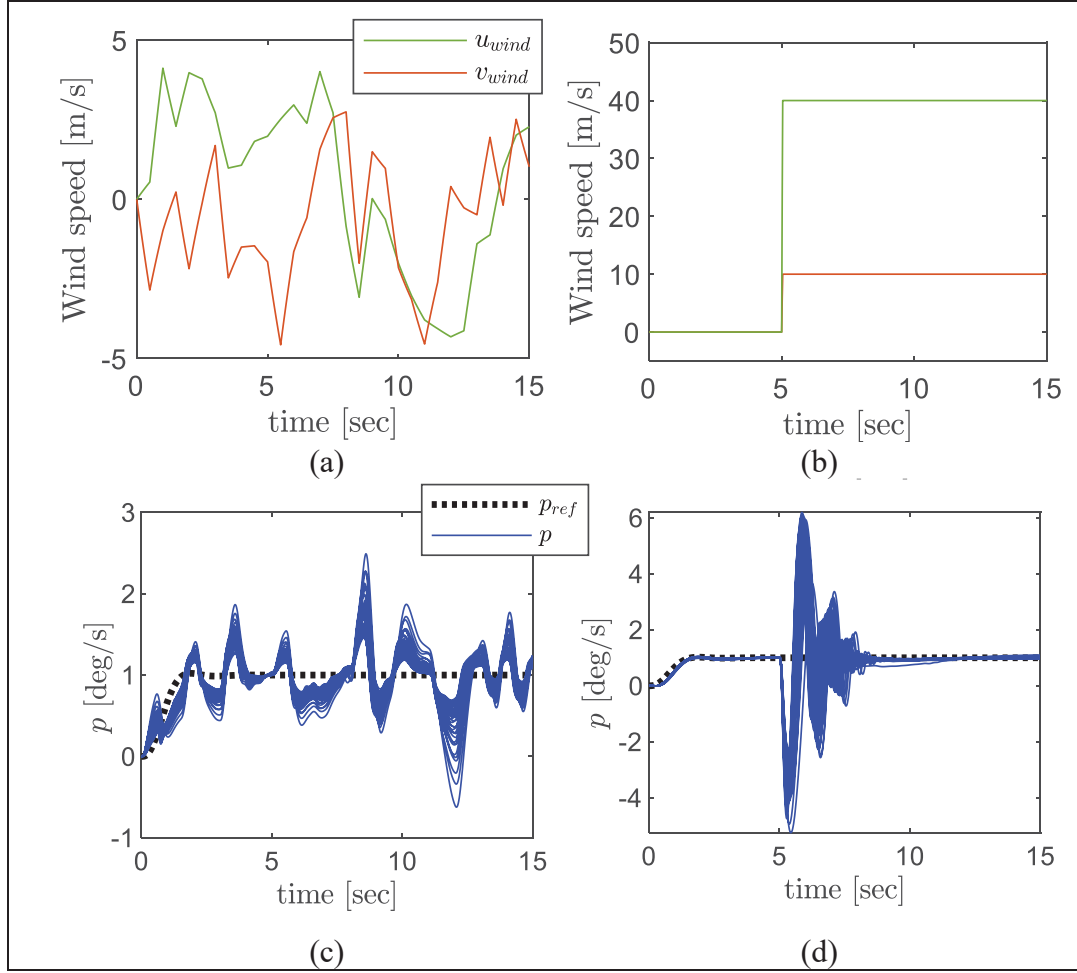


Figure 3.14 (a) Turbulence and (b) Wind Gust profiles and controller responses on roll rate for model reference signal tracking for 64 cruise conditions

3.4.2 Actuator Faults Simulations

The lateral controller was also tested under actuator faults. Such faults might be additive and/or multiplicative in nature, for example, partial loss of efficiency, or combined effects (Hussein et al., 2024 ; Yu, Zhang, Jiang, & Su, 2023). In this study, actuator Loss-of-Effectiveness (Figure 3.15a) and white noises as (Figure 3.15b). The effectiveness of the ailerons and rudders, represented by $C_l^{\delta a}$ and $C_l^{\delta r}$, were modified. These variables directly affect the

aerodynamic characteristics of the two control surfaces. The simulated failures consisted of adding bias and random multiplicative faults between 0 and 1. Despite these disturbances, the roll rate successfully maintained accurate tracking of the unit reference signal.

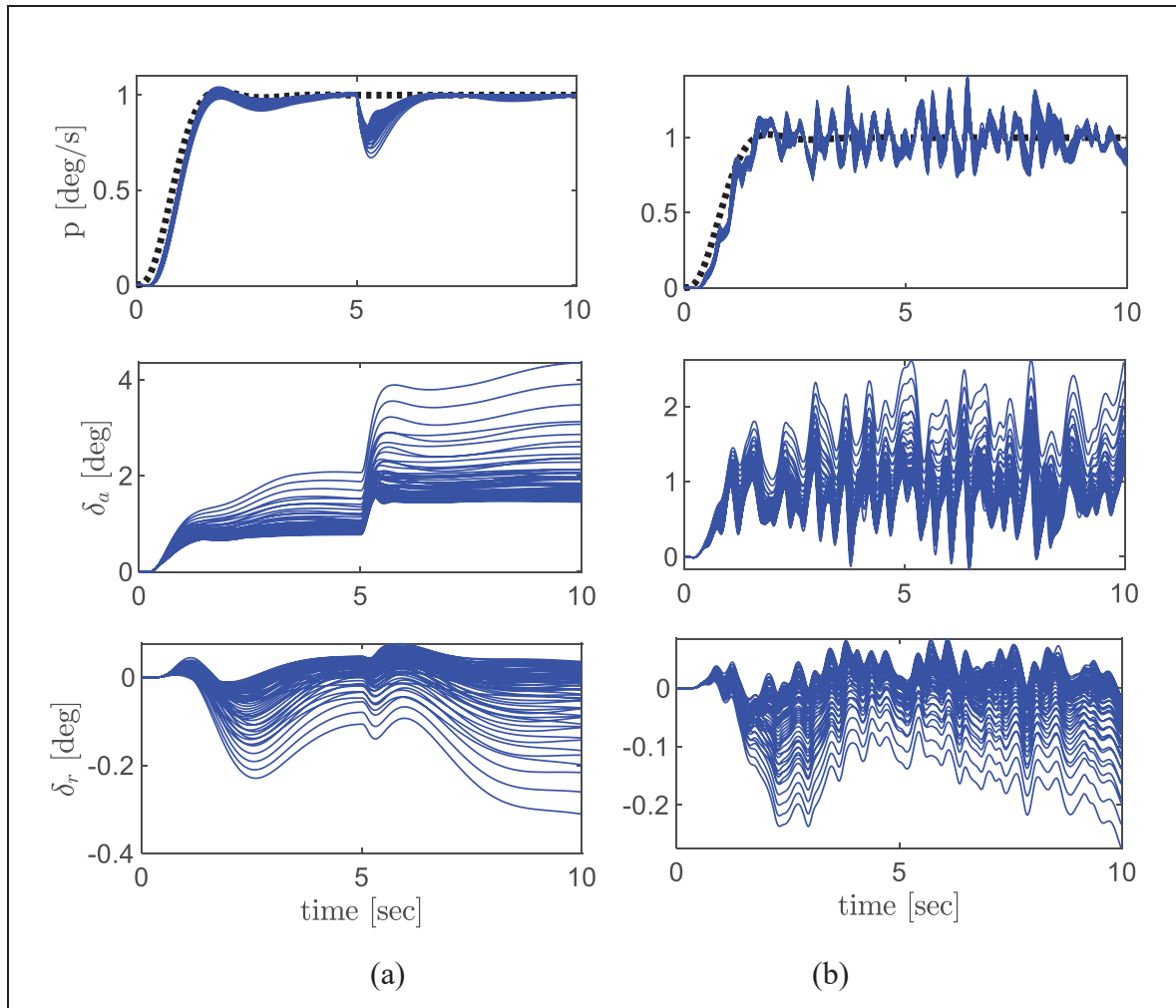


Figure 3.15 Lateral controller responses on roll rate for model reference signal tracking under (a) actuator Loss-of-Effectiveness and (b) actuator noise for 64 cruise conditions

3.5 Results discussion

The roll rate controller for the Cessna Citation X, shown in Figure 3.2, was successfully modelled and validated. The combined PID–RLS–NN structure ensured stability, convergence, and adaptability. The PID controller maintains stability, while the adaptive RLS

dynamic inversion and neural network addressed adaptability to model uncertainties at varying flight conditions. A Lyapunov stability analysis (Appendix) confirmed negative definiteness of the Lyapunov candidate derivative, ensuring boundedness of the tracking error and convergence of the control algorithm. Extensive parameter tuning (Table 3.3) demonstrated that the controller can adapt online to sudden changes in dynamics and disturbances. Across all configurations, no instability was observed.

3.5.1 Controller stability and convergence

The RLS inversion accurately estimated the control and state matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$. Convergence depended strongly on the covariance matrix $\mathbf{L}_k$, the forgetting factor κ and learning rates. Figure 3.5 illustrates the convergence of $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$, while Figure 3.7 confirms accurate estimation of lateral states for control purposes. The RLS estimates remained bounded throughout all simulation cases, and no divergence or numerical instability was observed, supporting the theoretical stability assumptions. Both the system states and the adaptive parameters remain bounded across all tested scenarios. In particular, the RLS-estimated parameters exhibited smooth and stable evolution without divergence or abrupt variations, indicating consistent convergence of the estimation process. Similarly, the neural network weights remain bounded and adapted progressively, confirming their role in compensating residual nonlinearities without destabilizing the closed-loop system.

In addition, all state variables, including roll and yaw rates, remain stable under disturbances and across the full set of flight conditions, with no evidence of oscillatory increase or instability. This observed boundedness of signals, and adaptive terms is consistent with the assumptions underlying the Lyapunov-based analysis and provides empirical support for the closed-loop stability of the proposed adaptive control architecture.

To mitigate potential singularities when computing the inverse term $\hat{\mathbf{G}}^{-1}$, from Eq. (3.35), bounding techniques were applied. Covariance matrix resetting was introduced to prevent numerical instabilities and ensure continued adaptation. Algorithm 1 resets the covariance

matrix $\mathbf{L}_k$ to a diagonal matrix $10^3 \mathbf{I}_{7,7}$ whenever the roll or yaw rate estimation errors exceed their thresholds, such that $|\epsilon_{p_k}| > 10^{-4}$ rad/s, or $|\epsilon_{r_k}| > 10^{-3}$ rad/s, respectively. As shown in Figure 3.6, this ensured stable convergence without degrading estimation accuracy.

The neural network controller adapted effectively throughout all simulations. Output weights $\mathbf{W}_a$ and $\mathbf{W}_r$ (on the order of 10^{-4}) remained significantly larger than the input weights $\mathbf{V}$ (on the order of 10^{-8}), as shown in Figure 3.5. These values remained stable, confirming correct implementation of the adaptation laws in Eq. (3.32).

3.5.2 Tuning process

The controller achieved precise tracking of multiple reference-model signals. Figure 3.12 shows representative roll rate tracking results over 10-second simulations. Across all 64 cruise conditions, the tracking error remained low, demonstrating robustness to varying operating points. Step response evaluations confirmed that the optimal configuration was obtained at $\kappa = 1.015$. As shown in Figure 3.9, higher learning rates improved accuracy but also introduced oscillations, indicating the need for thorough tuning. Figure 3.10 further illustrates the effect of turbulence and wind gusts on the sum of squared errors (SSE): while SSE initially decreased with increasing learning rates, overtraining effects caused degradation beyond the optimal point. The optimal configuration under disturbed conditions is summarized in Table 3.3.

3.5.3 Flight Quality Assessment

The controller was validated against Level 1 flight-quality (FQ) criteria defined in MIL-STD-1797A (Department of Defense, 1997). Roll and Dutch-roll dynamics were assessed across all 64 cruise conditions. The identified parameters — roll time constant τ , damping ratio ξ_{dr} and natural frequency ω_{dr} were obtained using nonlinear least square optimization (Figure 3.13). Fitting errors were consistently registered below 2%, confirming accurate system identification.

For Dutch-roll dynamics, the results closely matched the reference model (signal (a) in Table 3.4), with deviations observed for signals (b) and (c). Despite these differences, Level 1 requirements were achieved under all conditions (Figure 3.13). Step response evaluations (Table 3.5) further confirmed compliance with Category B cruise flight standards.

3.5.4 Controller robustness under disturbances

The controller demonstrated strong robustness to both external and internal disturbances. Under wind gusts, Dryden turbulence, and actuator faults, the lateral response remained stable and accurate (Figure 3.14). Simulations with actuator loss-of-effectiveness (LOE) in Figure 3.15 showed that the aileron and rudder deflections increased but did not saturate, ensuring continued controllability. This confirms that the proposed design maintains sufficient actuator margin under degraded conditions.

3.5.5 Computational Requirements

The computational requirements of the proposed controller remain compatible with those of real-time implementation. The PID controller adds negligible computational cost, while the RLS algorithm involves recursive updates of matrices of limited dimension, resulting in less demanding computational effort. The neural network consists of a single hidden layer with 30 neurons, leading to a limited number of arithmetic operations per control cycle. In addition, the simulations were implemented in MATLAB/Simulink and executed on a standard computing platform equipped with an Intel® Core™ i7-7700T CPU @ 2.90 GHz, demonstrating that the controller can be executed efficiently without requiring high-performance computing resources.

3.5.6 Future Work

The validation presented in this paper is limited to cruise flight conditions. However, the proposed controller has been evaluated under a wide range of nonlinearities, including aerodynamic variations across multiple flight conditions, external disturbances, and actuator

loss-of-effectiveness. These results provide a proof of concept for the effectiveness of the proposed adaptive control architecture on a business jet aircraft.

Although latency and time-delay effects may influence control performance, they are not explicitly considered in the present study, which focuses the model on validation within a high-fidelity nonlinear simulation environment. Future work will extend the proposed approach to other flight regimes, including high-angle-of-attack conditions and transient maneuvers, while investigating real-time implementation through hardware-in-the-loop simulations and experimental validation.

These developments will enable the assessment of computational constraints, latency effects, and system integration aspects, representing essential steps toward certification and deployment of adaptive control systems in safety-critical aerospace applications.

3.6 Conclusion

This paper presented a robust adaptive lateral control architecture for the Cessna Citation X, combining a baseline PID controller with Recursive Least Squares (RLS)-based dynamic inversion and neural network compensation. The objective is to improve tracking performance and robustness under nonlinear flight dynamics and varying operating conditions.

The simulation results demonstrate that the proposed control architecture achieves accurate roll-rate and yaw-rate tracking across the considered cruise flight envelope. The inclusion of RLS-based dynamic inversion enables adaptation to variations in system dynamics, while the neural network compensates for residual nonlinearities and modeling uncertainties. Compared to the baseline PID controller, the proposed approach provides improved tracking performance and enhanced robustness in the presence of disturbances and actuator loss-of-effectiveness. Across all tested scenarios, steady-state errors remain below 0.1%. The transient response is characterized by fast convergence, with settling times typically below 2 seconds and minimal

overshoot. The controller also maintains stable performance under external disturbances and actuator loss-of-effectiveness scenarios, with no degradation in tracking stability.

Control inputs remain within actuator limits, with deflections typically confined within $\pm 3^\circ$ around trim conditions and no saturation observed, indicating efficient control effort.

The closed-loop system stability was assessed using a Lyapunov-based analysis, and the simulation results support the boundedness of the signals and stable behaviour of the adaptive elements throughout the tested scenarios. The validation was conducted over 64 representative cruise flight conditions, demonstrating consistent performance without the need for gain scheduling. Although flight quality criteria are traditionally defined for linear systems, the results indicate that the proposed nonlinear adaptive controller can achieve performance consistent with Level 1 flying qualities within the tested conditions.

3.7 Appendix

This section assesses the controller's stability and convergence using Lyapunov criteria for adaptive systems. Adaptive systems are asymptotically stable if, for a given positive definite candidate $\mathcal{L}$, there exists its time derivative $\dot{\mathcal{L}}$ that is negative definite (Lavretsky & Wise, 2024), such that:

$$\mathcal{L} \geq 0 \text{ and } \dot{\mathcal{L}} \leq 0 \quad (3.39)$$

3.7.1 Roll rate and Yaw rate error dynamics

To analyze the stability of the adaptive controller, the closed-loop error dynamics are derived by substituting the control laws of Eqs. (3.17) and (3.19), together with the neural-network compensation terms introduced in Eq.(3.29), into the roll-rate and yaw-rate dynamics of Eq. (3.13). This yields the roll-rate error dynamics in Eq. (3.40). Since the roll dynamics are

coupled with the yaw channel through the rudder action and adaptive compensation terms, the yaw-related contribution is isolated in Eq. (3.41) to simplify the Lyapunov analysis.

Let us express the system error dynamics. Let $\dot{e}_p$ be the roll rate error dynamics, and $\dot{e}_r$ be the yaw rate error dynamics. Transforming Eq.(3.13) into discrete time, namely into Δp_{k+1} and Δr_{k+1} , and applying the control laws for $\Delta\delta_{a_k}$ and $\Delta\delta_{r_k}$ in Eq.(3.28), it is possible to derive the roll rate error dynamics $\dot{e}_p$ in Eq.(3.15) as:

$$\begin{aligned} \dot{e}_p = & (1 - K_{FF})\dot{p}_{ref} - \left(K_P + \frac{K_I}{S} + K_D S \right) e_p + \mathbf{C}_1 \widehat{\mathbf{G}}_{2k} \Delta\delta_{r_{k-1}} + \mathbf{W}_a^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \\ & + \mathbf{C}_1 \mathbf{G}_{2k} (\mathbf{C}_2 \widehat{\mathbf{G}}_{2k})^{-1} \left(-\frac{K_{WF} \cdot \tau S}{\tau S + 1} r + \mathbf{C}_2 \widehat{\mathbf{F}}_k \Delta \mathbf{x}_k + \mathbf{C}_2 \widehat{\mathbf{G}}_{1k} \Delta\delta_{a_{k-1}} \right. \\ & \left. + \mathbf{W}_r^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \right) \end{aligned} \quad (3.40)$$

Considering $\mathbf{M}_k = \mathbf{C}_1 \mathbf{G}_{2k} (\mathbf{C}_2 \widehat{\mathbf{G}}_{2k})^{-1}$, from Eq.(3.40), the yaw-channel contribution to the coupled error dynamics can be grouped into the term $\dot{e}_r$, defined as follows:

$$\dot{e}_r = \mathbf{M}_k \left(-\frac{K_{WF} \cdot \tau S}{\tau S + 1} r + \mathbf{C}_2 \widehat{\mathbf{F}}_k \Delta \mathbf{x}_k + \mathbf{C}_2 \widehat{\mathbf{G}}_{1k} \Delta\delta_{a_{k-1}} + \mathbf{W}_r^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \right) \quad (3.41)$$

where $\mathbf{M}_k$ is obtained from the RLS estimation and is a known scalar value.

Equation (3.41) does not introduce a separate independent system, but rather isolates the yaw-channel contribution embedded in Eq. (3.40). This separation is used only to simplify the boundedness and convergence analysis of the coupled lateral-directional dynamics.

3.7.2 Disturbance dynamics

Let us consider that the optimal weights $\mathbf{V}$ of the neural network contain some optimization error $\tilde{\mathbf{V}}$. The same logic applies to $\mathbf{W}_a$ and $\mathbf{W}_r$, so that:

$$\mathbf{V} = \hat{\mathbf{V}} + \tilde{\mathbf{V}} \quad ; \quad \mathbf{W}_a = \hat{\mathbf{W}}_a + \tilde{\mathbf{W}}_a \quad ; \quad \mathbf{W}_r = \hat{\mathbf{W}}_r + \tilde{\mathbf{W}}_r \quad (3.42)$$

where $\hat{\mathbf{W}}_a$, $\hat{\mathbf{W}}_r$ and $\hat{\mathbf{V}}$ are known weight values, while $\tilde{\mathbf{W}}_a$, $\tilde{\mathbf{W}}_r$ and $\tilde{\mathbf{V}}$ are unknown errors. The optimal weights $\mathbf{W}_a$, $\mathbf{W}_r$ and $\mathbf{V}$ are unknown.

Let us express the Taylor series approximation for a tangent sigmoid function σ around the current argument $\mathbf{V}^T \bar{\mathbf{x}}$ as:

$$\sigma(\mathbf{V}^T \bar{\mathbf{x}}) = \sigma(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) + \sigma'(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) \tilde{\mathbf{V}} \bar{\mathbf{x}} + \mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}}) \quad (3.43)$$

where σ' is the tangent sigmoid gradient, and $\mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}})$ is the unknown higher-order term. In the next equations, we make the following changes:

$$\sigma(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) = \hat{\sigma} \quad \text{and} \quad \sigma'(\hat{\mathbf{V}}^T \bar{\mathbf{x}}) = \hat{\sigma}' \quad (3.44)$$

The yaw rate error dynamics in Eq.(3.41) can then be expressed as follows:

$$\dot{e}_r = \mathbf{M}_k \left[-\frac{K_{WF} \cdot \tau_S}{\tau_S + 1} r + \mathbf{C}_2 \hat{\mathbf{F}}_k \Delta \mathbf{x}_k + \mathbf{C}_2 \hat{\mathbf{G}}_{1k} \Delta \delta_{a_{k-1}} + \hat{\mathbf{W}}_r^T \hat{\sigma} + \tilde{\mathbf{W}}_r^T \hat{\sigma} + \hat{\mathbf{W}}_r^T \hat{\sigma}' \tilde{\mathbf{V}} \bar{\mathbf{x}} + \mathbf{d}_{rk} \right] \quad (3.45)$$

where $\mathbf{d}_{rk}$ is the yaw rate disturbance term, so that:

$$\mathbf{d}_{rk} = \mathbf{M}_k (\tilde{\mathbf{W}}_r^T \hat{\sigma}' \tilde{\mathbf{V}} \bar{\mathbf{x}} + \mathbf{W}_r^T \mathcal{O}^2) \quad (3.46)$$

We should note that the linear component with respect to $\tilde{\mathbf{W}}_a$, $\tilde{\mathbf{W}}_r$ and $\tilde{\mathbf{V}}$ can be approximated with proper tuning of the learning rates (Narendra & Annaswamy, 1987 ; Rysdyk & Calise, 2005). The remaining higher-order terms in Eq.(3.41), containing two or more unknown values, are considered as disturbances.

Applying the notation in Eq.(3.44) into Eq.(3.40), the roll rate error dynamic becomes:

$$\begin{aligned}
\dot{e}_p = & (1 - K_{FF})\dot{p}_{ref} - \left(K_P + \frac{K_I}{S} + K_D S\right) e_p + \mathbf{C}_1 \widehat{\mathbf{G}}_{2\mathbf{k}} \Delta \delta_{r_{k-1}} + \widehat{\mathbf{W}}_a^T \widehat{\boldsymbol{\sigma}} \\
& + \mathbf{M}_k \left[-\frac{K_{WF} \cdot \tau S}{\tau S + 1} r + \mathbf{C}_2 \widehat{\mathbf{F}}_k \Delta \mathbf{x}_k + \mathbf{C}_2 \widehat{\mathbf{G}}_{1\mathbf{k}} \Delta \delta_{a_{k-1}} + \widehat{\mathbf{W}}_r^T \widehat{\boldsymbol{\sigma}} + \widetilde{\mathbf{W}}_r^T \widehat{\boldsymbol{\sigma}} \right. \\
& \left. + \widetilde{\mathbf{W}}_r^T \widehat{\boldsymbol{\sigma}}' \widetilde{\mathbf{V}} \bar{\mathbf{x}} \right] + \mathbf{d}_{p_k}
\end{aligned} \tag{3.47}$$

where $\mathbf{d}_{p_k}$ is the roll rate disturbance, which can be expressed as follows, by applying the same logic:

$$\mathbf{d}_{p_k} = \mathbf{W}_a^T \mathcal{O}^2 + \widetilde{\mathbf{W}}_a^T \widehat{\boldsymbol{\sigma}}' \widetilde{\mathbf{V}} \bar{\mathbf{x}} + \mathbf{d}_{r_k} \tag{3.48}$$

Replacing Eq.(3.46) into Eq.(3.48), we obtain:

$$\mathbf{d}_{p_k} = (\widetilde{\mathbf{W}}_a^T + \mathbf{M}_k \widetilde{\mathbf{W}}_r^T) \widehat{\boldsymbol{\sigma}}' \widetilde{\mathbf{V}} \bar{\mathbf{x}} + (\mathbf{W}_a^T + \mathbf{M}_k \mathbf{W}_r^T) \mathcal{O}^2 \tag{3.49}$$

3.7.3 Disturbance boundedness

As seen in Eq.(3.47), the roll rate error dynamics $\dot{e}_p$ depend upon the yaw rate error dynamics $\dot{e}_r$, and from Eq.(3.48), the roll rate disturbance $\mathbf{d}_{p_k}$ includes the yaw rate disturbance $\mathbf{d}_{r_k}$. The elements composing the disturbances $\mathbf{d}_{p_k}$ and $\mathbf{d}_{r_k}$ consists of the higher-order terms of the tangent sigmoid function in the neural network hidden layers. The value of $\mathbf{M}_k$ is obtained from the RLS estimation, and is bounded, as will be shown in the next subsection. Lewis et al. in (Lewis et al., 1995) assessed the disturbance boundedness for a tangent sigmoid function. They showed that the disturbance $\mathbf{d}_{r_k}$ can be bounded as given next:

$$\|\mathbf{d}_r\| \leq C_{00} + C_{01} \|\widetilde{\mathbf{V}}\| + C_{02} \|\widetilde{\mathbf{V}}\| \|e_r\| \tag{3.50}$$

where C_{00} , C_{01} and C_{02} are computable positive constants (Lewis et al., 1995). The same logic applies for the roll rate disturbance $\mathbf{d}_{p_k}$.

3.7.4 Lyapunov candidate and its derivative

The disturbance terms were bounded in the previous subsection. Let us now assess the stability of the overall system, by choosing the following Lyapunov candidate $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{2} e_p e_p + \frac{1}{2} \tilde{\Theta} \mathbf{L}^{-1} \tilde{\Theta} + \frac{1}{2} \tilde{\mathbf{W}}_a \Gamma_a^{-1} \tilde{\mathbf{W}}_a + \frac{1}{2} \tilde{\mathbf{W}}_r \Gamma_r^{-1} \tilde{\mathbf{W}}_r + \frac{1}{2} \tilde{\mathbf{V}} \Gamma_V^{-1} \tilde{\mathbf{V}} \quad (3.51)$$

where $\tilde{\Theta}$ is the RLS optimization error defined as:

$$\tilde{\Theta} = \Theta^* - \hat{\Theta} \quad (3.52)$$

with Θ^* , the optimal RLS parameter, and $\hat{\Theta}$, the estimated RLS parameter.

From Eq.(3.51), the candidate derivative $\dot{\mathcal{L}}$ can be written as:

$$\dot{\mathcal{L}} = \dot{e}_p e_p + \dot{\tilde{\Theta}} \mathbf{L}^{-1} \tilde{\Theta} + \dot{\tilde{\mathbf{W}}}_a \Gamma_a^{-1} \tilde{\mathbf{W}}_a + \dot{\tilde{\mathbf{W}}}_r \Gamma_r^{-1} \tilde{\mathbf{W}}_r + \dot{\tilde{\mathbf{V}}} \Gamma_V^{-1} \tilde{\mathbf{V}} \quad (3.53)$$

Replacing the neural network weight update from Eq.(3.32) and the expression of $\dot{e}_p$ from Eq.(3.47) into Eq.(3.53), the Lyapunov candidate derivative becomes :

$$\begin{aligned} \dot{\mathcal{L}} = & \left((1 - K_{FF}) \dot{p}_{ref} - \left(K_P + \frac{K_I}{S} + K_D S \right) e_p + \mathbf{C}_1 \hat{\mathbf{G}}_{2k} \Delta \delta_{r_{k-1}} + \hat{\mathbf{W}}_a^T \hat{\sigma} + \tilde{\mathbf{W}}_a^T \hat{\sigma} \right. \\ & + \hat{\mathbf{W}}_a^T \hat{\sigma}' \tilde{\mathbf{V}} \bar{\mathbf{x}} \\ & + \mathbf{M}_k \left[-\frac{K_{WF} \cdot \tau S}{\tau S + 1} r + \mathbf{C}_2 \hat{\mathbf{F}}_k \Delta \mathbf{x}_k + \mathbf{C}_2 \hat{\mathbf{G}}_{1k} \Delta \delta_{a_{k-1}} + \hat{\mathbf{W}}_r^T \hat{\sigma} \right] \\ & + \mathbf{d}_{p_k} \Big) e_p + \dot{\tilde{\Theta}} \mathbf{L}^{-1} \tilde{\Theta} \\ & + \left(-[(\hat{\sigma} - \hat{\sigma}' \tilde{\mathbf{V}}^T \bar{\mathbf{x}}) e_p + \lambda \|e_p\| \hat{\mathbf{W}}_a] \Gamma_a^{-1} \tilde{\mathbf{W}}_a \right. \\ & + \left(-[(\hat{\sigma} - \hat{\sigma}' \tilde{\mathbf{V}}^T \bar{\mathbf{x}}) e_r + \lambda \|e_r\| \hat{\mathbf{W}}_r] \Gamma_r^{-1} \tilde{\mathbf{W}}_r \right. \\ & \left. \left. + \left(-\Gamma_V [\bar{\mathbf{x}} e_p \hat{\mathbf{W}}_a^T \hat{\sigma}' + \bar{\mathbf{x}} e_r \hat{\mathbf{W}}_r^T \hat{\sigma}' + \lambda \|e_p + e_r\| \hat{\mathbf{V}}] \right) \Gamma_V^{-1} \tilde{\mathbf{V}} \right) \right) \end{aligned} \quad (3.54)$$

where the yaw rate error is chosen to minimize the sideslip derivative, such that $e_r = -\dot{\beta}$.

In order to assess the stability of the controller, the stability of the RLS inversion and of the NN controller are assessed separately.

3.7.5 RLS controller stability

First, the RLS controller stability is assessed. The stability of the RLS estimation relies on the boundedness of the covariance matrix and the regression signals. Under standard assumptions of bounded input/output signals and sufficient excitation, the estimation error remains bounded and $\hat{\Theta}$ converges to a neighborhood of the true parameters Θ^* . In the proposed control architecture, these conditions are satisfied supported by due to the bounded nature of the aircraft states and control inputs. Furthermore, assuming that the RLS state estimation is performed correctly, the RLS algorithm contributes to accurate local model estimation, while the overall closed-loop system stability is ensured through the Lyapunov-based control design as follows.

Let's consider the RLS Lyapunov candidate $\mathcal{L}_{\text{RLS}}$ and its derivative $\dot{\mathcal{L}}_{\text{RLS}}$ corresponding to the RLS adaptive filter, such that:

$$\mathcal{L}_{\text{RLS}} = \frac{1}{2} \tilde{\Theta} \mathbf{L}^{-1} \tilde{\Theta} \quad \text{and} \quad \dot{\mathcal{L}}_{\text{RLS}} = \tilde{\Theta} \mathbf{L}^{-1} \dot{\tilde{\Theta}} \quad (3.55)$$

where $\tilde{\Theta}$ is defined in Eq.(3.52), and $\mathbf{L}^{-1}$ is the inverse covariance matrix. $\mathcal{L}_{\text{RLS}}$ is here positive definite. The optimization error $\tilde{\Theta}$ can be expressed for two consecutive time steps, $k - 1$ and k , as follows:

$$\begin{aligned} \tilde{\Theta}_{k-1} &= \Theta^* - \hat{\Theta}_{k-1} \\ \tilde{\Theta}_k &= \Theta^* - \hat{\Theta}_k \end{aligned} \quad (3.56)$$

Recalling Eq.(3.26), $\hat{\Theta}_k$ can be expressed in a compact form as:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + \mathbf{K}_k \epsilon_k \quad (3.57)$$

where $\mathbf{K}_k$ is the Kalman gain vector as defined below:

$$\mathbf{K}_k = \frac{\mathbf{L}_{k-1} \mathbf{X}_k}{\kappa + \mathbf{X}_k \mathbf{L}_{k-1} \mathbf{X}_k} \quad (3.58)$$

From Eq.(3.57), the derivative $\dot{\tilde{\boldsymbol{\Theta}}}$ can be computed in discrete time as follows:

$$\dot{\tilde{\boldsymbol{\Theta}}} = \tilde{\boldsymbol{\Theta}}_k - \tilde{\boldsymbol{\Theta}}_{k-1} = -\mathbf{K}_k \boldsymbol{\epsilon}_k \quad (3.59)$$

Replacing Eq.(3.59) into Eq.(3.55), $\dot{\mathcal{L}}_{\text{RLS}}$ can be expressed as:

$$\dot{\mathcal{L}}_{\text{RLS}} = -\tilde{\boldsymbol{\Theta}} \mathbf{L}^{-1} \mathbf{K}_k \boldsymbol{\epsilon}_k \quad (3.60)$$

The values of $\tilde{\boldsymbol{\Theta}}$ can be positive or negative depending on the sign of $\boldsymbol{\Theta}^* - \hat{\boldsymbol{\Theta}}_k$. However, the norm of $\tilde{\boldsymbol{\Theta}}$ should decrease and tend to zero as $\hat{\boldsymbol{\Theta}}_k$ tends to $\boldsymbol{\Theta}^*$. In addition, the norm of the estimation error $\boldsymbol{\epsilon}_k$ in Eq.(3.60) also decreases.

The sign of $\boldsymbol{\epsilon}_k$ may still fluctuate between positive and negative values, but these fluctuations get shorter as $\hat{\boldsymbol{\Theta}}_k$ tends to $\boldsymbol{\Theta}^*$. The sign of the Kalman gain vector $\mathbf{K}_k$ may also fluctuate in order to minimize both estimation error $\boldsymbol{\epsilon}_k$ and optimization error $\tilde{\boldsymbol{\Theta}}$. However, we can consider that $\mathbf{K}_k$ is a vector that scales the correction applied to the parameter estimates rather than focusing solely on its sign, as shown in Eq.(3.57).

The inverse covariance matrix $\mathbf{L}^{-1}$ is positive definite, and so the definiteness of Eq.(3.60) relies on the signs of $\mathbf{K}_k$, $\boldsymbol{\epsilon}_k$ and $\tilde{\boldsymbol{\Theta}}$. Therefore, in order to guarantee that the RLS Lyapunov candidate derivative $\dot{\mathcal{L}}_{\text{RLS}}$ is negative definite, $\mathbf{K}_k$ is computed dynamically through the RLS equation update so that the product $\mathbf{K}_k \boldsymbol{\epsilon}_k$ aligns with the direction of $\tilde{\boldsymbol{\Theta}}$, leading to a positive dot product. However, the main role of the Kalman gain vector $\mathbf{K}_k$ consists of minimizing the

estimation error ϵ_k rather than enforcing a predefined direction. This would indicate that $\mathcal{L}_{\text{RLS}}$ decreases over time and tends to 0, and that the norm $\|\dot{\mathcal{L}}_{\text{RLS}}\|$ in Eq.(3.60) is bounded by a positive constant C_6 , and thus, $\mathcal{L}_{\text{RLS}}$ tends to zero over time, such that:

$$\|\dot{\mathcal{L}}_{\text{RLS}}\| \leq C_6 \quad (3.61)$$

Therefore, the RLS algorithm converges to stable values.

3.7.6 Boundedness and Convergence

The overall boundedness and definiteness of $\hat{\mathcal{L}}$ in Eq.(3.54) is assessed in this subsection. This analysis considers the assumptions, theorems and demonstrations stated in (Lewis et al., 1995).

ASSUMPTION 1: The norms of the neural network weights are bounded with known positive values, so that :

$$\|\mathbf{W}_a\| < W_m \quad \|\mathbf{W}_r\| < W_m, \text{ and } \|\mathbf{V}\| < V_m \quad (3.62)$$

ASSUMPTION 2: The desired trajectory is bounded, so that:

$$\left\| \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{bmatrix} \right\| \leq Q_d \quad (3.63)$$

where $Q_d \in \mathbb{R}$ is a known constant.

For each time t , the neural network input $\mathbf{x}$ is bounded as follows:

$$\|\mathbf{x}\| \leq c_1 Q_d + c_2 \|e_p\| \quad (3.64)$$

where c_1 and c_2 are computable positive constants.

THEOREM 1: The higher order term of the tangent sigmoid function may be bounded as follows:

$$\|\mathcal{O}^2(\tilde{\mathbf{V}}^T \bar{\mathbf{x}})\| \leq c_3 + c_4 Q_d \|\tilde{\mathbf{V}}\| + c_5 \|\tilde{\mathbf{V}}\| \|e_p\| \quad (3.65)$$

where c_3 , c_4 and c_5 are computable positive constants.

ASSUMPTIONS 3: The norms of the tangent sigmoid function and its gradient are known functions; therefore, they can be considered as constants at a given step time t :

$$\|\hat{\sigma}\| = c_{01} \quad \text{and} \quad \|\hat{\sigma}'\| = c_{02} \quad (3.66)$$

ASSUMPTIONS 4: $\mathbf{M}_k$ is computed using the RLS algorithm step time t , and is also considered as a constant value:

$$\begin{aligned} \mathbf{M}_k &= c_{11} \\ (1 + c_{11}) &= c_{12} \end{aligned} \quad (3.67)$$

Finally, let us define two norm properties:

THEOREM 2: The triangle inequalities are given as:

$$\|a - b\| \leq \|a\| + \|b\| \quad \text{and} \quad \|a + b\| \leq \|a\| + \|b\| \quad (3.68)$$

THEOREM 3 : The generalized Cauchy–Schwarz inequality is given as:

$$\|ab\| \leq \|a\| \cdot \|b\| \quad (3.69)$$

Using the above assumptions and theorems, the Lyapunov candidate derivative $\dot{\mathcal{L}}$ in Eq.(3.54) can be bounded as follows:

$$\begin{aligned}
\dot{\mathcal{L}} \leq & (1 - K_{FF})\dot{p}_{ref}e_p - \left(K_P + \frac{K_I}{s} + K_D s\right)e_p e_p - c_{11} \frac{K_{WF} \tau s}{\tau s + 1} r e_p \\
& + \mathbf{C}_1 \widehat{\mathbf{G}}_{2\mathbf{k}} \Delta \delta_{r_{k-1}} e_p + c_{11} \mathbf{C}_2 \widehat{\mathbf{F}}_{\mathbf{k}} \Delta \mathbf{x}_k e_p + c_{11} \mathbf{C}_2 \widehat{\mathbf{G}}_{1\mathbf{k}} \Delta \delta_{a_{k-1}} e_p \\
& + \|\dot{\mathcal{L}}_{\text{RLS}}\| \\
& + (c_{12} c_{01} W_m + V_m c_{02} c_1 Q_d \|\tilde{\mathbf{W}}_a\| + c_{02} c_1 Q_d W_m c_{11} \|\tilde{\mathbf{V}}\| \\
& + c_{02} c_1 Q_d \|\tilde{\mathbf{V}}\| \|\tilde{\mathbf{W}}_a^T\| + c_{12} W_m c_3 + c_{12} W_m c_4 Q_d \|\tilde{\mathbf{V}}\|) \|e_p\| \\
& + (c_{01} \|\tilde{\mathbf{W}}_r\| + V_m c_{02} c_1 Q_d \|\tilde{\mathbf{W}}_r\| + c_{02} c_1 Q_d W_m \|\tilde{\mathbf{V}}\|) \|e_r\| \\
& + (V_m \mathbf{C}_2 c_2 \|\tilde{\mathbf{W}}_a\| + c_{02} c_2 c_{11} W_m \|\tilde{\mathbf{V}}\| + c_{02} c_2 \|\tilde{\mathbf{V}}\| \|\tilde{\mathbf{W}}_a^T\| \\
& + c_{12} W_m c_5 \|\tilde{\mathbf{V}}\|) \|e_p\|^2 \\
& + (V_m \mathbf{C}_2 c_2 \|\tilde{\mathbf{W}}_r\| + c_{02} c_2 W_m \|\tilde{\mathbf{V}}\|) \|e_p\| \|e_r\| \\
& - \lambda \|e_p\| (W_m \|\tilde{\mathbf{W}}_a\| - \|\tilde{\mathbf{W}}_a\|^2) - \lambda \|e_r\| (W_m \|\tilde{\mathbf{W}}_r\| - \|\tilde{\mathbf{W}}_r\|^2) \\
& - \lambda (\|e_p\| + \|e_r\|) (V_m \|\tilde{\mathbf{V}}\| - \|\tilde{\mathbf{V}}\|^2)
\end{aligned} \tag{3.70}$$

Let us assess the boundedness of $\dot{\mathcal{L}}$ in Eq.(3.70) by verifying the Lyapunov stability requirements in Eq.(3.39). From Eq.(3.70), we can define the term $\dot{\mathcal{L}}_2$, such that:

$$\dot{\mathcal{L}}_2 = \mathbf{C}_1 \widehat{\mathbf{G}}_{2\mathbf{k}} \Delta \delta_{r_{k-1}} e_p + c_{11} \mathbf{C}_2 \widehat{\mathbf{F}}_{\mathbf{k}} \Delta \mathbf{x}_k e_p + c_{11} \mathbf{C}_2 \widehat{\mathbf{G}}_{1\mathbf{k}} \Delta \delta_{a_{k-1}} e_p \tag{3.71}$$

where $\dot{\mathcal{L}}_2$ is computed from the RLS approximation. As stated earlier, the RLS parameters are estimated in order to reduce the tracking error e_p . The previous commands on the rudders $\Delta \delta_{r_{k-1}}$ and ailerons $\Delta \delta_{a_{k-1}}$ are also outputted to reduce the current error e_p . Furthermore, it has been demonstrated that the RLS Lyapunov derivative $\dot{\mathcal{L}}_{\text{RLS}}$ is negative definite, so the RLS Lyapunov candidate $\mathcal{L}_{\text{RLS}}$ is decreasing, and the norm $\|\dot{\mathcal{L}}_{\text{RLS}}\|$ is bounded as in Eq.(3.61), and thus, the RLS filter is stable and converging (3.60). Therefore, the derivative $\dot{\mathcal{L}}_2$ in Eq.(3.71) also tends to zero over time, reducing the roll rate tracking error e_p . In addition, we can state that $\|\dot{\mathcal{L}}_2\|$ norm is also bounded by a positive constant C_{22} , so that:

$$\|\dot{\mathcal{L}}_2\| \leq C_{22} \tag{3.72}$$

Let us define the term $\dot{\mathcal{L}}_1$ from Eq.(3.70) such that:

$$\dot{\mathcal{L}}_1 = (1 - K_{FF})\dot{p}_{ref}e_p - \left(K_P + \frac{K_I}{s} + K_D s\right)e_p e_p - c_{11} \frac{K_{WF} \cdot \tau s}{\tau s + 1} r e_p \quad (3.73)$$

We should note that with a proper choice of the constant gains, the terms in Eq.(3.73) are negative definite. Finally, let us assess the boundedness of the roll rate error e_p with a case-by-case study by applying the condition in Eq.(3.39).

3.7.6.1 Case 1

The neural network approximation is performed properly, so the error norms $\|\tilde{\mathbf{W}}_a\|$, $\|\tilde{\mathbf{W}}_r\|$ and $\|\tilde{\mathbf{V}}\|$ are small and tend to 0. The roll rate error may then be bounded as follows:

$$\|e_p\| \leq \frac{(1 - K_{FF})\dot{p}_{ref} - c_{11} \frac{K_{WF} \cdot \tau s}{\tau s + 1} r + C_6 + (c_{12}c_{01}W_m + c_{12}W_m c_3)}{\left(K_P + \frac{K_I}{s} + K_D s\right)} \quad (3.74)$$

where C_6 is a positive constant.

3.7.6.2 Case 2

The neural network approximation is performed poorly, and so $\|\tilde{\mathbf{W}}_a\|$, $\|\tilde{\mathbf{W}}_r\|$ and $\|\tilde{\mathbf{V}}\|$ are large. Considering that the yaw rate control objective is fulfilled, such that $\|e_r\|$ tends to 0, and stating that the approximation errors $\|\tilde{\mathbf{W}}_a\|$, $\|\tilde{\mathbf{W}}_r\|$ and $\|\tilde{\mathbf{V}}\|$ remain bounded by positive values, such that:

$$\|\tilde{\mathbf{W}}_a\| < W_m ; \quad \|\tilde{\mathbf{W}}_r\| < W_m ; \quad \text{and} \quad \|\tilde{\mathbf{V}}\| < V_m \quad (3.75)$$

Then, the upper bound on $\|e_p\|$ can be found as follows:

$$\begin{aligned} \|e_p\| < \frac{(1 - K_{FF})\dot{p}_{ref} - c_{11} \frac{K_{WF} \tau S}{\tau S + 1} r + C_{01} - C_6 - \lambda (W_m \|\tilde{\mathbf{W}}_a\| - \|\tilde{\mathbf{W}}_a\|^2)}{\left(K_P + \frac{K_I}{S} + K_D S\right) - C_{02}} \\ + \frac{+\lambda (V_m \|\tilde{\mathbf{V}}\| - \|\tilde{\mathbf{V}}\|^2)}{\left(K_P + \frac{K_I}{S} + K_D S\right) - C_{02}} \end{aligned} \quad (3.76)$$

3.7.6.3 Case 3

The neural network approximation is performed poorly, and so $\|\tilde{\mathbf{W}}_a\|$, $\|\tilde{\mathbf{W}}_r\|$ and $\|\tilde{\mathbf{V}}\|$ are large. Considering that the roll rate control objective is fulfilled, such that $\|e_p\|$ tends to 0, then the inequation becomes:

$$\begin{aligned} \|e_r\| & \left((c_{01} \|\tilde{\mathbf{W}}_r\| + V_m c_{02} c_1 Q_d \|\tilde{\mathbf{W}}_r\| + c_{02} c_1 Q_d W_m \|\tilde{\mathbf{V}}\|) \right. \\ & \left. - \lambda (W_m \|\tilde{\mathbf{W}}_r\| - \|\tilde{\mathbf{W}}_r\|^2) - \lambda (V_m \|\tilde{\mathbf{V}}\| - \|\tilde{\mathbf{V}}\|^2) \right) \leq 0 \end{aligned} \quad (3.77)$$

Eq.(3.77) is true if λ is chosen as follows:

$$\lambda \geq \frac{(c_{01} \|\tilde{\mathbf{W}}_r\| + V_m c_{02} c_1 Q_d \|\tilde{\mathbf{W}}_r\| + c_{02} c_1 Q_d W_m \|\tilde{\mathbf{V}}\|)}{(W_m \|\tilde{\mathbf{W}}_r\| - \|\tilde{\mathbf{W}}_r\|^2) + (V_m \|\tilde{\mathbf{V}}\| - \|\tilde{\mathbf{V}}\|^2)} \quad (3.78)$$

Choosing a positive λ according to Eq.(3.78) ensures the robustness of the neural networks and the negative definiteness of the term $-\lambda (W_m \|\tilde{\mathbf{W}}_r\| - \|\tilde{\mathbf{W}}_r\|^2) - \lambda (V_m \|\tilde{\mathbf{V}}\| - \|\tilde{\mathbf{V}}\|^2)$.

In Cases 1 and 2, the roll rate error is bounded by an upper bound. This upper bound can be defined with proper choices of the fixed gains, namely K_{FF} , K_{WF} , K_P , K_I and K_D . As seen in Eq.(3.74), the roll rate error also depends on the yaw rate, dampened by a coefficient K_{WF} and scaled with a washout filter. Also, in Eqs.(3.74) and (3.76), the upper bound on $\|e_p\|$ is small if the PID gains K_P , K_I and K_D are chosen high enough.

In Cases 2 and 3, in addition to selecting the gains K_{FF} , K_{WF} , K_P , K_I and K_D , the error $\|e_p\|$ remains bounded with a proper choice of parameter λ , ensuring the robustness of the NN controller. In any case, the overall stability of the controller relies on the PID linear controller. Therefore, thorough tuning of and compromise on the PID gains should be performed in order to ensure the controller stability and robustness.

CHAPTER 4

ADAPTIVE AUTOPILOT DESIGN AND IMPLEMENTATION FOR THE CESSNA CITATION X

Rojo Princy Andrianantara ^a, Georges Ghazi^a, and Ruxandra Botez ^a, Hugo Roger, Louis Partaix, Daniel Mancera Coyotl

^a Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE), System Engineering Department, École de technologie supérieure, 1100 Rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3

Article submitted for publication in *MDPI Aerospace*, March 2026
<https://doi.org/10.3390/aerospace13040318>

Résumé

Dans le cadre de ce travail, deux pilotes automatiques adaptatifs ont été développés pour l'avion d'affaires Cessna Citation X. Ces stratégies de contrôle adaptatif sont appliquées garantissant une double adaptation, notamment grâce à un contrôleur à inversion dynamique et un contrôleur à réseau de neurones. L'objectif du contrôle consiste à suivre la vitesse verticale, l'altitude et les commandes de cap de l'avion. L'inversion dynamique est appliquée à chaque variable de sortie, puis le contrôleur à réseau de neurones est mis à jour à l'aide d'une loi adaptative, dérivée de la rétropropagation. L'inversion dynamique a été réalisée localement à l'aide d'un algorithme récursif des moindres carrés comme estimation d'état. La boucle de contrôle interne, qui comprend le contrôle du taux de tangage, du taux de roulis et du taux de lacet, a été intégré dans les pilotes automatiques. Les états longitudinaux ont été séparés des états latéraux afin de différencier le contrôle longitudinal du contrôle latéral. Des tests de robustesse ont été effectués dans des conditions de turbulence et de rafales de vent. Les résultats des pilotes automatiques ont été comparés aux données réelles des essais en vol d'un simulateur de vol de recherche de l'avion d'affaire Cessna Citation X. Les résultats ont montré que les pilotes automatiques étaient parfaitement capables de commander la vitesse verticale, l'altitude et le cap de l'avion, avec des signaux de référence donnés. La comparaison des essais

en vol a montré que les contrôleurs adaptatifs proposés étaient meilleurs que ceux actuellement embarqués à bord du Cessna Citation X.

Abstract

This paper presents the development of two adaptive autopilots for the Cessna Citation X business jet aircraft. The two adaptive control strategies, including a dynamic inversion controller and a neural network controller, provide dual adaptation. The control objective consists of tracking the vertical speed, altitude, and heading commands. The dynamic inversion is applied on each output variable, and then the Neural Network (NN) controller was updated using adaptive law, derived from backpropagation. Dynamic Inversion (DI) is achieved locally using a Recursive Least Square (RLS) algorithm for state estimation. An inner control loop for the pitch, roll and yaw rates are integrated within the autopilots. The longitudinal states were separated from the lateral states in order to differentiate between longitudinal and lateral control. Robustness tests were conducted under turbulence and wind gust conditions. The autopilot results were compared with flight simulation data from a Cessna Citation X research flight simulator. Results have shown that the autopilots accurately track the vertical speed, altitude and heading reference signals. The flight simulation comparison has shown that the proposed adaptive controllers were better than the one currently on board of the Cessna Citation X.

4.1 Introduction

Autonomous flight capabilities are becoming increasingly critical as aircraft manufacturers strive to reduce pilot workload and decrease computational demands. Autopilot systems enable aircraft to accurately follow predefined trajectories and perform maneuvers such as altitude changes or coordinated turns. As a result, the development of advanced flight control systems has become a central focus in modern aviation research, with the objective of improving aircraft performance and safety. In this scope, one of the main safety concerns in aviation is the Loss of Control In-Flight (LOC-I), which remains one of the major causes of fatal accidents. According to the Federal Aviation Administration, over 60% to 80% of these

accidents are attributed to human error (Federal Aviation Administration, 2009 ; Wiegmann & Shappell, 2001). Therefore, enhancing the autonomy and robustness of flight control systems through software and hardware improvements presents a cost-effective alternative to aerodynamic redesigns.

In general, Flight Control Systems (FCS) serve as the aircraft's "brain", managing subsystems to ensure safe operation. Current FCS architectures typically include Stability Augmentation Systems (SAS) in order to dampen oscillations, and Control Augmentation Systems (CAS), which increase aircraft controllability (Stevens et al., 2015). The SAS and CAS combination is denoted as the Control Stability Augmentation System or CSAS. Actual onboard autopilots include position angle control, airspeed control and load factor control, as well as advanced control architectures such as C^* controllers (R. P. Andrianantara, Mancera Coyotl, et al., 2024), which are especially relevant for high-speed vehicle control such as the Space Shuttle.

As a remainder, traditional autopilots largely depend on linear control techniques such as classic Proportional – Integral – Derivative (PID) controllers and modern Linear Quadratic Regulator (LQR) controllers (Stevens et al., 2015). Although certified and reliable, these linear methods highly depend on gain scheduling and accurate linear mathematical models (Reynolds et al., 1996). In fact, gain scheduling requires extensive effort, and linear control often neglects significant nonlinearities of complex aircraft dynamics (Stevens et al., 2015). Advanced linear methods, such as Linear Quadratic Gaussian (LQG) (Ferrier et al., 2018) and H_∞ Controllers (Efremov et al., 2021 ; A. Khalil & Fezans, 2021) address further robustness issues, but they still rely on linearized models that limit overall performance. In order to overcome the limitations of linear controllers, adaptive control strategies such as Dynamic Inversion (DI) (Horn, 2019 ; Moncayo et al., 2012 ; Steffensen et al., 2023) and Model Predictive Control (MPC) (Doff-Sotta et al., 2025) have gained high interest. DI control directly inverts system dynamics in order to compute control inputs, but it can suffer from zero-crossing singularities, while MPC controller (R. P. Andrianantara, Ghazi, et al., 2024) optimizes control inputs over prediction horizons. However, these two recent techniques also require reliable linear models.

In order to address these linearization issues, recent advances have shown the growing use of online estimation filters, such as Recursive Least Squares (RLS) (Haykin, 2013 ; Mahadi et al., 2022 ; Mohseni & Bernstein, 2022), Kalman Filtering (KF) (Xiaoqian et al., 2022), and intelligent methods, such as Fuzzy Logic (Grigorie, Botez, et al., 2012a ; Hashemi et al., 2020 ; S. M. Hosseini, Ghazi, et al., 2024a), and Neural Networks (Emami et al., 2022 ; Ge et al., 2013 ; Liu et al., 2025). These approaches enable real-time adaptive estimation of system dynamics, allowing controllers to locally linearize the flight dynamics and adapt to uncertainties. Hybrid architectures (Pedro & Meyer, 2007) combining traditional control with Neural Networks (NN) or Reinforcement Learning (RL) are promising, although challenges still remain in terms of data volume requirements, training times, and certification.

Recent studies have also investigated advanced adaptive nonlinear and fault-tolerant control strategies to improve controller robustness. Some research proposed a fast fixed-time incremental backstepping controller (Y. Li et al., 2025) and sliding mode controller (Dai, Feng, Zhao, Cui, & Wang, 2022) combined with a disturbance observer. Other works have (Ma, Liu, & Huang, 2025) explored robust nonlinear control based on adaptive or observer-based approaches. These approaches mainly focus on fault-tolerant stabilization of aircraft dynamics or damage recovery scenarios in the presence of model uncertainties and disturbances. In contrast, the present paper focuses on adaptive outer-loop autopilot design, combining dynamic inversion with online Recursive Least Squares (RLS) estimation and neural network compensation, which greatly improves trajectory tracking and robustness under varying flight conditions, and thus reducing dependence to gain scheduling.

Despite the positive results of new approaches, certification of adaptive controllers remains a significant hurdle. Regulatory organizations, such as the FAA, EASA, and NASA (Agogino et al., 2024 ; European Union Aviation Safety Agency, 2024 ; Federal Aviation Administration, 2024) require adaptive systems to have constrained adaptive elements and backup classical controllers due to potential non-deterministic behaviour. To date, adaptive controllers have primarily been deployed in unmanned aerial systems and simulation environments (Geronel, Botez, & Bueno, 2023 ; Yu et al., 2023). However, adaptive autopilots applied to high-level

trajectory tracking tasks (Yue et al., 2023), as in this paper, such as altitude command, vertical speed command, and heading command, offer a safer path toward the real application of adaptive control. These outer loop autopilots are designed by following tracking performance, robustness tests and validation.

4.1.1 Research Objectives

This paper presents the design and implementation of adaptive autopilots for vertical speed, altitude, and heading control, specifically adapted to the Cessna Citation X business jet. The control architecture integrates PID controllers, Dynamic Inversion, and Neural Network (NN)-based adaptation laws to satisfy precise control objectives. The heading autopilot includes an adaptive roll controller employing an NN-based Aileron-Rudder Interconnect (ARI) to enhance yaw damping. The adaptive autopilot incorporates validated inner loop controllers for pitch, roll, and yaw rates damping (R. P. Andrianantara et al., 2023a ; Quintin et al., 2024). These inner loop controllers, previously validated using flight quality criteria across the full flight envelope, achieve Level 1 flight quality for short-period, roll, and Dutch-roll dynamic modes.

The resulting autopilots demonstrate robustness against modelling uncertainties and eliminate the need for gain scheduling, enabling a single controller configuration to cover the validated cruise envelope for the Cessna Citation X. The design emphasizes minimizing complexity to maintain system reliability, which is in line with best practices in aircraft control design.

4.1.2 Related Work

Previous research has explored various adaptive controllers for the Cessna Citation X, including Fuzzy Logic (S. M. Hosseini, Ghazi, et al., 2024a), recurrent neural networks, Model Predictive Control (R. P. Andrianantara, Ghazi, et al., 2024), Dynamic Inversion control (R. P. Andrianantara et al., 2023a ; Quintin et al., 2024), and Neural Network-based controllers. Robust control approaches such as H_∞ and LQG (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Ghazi & Botez, 2015b) have also been studied for longitudinal and lateral dynamic models.

These designs are generally validated using sensitivity analysis focused on stability, performance, and tracking accuracy. Also, the highly nonlinear C^* parameter has been successfully controlled on the Cessna Citation X flight dynamics using dynamic inversion (R. P. Andrianantara, Mancera Coyotl, et al., 2024) and neural network techniques, thus improving model flight qualities. Other research from NASA's Intelligent Flight Control System project (N. Nguyen, 2011 ; Park et al., 2021) demonstrated that adaptive neural networks and dynamic inversion methods can be implemented on experimental aircraft, confirming their practical applicability.

4.2 Validation methods

The autopilots are validated in this paper using an accurate MATLAB/Simulink flight dynamic model derived from high-fidelity simulation data (Ghazi & Botez, 2015a) obtained from a Cessna Citation X Research Aircraft Flight Simulator (RAFS). The aircraft's cruise envelope is studied and tested for altitude capture, vertical speed control and coordinated turn maneuvers. Robustness tests under Dryden turbulence and wind gust disturbances confirm the controllers' ability to maintain its objectives.

A modified Recursive Least Squares filter is employed with covariance matrix updates, triggered by threshold-based estimation errors. Additional logic prevents instability due to divisions by zero by enforcing minimum limits and absolute value corrections. A parameter optimization process accounts for environmental disturbances to enhance controller robustness, evaluating model performance using Sum Squared Errors (SSE) metric across simulations.

The proposed autopilots achieve precise tracking of reference commands for vertical speed, altitude, and heading, including dual-objective scenarios. Control surface deflections remain effective without being saturated, and stability is maintained across varying flight conditions. The adaptive approach reduces pilot workload, enhances safety, and simplifies the autopilot architecture into a single robust configuration.

This paper is structured as follows: Section 2 details the simulation setup. The adaptive autopilot architecture and the development of command laws are described in Section 3. The simulation results and their validation are presented in Section 4, followed by a robustness assessment in Section 5. Finally, in Section 6, the results are discussed.

4.3 Aircraft Description and Simulation platform

4.3.1 Cessna Citation X Aircraft Description

The Cessna Citation X (CCX) is a high-performance conventional aircraft introduced in 1996, known for its high-speed and long-range capabilities. Designed for corporate and private aviation, it balances speed, comfort, and efficiency. The aircraft features a low-swept wing optimized for high-speed cruise, with a maximum operating Mach number of 0.92. Its pressurized cabin accommodates up to 12 passengers and 3 crew members. The CCX can operate at altitudes of up to 51,000 ft (MSL). Powered by two Rolls-Royce AE 3007C-1 turbofan engines, the Cessna Citation X cruises at approximately 350 knots and offers a range of about 3,460 nautical miles. The flight control system uses actuators and supports advanced autopilot functions. Inner control loops regulate pitch rate, roll rate, and yaw rate dynamics, while outer loops manage altitude, vertical speed, and heading.

For the development and validation of the proposed autopilots, flight simulated data from a Research Aircraft Flight Simulator (RAFS) with Level D flight dynamics, have been gathered in order to model the Cessna Citation X flight dynamics. The RAFS is therefore a valuable platform for flight control studies due to its performance and realistic system data.



Figure 4.1 Research Aircraft Flight Simulator of the Cessna Citation X available at LARCASE laboratory

Table 4.1 Cessna Citation X performance characteristics

Performance	Maximum Value
Cruise speed	648 km/h [350 knots]
Mach number	0.92
Range	5725 km [3,091 nm]
Altitude	15 544 m [51,000 ft]
Takeoff Weight	16,193 kg [35,700 lb]
Zero Fuel weight	11,067 kg [24,400 lb]
Gross thrust	28 655 N [6442 lbf]

4.3.2 Cessna Citation X Simulink platform

The Simulink platform of the Cessna Citation X was developed in earlier studies and validated against flight simulation data gathered from the RAFS (Ghazi & Botez, 2015a). The platform incorporates high-fidelity aerodynamic and engine models, nonlinear actuator dynamic models, and equations of motion that accurately represent the aircraft's flight dynamics. It also integrates environmental modules, such as wind models, as well as failure scenarios involving engine failures or aerodynamic effects. These components were validated against RAFS flight-

test data with less than 5% error (Ghazi & Botez, 2015a). The RAFS, manufactured by CAE Inc., is qualified to Level D for flight dynamics. According to the FAA, Level D is the highest qualification level for flight simulators, ensuring that simulation outputs closely match real-flight behaviour within certified tolerances. This certification supports the generalizability and practical relevance of the proposed control algorithms when evaluated in this environment.

Additionally, the Simulink model includes linearization capabilities at specified calibrated airspeeds (CAS) and altitudes, supporting a variety of control design approaches. This platform has been successfully used in prior work for designing controllers such as Linear Quadratic Regulators (LQR) (Y. Boughari, Ghazi, et al., 2017a, 2017b), fuzzy controllers (S. M. Hosseini, Ghazi, & Botez, 2026 ; S. M. Hosseini, Ghazi, et al., 2024a), sliding mode controllers (S. M. Hosseini, Bematol, et al., 2024b), hybrid adaptive controllers (R. P. Andrianantara, Ghazi, & Botez, 2026) and hybrid MPC controller (R. P. Andrianantara, Ghazi, et al., 2024).

4.4 Design methodology

4.4.1 Inner loop controllers and Autopilots design requirements

The autopilot system for the Cessna Citation X is designed to satisfy key performance and safety requirements derived from certification standards. The CSAS and autopilots are designed to operate over an altitude range of 1,500 ft (MSL) to 51,000 ft, covering the aircraft's certified service ceiling and ensuring flight-envelope protection under all operating conditions. Throughout this altitude range, the system maintains the load factor within certified structural limits, thereby preserving airframe integrity and ensuring passenger comfort during cruise and flight-level changes. In addition, the CSAS complies with the Level 1 flying-qualities given in MIL-STD-1797A (Department of Defense, 1997). The design requirements are summarized in Table 4.2.

Table 4.2 CSAS and Autopilots design specifications

CSAS/Autopilot	Controlled state	Limits	Specifications
Pitch rate CSAS	q	$\pm 1.2^\circ/\text{s}$ [± 0.0017 rad/s]	Smooth and controlled longitudinal maneuvers; Level 1 flight quality.
Vertical Speed Autopilot	V_z	± 9.14 m/s [± 1800 ft/min]	Ensures safe climb and descent rates while allowing rapid altitude changes in emergencies.
Altitude Autopilot	h	no overshoot	Enhances safety and passenger comfort during climb/descent transitions.
Roll Rate CSAS	p	$\pm 2.5^\circ/\text{s}$ [0.43 rad/s]	Guarantees smooth, controlled lateral maneuvers; Level 1 flight quality.
Roll Angle CSAS	ϕ	$\pm 27.5^\circ$ [0.47 rad]	Maintains safe load factors and prevents structural overstress.
Heading Autopilot	ψ	heading rate of $3^\circ/\text{s}$ [0.052 rad/s]	Complies with civil and military standard turn rates.

4.4.2 Adaptive Autopilot Architectures

The complete controller and autopilot architectures are illustrated in Figure 4.2 through Figure 4.5. Figure 4.2 presents the pitch rate controller in the inner loop and the vertical speed autopilot in the outer loop. Figure 4.3 illustrates the altitude autopilot. Figure 4.4 and Figure 4.5 detail the lateral control loops, including the roll rate and roll angle CSAS and the heading autopilot.

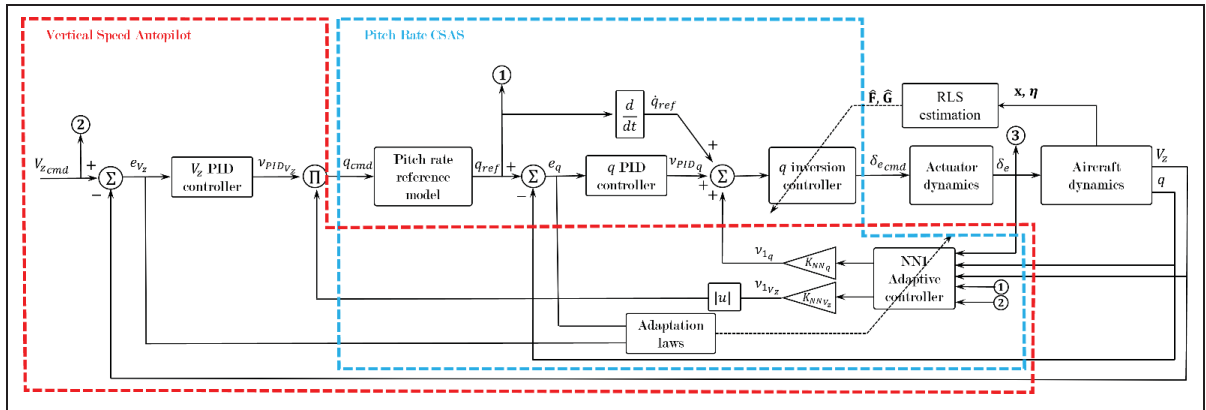


Figure 4.2 Adaptive Pitch rate CSAS and Adaptive Vertical Speed Autopilot architectures

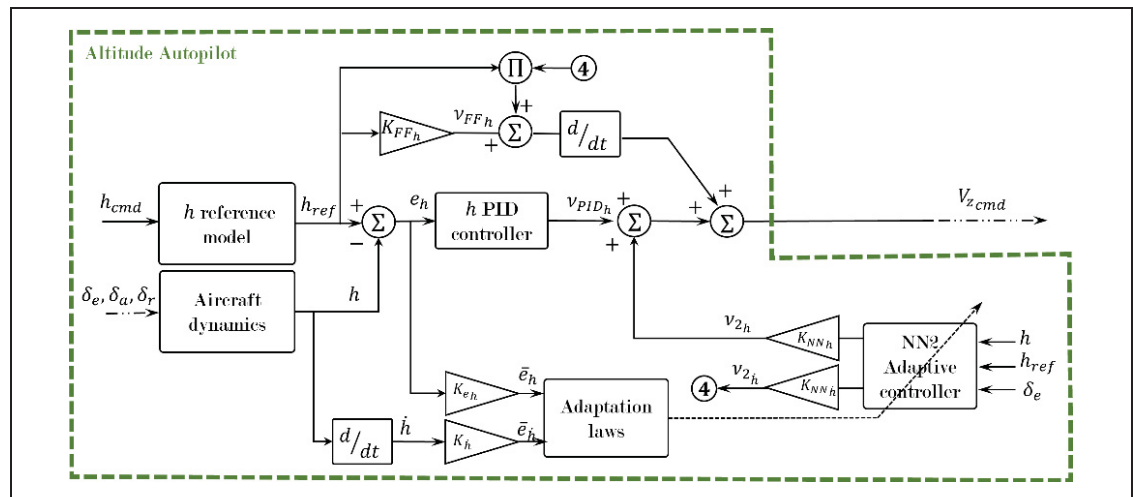


Figure 4.3 Adaptive Altitude autopilot architecture

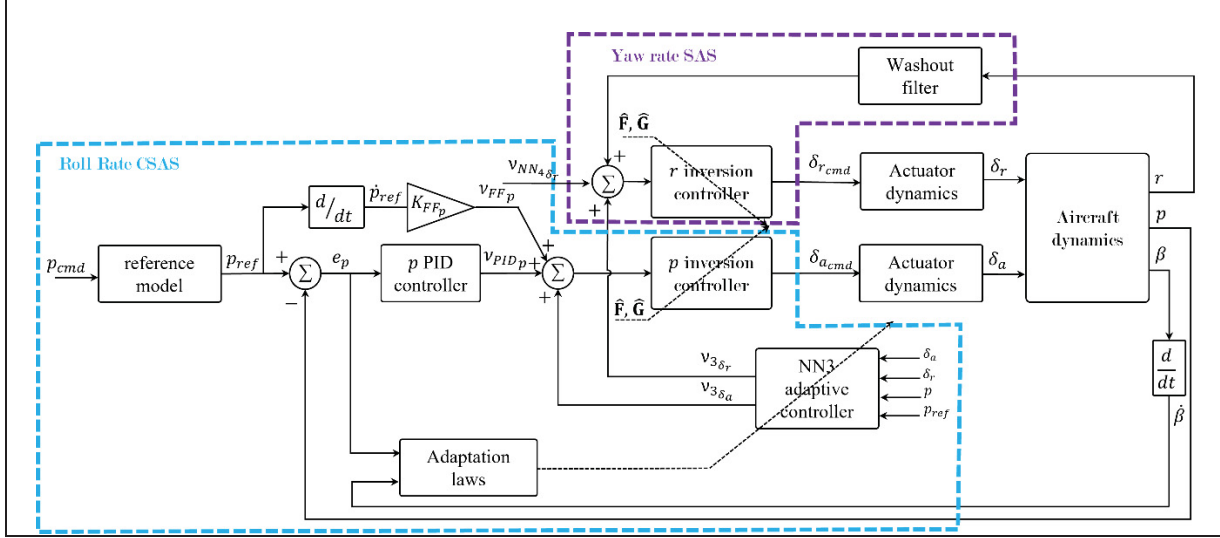


Figure 4.4 Adaptive Yaw rate stabilization and adaptive roll rate CSAS architectures

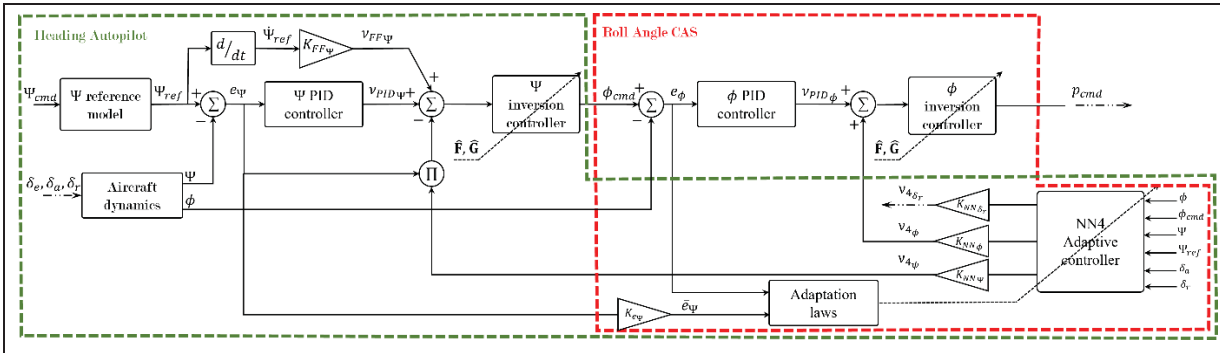


Figure 4.5 Adaptive Roll angle CSAS Controller and adaptive heading autopilot architectures

4.4.3 Aircraft Nonlinear Dynamics

This study considers the longitudinal and lateral aircraft dynamics. The aircraft nonlinear dynamics can be represented in the following form:

$$\begin{aligned}\dot{\mathbf{x}} &= \boldsymbol{\varphi}(\mathbf{x}, \boldsymbol{\eta}) \\ y &= \mathbf{\kappa}(\mathbf{x})\end{aligned}\tag{4.1}$$

where the state vector $\mathbf{x}$ and the control inputs vector $\boldsymbol{\eta}$ are defined as follows:

$$\begin{aligned}\mathbf{x} &= [\mathbf{x}_{\text{lon}} \ \mathbf{x}_{\text{lat}}]^T \\ \boldsymbol{\eta} &= [\delta_e \ \delta_a \ \delta_r]^T\end{aligned}\quad (4.2)$$

with

$$\mathbf{x}_{\text{lon}} = [u \ w \ q \ \theta \ V_z \ h]^T \text{ and } \mathbf{x}_{\text{lat}} = [v \ p \ r \ \phi \ \Psi]^T \quad (4.3)$$

Here, $\mathbf{x}_{\text{lon}}$ is the longitudinal state vector, including the longitudinal and vertical velocities u and w , the pitch rate q , the pitch angle θ , the vertical speed V_z , and the altitude h . The lateral state vector $\mathbf{x}_{\text{lat}}$ includes the lateral velocity v , the roll and yaw rates p and r , the roll angle ϕ , and the heading angle Ψ . The control input vector $\boldsymbol{\eta}$ includes the elevator δ_e , aileron δ_a , and rudder δ_r angles. The functions $\boldsymbol{\varphi}$ and $\mathbf{x}$ represent the unknown nonlinear dynamics and output mappings, respectively.

The nonlinear formulation in Eq. (4.1) captures the full state and control input variations of the aircraft, distinguishing it from traditional linear state-space models, and making it well suited for designing nonlinear controllers and autopilots.

4.4.4 Nonlinear Actuator Dynamics

The actuator models were implemented in the Simulink model using a second-order nonlinear transfer function, which captures the higher-order nonlinearities inherent in their real actuator dynamics. The key parameters defining the actuator dynamics are given in Table 4.3.

Table 4.3 Second Order Nonlinear Actuator dynamics

Characteristics	Values
Natural frequency	20 rad/s
Maximum deflection	14°
Minimum deflection	-19°
Damping ratio	0.8
Rate limit	20°/sec

4.4.5 Mathematical foundations and control elements

This section presents the controller and autopilot designs, and reviews the inner loop controllers, including the pitch and roll rate controllers, and the yaw rate stabilization. The control loops and command laws for the vertical speed, altitude, and heading autopilots are described separately in Figure 4.2 to Figure 4.5. The designed control system consists of multi-stage adaptive control loops that combine Adaptive Neural Networks (NN), Recursive Least Squares (RLS) dynamic inversion, and conventional linear PID controllers. Additional elements, such as feedforward filters, were incorporated to enhance transient response and overall performance.

Four neural networks, respectively labelled NN1, NN2, NN3 and NN4, were developed and integrated, as shown in Figure 4.2 to Figure 4.5. The adaptive signals produced by the neural network controllers were denoted according to their corresponding inputs and outputs. For example, $v_{4\delta_r}$ denotes the adaptive control signal generated by NN4 for the rudder deflection δ_r . Adaptive laws are used to update the neural network weights and the RLS parameters, ensuring convergence and stability. These laws have been derived to minimize tracking errors while guaranteeing the boundedness of the adaptive signals. Additional feedforward filters and disturbance rejection strategies were incorporated to enhance transient response and robustness in the presence of uncertainties.

These control elements and their mathematical foundations enable the design of robust adaptive autopilots capable of operating effectively across the cruise flight envelope of the CCX, while maintaining performance in the presence of uncertainties and nonlinearities such as turbulence and wind gusts.

4.4.5.1 Dynamic Inversion Controller

The CCX nonlinear dynamics can be represented as an affine function with respect to the control input:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\boldsymbol{\eta} \quad (4.4)$$

where $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are nonlinear functions describing the aircraft dynamics. The measured output y_i is expressed as:

$$y_i = \mathbf{C}_i \mathbf{x} \quad (4.5)$$

Dynamic Inversion controller seeks to compute the control input $\boldsymbol{\eta}$ that will directly achieve the desired state derivative $\dot{\mathbf{x}}$, given by the following inverse function:

$$\boldsymbol{\eta} = \boldsymbol{\varphi}^{-1}(\dot{\mathbf{x}}, \mathbf{x}) \quad (4.6)$$

Let us linearize the aircraft dynamics at a nominal operating point $(\mathbf{x}_0, \boldsymbol{\eta}_0)$. This local linearization, expressed in terms of small variations, may be obtained through Taylor series expansion and derivations. Thus, the incremental state dynamics $\Delta\dot{\mathbf{x}}$ can then be approximated locally as follows:

$$\Delta\dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta\mathbf{x} + \mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0} \Delta\boldsymbol{\eta} \quad (4.7)$$

where $\mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0}$ and $\mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0}$ are the Jacobian matrices at $(\mathbf{x}_0, \boldsymbol{\eta}_0)$ defined as:

$$\mathbf{F}_{\mathbf{x}_0, \boldsymbol{\eta}_0} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \boldsymbol{\eta} \right)_{\mathbf{x}_0, \boldsymbol{\eta}_0} ; \mathbf{G}_{\mathbf{x}_0, \boldsymbol{\eta}_0} = \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \boldsymbol{\eta}_0} \quad (4.8)$$

From the output relation in Eq. (4.5) and the linearized system shown in Eq. (4.7), the incremental control input $\Delta\eta$ required to track the reference signal r_s can be computed using the following control law :

$$\Delta\eta = (\mathbf{C}\mathbf{G}_{\mathbf{x}_0, \eta_0})^{-1} (\dot{r}_s + v - \mathbf{C}\mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta\mathbf{x}) \quad (4.9)$$

where $\mathbf{C}$ is the output matrix, and the signal v is chosen to reduce the error as follows (Stevens et al., 2015):

$$\dot{e} = -v \quad (4.10)$$

This formulation allows the error e to be stabilized by an appropriate choice of v , ensuring that the system tracks the desired reference signal with guaranteed stability and convergence.

4.4.5.2 Online State Estimation

The state variation at time step $k + 1$, denoted by $\Delta\mathbf{x}_{k+1}$, depends on the current state variation $\Delta\mathbf{x}_k$ and the control input variation $\Delta\eta_k$. Based on Eqs. (4.7), the state equation can be written as:

$$\Delta\hat{\mathbf{x}}_{k+1} = [\Delta\mathbf{x}_k \quad \Delta\eta_k] \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix} \quad (4.11)$$

where $\hat{\mathbf{F}}_k^T$ and $\hat{\mathbf{G}}_k^T$ are the Jacobian matrix estimates.

Let us consider $\hat{\boldsymbol{\Theta}}_k = \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix}$ as the vector of unknown parameters and $\mathbf{X}_k = [\Delta\mathbf{x}_k \quad \Delta\eta_k]^T$ as the vector of past increments. With these definitions, Eq. (4.11) can be compactly rewritten as:

$$\Delta\hat{\mathbf{x}}_{k+1} = \mathbf{X}_k^T \hat{\boldsymbol{\Theta}}_k \quad (4.12)$$

in which $\hat{\Theta}_k$ can be estimated online using the Recursive Least Squares (RLS) algorithm.

We can define the RLS prediction error ϵ_k as follows:

$$\epsilon_k = \Delta \mathbf{x}_k - \Delta \hat{\mathbf{x}}_k \quad (4.13)$$

where $\Delta \mathbf{x}_k$ is the measured state variation and $\Delta \hat{\mathbf{x}}_k$ is its estimated state variations.

Following the method described in (Haykin, 2013), the parameter vector $\hat{\Theta}_k$ is computed as:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + \frac{\mathbf{L}_{k-1} \mathbf{X}_k}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \epsilon_k \quad (4.14)$$

where, $\mathbf{L}_k$ is the covariance matrix, which is numerically computed as follows:

$$\mathbf{L}_k = \frac{1}{\kappa} \left(\mathbf{L}_{k-1} - \frac{\mathbf{L}_{k-1} \mathbf{X}_k \mathbf{X}_k^T \mathbf{L}_{k-1}}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \right) \quad (4.15)$$

Equations (4.14) and (4.15) are derived from the matrix inversion lemma, as detailed in (Haykin, 2013). Note that in this RLS formulation, $\hat{\mathbf{F}}_k^T$ and $\hat{\mathbf{G}}_k^T$ depend on the increment vector $\mathbf{X}_k$, rather than on the absolute state and control values.

Before running the algorithm, Θ_0 , $\mathbf{L}_0$, and κ must be initialized. In this work, in order to improve stability, the covariance matrix $\mathbf{L}_k$ is updated conditionally: $\mathbf{L}_k$ is reset when the prediction error ϵ_k exceeds a prescribed threshold, i.e., $\epsilon_k > \epsilon_{threshold}$. Here below is given a pseudo-code for the covariance matrix update :

Table 4.4 RLS estimation algorithm and covariance matrix update

Initialize $\Theta_0, \mathbf{L}_0$ Set κ and $\epsilon_{threshold}$ Compute $\hat{\Theta}_k$ using Eq.(4.14) Compute $\Delta \hat{\mathbf{x}}_{k+1}$ using Eq. (4.12) Compute ϵ_k using Eq. (4.13) Update $\mathbf{L}_k$ if $\text{abs}(\epsilon_k) \geq \epsilon_{threshold}$ then $\mathbf{L}_k = 10^3 \mathbf{I}$ else compute $\mathbf{L}_k$ using Eq.(4.15)

4.4.5.3 NN Controller

Using the errors estimations in Eq. (4.13) often leads to prediction errors in real-world applications (Soares & Burken, 2006). Therefore, the actual output variation at the time step $k + 1$ can be expressed in discrete form as:

$$\Delta \hat{y}_{k+1} = \Delta y_{k+1} - \varepsilon_1 \quad (4.16)$$

where $\Delta \hat{y}_{k+1}$ is the estimated output and ε_1 is the RLS estimation error.

The control input variation $\Delta \eta_k$ in Eq. (4.9) can then be rewritten as follows:

$$\Delta \eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} [\dot{r}_s + v - \mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x} - \varepsilon_1] \quad (4.17)$$

where $\mathbf{C}\hat{\mathbf{G}}_k$ and $\mathbf{C}\hat{\mathbf{F}}_k$ describe the aircraft's local dynamics. An adaptive term is used to compensate for RLS estimation errors, such that:

$$\varepsilon_1 = v_{ad} \quad (4.18)$$

A Feedforward Neural Network (FFNN) is used to compute v_{ad} as an adaptive component.

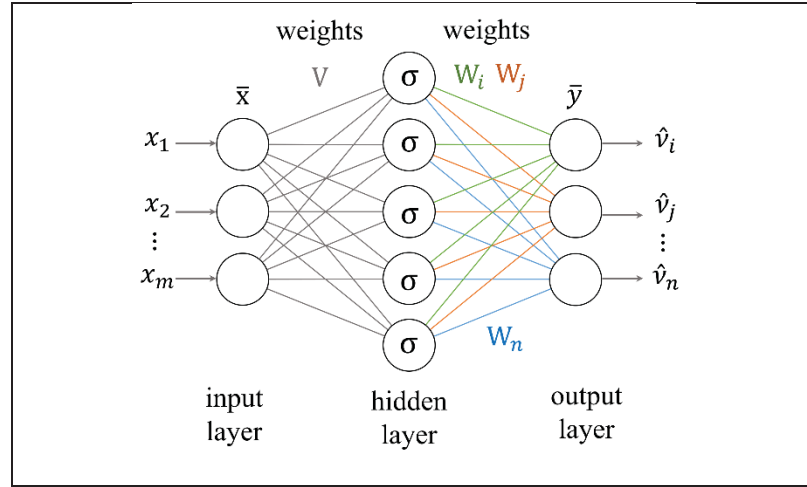


Figure 4.6 Single Layer Feedforward Neural Network architecture

Figure 4.6 depicts the architecture of the FFNN used in the CSAS and in the autopilots. The NN's performance depends on the number of neurons, the number of hidden layers, the activation function, and the weighting vectors. The adaptive output signal from the neural network can be expressed as follows (Stevens et al., 2015):

$$\hat{v}_i = \mathbf{W}_i^T \boldsymbol{\sigma}(\mathbf{V}^T \bar{\mathbf{x}}) \quad (4.19)$$

where $\boldsymbol{\sigma} \in \mathbb{R}^{N \times 1}$ is the activation function applied elementwise over N neurons in the hidden layer. The input vector is defined as $\bar{\mathbf{x}} = [x_1 \ x_2 \ \cdots \ x_m]^T$. The weight vectors are $\mathbf{W} \in \mathbb{R}^{N \times n}$ and $\mathbf{V} \in \mathbb{R}^{m \times N}$, with m inputs and n outputs. The tangent sigmoid activation function $\boldsymbol{\sigma}$ is defined as:

$$\sigma_i(z) = \frac{1}{1 + e^{-z}} \quad (4.20)$$

A linear function is used for both the input and the output layers. As seen in Figure 4.2 to Figure 4.5, the neural network adaptive signals are scaled by a gain factor K_{NNi} , thus controlling the amplitude of the adaptive correction term v_{ad} .

4.4.5.4 NN Adaptation laws

The neural network weights are dynamically updated online using adaptation laws issued from backpropagation algorithm (Narendra & Annaswamy, 1987 ; Smith et al., 2010 ; Williams-Hayes, 2005):

$$\begin{aligned}\dot{\mathbf{W}}_i &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}^T \bar{\mathbf{x}}) e_i^T + \lambda \|e_i\| \mathbf{W}_i] \boldsymbol{\Gamma}_i \\ \dot{\mathbf{V}} &= -\boldsymbol{\Gamma}_V [\bar{\mathbf{x}} e_i^T \mathbf{W}_i^T \boldsymbol{\sigma}' + \lambda \|e_i\| \mathbf{V}]\end{aligned}\quad (4.21)$$

Here, e_i is the tracking error vector corresponding to the i -th controlled variable, $\boldsymbol{\Gamma}_i$ is the learning rate matrix corresponding to the i -th output of the NN, $\boldsymbol{\Gamma}_V$ is the input layer learning rate, and $\boldsymbol{\sigma}'$ is the gradient of the activation function $\boldsymbol{\sigma}$. The scalar λ is set to 1. The additional terms $\lambda \|e_i\| \mathbf{W}_i$ and $\lambda \|e_i\| \mathbf{V}$ are defined to guarantee robustness and stability, as demonstrated through Lyapunov's direct method. These update laws are based on the backpropagation theory (Narendra & Annaswamy, 1987 ; Smith et al., 2010 ; Williams-Hayes, 2005).

4.4.5.5 PID Controller

From the inversion control law, a linear feedback controller v is required (Stevens et al., 2015). The feedback term v is solely designed to ensure system stability. To model v , a basic Proportional – Integral – Derivative (PID) controller is typically sufficient and can be expressed as follows (Stevens et al., 2015):

$$v_i = \left(K_P + \frac{K_I}{s} + K_D s \right) e_i \quad (4.22)$$

where K_P, K_I and K_D are the proportional, integral, and derivative gains, s is the Laplace operator, and i indexes the controlled variable.

4.4.6 Longitudinal and lateral CSAS inner loops

In this research, the longitudinal and lateral Control Stability Augmentation System (CSAS) consist of low-level inner loop controllers that ensure robust and stable tracking, which are included within higher-level adaptive autopilots.

The longitudinal CSAS is designed to control the pitch rate q and to stabilize short-period dynamics. It provides a rapid response feedback loop that maintains pitch stability and reduces pilot workload. The longitudinal autopilot enhances this by controlling vertical speed and altitude through precise pitch rate commands, ensuring smooth vertical flight path tracking even under disturbances.

Regarding the lateral dynamics, the CSAS focuses on controlling the roll rate p and stabilizing the yaw rate r by effectively damping the Dutch roll effect. This fact is critical for maintaining directional stability. Then, the lateral autopilots command the roll angle φ and heading Ψ , allowing coordinated control that enables accurate heading tracking and bank angle control. A notable feature in this research is the incorporation of an adaptive Aileron-Rudder Interconnect (ARI) using neural networks, which improves yaw damping by adaptively coordinating aileron and rudder inputs.

These CSAS consist of model references controllers designed with given reference signals. The CSAS have been previously validated for Level 1 flight qualities for the Cessna Citation X operational envelope (R. P. Andrianantara, Ghazi, & Botez, 2025). Their proven performance provides a reliable foundational work for the outer loop adaptive autopilots developed in this study, ensuring their stability and responsiveness during complex maneuvers even in the presence of uncertainties.

4.4.7 Pitch rate CSAS

For pitch rate and roll rate control, a second-order transfer function is used as a reference model, which approximates the short period dynamics with specific frequency, damping ratio

and time constant values, ω_{sp} , ξ_{sp} and T_{θ_2} , respectively. Their values have been set to satisfy Level 1 flight qualities according to MIL-STD-1797A (Department of Defense, 1997). Table 4.5 summarizes the mathematical design equations of the pitch rate controller.

Table 4.5 Pitch rate CSAS controller design equations

Control elements	Equations	
Pitch rate reference model	$\frac{q_{ref}}{q_{cmd}} = \frac{\omega_{sp}^2 (1 + T_{\theta_2} s)}{s^2 + 2\xi_{sp}\omega_{sp}s + \omega_{sp}^2}$	
Pitch rate error dynamics	$e_q = q_{ref} - q$	
NN1 controller input vector	$\bar{\mathbf{x}}_1 = [q \quad q_{ref} \quad V_z \quad V_{zcmd} \quad \delta_e]^T$	
NN1 adaptation laws	$\begin{aligned} \dot{\mathbf{W}}_q &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_1^T \bar{\mathbf{x}}_1) e_{V_z}^T + \lambda \ e_{V_z}\ \mathbf{W}_q] \boldsymbol{\Gamma}_q \\ \dot{\mathbf{W}}_{V_z} &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_1^T \bar{\mathbf{x}}_1) e_{V_z}^T + \lambda \ e_{V_z}\ \mathbf{W}_{V_z}] \boldsymbol{\Gamma}_{V_z} \\ \dot{\mathbf{V}}_1 &= -\boldsymbol{\Gamma}_{V_1} [\bar{\mathbf{x}}_1 e_{V_z} \mathbf{W}_{V_z}^T \boldsymbol{\sigma}' + \bar{\mathbf{x}}_1 e_q \mathbf{W}_q^T \boldsymbol{\sigma}' \\ &\quad + \lambda \ e_{V_z} + e_q\ \mathbf{V}_1] \end{aligned}$	
PID controller	$v_{PIDq} = \left(K_P + \frac{K_I}{s} + K_D s \right) e_q$	
NN1 output signals	$v_{1q} = K_{NNq} \mathbf{W}_q^T \boldsymbol{\sigma}(\mathbf{V}_1^T \bar{\mathbf{x}}_1)$ $v_{1V_z} = K_{NNV_z} \mathbf{W}_{V_z}^T \boldsymbol{\sigma}(\mathbf{V}_1^T \bar{\mathbf{x}}_1)$	
Elevator control law	$\begin{aligned} \Delta \delta_e &= (\mathbf{C}_q \hat{\mathbf{G}}_k)^{-1} \left[\dot{q}_{ref} + \left(K_P + \frac{K_I}{s} + K_D s \right) e_q \right. \\ &\quad \left. - \mathbf{C}_q \hat{\mathbf{F}}_k \Delta \mathbf{x}_k - K_{NNq} \mathbf{W}_q^T \boldsymbol{\sigma}(\mathbf{V}_1^T \bar{\mathbf{x}}_1) \right] \end{aligned}$	(4.23)

4.4.8 Vertical Speed autopilot

The vertical speed autopilot is designed to control V_z . The designed autopilot uses a PID controller and an adaptive neural network NN1.

Table 4.6 Vertical Speed Autopilot design equations

Control elements	Equations	
Error dynamics	$e_{V_z} = V_{zcmd} - V_z$	
PID controller output	$v_{PIDV_z} = \left(K_P + \frac{K_I}{s} + K_D s \right) e_{V_z}$	
NN1 controller	See Table 4.5	
Pitch rate command	$q_{cmd} = v_{1V_z} v_{PIDV_z}$	(4.24)

4.4.9 Altitude Autopilot

An adaptive neural network NN2 and a PID are used to track the desired altitude. The neural network receives altitude error and its derivatives as inputs in the adaptation laws, enabling it to reduce tracking errors, stabilize altitude, and dampen oscillations during transient phases effectively.

A ramp reference model is used to generate altitude commands. This model ensures smooth transitions between climbing and cruise phases by shaping the altitude reference signal with a first-order transfer function. The ramp is designed to constrain climb and descent rates within ± 1800 ft/min, thereby indirectly controlling the vertical speed and ensuring safe flight level changes.

Table 4.7 Altitude Autopilot design equations

Control elements	Equations	
Error dynamics	$\bar{e}_h = K_{e_h}(h_{ref} - h)$ $\bar{e}_{\dot{h}} = K_{e_{\dot{h}}}\dot{h}$	
NN2 inputs	$\bar{\mathbf{x}}_2 = [h \quad h_{ref} \quad \delta_e]^T$	
NN2 adaptation laws	$\dot{\mathbf{W}}_h = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_2^T \bar{\mathbf{x}}_2) \bar{e}_h^T + \lambda \ \bar{e}_h\ \mathbf{W}_h] \boldsymbol{\Gamma}_h$ $\dot{\mathbf{W}}_{\dot{h}} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_2^T \bar{\mathbf{x}}_2) \bar{e}_{\dot{h}}^T + \lambda \ \bar{e}_{\dot{h}}\ \mathbf{W}_{\dot{h}}] \boldsymbol{\Gamma}_{\dot{h}}$ $\dot{\mathbf{V}}_2 = -\boldsymbol{\Gamma}_{V_2} [\bar{\mathbf{x}}_2 \bar{e}_h \mathbf{W}_h^T \boldsymbol{\sigma}' + \bar{\mathbf{x}}_2 \bar{e}_{\dot{h}} \mathbf{W}_{\dot{h}}^T \boldsymbol{\sigma}' + \lambda \ \bar{e}_h + \bar{e}_{\dot{h}}\ \mathbf{V}_2]$	
PID controller	$v_{PID_h} = \left(K_P + \frac{K_I}{s} + K_D s \right) e_h$	
Feedforward controller	$v_{FF_h} = K_{FF_h} h_{ref}$	
NN2 output signals	$v_{2_h} = K_{NN_h} \mathbf{W}_h^T \boldsymbol{\sigma}(\mathbf{V}_2^T \bar{\mathbf{x}}_2)$ $v_{2_{\dot{h}}} = K_{NN_{\dot{h}}} \mathbf{W}_{\dot{h}}^T \boldsymbol{\sigma}(\mathbf{V}_2^T \bar{\mathbf{x}}_2)$	
Command law	$V_{z_{cmd}} = (K_{FF_h} + v_{ad_{\dot{h}}}) \dot{h}_{ref} + v_{PID_h} + \frac{d}{dt} v_{2_h}$	(4.25)

4.4.10 Roll rate CSAS and yaw rate stabilization

The roll rate CSAS controller uses a Model Reference Adaptive Control (MRAC) approach, where a reference model captures the desired roll and Dutch-roll dynamics based on flight quality specifications MIL-STD-1797A (Department of Defense, 1997), specifically the roll time constant τ_r , damping ξ_{dr} and natural frequency ω_{dr} . The controller combines a traditional PID controller with Dynamic Inversion and an adaptive neural network NN3. A feedforward term helps to anticipate changes in roll dynamics, and a washout filter stabilizes the yaw rate during lateral maneuvers. Input commands are generated for both ailerons and rudders, considering actuator dynamics and coupling effects, ensuring coordinated and stable lateral control.

While stabilizing the yaw rate, the derivative of the sideslip angle $\dot{\beta}$ is also damped, and thus it tends to 0. The value of $\dot{\beta}$ can be approximated by the following kinetic relationship (Stevens et al., 2015):

$$\dot{\beta} \approx \frac{\dot{v}}{\text{TAS}} \quad (4.26)$$

where $\dot{v}$ is the lateral acceleration and TAS is the true airspeed. Setting $\dot{\beta}$ to 0 allows to fulfill a coordinated turn with a dampened Dutch-roll effect.

Table 4.8 Roll rate CSAS controller design equations

Control elements	Equations	
Roll rate reference model	$\frac{p_{ref}}{p_{cmd}} = \frac{\omega_{dr}^2}{s^2 + 2 \xi_{dr} \omega_{dr} s + \omega_{dr}^2} \frac{1}{\tau s + 1}$	
Error dynamics	$e_p = p_{ref} - p$ $e_r = \dot{\beta}$	
NN3 inputs	$\bar{\mathbf{x}}_3 = [p \quad p_{ref} \quad \delta_a \quad \delta_r]^T$	
NN3 adaptation laws	$\dot{\mathbf{W}}_{3\delta_a} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_3^T \bar{\mathbf{x}}_3) e_p + \lambda \ e_p\ \mathbf{W}_{\delta_a}] \boldsymbol{\Gamma}_{\delta_a}$ $\dot{\mathbf{W}}_{3\delta_r} = -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_3^T \bar{\mathbf{x}}_3) e_r + \lambda \ e_r\ \mathbf{W}_{3\delta_r}] \boldsymbol{\Gamma}_{\delta_r}$ $\dot{\mathbf{V}}_3 = -\boldsymbol{\Gamma}_{V_3} [\bar{\mathbf{x}} e_p \mathbf{W}_a^T \boldsymbol{\sigma}' + \bar{\mathbf{x}} e_r \mathbf{W}_r^T \boldsymbol{\sigma}' + \lambda \ e_p + e_r\ \mathbf{V}_3]$	
PID controller	$v_{PID_p} = \left(K_P + \frac{K_I}{s} + K_D s \right) e_p$	
Feedforward controller	$v_{FF_p} = \mathbf{K}_{FF} \dot{p}_{ref}$	
Yaw rate washout filter	$F(s) = \frac{K_{WF} \cdot \tau s}{\tau s + 1}$	
NN3 outputs	$v_{3\delta_a} = \mathbf{W}_a^T \boldsymbol{\sigma}(\mathbf{V}_3^T \bar{\mathbf{x}}_3)$ $v_{3\delta_r} = \mathbf{W}_{3\delta_r}^T \boldsymbol{\sigma}(\mathbf{V}_3^T \bar{\mathbf{x}}_3)$	
Ailerons control law	$\Delta \delta_{a_k} = (\mathbf{C}_p \hat{\mathbf{G}}_{\delta_{a_k}})^{-1} \left[\mathbf{K}_{FF} \dot{p}_{ref} + \left(K_P + \frac{K_I}{s} + K_D s \right) e_p - \mathbf{C}_p \hat{\mathbf{F}}_k \Delta \mathbf{x}_k - \mathbf{C}_p \hat{\mathbf{G}}_{\delta_{r_k}} \Delta \delta_{r_{k-1}} - \mathbf{W}_a^T \boldsymbol{\sigma}(\mathbf{V}_3^T \bar{\mathbf{x}}_3) \right]$	(4.27)
Rudders control law	$\Delta \delta_{r_k} = (\mathbf{C}_r \hat{\mathbf{G}}_{\delta_{r_k}})^{-1} \left[\frac{K_{WF} \cdot \tau s}{\tau s + 1} r - \mathbf{C}_r \hat{\mathbf{F}}_k \Delta \mathbf{x}_k - \mathbf{C}_r \hat{\mathbf{G}}_{\delta_{a_k}} \Delta \delta_{a_{k-1}} - \mathbf{W}_{3\delta_r}^T \boldsymbol{\sigma}(\mathbf{V}_3^T \bar{\mathbf{x}}_3) + v_{4\delta_{r_k}} \right]$	(4.28)

K_{WF} is the washout filter gain and τ is its time constant; $\mathbf{C}_i$ is the output matrix in respect to the i -th variable state.

4.4.11 Roll angle CSAS

The roll controller aims to regulate the aircraft's roll angle ϕ in order to follow a command ϕ_{cmd} . The control loop incorporates PID control, RLS dynamic inversion and adaptive neural network controller NN4.

Table 4.9 Roll angle CSAS controller design equations

Control elements	Equations	
Error dynamics	$e_\phi = \phi_{cmd} - \phi$	
NN4 inputs	$\bar{\mathbf{x}}_4 = [\phi \quad \phi_{cmd} \quad \Psi \quad \Psi_{ref} \quad \delta_a \quad \delta_r]^T$	
PID controller	$v_{PID\phi} = \left(K_P + \frac{K_I}{s} + K_D s\right) e_\phi$	
NN4 adaptation laws	$\dot{\mathbf{W}}_{4\delta_r} = -\left[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_4^T \bar{\mathbf{x}}_4) e_\phi^T + \lambda \ e_\phi\ \mathbf{W}_{4\delta_r}\right] \boldsymbol{\Gamma}_{4\delta_r}$ $\dot{\mathbf{W}}_\phi = -\left[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_4^T \bar{\mathbf{x}}_4) e_\phi^T + \lambda \ e_\phi\ \mathbf{W}_\phi\right] \boldsymbol{\Gamma}_\phi$ $\dot{\mathbf{W}}_\Psi = -\left[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}_4^T \bar{\mathbf{x}}_4) \bar{e}_\Psi^T + \lambda \ \bar{e}_\Psi\ \mathbf{W}_\Psi\right] \boldsymbol{\Gamma}_\Psi$ $\dot{\mathbf{V}}_4 = -\boldsymbol{\Gamma}_{V_4} [\bar{\mathbf{x}}_4 e_\phi \mathbf{W}_\phi^T \boldsymbol{\sigma}' + \bar{\mathbf{x}}_4 \bar{e}_\Psi \mathbf{W}_\Psi^T \boldsymbol{\sigma}' + \lambda \ e_\phi + \bar{e}_\Psi\ \mathbf{V}_4]$	
NN4 outputs	$v_{4\delta_r} = K_{NN\delta_r} \mathbf{W}_{4\delta_r}^T \boldsymbol{\sigma}(\mathbf{V}_4^T \bar{\mathbf{x}}_4)$ $v_{4\phi} = K_{NN\phi} \mathbf{W}_\phi^T \boldsymbol{\sigma}(\mathbf{V}_4^T \bar{\mathbf{x}}_4)$ $v_{4\Psi} = K_{NN\Psi} \mathbf{W}_\Psi^T \boldsymbol{\sigma}(\mathbf{V}_4^T \bar{\mathbf{x}}_4)$	
Command law	$p_{cmd} = (\mathbf{C}_\phi \hat{\mathbf{G}}_{\delta_{ak}})^{-1} \left[v_{PID\phi} - \mathbf{C}_\phi \hat{\mathbf{F}}_k \Delta \mathbf{x}_k + v_{4\phi} \right]$	(4.29)

4.4.12 Heading autopilot

Finally, the heading autopilot is designed. The autopilot uses a PID controller, an RLS Dynamic Inversion and an adaptive neural network. A feedforward term is added to improve tracking of the heading reference signal. This formulation ensures precise and robust control of the aircraft's heading by integrating model-based control with adaptive neural network compensation.

Table 4.10 Heading Autopilot design equations

Control elements	Equations	
Error dynamics	$e_\Psi = \Psi_{ref} - \Psi$ $\dot{e}_\Psi = K_{e_\Psi} e_\Psi$	
PID controller	$v_{PID\Psi} = \left(K_P + \frac{K_I}{s} + K_D s \right) e_\Psi$	
Feedforward controller	$v_{FF\Psi} = K_{FF\Psi} \dot{\Psi}_{ref}$	
Command law	$\phi_{cmd} = (\mathbf{C}_\Psi \hat{\mathbf{G}}_{\delta_{ak}})^{-1} [v_{FF\Psi} + v_{PID\Psi} - \mathbf{C}_\Psi \hat{\mathbf{f}}_k \Delta \mathbf{x}_k + v_{4\Psi}]$	(4.30)

4.5 Results

This section presents the control applications of the designed CSAS and the adaptive autopilots. The simulations include parameter tuning in nominal tests and robustness tests in the presence of wind and turbulence. Adaptive parameter variations are also shown, and the overall flight envelope of the Cessna Citation X is studied.

4.5.1 Control parameters

The control parameters presented in the previous equations were carefully tuned over the aircraft's flight envelope using parameter sensitivity analysis, with their values summarized in Table 4.10 and Table 4.11.

Table 4.11 Longitudinal CSAS and Autopilot Design Parameters

Control Elements	Parameters	Values
$\mathbf{q}$ PID controller	K_P, K_I, K_D	$-1, -1, -0.1$
RLS longitudinal state estimation	$\kappa, \mathbf{L}_0, \hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$ $\epsilon_{q_{threshold}}, \epsilon_{\theta_{threshold}}$ $\mathbf{L}_{update}$	$1.015, 10^7 \mathbf{I}_{5,5}, \mathbf{I}_{4,4}, -\mathbf{0.1}_{4,1}$ $0.0001 \text{ rad/s}, 0.005 \text{ rad}$ 10^4
NN1 controller	K_{NN_q} , learning rate $\Gamma_{\mathbf{q}}$ $K_{NN_{V_z}}$, learning rate Γ_{V_z} hidden layers, neurons	$5, 1$ $0.02, 0.8$ $1, 30$
Pitch Rate Reference Model	$\xi_{sp}, \omega_{sp}, T_{\theta_2}$	$0.69, 4 \text{ rad/s}, 0.35 \text{ sec}$
V_z PID controller	K_P, K_I, K_D	$-0.1, 0, 0$
h PID controller	K_P, K_I, K_D	$-0.3, 0, 0$
h feedforward controller	K_{FF_h}	-0.2
NN2 Controller	$K_{NN_h}, K_{NN_{\dot{h}}}$, learning rates $\Gamma_h, \Gamma_{\dot{h}}, \Gamma_{V_2}$ $K_{e_h}, K_{\dot{h}}$, hidden layers, neurons	$2, 0.05$ $1, 1, 2$ $0.001, 0.0002$ $1, 30$

Table 4.12 Lateral CSAS and Autopilot Design Parameters

Control Elements	Parameters	Values
p PID Controller	K_P, K_I, K_D	-1, -1, 0
p Feedforward controller	K_{FF_p}	-0.1
Yaw damper, Washout Filter	K_{WF}, τ	3, 2 sec
NN3 controller	Learning rates $\Gamma_a, \Gamma_r, \Gamma_{v_3}, \lambda$ Number of layers Number of neurons	0.1, 0.3, 2, 1 1 30
p Reference Model	$\xi_{dr}, \omega_{dr}, \tau$	0.5, 3 rad/s, 0.5
RLS lateral state estimation	$\kappa, \mathbf{L}_0, \hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$ $\epsilon_{p_threshold}, \epsilon_{r_threshold}$ $\mathbf{L}_{update}$	1.015, $10^4 \mathbf{I}_{7,7}$, $\mathbf{I}_{5,5}$, $\mathbf{0.1}_{5,2}$ 0.0001 rad/s, 0.001 rad/s $10^4 \mathbf{I}_{7,7}$
ϕ PID controller	K_P, K_I, K_D	0.03, 0, 0
Ψ PID controller	K_P, K_I, K_D	0.2, 0, 0.1
Ψ feedforward controller	$K_{FF\Psi}$	
NN4 controller	$K_{NN_{\delta_r}}, K_{NN_{\phi}}, K_{NN_{\psi}}, K_{e_{\psi}}$ Learning rates $\Gamma_{\delta_r}, \Gamma_{\phi}, \Gamma_{\psi}, \Gamma_{v_{NN4}}, \lambda$ hidden layers, neurons	0.001, 0.0001, 0.005, 0.1 0.5, 4, 4, 1, 1, 1 1, 30

4.5.2 Vertical speed Autopilot Tuning

This section presents the tuning of the Vertical Speed autopilot. The tuning was performed using parameter sensitivity analysis. The adaptive features were tuned for 64 cruise flights covering the Cessna Citation X operational envelope.

First, the PID controller was tuned. A single PID configuration was set in order to ensure controller stability and convergence. The adaptive features were then tuned. In the case of the vertical speed autopilot, the neural networks' learning rates Γ_{v_z} and gain $K_{NN_{v_z}}$ were set. Figure 4.7 shows the parameters' effect on tracking performance. A suitable combination of parameters was found and summarized in the above Table 4.11.

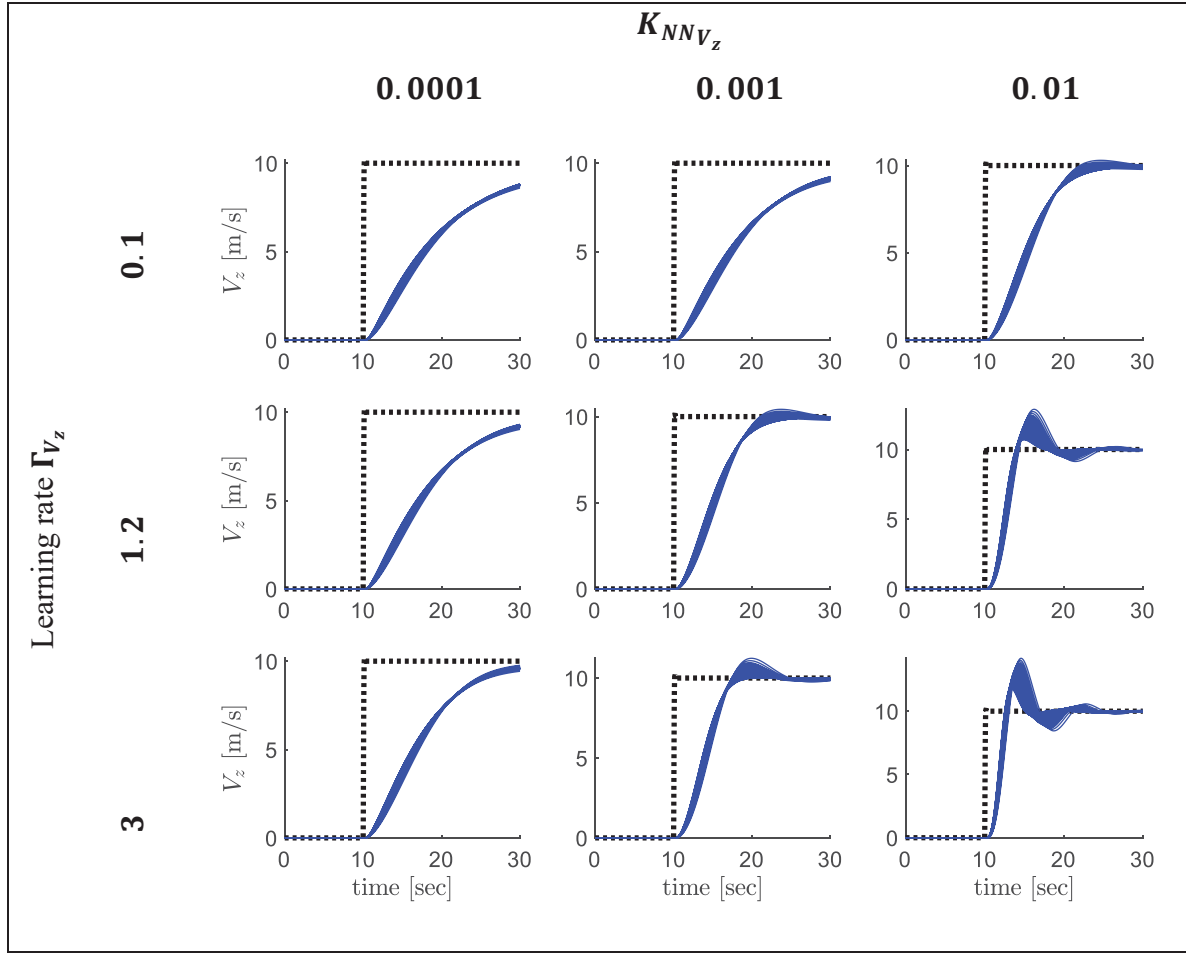


Figure 4.7 Vertical Speed responses for varying learning rates and NN gain for 64 representative cruise operating conditions

The NN learning rates and gains were optimized in the presence of uncertainties. In fact, incorporating uncertainties during the tuning process ensures very good robustness.

The optimization was fulfilled using the objective function such as the overall Sum Squared Error denoted as *Overall SSE* and computed as follows:

$$overall\ SSE = \sum_{i=1}^{i=N} SSE_i ; \quad SSE = \Delta t \sum_{t=0}^{t=T} (V_{zcmd_t} - V_{z_t})^2 \quad (4.31)$$

where N is the number of simulated cruise scenarios. The SSE is computed for each flight scenario, V_{zcmd_t} and V_{z_t} are the reference vertical speed and the vertical speed values at a given

time step t , T is the simulation time length for one flight condition, and $\Delta t = 0.02$ sec is the sample time.

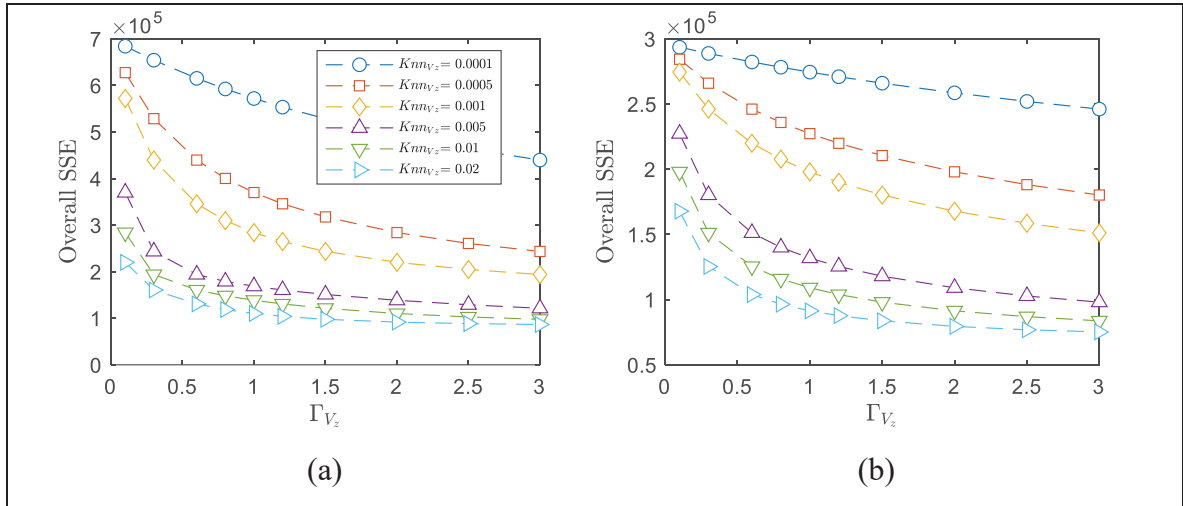


Figure 4.8 Overall Sum Squared Error SSE for 64 flight conditions vs learning rates Γ_{V_z} and K_{NNV_z} in the presence of (a) wind gust and (b) turbulence for different learning rate and NN gain variations

The remaining CSASs and autopilots such as the altitude autopilot, roll controller and heading autopilot were tuned following the same procedure described above. The parameters for each control algorithm are summarized in Table 4.11 and Table 4.12.

4.5.3 Altitude and Heading Command at 35,000 ft and 290 knots

Once the CSASs and the autopilots were properly tuned, they were tested for flight simulation scenarios. First, a combined altitude and heading command simulation was conducted under cruise conditions at an altitude of 35,000 ft and a calibrated airspeed of 290 knots. As seen in Figure 4.9, the altitude tracking consists of a flight level change of 300 m [$\approx 1,000$ ft], with a climb rate of 9.14 m/s [1,800 ft/min], allowing an indirect vertical speed command. Upon reaching the target altitude, the altitude capture mode engages automatically to ensure a smooth transition with minimal overshoot.

Simultaneously, a heading command of 90° is fulfilled, allowing a coordinated turn with a constant turn rate of $3^\circ/\text{sec}$. This scenario allows the controller's performance during coordination between the longitudinal and lateral control loops to be assessed. Figure 4.10 and Figure 4.11 show the adaptive parameter variations, including the RLS state estimation, the RLS parameters' variations including $\mathbf{F}$ and $\mathbf{G}$, the covariance matrices $\mathbf{L}_k$ and the estimation errors ϵ_i , the NN inputs and weights $\mathbf{V}$ and $\mathbf{W}$.

The RLS parameters numerical values are given below in Eq. (4.32), showing their orders and dimensions for both longitudinal and lateral controllers. The values were registered at simulation time $t = 15$ sec. $\hat{\mathbf{F}}_{\text{lon}}$ is a 6×6 state matrix, corresponding to 6 longitudinal states included in the state vector $\mathbf{x}_{\text{lon}}$; $\hat{\mathbf{G}}_{\text{lon}}$ is a 6×1 control matrix, corresponding to the elevator control input δ_e , and the covariance matrix is a 7×7 matrix, corresponding to the sum of states and control input. The same logic applies to the lateral parameters $\hat{\mathbf{F}}_{\text{lat}}$, $\hat{\mathbf{G}}_{\text{lat}}$ and $\mathbf{L}_{\text{lat}}$, where the number of lateral states is 5, and the control inputs are the ailerons δ_a and the rudders δ_r . As we can see from Eq. (4.32), the state matrix $\hat{\mathbf{F}}_{\text{lon}}$ and $\hat{\mathbf{F}}_{\text{lat}}$ are close to the identity matrix, which is the right state space representation in discrete time, thus showing that the RLS estimation is performed correctly. Also, the diagonal values of the covariance matrices $\mathbf{L}_{\text{lon}}$ and $\mathbf{L}_{\text{lat}}$ are high and positive, showing the positive definiteness of the matrices during estimation. Example values of the NN weights are not given here, as their dimensions are quite big and reach 7×30 , depending on the number of neural network inputs, outputs and the number of neurons.

$$\begin{aligned}
\hat{\mathbf{F}}_{\text{lon}} &= \begin{pmatrix} 0.985 & -0.0049 & -0.006 & -0.0054 & -0.0015 & -0.0013 \\ 0.245 & 1.04 & 0.0973 & 0.0924 & 0.0108 & 0.0107 \\ 0.0006 & -0.0003 & 0.999 & -0.0011 & 0 & 0 \\ -0.0002 & -0.0002 & -0.0012 & 0.999 & 0.0001 & 0 \\ -0.0312 & -0.0197 & 0.0893 & 0.0926 & 1 & -0.0038 \\ -0.0178 & 0.0002 & -0.0534 & -0.0559 & -0.0181 & 1 \end{pmatrix} \\
\hat{\mathbf{G}}_{\text{lon}} &= \begin{pmatrix} -0.0974 \\ -0.139 \\ -0.101 \\ -0.0994 \\ -0.138 \\ -0.0818 \end{pmatrix} \\
\mathbf{L}_{\text{lon}} &= \begin{pmatrix} 225 & 7.57 & -44.1 & -11.8 & -22.6 & -2.86 & 16.8 \\ 7.57 & 0.557 & -12.1 & -6.35 & -0.570 & 0.0019 & 5.46 \\ -44.1 & -12.1 & 963 & -0.736 & -4.08 & -2.13 & 1.01 \\ 11.8 & -6.35 & -0.736 & 965 & 5.42 & -0.807 & 0.322 \\ -22.6 & -0.57 & -4.08 & 5.42 & 2.47 & 0.364 & -0.0424 \\ -2.86 & 0.0019 & -2.13 & -0.807 & 0.364 & 0.0745 & 0.333 \\ 16.8 & 5.46 & 1.01 & 0.322 & -0.0424 & 0.333 & 965 \end{pmatrix} \quad (4.32) \\
\hat{\mathbf{F}}_{\text{lat}} &= \begin{pmatrix} 0.977 & 0.241 & -0.0595 & 0.07 & -0.123 \\ -0.0018 & 0.97 & -0.0103 & 0.0062 & -0.0065 \\ -0.0029 & -0.0232 & 0.924 & 0.0138 & -0.0122 \\ -0.0013 & 0.0002 & -0.0184 & 1.01 & -0.009 \\ -0.0015 & -0.0187 & -0.0195 & 0.0105 & 0.989 \end{pmatrix} \\
\hat{\mathbf{G}}_{\text{lat}} &= \begin{pmatrix} 0.0181 & -0.188 \\ 0.0996 & 0.0559 \\ 0.110 & 0.0542 \\ 0.103 & 0.0211 \\ 0.105 & 0.0132 \end{pmatrix} \\
\mathbf{L}_{\text{lat}} &= \begin{pmatrix} 10.3 & -42.7 & -4.99 & -25.3 & 35.5 & -3.22 & -29.4 \\ -42.7 & 527 & -4.78 & 58.2 & -47.2 & -8.99 & -27.3 \\ -4.99 & -4.78 & 580 & -16.7 & -22.1 & -0.988 & -2.49 \\ -25.3 & 58.2 & -16.7 & 77.9 & -118 & 18.5 & 39.7 \\ 35.5 & -47.2 & -22.1 & -118 & 190 & -20 & -22.5 \\ -3.22 & -8.99 & -0.988 & 18.5 & -20 & 579 & -5.51 \\ -29.4 & -27.3 & -2.49 & 39.7 & -22.5 & -5.51 & 565 \end{pmatrix}
\end{aligned}$$

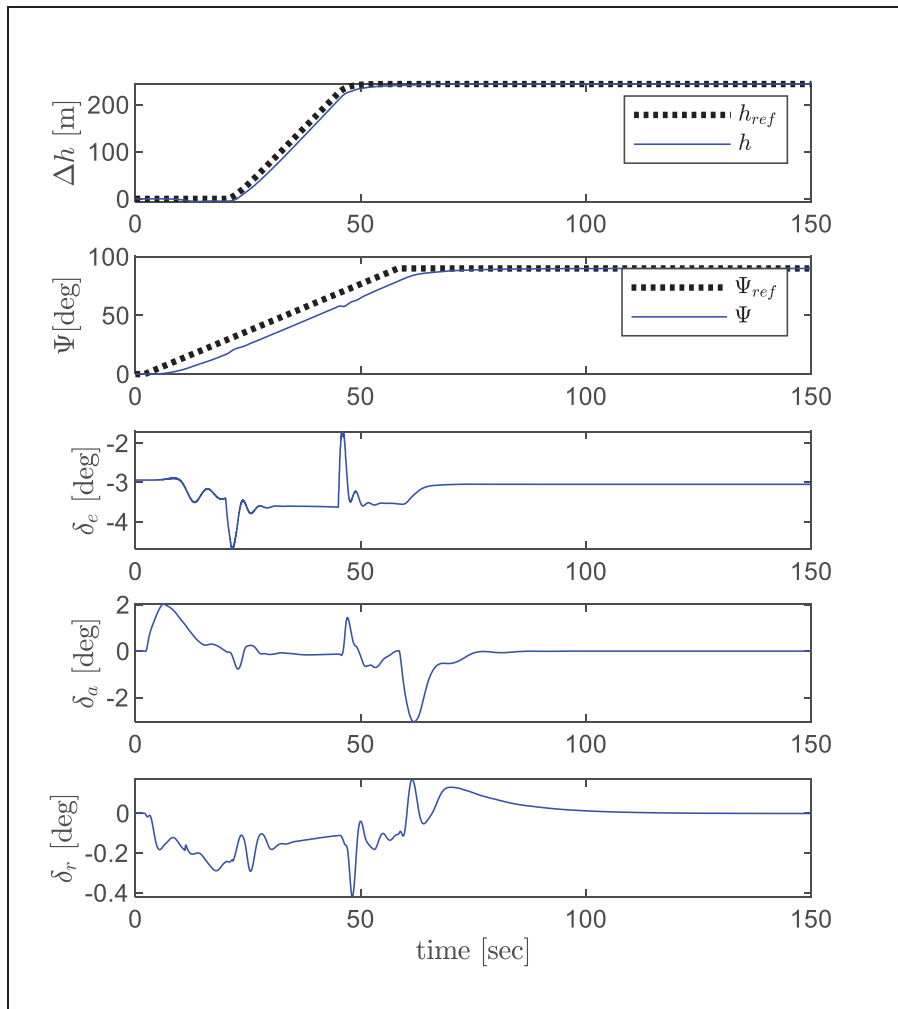


Figure 4.9 Altitude and heading command and control surface responses for coordinated turn and climb at 35,000 ft and 290 knots

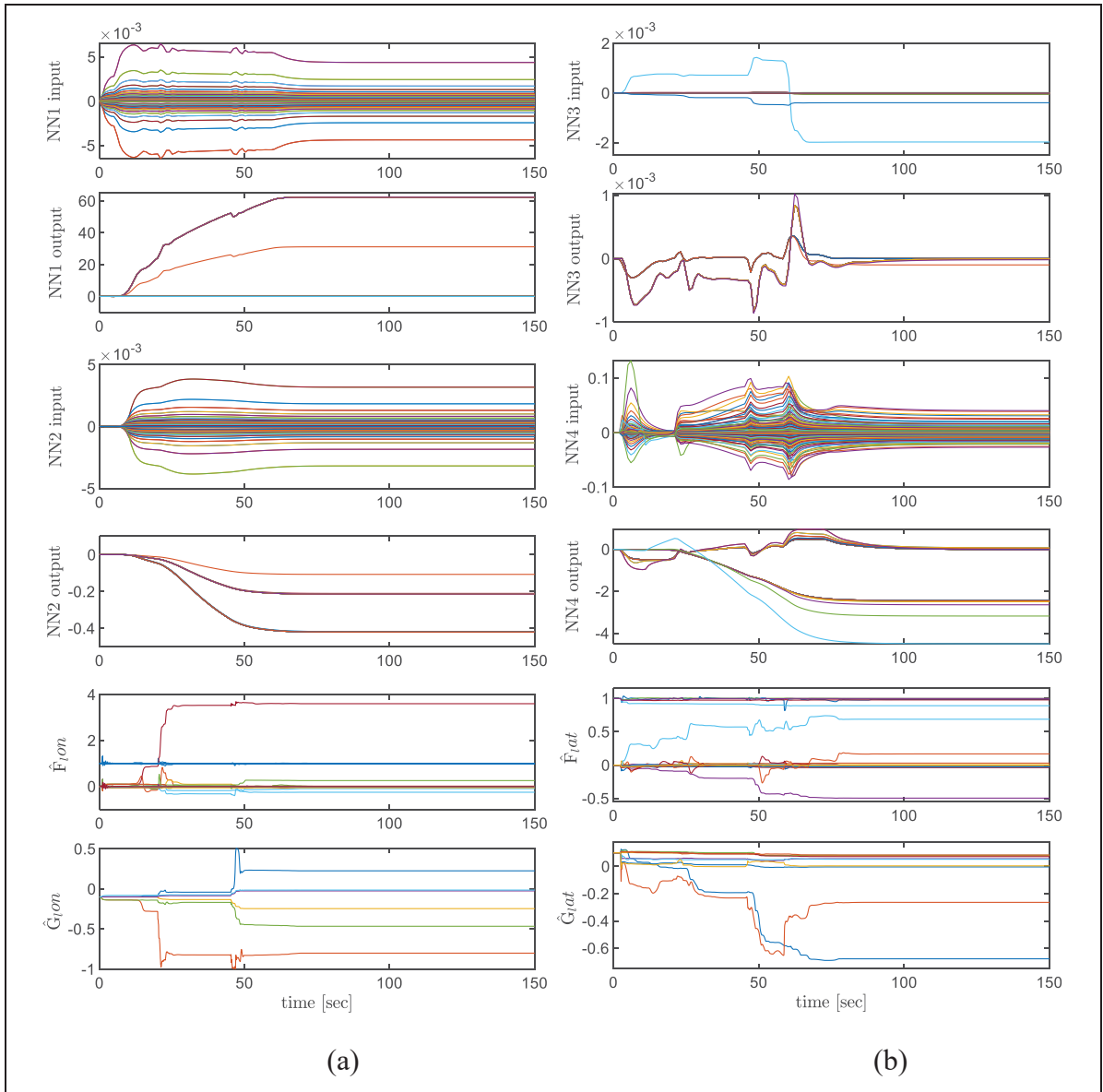


Figure 4.10 Adaptive hyperparameter variations for (a) longitudinal controllers and (b) lateral controllers

For longitudinal and lateral controllers, the adaptive hyperparameters are the NN weight matrices $\mathbf{V}$ and $\mathbf{W}$ and the RLS estimated matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$. Each coloured curve corresponds to the variation of an element in the matrices for the simulation length. The number of curves corresponds to the number of elements in the matrices.

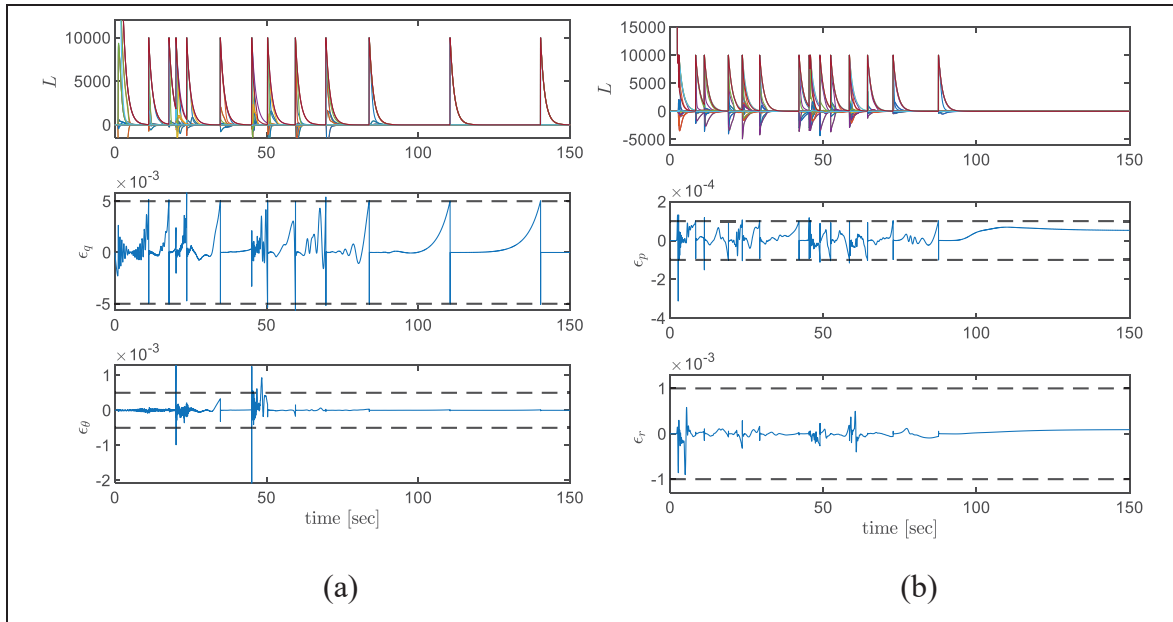


Figure 4.11 RLS hyperparameter variations during Altitude and Heading command for
(a) longitudinal and (b) lateral states

Each colored curve corresponds to an element of the covariance matrices $\mathbf{L}$. The number of curves corresponds to the number of elements included in the covariance matrix.

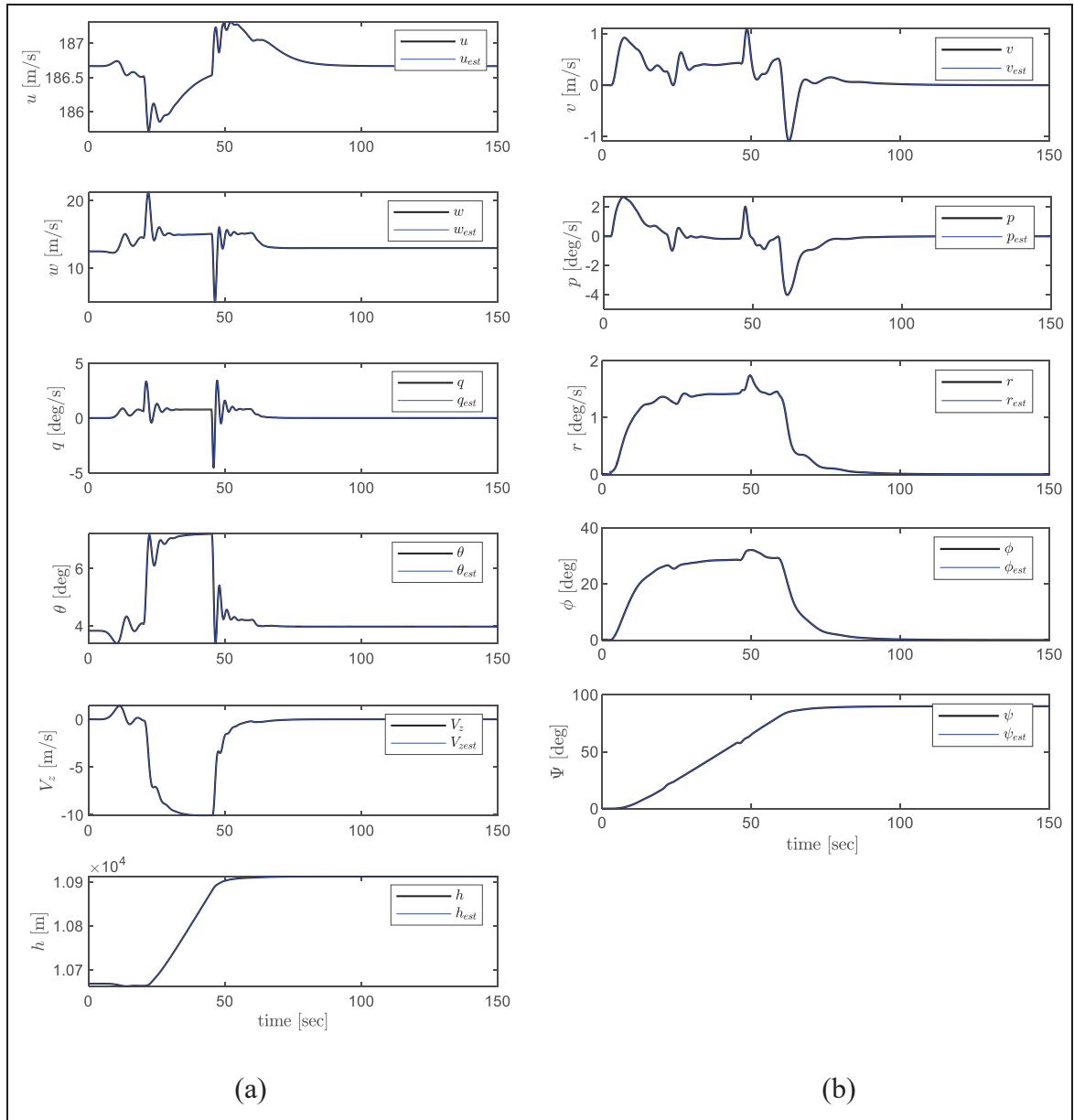


Figure 4.12 RLS estimation during Altitude and Heading command for (a) longitudinal and (b) lateral states

Blue lines correspond to RLS estimated states, while black lines correspond to actual states.

4.5.4 Validation for the Cessna Citation X cruise flight envelope

The CSAS and the autopilots were tested across the Cessna Citation X cruise operational envelope. The cruise flight envelope was discretized into a set of flight points (trim conditions). As seen in Figure 4.13, a total of 64 representative cruise flight conditions were defined for altitudes ranging from 5,000 ft to 40,000 ft, while the calibrated airspeed was varied from 170 to 330 knots. Each CSAS and autopilot were tested, with and without their adaptive elements, as seen in Figure 4.14 through Figure 4.19. Overall, the controllers with adaptive elements exhibit improved tracking performance, following the reference commands more closely than their non-adaptive counterparts.

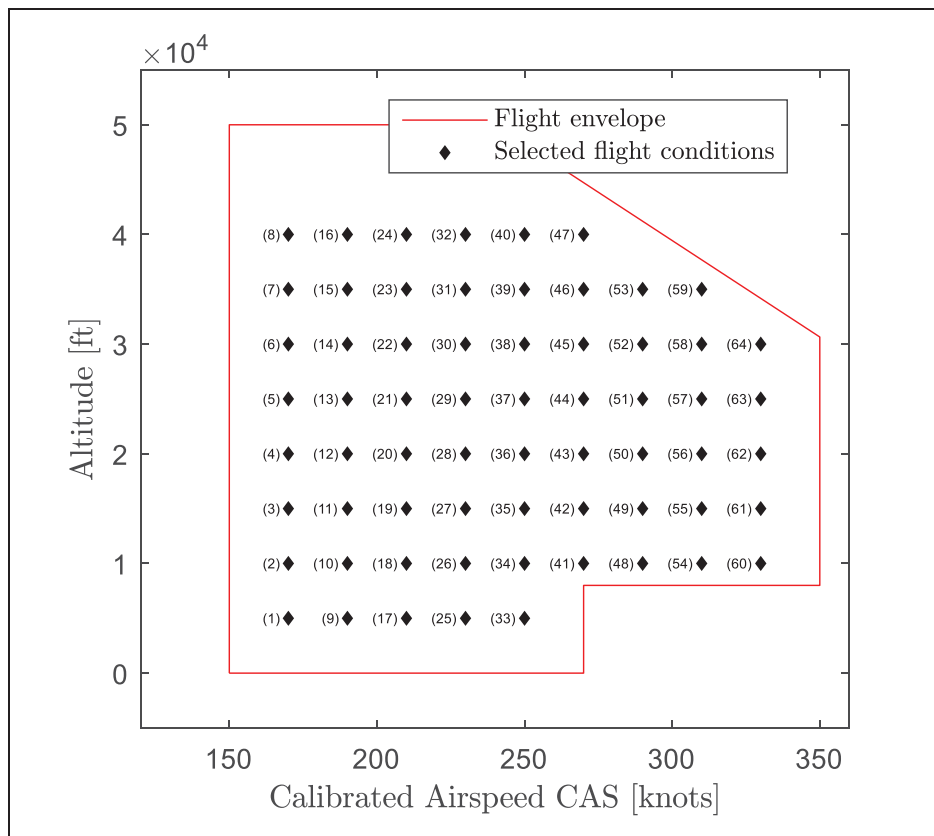


Figure 4.13 Cessna Citation X flight envelope expressed in terms of 64 representative cruise conditions considered for validation

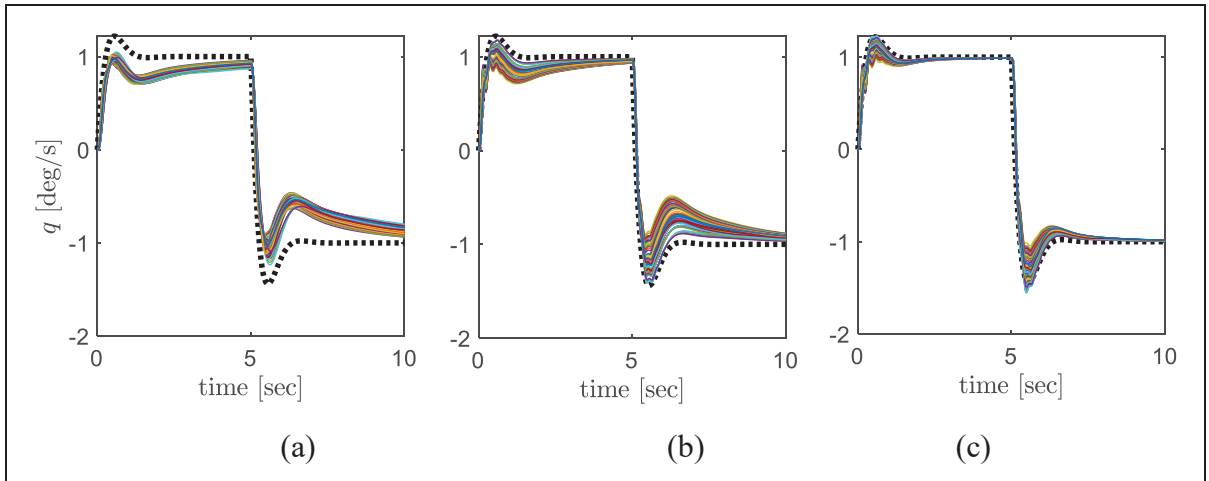


Figure 4.14 Pitch rate q variation with time responses for 64 cruise conditions for: (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations

The dotted line corresponds to the reference pitch rate signal, while the colored lines correspond respectively to the pitch rate response for each flight condition. The same reference signal was applied throughout the 64 cruise conditions.

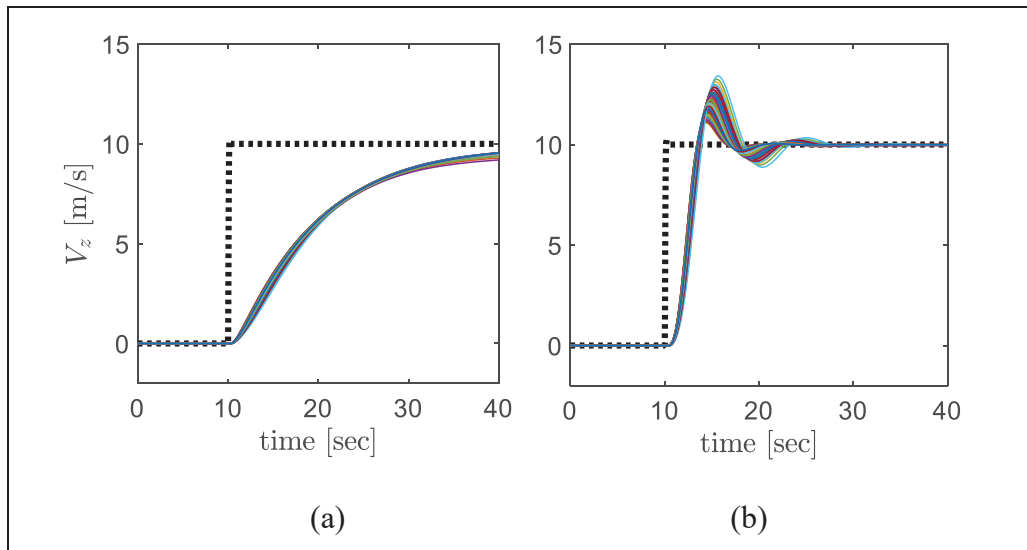


Figure 4.15 Vertical speed variations with time responses for 64 cruise conditions using: (a) PID, (b) PID – NN controller configurations

The dotted line corresponds to the reference vertical speed signal, while the colored lines correspond respectively to the vertical speed response for each flight condition.

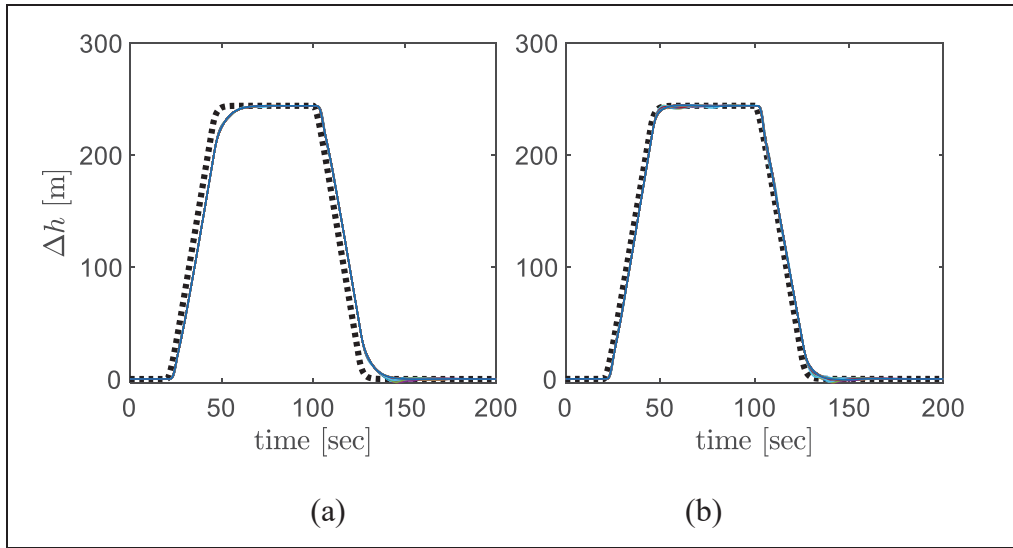


Figure 4.16 Altitude variations with time responses for 64 cruise conditions using: (a) PID, (b) PID – NN controller configurations

The dotted line corresponds to the reference altitude signal, while the colored lines correspond respectively to the altitude response for each flight condition.

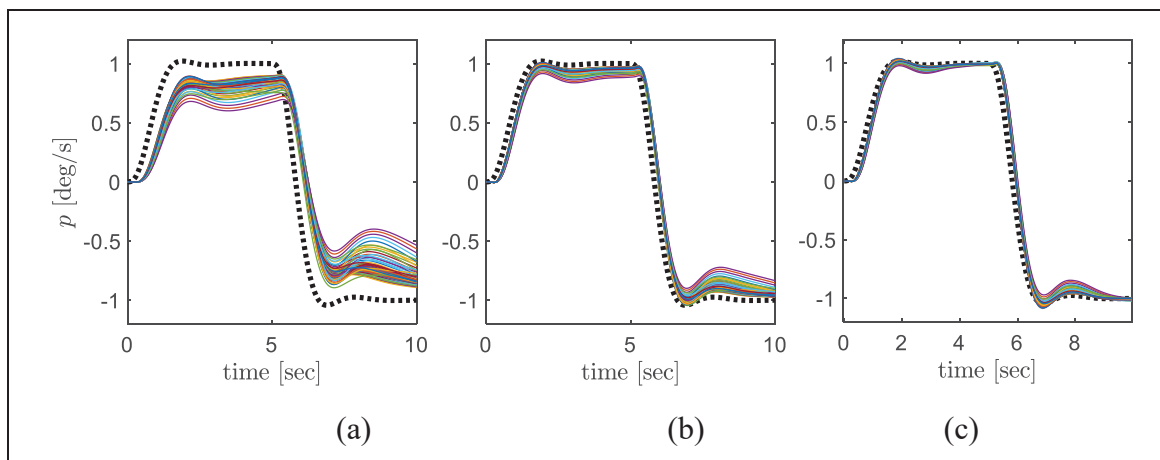


Figure 4.17 Roll rate p variations with time responses for 64 cruise conditions using : (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations

The dotted line corresponds to the reference roll rate signal, while the colored lines correspond respectively to the roll rate response for each flight condition.

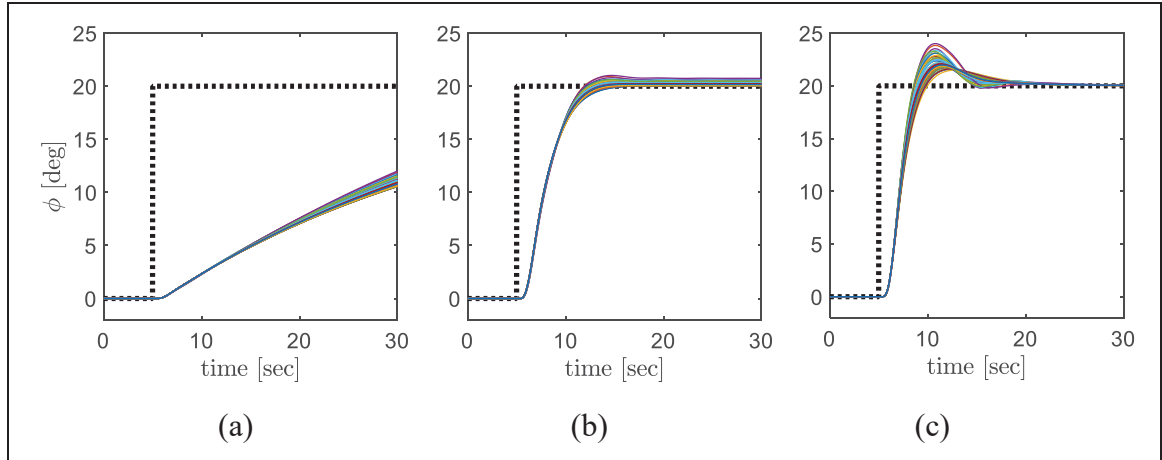


Figure 4.18 Roll ϕ variation with time responses for 64 representative cruise conditions: (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations

The dotted line corresponds to the reference roll angle signal, while the colored lines correspond respectively to the roll angle response for each flight condition.

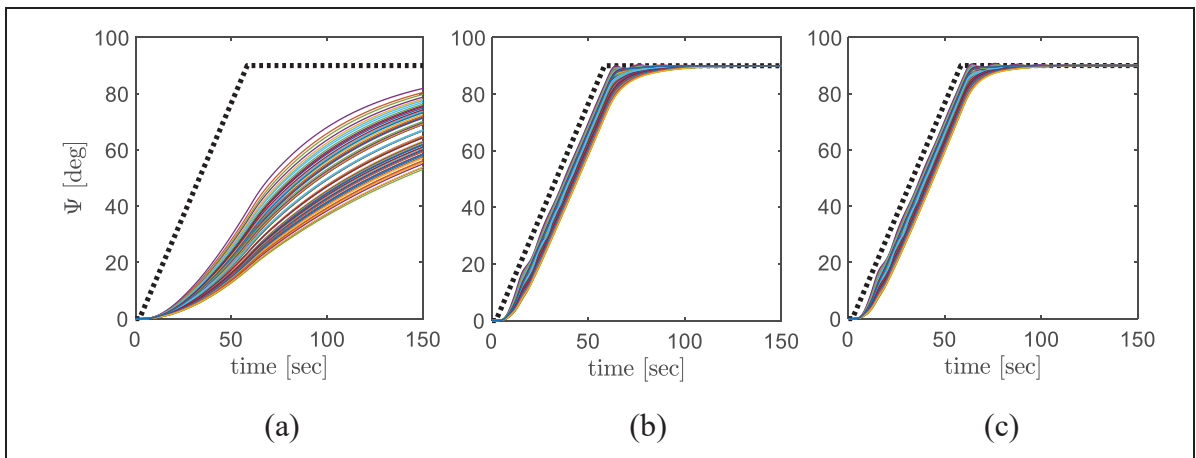


Figure 4.19 Heading control for 64 cruise conditions for : (a) PID, (b) PID and DI and (c) PID – DI and NN controller configurations

The dotted line corresponds to the reference heading signal, while the colored lines correspond respectively to the heading response for each flight condition. The same reference signal was applied for the three configurations (a), (b) and (c).

4.5.5 Performance results

The CSAS and Autopilot performances have been summarized in the following Table 4.13. The inner loop CSAS, including the pitch rate and roll rate responses, present damping and frequency characteristics, as they are oscillatory motions related to the short period and Dutch roll dynamics. Comparing the results with the flight quality specifications provided by (Department of Defense, 1997), we can conclude the inner loop controllers respect the Level 1 flight quality and are therefore suitable to implement within the trajectory tracking controller. The vertical speed overshoots between 11% to 34%, depending on the flight conditions; however, the settling and rising time are small enough for longitudinal maneuver. Also, the altitude and the heading autopilots present negligible overshoots and small enough settling times, which are the requirement for smooth flight-level change and coordinated turns, thus ensuring passenger comfort.

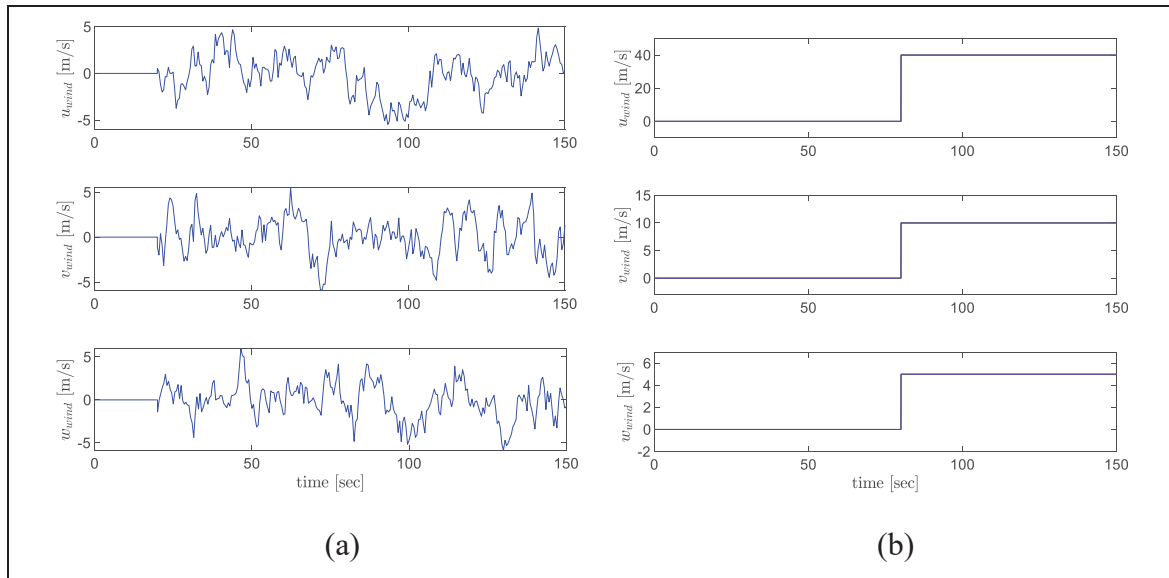
Table 4.13 CSAS and Autopilot transient performance results

Signal	Damping	Frequency, rad/s	Rising time, sec	Settling time, sec	Overshoot, %
Pitch rate controller	0.59 – 0.77	4.3 – 5.3	0.25 – 0.29	1 – 1.85	6 – 28
Roll rate controller	0.33– 0.47	2.8 – 3.03	1.37 – 1.49	1.5 – 3.5	4.3 %<
Roll angle controller	-	-	8 – 10	13.2 – 15.2	7.2 – 20
Vertical speed autopilot	-	-	5 – 8	10 – 15	11 – 34
Altitude autopilot	-	-	2.8 – 4.8	3.4 – 6.7	1%<
Heading autopilot	-	-	3.5 – 23	3 – 31	1%<

4.6 Robustness Tests

4.6.1 Wind and Turbulence Tests

The CSASs and the autopilots were tested in the presence of external disturbances such as wind gusts and turbulence. The disturbances were simulated using a Dryden turbulence model, and they were directly implemented on the aircraft model dynamics. Figure 4.20 illustrates the wind and turbulence profiles, while Figure 4.21 shows the coordinated altitude and heading tracking for 64 flight conditions.



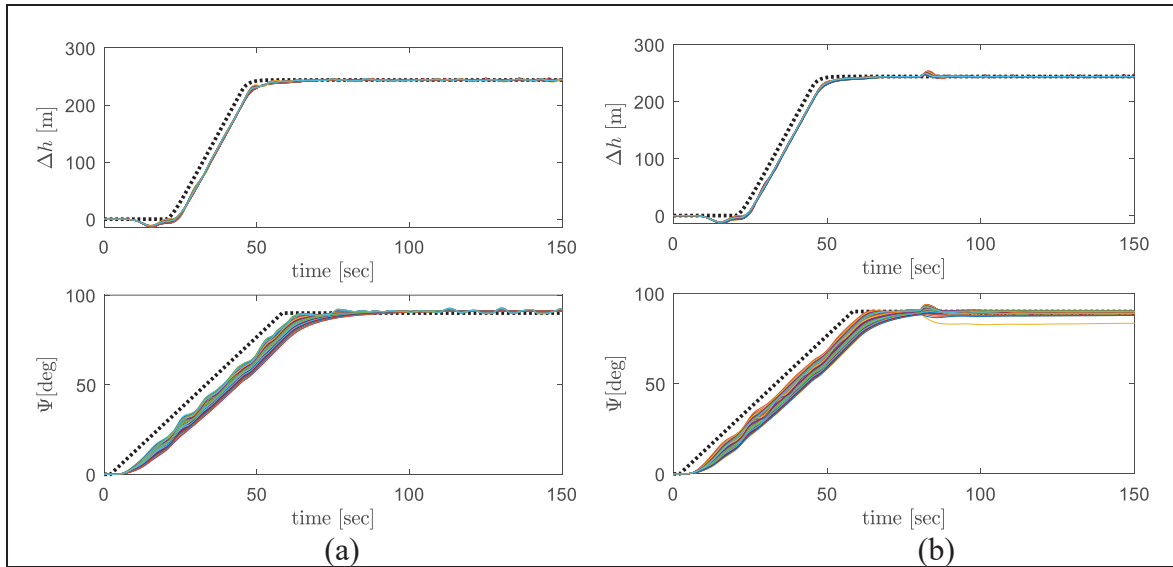


Figure 4.21 Dual altitude and heading command for 64 representative cruise conditions in the presence of (a) turbulence and (b) wind gusts

4.6.2 Flight Tests Comparison on the RAFS

Finally, the altitude and heading autopilots were compared with those of the real Cessna Citation X using flight tests on the Research Aircraft Flight Simulator (RAFS), and the comparison results are given in Figure 4.22. Saturators were added according to the aircraft performance limitations as specified in Table 4.2.

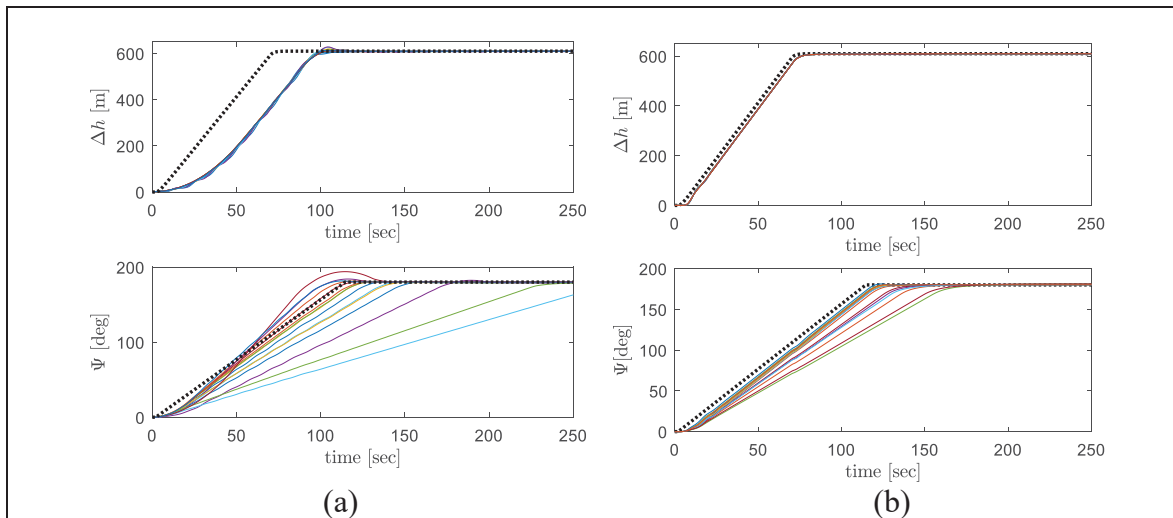


Figure 4.22 Altitude and heading time variations comparison between (a) flight tests on the RAFS and (b) Simulink simulations

4.7 Discussion of Results

4.7.1 CSAS and Autopilot architectures

The CSAS and autopilots developed for the Cessna Citation X contained three inner loop controllers for the pitch rate, roll rate, and roll angle control, and three outer loop autopilots for the vertical speed, altitude, and heading command. Their respective architectures were illustrated in Figure 4.2 to Figure 4.5, where Figure 4.2 shows the pitch rate and vertical speed autopilot configuration, Figure 4.3 shows the altitude autopilot, and Figure 4.4 and Figure 4.5 show the complete set of autopilot architectures, including the heading autopilot with the roll angle control and yaw rate stabilization. The CSAS design objectives were established to satisfy MIL-STD-1797A Level 1 flight quality criteria across the CCX cruise flight envelope (Department of Defense, 1997). The longitudinal system architecture included the pitch rate, vertical speed, and altitude autopilots, whereas the lateral directional system architecture integrated roll rate, roll angle, and heading control with a yaw rate stabilization. In addition, the yaw stabilization was fulfilled through dampening of the sideslip angle derivative.

The CSAS and e hybrid adaptive architectures combining PID control for baseline stability, Recursive Least Squares (RLS) based dynamic inversion, and Neural Network (NN) adaptation for residual error compensation. The Dynamic Inversion (DI) was applied to the CSAS controllers and heading autopilots, while the altitude and vertical speed autopilots used PID–NN configurations. The DI was not applied on the altitude and vertical speed command due to small value inversion, which may lead to zero division, and thus, numerical instabilities when inverting the control matrices $\mathbf{G}_{V_z}$ and $\mathbf{G}_h$.

4.7.2 Tuning and Adaptation

In sum, the overall controllers and autopilots were tuned and validated for 64 representative cruise flight conditions with varying altitudes from 1,500 m [5,000 ft] to 12,200 m [40,000 ft] and calibrated airspeeds between 314 km/h [170 knots] and 611 km/h [330 knots], as shown in Figure 4.13. Parameter tuning was fulfilled using sensitivity analysis and global

optimization, as shown in Figure 4.7 and Figure 4.8, to identify a single robust controller configuration. The RLS estimator provided accurate local linearization, with updates triggered only when the estimation error exceeded a predefined threshold to avoid divergence, as indicated in Figure 4.11. The influence of the forgetting factor κ was analyzed; small κ values yielded faster but noisier updates, while higher κ values improved stability. The convergence of RLS estimated states, covariance matrices, and hyperparameters are shown in Figure 4.11 to Figure 4.13, confirming reliable estimation of both longitudinal and lateral dynamics. The neural network compensates for modelling errors from the RLS estimations. The normalization coefficient K_{e_i} was applied to the NN i^{th} input to keep error magnitudes near zero and stabilize weight adaptation, as depicted in Figure 4.14. The NN demonstrated effective learning for both single-loop and coupled-loop configurations, thus accelerating transient responses and reducing steady-state tracking errors.

Regarding numerical singularities associated with nonlinear dynamic inversion, they may arise when the control effectiveness matrix becomes ill-conditioned. In fact, as seen in the numerical values of $\hat{\mathbf{G}}_{lon}$ and $\hat{\mathbf{G}}_{lat}$, from Eq. (4.32), their values can be very small and cross zero, as they constantly change through time. In the proposed adaptive autopilot architecture, this issue is mitigated by operating around trimmed flight conditions, and a covariance matrix update is performed. The inversion process is therefore performed on matrices whose determinants remain sufficiently far from zero, ensuring numerical stability and reliable control allocation. Furthermore, the adaptive neural network compensation and feedback control loops contribute to maintaining bounded system responses, which prevents the controller from driving the system toward singular configurations during the evaluated cruise flight conditions.

4.7.3 CSAS and Autopilot Performance

Three controller configurations were assessed: PID, PID–DI, and the full hybrid PID–DI–NN. As shown in Figure 4.14 to Figure 4.19, the hybrid adaptive design consistently achieved the best tracking and disturbance rejection performance. Pitch rate and roll rate controllers reached steady state within 1.9 s with errors below 0.1 %, meeting Level 1 flight quality by closely

following reference signals. For vertical speed control, response times ranged from 5 s to 8 s, and settling times from 10 s to 15 s, while NN adaptation reduced transient response time by about 30%. The altitude autopilot achieved smooth altitude capture with minimal overshoot, improved by derivative-based NN weight updates near the target altitude. Roll control performance improved, notably by the use of dynamic inversion, and the addition of NN further decreased the settling time from 12 to 15 s to 8–10 s and eliminated residual steady state errors. The heading autopilot accurately tracked a $3^\circ/\text{s}$ standard turn rate, maintaining altitude-heading maneuver coordination. The combined altitude-heading maneuver at 35,000 ft and 290 knots was illustrated in Figure 4.21, and it confirms the controller's ability to coordinate longitudinal and lateral control.

4.7.4 Robustness validation

Robustness testing under Dryden turbulence and wind-gust disturbances shown in Figure 4.21 has validated the system's resilience to uncertainties. Controller stability was preserved across all conditions without retuning or reconfiguration, and control-surface deflections remained within their physical required limits. Furthermore, the adaptive control elements demonstrated numerical stability and convergence. Thus, the adaptive autopilots maintained stable dynamics even when reference models were changed, confirming structural robustness.

Finally, a comparative validation using the Level D Research Aircraft Flight Simulator was shown in Figure 4.22. The proposed controller achieved faster and smoother trajectory tracking than the actual baseline Cessna Citation X autopilot. Despite minor actuator saturation limits in real flight, the adaptive-controller responses were nearly identical, demonstrating the realism and generalizability of the simulation results.

4.7.5 Stability assessment

The stability and convergence of the proposed adaptive control architecture follow the Lyapunov-based framework commonly used in adaptive neural network control. According to the Lyapunov stability criterion, a positive definite candidate function $\mathcal{L}$ is selected such that

$\mathcal{L} \geq 0$ and its time derivative satisfies $\dot{\mathcal{L}} \leq 0$, which guarantees boundedness of the closed-loop signals and convergence of the tracking error (Lavretsky & Wise, 2024). Neural network adaptive control laws derived using Lyapunov stability theory have been extensively established in the literature, where the weight adaptation laws are designed to ensure the negative semi-definiteness of the Lyapunov derivative and the boundedness of the neural network approximation error (Kim & Calise, 1997 ; Lewis et al., 1995 ; Narendra & Annaswamy, 1987). In addition, the parameter estimation used in the proposed controller relies on the Recursive Least Squares (RLS) algorithm, whose convergence properties and boundedness of the estimation error are well established in estimation theory (Haykin, 2013). Under appropriate excitation conditions, the RLS estimator ensures convergence of the estimated parameters and bounded estimation errors. Consequently, in this work, we assume that the combination of Lyapunov-based neural network adaptation and RLS parameter estimation guarantees bounded closed-loop signals and convergence of the tracking error for the proposed adaptive control system.

4.7.6 General remarks

In sum, the designed controller's main advantage is that it does not require any prior knowledge of the aircraft's mathematical model. Control inputs are solely generated from online RLS estimated local dynamics, allowing dual adaptation using dynamic inversion and NN to maintain controller high accuracy and robustness throughout the cruise flight envelope. By combining the proven reliability of classical PID control with modern adaptive estimation and learning, the proposed architecture achieved stable, certifiable, and high-performance flight path control suitable for advanced business jet autopilot systems.

4.7.7 Future works

Although the proposed adaptive autopilot architecture has been validated across 64 representative cruise flight conditions and evaluated under wind gust and Dryden turbulence disturbances, further robustness assessment could be conducted using Monte Carlo simulations. Such an approach would allow a systematic statistical analysis of controller

performance under randomized variations in aerodynamic parameters, actuator dynamics, sensor uncertainties, and environmental disturbances. In addition, the present work focuses primarily on cruise flight conditions. Future research will extend the proposed control framework to other flight regimes, including climb, descent, takeoff, and landing phases, where aircraft dynamics and operational constraints differ significantly. Investigating controller performance across the full flight envelope would further demonstrate the applicability of the adaptive architecture to complete flight missions. Finally, future developments may explore the integration of fault-tolerant control strategies to enhance system resilience in the presence of actuator or sensor failures. The inclusion of additional secondary control surfaces, such as spoilers, could also be investigated to improve control allocation and overall aircraft maneuverability.

4.8 Conclusions

4.8.1 Research Summary

This research mainly aims to:

- Address Loss of Control In-Flight risks through enhanced flight control robustness.
- Reduce the strong dependence of conventional controllers on gain scheduling and linear models.
- Improve adaptability across the cruise flight envelope, while maintaining Level 1 flight quality requirements.
- Bridge the gap between advanced adaptive control methods and practical implementation constraints, including flight quality requirements.
- The key contributions can be enumerated as follows:
- Designing hybrid adaptive controllers and autopilots combining PID, RLS-based dynamic inversion and neural network controllers.
- Eliminating gain scheduling by enabling a single controller configuration that is valid across the cruise flight envelope.

- Validating the designed controllers and autopilots throughout the cruise envelope of the Cessna Citation X by using a high-fidelity nonlinear flight dynamic model.
- Comparing the adaptive autopilots with the actual autopilots mounted on the Cessna Citation X.

4.8.2 Conclusions

From the simulation results, the following conclusions can be formulated:

- The hybrid adaptive controller and autopilots were successfully developed for Cessna Citation X flight dynamics. The proposed system included inner-loop CSAS controllers for pitch rate, roll rate, and yaw rate stabilization, and outer-loop autopilots for vertical speed, altitude, and heading control.
- The controller achieves MIL-STD-1797A Level 1 flight quality across 64 cruise conditions using a single configuration without gain scheduling.
- The RLS estimator and NN adaptation enable accurate online modelling and compensation of uncertainties, ensuring robust performance in the presence of wind gusts and turbulence.
- The proposed controller outperforms baseline PID and PID–RLS designs, providing faster response, improved tracking accuracy, and effective disturbance rejection.
- Validation with high-fidelity RAFS data confirms the practical feasibility, robustness, and real-world applicability of the approach.

CHAPTER 5

DISCUSSIONS AND RECOMMENDATIONS

5.1 Performance assesment

The performance of the proposed adaptive control architecture is quantitatively assessed using time-domain metrics, frequency-domain characteristics, and handling-quality criteria. The evaluation is conducted over multiple trim conditions spanning the cruise flight envelope, and the results are compared against baseline controllers and reference model requirements.

For the pitch-rate control loop, the closed-loop response exhibits a rise time between 0.35 s and 0.55s, with a settling time (2% criterion) consistently below 2.0 s across all tested conditions. The maximum overshoot remains below 8%, which is well within the typical Level 1 handling-quality recommendation of less than 10% for short-period dynamics. In comparison, the baseline non-adaptive controller shows overshoot values reaching 15–18% under the same condition.

Steady-state tracking errors are negligible, with mean absolute error values below 0.02 rad/s for pitch-rate commands and below 0.15 m/s for vertical-speed tracking. These results confirm that the adaptive elements effectively compensate for modeling uncertainties and flight-condition variations without introducing bias or drift.

The identified closed-loop short-period dynamics present a natural frequency ranging from 1.8 to 2.6 rad/s and a damping ratio between 0.45 and 0.75, depending on the flight condition. All operating points fall strictly within the MIL-STD-1797A Level 1 short-period boundaries, with a minimum damping margin of approximately 20% relative to the lower bound.

Under parametric variations and external disturbances (e.g., turbulence and aerodynamic coefficient perturbations), the adaptive controller maintains stable behavior with bounded adaptive signals. The Lyapunov-based adaptation law ensures that the estimated parameters

converge to bounded regions, with adaptation transients settling within 3–5 seconds after disturbance onset.

Quantitatively, disturbance-induced pitch-rate deviations are reduced by approximately 40–55% compared to the non-adaptive controller. Similarly, vertical-speed deviations during gust encounters are reduced by up to 50%, demonstrating a significant improvement in disturbance rejection capability.

Flight-quality metrics derived from simulation data confirm Level 1 handling qualities for all evaluated conditions. The Control Anticipation Parameter (CAP) remains within acceptable bounds, and the dropback magnitude is consistently below 0.1, indicating well-balanced short-period response and pilot-acceptable transient behavior.

The adaptive controller also reduces the dispersion of handling-quality metrics across the flight envelope. For instance, the standard deviation of short-period damping across trim points is reduced by approximately 35%, highlighting the benefit of adaptation in minimizing gain-scheduling requirements.

The quantitative results demonstrate that the proposed adaptive control architecture not only meets but exceeds standard flight-quality and robustness requirements. Compared to baseline fixed-gain controllers, the adaptive approach yields faster response, reduced overshoot, improved robustness margins, and significantly enhanced disturbance rejection, while maintaining strict compliance with MIL-STD-1797A Level 1 criteria across the entire tested envelope.

5.2 Thesis Discussions

This thesis addresses fundamental limitations identified in the literature on flight control systems, namely the strong dependence of classical and modern controllers on accurate modeling, extensive gain scheduling, and expert-driven tuning, as well as the lack of

certification-oriented validation methods for adaptive and intelligent controllers. The results obtained following the longitudinal, lateral-directional, and autopilot control studies demonstrate that these limitations can be effectively overcome through a hybrid adaptive control architecture that combines fixed-gain PID stabilization, Recursive Least Square-based dynamic inversion, and neural network adaptation. Rather than proposing incremental improvements to existing methodologies, this research establishes a coherent and scalable control method capable of operating across the full cruise flight envelope of a commercial business jet aircraft. This method could further be adapted to another similar type of business aircraft.

A central outcome of this research consists in the demonstration that a single controller configuration can provide stable, robust, and high-performance control across 64 cruise conditions without gain scheduling. This result represents a significant achievement in conventional flight control design practices, where envelope coverage is obtained by trimming the aircraft over the flight domain, and by designing its multiple controllers. The proposed architecture replaces gain scheduling with online estimation and adaptation, allowing the controller to continuously reconstruct the aircraft local dynamics model and compensate for real time uncertainties. The elimination of gain scheduling reduces design complexity, memory requirements, and verification effort, while simultaneously improving controller stability and maneuverability.

The effectiveness of the hybrid PID–RLS–NN architecture lies in the clear functional separation between its components. The fixed-gain PID controller ensures baseline stability and predictable behavior, thereby preserving classical control and safety. The RLS-based dynamic inversion provides online identification of local system dynamics using only variable increments, enabling adaptive inversion without reliance on a global aircraft model. The neural network layer complements this process by compensating residual inversion errors, actuator nonlinearities, and unmodeled dynamics that cannot be captured only by local linearization alone. Sensitivity and comparative studies consistently showed that neither RLS nor neural network adaptation alone is sufficient to achieve the desired level of robustness and

performance; instead, their combination yields a dual adaptation mechanism that enhances convergence, disturbance rejection, and tracking accuracy.

This thesis also contributes to practical strategies that are essential for the reliable deployment of adaptive control algorithms. Numerical stabilization mechanisms including threshold-based RLS updates, covariance matrix reinitialization, and bounding strategies during matrix inversion, proved their efficiency in preventing divergence, and in ensuring stable convergence over long time simulation and under disturbances. These measures enabled the controller to remain numerically bounded across a wide range of operating points, solving a common weakness of online identification and inversion methods.

A major distinguishing feature of this thesis is the explicit validation of adaptive controllers using certification-relevant flying quality metrics. Rather than evaluating performance solely in terms of tracking error or transient response, the proposed controllers were systematically assessed by use of MIL-STD-1797A Level 1 flight quality requirements for short-period, roll, and Dutch-roll dynamics. Across all tested conditions, the adaptive controllers consistently met Level 1 criteria while maintaining low steady-state errors, acceptable settling times, and unsaturated actuator behavior. This result demonstrates that nonlinear adaptive controllers can be designed and tuned to satisfy established flying quality standards, thereby bridging the gap between advanced adaptive control research and certification-oriented flight control design.

The scalability of the proposed architecture was further demonstrated through its integration into a complete control stability augmentation system and an outer-loop autopilot. The successful coordination of the pitch rate, roll rate, yaw stabilization, vertical speed, altitude, and heading control confirms that the hybrid adaptive method is not applied only to isolated inner-loop controllers, but it can support higher-level flight path control objectives. Design decisions were made to preserve the model numerical robustness, such as applying dynamic inversion where estimation sensitivity was acceptable and retaining PID–NN structures, for which matrix inversion became ill-conditioned.

The results show that the hybrid PID–RLS–NN dynamic inversion architecture provides a robust, scalable, and certifiable solution for adaptive flight control of a business aircraft. By combining the reliability and stability of classical control with online estimation and learning-based adaptation, this thesis demonstrates a practical pathway toward intelligent flight control systems that meet both performance and certification requirements.

5.3 Recommendations

Future work should extend the proposed control architecture beyond cruise flight into more demanding operating flight regimes, such as takeoff, climb, descent, approach, landing, and high-angle-of-attack maneuvers. These flight phases involve stronger nonlinearities, tighter actuator constraints, and more pronounced coupling effects, which will further challenge the estimation, inversion, and adaptation mechanisms. Extending model validation to these regimes would significantly strengthen the generality and operational relevance of the proposed control method.

While this thesis established compliance with Level 1 flying qualities through quantitative metrics, a complete assessment of aircraft handling qualities requires pilot-in-the-loop evaluations. Future studies should therefore incorporate piloted simulation campaigns using objective rating methodologies, such as Cooper–Harper, Neal–Smith, or bandwidth-based criteria. These evaluations would provide critical insight into pilot workload, controllability, and handling qualities, thereby complementing the quantitative flying quality analysis and supporting certification-oriented validation.

To further reduce the simulation-to-reality gap, hardware-in-the-loop validation is strongly recommended. Flight control real time hardware would allow assessment of computational latency, discretization effects, sensor noise, actuator rate limits, and real-time scheduling constraints. This work is essential to demonstrate that the proposed adaptive controllers can operate reliably under realistic implementation conditions, and to identify any additional constraints required for real-time deployment.

The demonstrated robustness of the proposed controllers under actuator loss-of-effectiveness motivates the extension of this work toward explicit fault-tolerant and reconfigurable control architectures. Integrating fault detection and isolation mechanisms with the adaptive control method would enable systematic handling of failures (such as sensors, actuators, aerodynamics, etc.) while preserving stability and controllability. The existing hybrid controller structure is particularly well suited for such extensions, as the adaptive layers can compensate for degraded dynamics while the PID baseline controller ensures stability during fault detection and transition phases.

Finally, future research should focus on finalizing certification-oriented design and validation procedures for adaptive flight control systems. This research includes defining robustness margins relative to MIL-STD-1797A boundaries, documenting repeatable tuning methodologies for adaptive parameters, and establishing verification processes that are verifiable. Such developments would help transform adaptive control from a research topic into an accepted and standardized hardware of safety-critical flight control systems.

CONCLUSION

This thesis investigated the design, validation, and certification relevance of adaptive flight control systems for commercial aircraft, with a particular focus on the Cessna Citation X business jet. Motivated by the limitations of classical and modern flight controllers identified in the literature, namely their reliance on accurate modeling, extensive gain scheduling, and expert-driven tuning, this research proposed and validated a hybrid adaptive control architecture capable of operating across the full cruise flight envelope without gain scheduling while maintaining certification-compatible performance.

The main contribution of this thesis is the development of a unified hybrid control method that combines fixed-gain PID controllers, Recursive Least Square-based dynamic inversion, and neural network adaptation. This architecture preserves the stability, transparency, and reliability of classical controllers while introducing online adaptation mechanisms that compensate for modeling uncertainties, nonlinearities, and time-varying dynamics. Unlike purely model-based or purely data-driven approaches, the proposed method achieves adaptation through a structured and interpretable combination of estimation, inversion, and learning, making it particularly well suited for safety-critical aerospace applications.

The proposed control architecture was systematically validated through a series of longitudinal, lateral-directional, and integrated autopilot control studies. Inner-loop controllers for pitch rate, roll rate, and yaw stabilization, as well as outer-loop autopilots for vertical speed, altitude, and heading control, were designed and evaluated using a high-fidelity nonlinear simulator of the Cessna Citation X. Across 64 cruise flight conditions spanning wide ranges of altitudes and airspeeds, the controllers consistently achieved accurate tracking, rapid convergence, and robust disturbance rejection using a single controller configuration. This result demonstrates that envelope-wide adaptive flight control can be achieved without gain scheduling, significantly reducing design complexity and improving scalability.

A key outcome of this research is the explicit validation of adaptive controllers using MIL-STD-1797A Level 1 flying quality requirements. Short-period, roll, and Dutch-roll dynamics were quantitatively assessed and shown to remain within Level 1 boundaries across the entire cruise envelope, even in the presence of wind gusts, Dryden turbulence, actuator noise, and actuator loss-of-effectiveness. Lyapunov-based stability analyses further confirmed boundedness and convergence of adaptive elements. By embedding certification-relevant performance metrics into the validation process, this thesis demonstrates that adaptive and learning-based controllers can be designed accordingly with established airworthiness standards.

From an implementation perspective, this thesis also addressed practical challenges associated with online estimation and dynamic inversion. Numerical stabilization strategies, including threshold-based RLS updates, covariance matrix reinitialization, and bounded inversion techniques, were shown to be essential for ensuring reliable convergence and preventing numerical instability. These contributions highlighted that successful adaptive control design requires not only theoretical soundness but also careful attention to numerical robustness and implementation details.

The scalability of the proposed architecture was demonstrated through its integration into a complete control stability augmentation system and autopilots. The coordinated control of longitudinal and lateral-directional dynamics model, including combined maneuvers such as simultaneous climb and turn, has confirmed that the hybrid adaptive method can support higher-level flight path control objectives. Comparative validation of the developed controller results using a Level D research flight simulator further demonstrated that the proposed adaptive autopilot achieved smoother and faster responses than the baseline onboard controller, reinforcing the practical relevance of the results.

In summary, this thesis provides a comprehensive and certification-aware contribution to the field of adaptive flight control. It demonstrates that a hybrid PID–RLS dynamic inversion–neural network architecture can deliver robust, high-precision, and envelope-wide flight

control for business aircraft while remaining compatible with established flying quality standards. By bridging the gap between classical control reliability and modern adaptive capabilities, this thesis establishes a credible pathway toward the deployment of adaptive flight control systems in safety-critical aviation environments and lays a solid foundation for future developments toward fault-tolerant, autonomous, and experimentally validated flight control architectures.

ANNEX I

MODEL PREDICTIVE CONTROLLER WITH ADAPTIVE NEURAL NETWORKS AND ONLINE STATE ESTIMATION FOR PITCH RATE CONTROL OF THE CESSNA CITATION X

Rojo Princy Andrianantara ^a, Georges Ghazi^a, and Ruxandra Botez ^a

^a Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE), System Engineering Department, École de technologie supérieure, 1100 Rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3

Article published in *AIAA SciTech 2024 Forum*, January 2024

<https://doi.org/10.2514/6.2024-0118>

Résumé

Cette recherche présente une technique de conception d'un contrôleur non linéaire adaptatif pour la vitesse de tangage de l'avion d'affaires Cessna Citation X. L'algorithme de contrôle proposé comprend trois composants principaux, à savoir un contrôleur à modèle de prédictif (MPC), une estimation d'état en ligne et un contrôleur adaptatif à réseau de neurones. L'estimation de la dynamique locale de l'avion est effectuée à l'aide de la méthode des moindres carrés récurrents (RLS). L'algorithme RLS consiste en un filtre adaptatif adapté à l'approximation d'état qui ne nécessite que des états et des incréments de contrôle. Le contrôleur MPC consiste en une optimisation de la trajectoire de contrôle sur un horizon de contrôle et un horizon de prédiction donnés. Une optimisation non linéaire est réalisée, incluant des contraintes, pour lequel une fonction de coût quadratique par rapport à la trajectoire de contrôle est minimisée. Enfin, un réseau de neurones adaptatif est ajouté, qui compense les erreurs dans les estimations d'état et améliore les performances globales du contrôleur. Le composant adaptatif consiste en un réseau neuronal à propagation directe à une couche cachée, dont les hyperparamètres sont mis à jour en ligne à l'aide du principe de rétropropagation sur les poids des couches d'entrée et de sortie. Les résultats de la simulation ont montré que l'algorithme de contrôle proposé était parfaitement capable de suivre un signal de référence donné définissant une dynamique de vol souhaitée pour le contrôle du taux de tangage. Le contrôleur de vol a été testé dans 63 conditions de vol différentes sur l'ensemble de l'enveloppe de vol du Cessna Citation X et a démontré de bonnes performances d'adaptation. L'algorithme de contrôle a ensuite été testé avec différents paramètres des éléments du contrôleur, et de très bonnes performances ont été obtenues pour des signaux de référence donnés sur le taux de tangage.

Abstract

This research presents a technique for designing an adaptive nonlinear controller for the pitch rate of the Cessna Citation X business jet aircraft. The proposed control algorithm includes 3

major components, namely a Model Predictive Controller (MPC), an online state estimation, and an adaptive neural network controller. The estimation of the aircraft local dynamics is performed using Recursive Least Square (RLS) method. The RLS algorithm consists of an adaptive filter suited for state approximation which only requires state and control increments. The MPC controller consists of an optimization of the control trajectory over a given control and predicted horizons. A constrained nonlinear optimization is fulfilled to solve the optimization problem, for which a quadratic cost function with respect to the control trajectory is minimized. Finally, an adaptive neural network is added, which compensates for the errors in the state estimations and improves the overall performance of the controller. The neural network component consists of a single hidden layer feedforward neural network, for which the hyperparameters are updated online using backpropagation principle on the weights of the input and output layers. The simulation results showed that the proposed control algorithm was perfectly capable of tracking a given reference signal defining a desired flight dynamic for the pitch rate control. The flight controller was tested on 63 different flight conditions across the flight envelope of the Cessna Citation X and demonstrated good adaptation performance. The control algorithm was then tested with different parameters of the controller elements, and very good performance was obtained for given reference signals on the pitch rate.

I.1 Introduction

Flight controllers remain one of the major components of aircraft systems, without which it is impossible for an aircraft to perform any maneuver. In fact, actual aircraft manufacturers are investing in aircraft flight controls in order to improve aircraft performance, instead of changing the overall structure of the aircraft. A flight controller should meet several requirements, one of which consists of its ability to adapt to changing environment. Indeed, aircraft are constantly subjected to uncertainties, which may affect their stability during flight operation. However, one of the major issues in flight controls consists of implementing such controllers, which can adapt themselves to any changes. An adaptive controller could compensate for the loss of effectiveness of control surfaces, such as elevators.

Actual flight controllers are still based on classical and modern control, such as Proportional Integral Derivative (PID) controllers or Linear Quadratic Regulator (LQR) (Nelson, 1998). Even though classical and modern control are powerful in terms of stability and robustness, they are not suited for adaptation. Indeed, one of their characteristics consists in designing the flight controller for a specific range of flight operating conditions. Firstly, the linearization of aircraft dynamics may neglect some important nonlinearities. Secondly, several configurations of the flight controllers are required to cover a large range of flight conditions. And thirdly, their application is only possible for specific conditions. Therefore, for a PID or an LQR controller, a gain scheduling is often required, which is a time-consuming task (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017).

More advanced control methods were proposed in order to deal with controller adaptation. Among them, we can cite the Dynamic Inversion (DI) (Lane & Stengel, 1988) and Model Predictive Control (MPC) (Eren et al., 2017) in the 1990's. These methods were able to deal

with the adaptation problem but require a precise knowledge of the aircraft dynamics. The Dynamic Inversion consists of expressing the control input with respect to the aircraft states and their derivatives (Ducard & Geering, 2008 ; Horn, 2019), while the MPC controller consists of an optimization of the control input in a given optimization window. The MPC controller may be related to the LQR control, in the fact that optimization is fulfilled by minimizing a given cost function. However, the DI and MPC methods are difficult to apply when the aircraft dynamics are complex. In fact, for Dynamic Inversion, the method requires the derivation of the output signal, while for the MPC controller, it requires a precise model of the aircraft in order to compute its optimal control signal (Camacho & Bordons, 2007). In both cases, a linearization of the aircraft dynamics is required for the design of the controller, which is not always applicable for nonlinear dynamics model.

One solution consists of using online estimates, which can be fulfilled using adaptive filters such as Recursive Least Square (RLS) methods (Haykin, 2013 ; Zhou et al., 2018). For nonlinear systems, the RLS algorithm allows to estimate the local dynamics of the aircraft, which can be further combined with Dynamic Inversion and MPC controllers (R. P. Andrianantara, Ghazi, et al., 2024). However, in previous studies, when combined with Dynamic Inversion, the parameters of the RLS algorithm were highly sensitive to its initialization parameters, as the Dynamic Inversion may conduct numerical instabilities. Therefore, additional mechanisms on the covariance matrix were necessary in order to guarantee its stability.

However, in this study, the RLS algorithm is combined with an MPC controller and only plays the role of state estimator. The use of the RLS filter was then much simpler than in the previous studies. Even though the proposed method shows very good closed-loop performance, it still presents a steady-state error, which may be due to the estimation error obtained from the RLS algorithm. One solution consists of adding an adaptation component, such as neural networks (N. Nguyen & Krishnakumar, 2007). In fact, Artificial Intelligence has been combined with classical methods in order to improve their performance, such as fuzzy logic (Grigorie, Botez, et al., 2012a ; S. Hosseini, Ghazi, & Botez, 2023a, 2023b), neural networks (S. Hosseini, Inga, et al., 2023 ; Mosbah et al., 2016) or evolutionary algorithms (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017). Artificial Neural Networks are universal approximators that are extensively used in regression problems, such as control. In the domain of system identification, neural networks were applied for the identification of complex flutter dynamics (Boely et al., 2011 ; Boely & Botez, 2010), aircraft dynamics (Zohreh Nejad et al., 2023) and nonlinear engine dynamics from flight tests (R. P. Andrianantara et al., 2021, 2023b ; Rojo Princy Andrianantara et al., 2021 ; Zaag et al., 2019). For control purposes, neural networks were able to adapt their hyperparameters in order to follow a reference signal (Bosworth & Williams-Hayes, 2007 ; Johnson & Calise, 2003 ; Soares & Burken, 2006).

Therefore, this paper presents the design of a Model Predictive Controller combined with a Neural network controller and an online state estimation for the pitch rate control of the Cessna Citation X business jet aircraft. The online state estimation is fulfilled using the RLS method, and the estimated dynamics are fed into the MPC controller for optimization. A constrained

nonlinear optimization algorithm is used in order to compute the optimal control input, and a feedforward neural network is used as an adaptive element. The controller was tested on several flight conditions, throughout the Cessna Citation X flight envelope, for which the objective consists in tracking a reference signal on the pitch rate.

The remainder of this paper is structured as follows. A description of the Cessna Citation X business jet aircraft and its longitudinal dynamics is given in Section I.2. The design methodology and the implementation of the control algorithm is given in Section I.3. Finally, simulation results are presented in Section I.4.

I.2 Aircraft Model and Simulation Platform

Cessna Citation X Research Flight Simulator and Simulink Interface

I.2.1 The Cessna Citation X is one of the fastest business jet aircraft in the world. At the LARCASE laboratory, a high-fidelity nonlinear simulation platform of the Cessna Citation X was developed in Matlab/Simulink for research purposes from a combination of complex flight dynamics modules (Ghazi & Botez, 2015a). This simulation platform includes a high-fidelity aerodynamic and propulsion model, various sensors and actuator models, and an atmosphere and wind model. Most of these modules included in the simulation platform were developed based on data acquired from a highly qualified Level D Research Aircraft Flight Simulator (RAFS) manufactured by CAE Inc. Thus, the MATLAB/Simulink simulation platform can be considered as a high precision tool for the development and validation of our proposed control algorithm.

In addition, the simulation platform includes various routines to trim and linearize the aircraft dynamics for specific flight conditions. Previous research conducted by the LARCASE research team has demonstrated that a high-fidelity model of the Cessna Citation X business jet allows for the design of effective LQR and H_∞ controllers for the entire flight envelope (Y. Boughari, Ghazi, et al., 2017a, 2017b ; Yamina Boughari, Botez, et al., 2017 ; Yamina Boughari, Ghazi, et al., 2017 ; Ghazi & Botez, 2015b).

Aircraft Nonlinear Dynamics Model

The longitudinal dynamics of the Cessna Citation X are considered in order to design the pitch rate control, for which only elevator commands are used. Also, the actuator is modeled using a second-order nonlinear dynamic. Based on these assumptions, the longitudinal dynamics of the aircraft can mathematically be represented as follows:

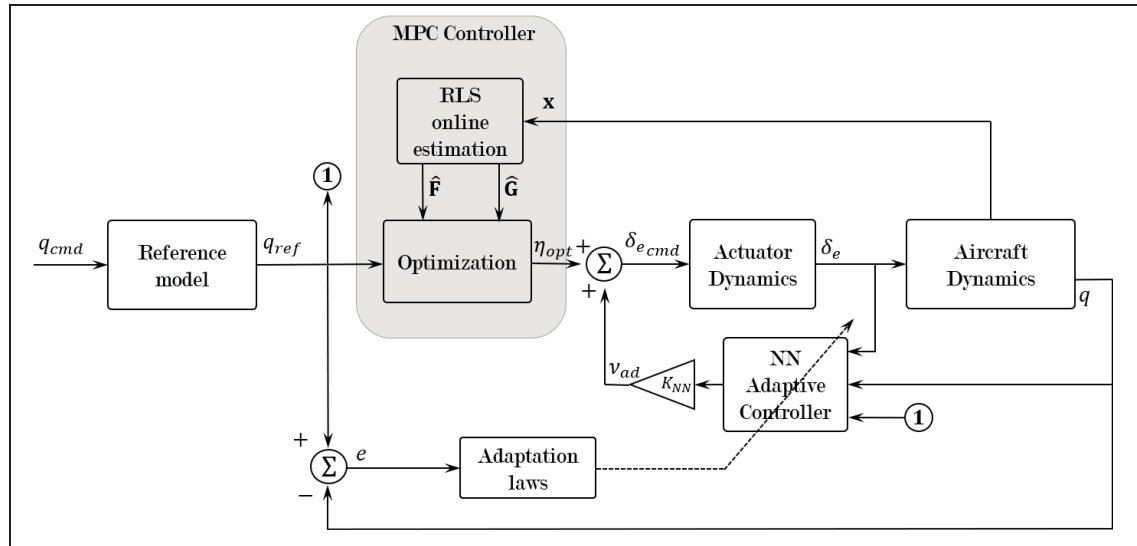
$$\begin{aligned}\dot{\mathbf{x}} &= \boldsymbol{\varphi}(\mathbf{x}, \eta) \\ y &= h(\mathbf{x})\end{aligned}\tag{A I-1}$$

where $\mathbf{x} = [u \quad w \quad q \quad \theta]^T$ is the state vector composed of the longitudinal component of the velocity, the vertical component of the velocity, the pitch rate, and the pitch angle, respectively,

$\eta = \delta_e$ is the elevator command, $\boldsymbol{\varphi}$ is a function describing the nonlinear longitudinal dynamics of the aircraft, y is the output of the model to be controlled, and h is a nonlinear function describing the relationship between the output of interest (i.e., the pitch rate) and the state vector $\mathbf{x}$. Equation (A I-1) is the nonlinear form of the state space representation of the aircraft dynamics, which also includes the actuator dynamics, and for which the nonlinear function $\boldsymbol{\varphi}$ is unknown or uncertain.

I.3 Design Methodology

The proposed control algorithm consists of a Model Reference Adaptive Controller (MRAC), whose architecture is shown in Figure A I-1. As seen in this figure, the control algorithm is composed of a Model Predictive Controller, an RLS state estimation and an adaptive controller, which is modeled using neural networks.



I.3.1

Figure A I-1 Architecture for the MPC Controller using Recursive Least Square Estimation and Adaptive Neural Network Controller

Aircraft Dynamics linearization

For aircraft systems, the local dynamics may be written in the state space form, where the state derivatives may be expressed in affine form with respect to the control input η (Stevens et al., 2015). Therefore, Eq. (A I-1) can be then rewritten as follows:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\eta \\ y &= q = \mathbf{C} \cdot \mathbf{x}\end{aligned}\tag{A I-2}$$

where $\mathbf{f}$ and $\mathbf{g}$ are nonlinear functions that represent the aircraft dynamics, and the output equation y corresponds to the pitch rate q . Finally, $\mathbf{C}$ is the output matrix corresponding to the pitch rate q and is equal to $[0 \ 0 \ 1 \ 0]$.

The next development consists of linearizing the aircraft dynamics at a given sample time applied in Figure A I-1.

Let's consider the affine relationship presented in Eq. (A I-1) and Eq. (A I-2) as follows:

$$\dot{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}, \eta) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\eta \quad (\text{A I-3})$$

Thus, the first order Taylor series expansion of Eq. (A I-3) around an initial point $(\mathbf{x}_0, \eta_0)$ may be written as follows:

$$\boldsymbol{\varphi}(\mathbf{x}, \eta) = \boldsymbol{\varphi}(\mathbf{x}_0, \eta_0) + \boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}(\eta - \eta_0) \quad (\text{A I-4})$$

where $\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0)$ is the value of $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ at $(\mathbf{x}_0, \eta_0)$, $\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}$ and $\boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}$ are the partial derivatives of $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ with respect to the states $\mathbf{x}$ and to the input η , respectively. From Eq. (A I-3), $\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0)$ can be obtained with Eq. (4.1). Then, the derivatives of $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ with respect to states $\mathbf{x}$ and to the input η are obtained from Eq. (A I-4) as follows:

$$\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta_0 \quad (\text{A I-4.a})$$

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\eta \quad (\text{A I-4.b})$$

$$\boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \eta} + \frac{\partial [\mathbf{g}(\mathbf{x})\eta]}{\partial \eta} = \mathbf{g}(\mathbf{x}) \quad (\text{A I-4.c})$$

Thus, rewriting the states and input in terms of their variation $\Delta \mathbf{x} = (\mathbf{x} - \mathbf{x}_0)$ and $\Delta \eta = (\eta - \eta_0)$ and replacing all terms obtained in Eq. (4.1) (4.2) and (4.3) in Eq. (A I-4), we obtain Eq. (A I-5):

$$\boldsymbol{\varphi}(\mathbf{x}, \eta) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta_0 + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\eta_0 \right) \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A I-5})$$

Let's rewrite the time derivative of the state vector $\mathbf{x}$ in Eq. (A I-3) as the sum state derivation at initial condition $(\mathbf{x}_0, \eta_0)$ and perturbation around.

$$\dot{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}, \eta) = \dot{\mathbf{x}}_0 + \Delta \dot{\mathbf{x}} \quad (\text{A I-6})$$

As $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ is given by both Eq. (A I-5) and Eq. (A I-6), and $\dot{\mathbf{x}}_0 = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta_0$, and from the variation of the state derivative $\Delta \dot{\mathbf{x}}$ can be expressed as follows:

$$\Delta \dot{\mathbf{x}} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \eta \right)_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A I-7})$$

Considering the following notations:

$$\mathbf{F}_{\mathbf{x}_0, \eta_0} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \eta \right)_{\mathbf{x}_0, \eta_0} ; \mathbf{G}_{\mathbf{x}_0, \eta_0} = \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \quad (\text{A I-8})$$

Then, Eq. (A I-7) can be rewritten in the following compact form:

$$\Delta \dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{G}_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A I-9})$$

Online State Estimation

I.3.2

In practice, the computation of $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ and $\mathbf{G}_{\mathbf{x}_0, \eta_0}$ in Eq. (A I-8) is analytically difficult, especially for nonlinear functions, and even more difficult when the functions $\mathbf{f}$ and $\mathbf{g}$ are unknown. Therefore, one method consists of estimating these values by using numerical methods such as the Recursive Least Square algorithm, which allows to approximate these values.

Let's rewrite Eq. (A I-9) in the discrete form as follows:

$$\Delta \mathbf{x}_{k+1} = \tilde{\mathbf{F}}_k \Delta \mathbf{x}_k + \tilde{\mathbf{G}}_k \Delta \eta_k \quad (\text{A I-10})$$

where $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{G}}_k$ are now the discrete forms of $\mathbf{F}$ and $\mathbf{G}$. Writing Eq. (A I-10) in a compact form, we obtain the following matrices relationship:

$$\Delta \mathbf{x}_{k+1} = [\Delta \mathbf{x}_k \quad \Delta \eta_k] \begin{bmatrix} \tilde{\mathbf{F}}_k^T \\ \tilde{\mathbf{G}}_k^T \end{bmatrix} \quad (\text{A I-11})$$

By replacing $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{G}}_k$ with their estimates $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$, we obtain:

$$\Delta \hat{\mathbf{x}}_{k+1} = [\Delta \mathbf{x}_k \quad \Delta \eta_k] \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix} \quad (\text{A I-12})$$

Considering the matrix of parameters $\hat{\Theta}_k = \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix}$ and the vector of previous increments $\mathbf{X}_k = [\Delta \mathbf{x}_k \quad \Delta \eta_k]^T$, the state variation $\Delta \hat{\mathbf{x}}_{k+1}$ may be predicted as follows:

$$\Delta \hat{\mathbf{x}}_{k+1} = \mathbf{X}_k^T \hat{\boldsymbol{\Theta}}_k \quad (\text{A I-13})$$

where $\hat{\boldsymbol{\Theta}}_k$ corresponds to the estimates of $\boldsymbol{\Theta}_k$ and $\Delta \hat{\mathbf{x}}_{k+1}$ corresponds to the estimates of $\Delta \mathbf{x}_{k+1}$. The vector of parameters $\hat{\boldsymbol{\Theta}}_k$ can be determined online using the RLS method. Let's consider the error matrix $\boldsymbol{\epsilon}_k$, also called innovation error, as the difference between the actual states vector and its estimate:

$$\boldsymbol{\epsilon}_k = \Delta \mathbf{x}_k - \Delta \hat{\mathbf{x}}_k \quad (\text{A I-14})$$

Then, the vector of parameters $\hat{\boldsymbol{\Theta}}_k$ may be updated numerically as follows (Haykin, 2013):

$$\hat{\boldsymbol{\Theta}}_k = \hat{\boldsymbol{\Theta}}_{k-1} + \frac{\mathbf{L}_{k-1} \mathbf{X}_k}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \boldsymbol{\epsilon}_k \quad (\text{A I-15})$$

where

$$\mathbf{L}_k = \frac{1}{\kappa} \left(\mathbf{L}_{k-1} - \frac{\mathbf{L}_{k-1} \mathbf{X}_k \mathbf{X}_k^T \mathbf{L}_{k-1}}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \right) \quad (\text{A I-16})$$

where $\mathbf{L}_k$ is the covariance matrix, and κ is the forgetting factor. The two matrices $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ in $\hat{\boldsymbol{\Theta}}_k$ represent the local behavior of the aircraft dynamics.

A key feature of the RLS method lies in the fact that the estimation of $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ is fully based on the increments of states and control inputs, thus making it suitable for online estimation.

The RLS parameters to be initialized are the forgetting factor κ , the matrix of covariance $\mathbf{L}_0$, and the matrix of parameter $\boldsymbol{\Theta}_0$.

Model Predictive Controller

For the design of a pitch rate controller, a Model Predictive Controller (MPC) is proposed, for which an online estimation of the aircraft dynamics is applied using the RLS algorithm. In fact, an MPC controller consists of optimizing the control trajectory $\Delta \mathbf{U}$ of length N_c over an optimization window N_p , where N_c and N_p are also referred as the control horizon and the predicted horizon, respectively. Therefore, the control trajectory $\Delta \mathbf{U}$ consists of a vector of future control inputs at sample time $k, k+1, \dots, k+N_c-1$, which is calculated using an optimization process. Thus, the vector $\Delta \mathbf{U}$ may be defined as follows (Wang, 2009).

$$\Delta \mathbf{U} = [\Delta \eta_k \ \Delta \eta_{k+1} \ \dots \ \Delta \eta_{k+N_c-1}]^T \quad (\text{A I-17})$$

where $\Delta \eta_k$ is the calculated control input at a sample time k .

The corresponding future state variables may be set as $\mathbf{x}_k|_k, \mathbf{x}_{k+1}|_k, \dots, \mathbf{x}_{k+m}|_k, \dots, \mathbf{x}_{k+N_p}|_k$ of length N_p . In that case, $\mathbf{x}_{k+m}|_k$ is the predicted state variable at a sample time $k + m$ with given current system information $\mathbf{x}_k$.

The future state variables $\mathbf{x}_{k+m}$ are calculated sequentially with respect to the future control parameters $\Delta\mathbf{U}$, using the local estimated state space model $(\hat{\mathbf{F}}_k, \hat{\mathbf{G}}_k)$ at a sample time k , where $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ were obtained using the RLS estimation method explained in the previous section. The future state variables may be computed sequentially as follows (Wang, 2009):

$$\begin{aligned}
 \mathbf{x}_{k+1}|_k &= \hat{\mathbf{F}}_k \mathbf{x}_k + \hat{\mathbf{G}}_k \Delta\eta_k \\
 \mathbf{x}_{k+2}|_k &= \hat{\mathbf{F}}_k \mathbf{x}_{k+1} + \hat{\mathbf{G}}_k \Delta\eta_{k+1} \\
 &= \hat{\mathbf{F}}_k^2 \mathbf{x}_k + \hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \Delta\eta_k + \hat{\mathbf{G}}_k \Delta\eta_{k+1} \\
 &\vdots \\
 \mathbf{x}_{k+N_p}|_k &= \hat{\mathbf{F}}_k^{N_p} \mathbf{x}_k + \hat{\mathbf{F}}_k^{N_p-1} \hat{\mathbf{G}}_k \Delta\eta_k \\
 &\quad + \hat{\mathbf{F}}_k^{N_p-2} \hat{\mathbf{G}}_k \Delta\eta_{k+1} + \dots + \hat{\mathbf{F}}_k^{N_p-N_c} \hat{\mathbf{G}}_k \Delta\eta_{k+N_c-1}
 \end{aligned} \tag{A I-18}$$

From the predicted state variables, the predicted output variables are obtained by introducing the output matrix, $\mathbf{C} = [0 \ 0 \ 1 \ 0]$ corresponding to the pitch rate.

$$\begin{aligned}
 y_{k+1}|_k &= \mathbf{C} \hat{\mathbf{F}}_k \mathbf{x}_k + \mathbf{C} \hat{\mathbf{G}}_k \Delta\eta_k \\
 y_{k+2}|_k &= \mathbf{C} \hat{\mathbf{F}}_k^2 \mathbf{x}_k + \mathbf{C} \hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \Delta\eta_k + \mathbf{C} \hat{\mathbf{G}}_k \Delta\eta_{k+1} \\
 &\vdots \\
 y_{k+N_p}|_k &= \mathbf{C} \hat{\mathbf{F}}_k^{N_p} \mathbf{x}_k + \mathbf{C} \hat{\mathbf{F}}_k^{N_p-1} \hat{\mathbf{G}}_k \Delta\eta_k \\
 &\quad + \mathbf{C} \hat{\mathbf{F}}_k^{N_p-2} \hat{\mathbf{G}}_k \Delta\eta_{k+1} + \dots + \mathbf{C} \hat{\mathbf{F}}_k^{N_p-N_c} \hat{\mathbf{G}}_k \Delta\eta_{k+N_c-1}
 \end{aligned} \tag{A I-19}$$

As we can see, all predicted variables are formulated in terms of current state variable information $\mathbf{x}_k$ and the future control trajectory $\Delta\eta_{k+i}$, where $i = 0, 1, \dots, N_c - 1$

Let's define the vector of future outputs $\mathbf{Y}$ such as:

$$\mathbf{Y} = \begin{bmatrix} y_k|_k & y_{k+1}|_k & \dots & y_{k+N_p}|_k \end{bmatrix}^T \tag{A I-20}$$

Considering the vector of future control inputs defined in Eq. (A I-17), then Eq. (A I-19) can be written in the following matrix form:

$$\mathbf{Y} = \mathbf{E} \mathbf{x}_k + \Phi \Delta\mathbf{U} \tag{A I-21}$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{C}\hat{\mathbf{F}}_k \\ \mathbf{C}\hat{\mathbf{F}}_k^2 \\ \vdots \\ \mathbf{C}\hat{\mathbf{F}}_k^{N_p} \end{bmatrix}; \quad \Phi = \begin{bmatrix} \mathbf{C}\hat{\mathbf{G}}_k & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\hat{\mathbf{F}}_k\hat{\mathbf{G}}_k & \mathbf{C}\hat{\mathbf{G}}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\hat{\mathbf{F}}_k^2\hat{\mathbf{G}}_k & \mathbf{C}\hat{\mathbf{F}}_k\hat{\mathbf{G}}_k & \mathbf{C}\hat{\mathbf{G}}_k & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\hat{\mathbf{F}}_k^{N_p-1}\hat{\mathbf{G}}_k & \mathbf{C}\hat{\mathbf{F}}_k^{N_p-2}\hat{\mathbf{G}}_k & \mathbf{C}\hat{\mathbf{F}}_k^{N_p-3}\hat{\mathbf{G}}_k & \cdots & \mathbf{C}\hat{\mathbf{F}}_k^{N_p-N_c}\hat{\mathbf{G}}_k \end{bmatrix} \quad (\text{A I-22})$$

Optimal control solution

For a given reference signal r_k at a sample time k , within a prediction horizon N_p , the objective of the control system is to bring the predicted output as close as possible to the reference signal r_k , where it is assumed that the reference signal remains constant in the optimization window N_p . In other terms, the objective consists of minimizing the squared error $(\mathbf{R}_s - \mathbf{Y})^T(\mathbf{R}_s - \mathbf{Y})$ between the vector of reference signal $\mathbf{R}_s$ and the vector of future outputs $\mathbf{Y}$. This error may be minimized by acting on the control input vector $\Delta\mathbf{U}$. Therefore, the objective function may be translated into an optimization design to find the optimal control parameter $\Delta\mathbf{U}$ that minimizes the error between the reference signal and the output variable. Assuming that the vector that contains the reference signal $\mathbf{R}_s$ at a sample time k is given as follows:

$$\mathbf{R}_s = [1 \ 1 \ \dots \ 1]^T r_k = \bar{\mathbf{R}}_s r_k \quad (\text{A I-23})$$

where $\bar{\mathbf{R}}_s = [1 \ 1 \ \dots \ 1]^T$

Then, the cost function J that reflects the control objective stated above may be written such as:

$$J = (\mathbf{R}_s - \mathbf{Y})^T(\mathbf{R}_s - \mathbf{Y}) + \Delta\mathbf{U}^T \bar{\mathbf{R}} \Delta\mathbf{U} \quad (\text{A I-24})$$

where the first term is linked to the objective of minimizing the errors between the predicted output and the reference signal, while the second term takes into account the influence of the control trajectory $\Delta\mathbf{U}$ when the objective function J is made to be as small as possible. $\bar{\mathbf{R}}$ is a diagonal matrix in the form that :

$$\bar{\mathbf{R}} = r_\omega \mathbf{I}_{N_c \times N_c} \quad (\text{A I-25})$$

where $r_\omega \geq 0$ is used as a tuning parameter for obtaining the desired closed-loop performance. In the case where $r_\omega = 0$, the cost function J does not take into account the size of the control input $\Delta\mathbf{U}$. In the case where $r_\omega \gg 1$, i.e. large, the cost function depends on the size of $\Delta\mathbf{U}$ in order to reduce the error $(\mathbf{R}_s - \mathbf{Y})^T(\mathbf{R}_s - \mathbf{Y})$.

To find the optimal $\Delta\mathbf{U}$ that minimizes the cost function J , Eq. (A I-21) is substituted into Eq. (A I-24). Then, the cost function J may be expressed as follows:

$$J = (\mathbf{R}_s - \mathbf{E}\mathbf{x}_k)^T(\mathbf{R}_s - \mathbf{E}\mathbf{x}_k) - 2\Delta\mathbf{U}^T\Phi^T(\mathbf{R}_s - \mathbf{E}\mathbf{x}_k) + \Delta\mathbf{U}^T(\Phi^T\Phi + \bar{\mathbf{R}})\Delta\mathbf{U} \quad (\text{A I-26})$$

Minimizing Eq. (A I-26) consist in finding a vector $\Delta\mathbf{U}$ that satisfies the following Eq. (A I-27).

$$\frac{\partial J}{\partial \Delta\mathbf{U}} = 0 \quad (\text{A I-27})$$

Therefore, the optimal solution for the control signal obtained from Eq. (A I-27) is the following:

$$\Delta\mathbf{U} = (\Phi^T\Phi + \bar{\mathbf{R}})^{-1}\Phi^T(\mathbf{R}_s - \mathbf{E}\mathbf{x}_k) \quad (\text{A I-28})$$

with the assumption that the inversed matrix $(\Phi^T\Phi + \bar{\mathbf{R}})^{-1}$ exists. The inversed matrix $(\Phi^T\Phi + \bar{\mathbf{R}})^{-1}$ is also called the ‘‘Hessian’’. Taking into account the changes in Eq. (A I-23), the optimal solution $\Delta\mathbf{U}$ may be expressed in terms of the reference signal r_k and state variable $\mathbf{x}_k$ as follows:

$$\Delta\mathbf{U} = (\Phi^T\Phi + \bar{\mathbf{R}})^{-1}\Phi^T(\bar{\mathbf{R}}_s r_k - \mathbf{E}\mathbf{x}_k) \quad (\text{A I-29})$$

Constrained Nonlinear Optimization

Finding $\Delta\mathbf{U}$ in Eq. (A I-29) analytically may be a complex task. However, numerical methods may be used in order to find a vector $\Delta\mathbf{U}$ that minimizes the cost function J in Eq. (A I-26). These methods consist of constrained nonlinear optimization as the Sequential Quadratic Programming algorithm (SQP) (Bonnans, Gilbert, Lemaréchal, & Sagastizábal, 2006), available in MATLAB, which aims to find an optimal vector $\Delta\mathbf{U}$ that minimizes a given cost function J with given constraints.

Cost function

From Eq.(A I-26), let’s define the following notations:

$$\begin{aligned} \mathbf{f} &= -2\Phi^T(\bar{\mathbf{R}}_s r_k - \mathbf{E}\mathbf{x}_k) \\ \mathbf{H} &= 2(\Phi^T\Phi + \bar{\mathbf{R}}) \end{aligned} \quad (\text{A I-30})$$

Therefore, minimizing Eq.(A I-26), consists of finding a vector $\Delta\mathbf{U}$ that satisfies the following equation in partial derivatives with respect to $\Delta\mathbf{U}$

$$\frac{\partial J}{\partial \Delta\mathbf{U}} = \mathbf{f} + \mathbf{H}\Delta\mathbf{U} = 0 \quad (\text{A I-31})$$

whose analytical solution may be stated as follows:

$$\Delta \mathbf{U} = -\mathbf{H}^{-1}\mathbf{f} \quad (\text{A I-32})$$

Finding the minimum of the cost function in Eq. (A I-26) is equivalent to finding the minimum of the following quadratic cost function with respect to $\Delta \mathbf{U}$:

$$J \approx \frac{1}{2} \Delta \mathbf{U}^T \mathbf{H} \Delta \mathbf{U} + \mathbf{f}^T \Delta \mathbf{U} \quad (\text{A I-33})$$

Equality constraint

The objective function, which consists in reducing the error between the reference signal and the output signal is transformed into an equality constraint with respect to $\Delta \mathbf{U}$. In fact, the tracking of the reference signal is fulfilled by respecting the equality constraint, which is defined as follows:

$$y_{k+2} - r_k = 0 \quad (\text{A I-34})$$

In our case, the equality constraint is imposed on the future output y_{k+2} , which is the predicted pitch rate at a sample time $k + 2$. In fact, after testing different objective functions on y_{k+1} and y_{k+3} , the results have shown that imposing an equality constraint on y_{k+2} guaranteed the best results for the closed-loop performance of the controller.

In the case of the SQP algorithm, the equality constraint is imposed on the vector $\Delta \mathbf{U}$. Let's transform Eq. (A I-34) in the way that $\Delta \mathbf{U}$ is explicitly expressed as follows:

$$\mathbf{A}_{\text{eq}} \Delta \mathbf{U} = \mathbf{b}_{\text{eq}} \quad (\text{A I-35})$$

where $\mathbf{A}_{\text{eq}}$ and $\mathbf{b}_{\text{eq}}$ are matrices to be determined. From Eq. (A I-19), we can see that:

$$y_{k+2}|_k = \mathbf{C}\hat{\mathbf{F}}_k^2 \mathbf{x}_k + \mathbf{C}\hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \Delta \eta_k + \mathbf{C}\hat{\mathbf{G}}_k \Delta \eta_{k+1} \quad (\text{A I-36})$$

It may be rewritten in the following form by introducing $\Delta \mathbf{U}$ given by Eq. (A I-17)

$$y_{k+2}|_k = \mathbf{C}\hat{\mathbf{F}}_k^2 \mathbf{x}_k + [\mathbf{C}\hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \quad \mathbf{C}\hat{\mathbf{G}}_k \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \Delta \mathbf{U} \quad (\text{A I-37})$$

Therefore, by replacing Eq. (A I-37) in Eq. (A I-34) :

$$\text{or} \quad \mathbf{C}\hat{\mathbf{F}}_k^2 \mathbf{x}_k + [\mathbf{C}\hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \quad \mathbf{C}\hat{\mathbf{G}}_k \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \Delta \mathbf{U} - r_k = 0 \quad (\text{A I-38})$$

$$[\mathbf{C}\hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \quad \mathbf{C}\hat{\mathbf{G}}_k \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \Delta \mathbf{U} = r_k - \mathbf{C}\hat{\mathbf{F}}_k^2 \mathbf{x}_k$$

From Eq. (A I-35) and by identification in Eq. (A I-38), $\mathbf{A}_{\text{eq}}$ and $\mathbf{b}_{\text{eq}}$ are obtained as follows:

$$\mathbf{A}_{\text{eq}} = [\mathbf{C}\hat{\mathbf{F}}_k \hat{\mathbf{G}}_k \quad \mathbf{C}\hat{\mathbf{G}}_k \quad \mathbf{0} \quad \dots \quad \mathbf{0}]$$

$$\mathbf{b}_{\text{eq}} = r_k - \mathbf{C}\hat{\mathbf{F}}_k^2 \mathbf{x}_k \quad (\text{A I-39})$$

Adaptive Neural Network Controller

- I.3.4 The designed MPC controller takes into account the estimated local dynamics model from the RLS algorithm. Therefore, as long as we deal with estimates, there may be errors in the state predictions (Haykin, 2013), which may lead to poor control. However, from the simulation results in Figure A I-10, it can be seen that the MPC controller combination with the RLS is not enough to track a desired reference signal. There always exists a steady state error, which should be corrected. One solution consists of introducing an adaptive element, such as a neural network controller, which updates its parameters in order to reduce the steady-state error between the reference signal and the pitch rate.

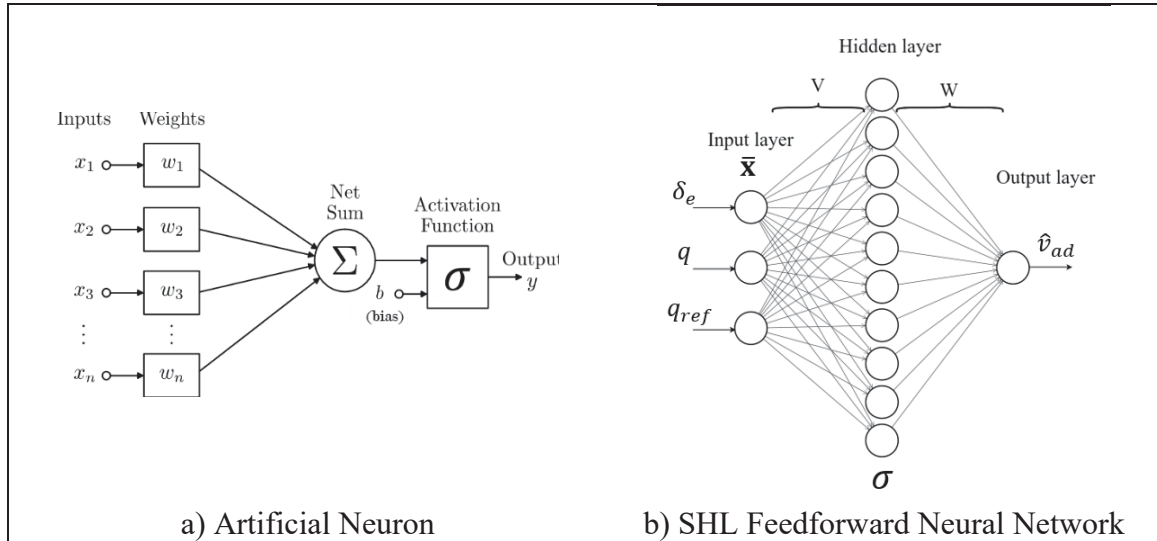


Figure A I-2 Representation of an Artificial Neuron (a) and a SHL Feedforward Neural Network (b)

In Figure A I-1, the neural network is sending an additional adaptive signal v_{ad} to the elevator, in order to compensate for the steady-state error. In our case, a basic Single Hidden Layer (SHL) feedforward neural network is used, which is the simplest structure among neural

networks, and is enough for our control purpose. As shown in Figure A I-2b, a SHL feedforward neural network is composed of a vector of input layer $\bar{\mathbf{x}}$ of size $m \times 1$, one output layer $\hat{v}_{ad}$ of size $n \times 1$ and one hidden layer of size $N \times 1$. m, n and N are the number of inputs, outputs and neurons in the hidden layer, respectively. The input layer $\bar{\mathbf{x}}$ and the hidden layer are then weighted with the weight matrices $\mathbf{V}$ of size $m \times N$ and $\mathbf{W}$ of size $N \times n$, respectively. The hidden layer is composed of neurons, whose activation function is set to tangent sigmoid described by the following expression:

$$\sigma_i(z) = \frac{1}{1 + \exp(-z)} \quad (\text{A I-40})$$

while a linear function was used on the input and output layers.

Therefore, the output of the neural network in Figure A I-2b may be summarized mathematically as follows:

$$\hat{v}_{ad(n \times 1)} = \mathbf{W}_{(N \times n)}^T \sigma_{(N \times N)}(\mathbf{V}_{(m \times N)}^T \bar{\mathbf{x}}_{(m \times 1)}) \quad (\text{A I-41})$$

where the weights $\mathbf{V}$ and $\mathbf{W}$ were updated online using the following adaptation laws given in Eq. (A I-42), which were obtained from the backpropagation principle, as show in detail in (Johnson & Calise, 2003 ; Stevens et al., 2015) :

$$\begin{aligned} \dot{\mathbf{W}} &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}^T \bar{\mathbf{x}})e^T + \lambda \|e\| \mathbf{W}] \boldsymbol{\Gamma} \\ \dot{\mathbf{V}} &= -\boldsymbol{\Gamma} [\bar{\mathbf{x}} e^T \mathbf{W}^T \boldsymbol{\sigma}' + \lambda \|e\| \mathbf{V}] \end{aligned} \quad (\text{A I-42})$$

where e is the error between the refence signal and the pitch rate, $\boldsymbol{\Gamma}$ is the learning rate, $\boldsymbol{\sigma}'$ is the derivative of $\boldsymbol{\sigma}$, λ is a scalar used for the e-modification of the backpropagation algorithm (Johnson & Calise, 2003 ; Stevens et al., 2015) and is set to 1. As seen on Figure A I-1, the signal from the neural network $\hat{v}_{ad}$ is amplified with a gain K_{NN} , which gives more relevance to the adaptation signal. Therefore, the adaptive signal may be set as follows:

I.3.5

$$v_{ad} = K_{NN} \hat{v}_{ad} = K_{NN} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}) \quad (\text{A I-43})$$

Control Law

The optimal control signal $\Delta \eta_{opt}$ from the MPC controller consists of the first element of the vector $\Delta \mathbf{U}$ in Eq. (A I-17). Therefore, at a given sample time k ,

$$\Delta \eta_{opt} = \Delta \eta_k \quad (\text{A I-44})$$

Adding the adaptive control signal from the neural network in Eq. (A I-43), the overall command signal to the elevators $\Delta \delta_{cmd}$ may be summed as follows:

$$\Delta\delta_{e_{cmd}} = \Delta\eta_{opt} + v_{ad} \quad (\text{A I-45})$$

Reference Model

I.3.6 The designed controller consists of a model reference adaptive controller, which follows a given reference signal with given characteristics. The reference signal is set by following the design requirements given by handling qualities on short period dynamics, which means that a specific damping ξ and natural frequency ω_n (Nelson, 1998) are given, and the following filter is applied on the reference signal as shown in Figure A I-1.

$$\frac{q_{ref}}{q_{cmd}} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (\text{A I-46})$$

where s is the Laplace operator.

I.4 Model Simulation and Validation Results

I.4.1 Controller Tuning Parameters

The control architecture shown in Figure A I-1 was applied to the pitch rate of the Cessna Citation X business jet aircraft. The tuning parameters of the controller are summarized in the following Table A I-1. A random step command signal was imposed on the pitch rate q with values between $\pm 1^\circ/\text{sec}$ for a period of 5 seconds, as shown in Figure A I-3.

Table A I-1 Design Parameters for the Controller

Controller Element	Parameters	Value
MPC Controller	N_c, N_p, r_ω	3, 5, 300
RLS online estimation	$\kappa, \mathbf{L}_0, \hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$	0.85, $10^7 \mathbf{I}_{5,5}$, $\mathbf{I}_{4,4}$, $-\mathbf{0.1}_{4,1}$
NN Controller	K_{NN} , learning rate Γ number of layers number of neurons	10, 1.2 1 30
Reference Model	ξ, ω_n	0.69, 4 rad/s

Figure A I-3 shows the simulation results for the control of the pitch rate for a cruise phase at an altitude of 35,000 ft and a Calibrated Airspeed (CAS) of 290 knots. The weight of the aircraft was set at 33,000lb and the center of gravity was set at 25% of the mean aerodynamic chord. As we can see, the pitch rate is perfectly following the reference signal, and the elevator deflection varies around $\pm 2^\circ$.

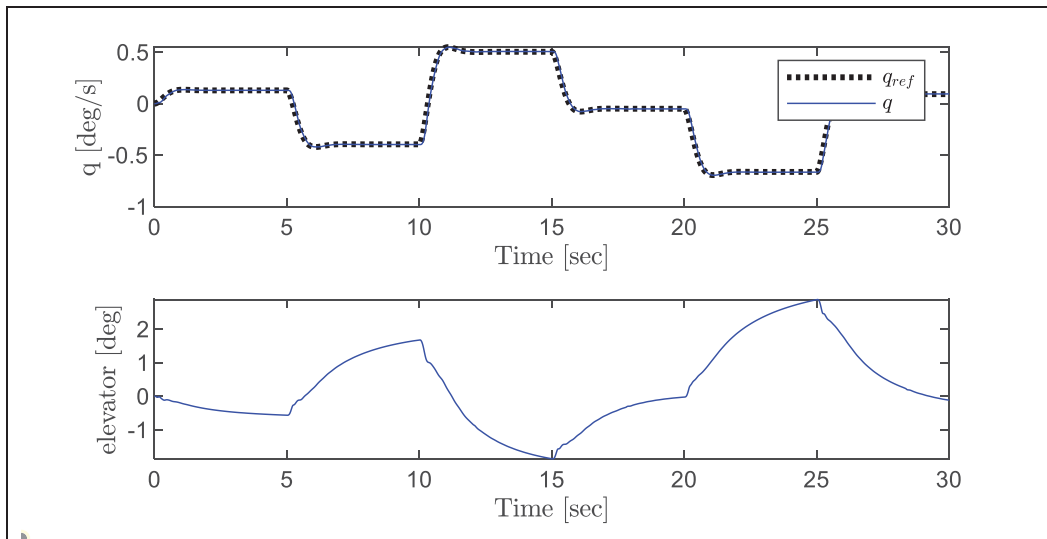


Figure A I-3 Pitch Rate and Elevator Responses for Simulation at 35,000 ft and 290 knots for the cruise phase.

The weights of the neural network and the variation of matrices $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ given by the RLS method are shown in Figure A I-4. It is seen that the values of $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ matrices are converging to stable signals, which means that the state estimation is fulfilled correctly using the RLS algorithm.

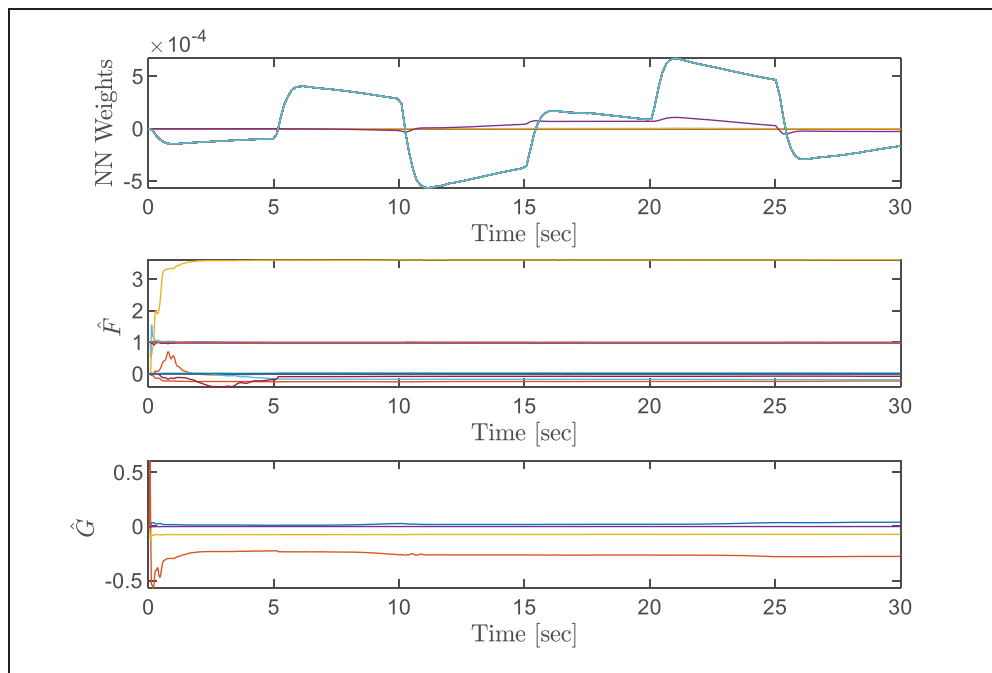


Figure A I-4 Parameter Responses for Simulation at 35,000 ft and 290 knots for the cruise phase.

Validation of the Control Algorithm over the Flight Envelope

I.4.2

The proposed control algorithm was validated for 63 flight conditions across the Cessna Citation X flight envelope. In fact, the flight conditions were set in terms of altitude h and Calibrated Airspeed (CAS) in order to cover the entire flight envelope. The center of gravity was set at 25% of the mean aerodynamic chord and the weight at 33,000 lb. The altitude varied between 5 000 and 40,000 ft with a step of 5 000 ft while the calibrated airspeed was set between 170 kts and 350 kts with a step of 20 knots.

Selection of r_ω

The first challenge consisted of tuning of the parameter r_ω given in Eq. (A I-25). For this purpose, the control algorithm was tested on the 63 flight conditions with different value of r_ω , and fixed values of N_c and N_p as shown in Table A I-1. The overall Sum Squared Error (SSE), the number of iterations and the number of functions count were outputted. In fact, as we deal with optimization, the optimization time is an important factor, which is reflected by the number of iterations and the number of times that the cost function in Eq. (A I-33) was evaluated. Therefore, the value of r_ω was set, in the way that the overall SSE was small, and the number of iteration and number of function counts were optimal. As shown in Figure A I-5, when the value of r_ω increases, the number of iterations decreases while the number of function count increases. However, the overall SSE reaches a stable value when r_ω is greater than 100. Therefore, r_ω was set at 300 for the rest of the simulations, for which the optimization time was optimal.

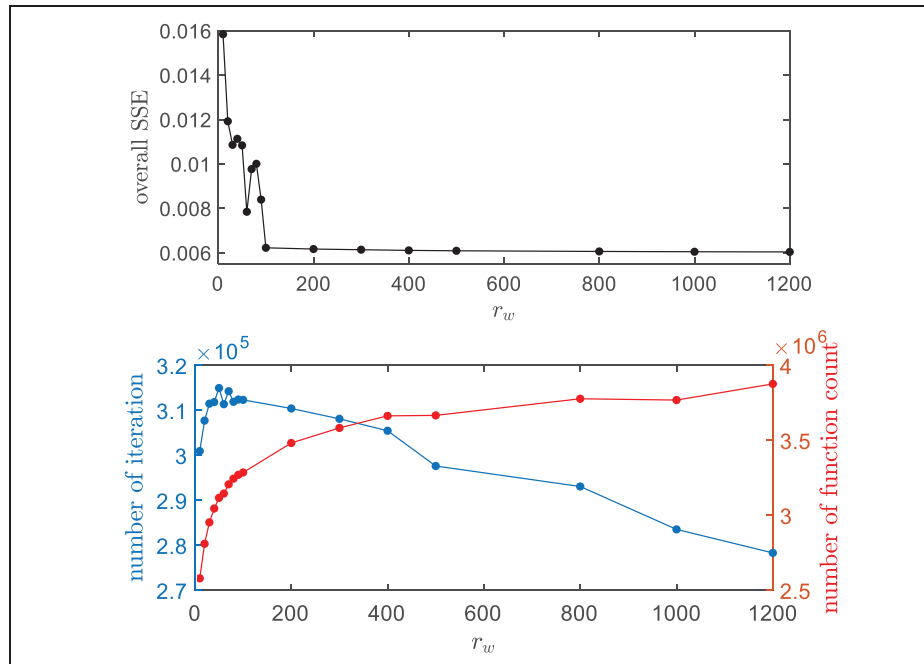


Figure A I-5 Selection of the parameter r_ω

Figure A I-6 shows the pitch rate q response for 63 flight conditions for 3 different values of r_ω . As we can see, a higher r_ω results in a better closed-loop performance.

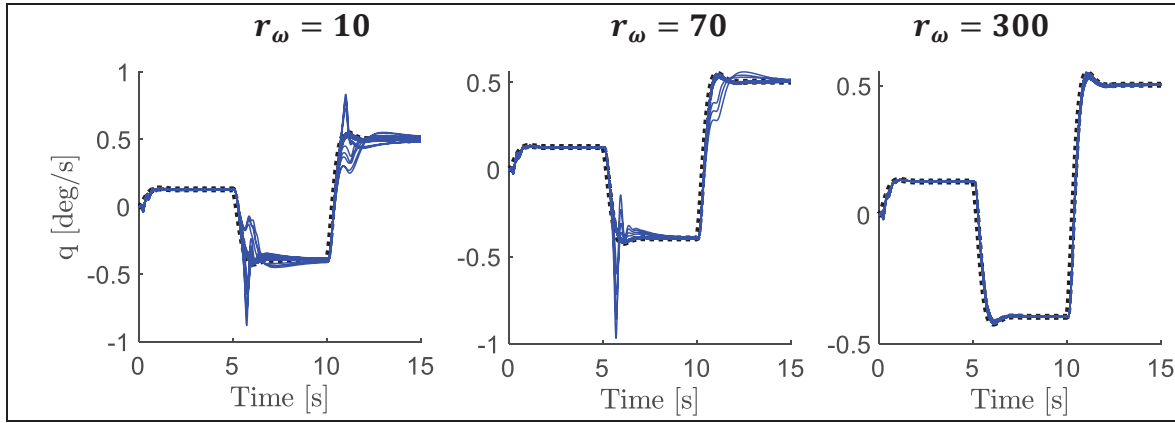


Figure A I-6 Comparison of 3 values of r_ω for 63 flight conditions

Selection of control the horizon N_c and predicted horizon N_p

The next step consisted of selecting the control horizon N_c and the predicted horizon N_p . As seen on Figure A I-7, higher values of N_c and N_p result in poorer closed-loop performance. In fact, if the optimization window N_p is too large, the estimated local dynamics ($\hat{\mathbf{F}}_k, \hat{\mathbf{G}}_k$) from the RLS algorithm may not be representative of the dynamics for a large optimization window N_p and oscillations may occur. On the other hand, smaller values of N_c and N_p guarantee that the optimization is fulfilled locally at a given set-point with given local dynamics $\hat{\mathbf{F}}_k, \hat{\mathbf{G}}_k$.

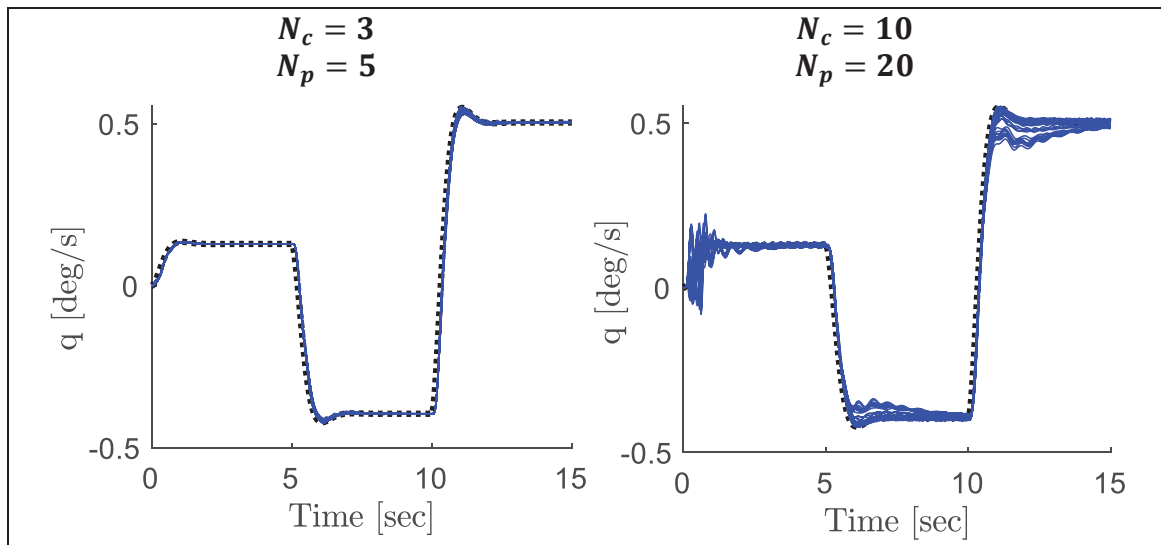


Figure A I-7 Comparison of 2 different combinations of N_c and N_p for 63 flight conditions

Selection of Learning Rate Γ

The final step consists of choosing the learning rate Γ for the neural network controller, in order to update the neural network weights in Eq. (A I-42). In this case, the parameters of the RLS algorithm and MPC controller were fixed as per Table A I-1. As stated before, the neural network provides additional adaptive control signals that improve the closed-loop performance of the controller. As shown on Figure A I-8, a low learning rate results in poor adaptation while a high learning rate resulting in better adaptation.

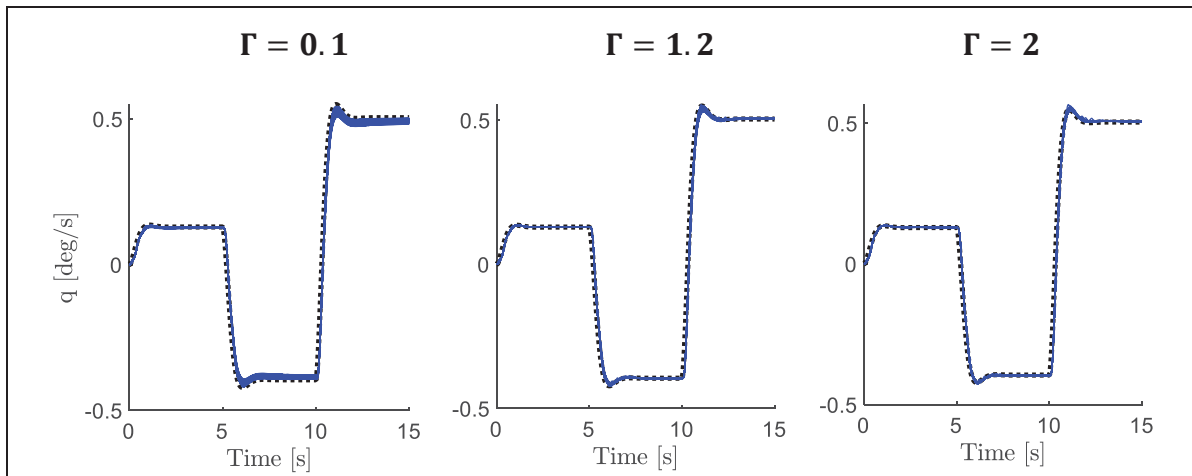


Figure A I-8 Pitch Rate Response for 3 different Learning Rates Γ and 63 flight conditions

Comparison with different configurations

The proposed controller MPC + NN (in red) was compared with different configurations as seen on Figure A I-9 and Figure A I-10, including an MPC alone (in blue), a MPC + PID (in purple), a NN (in orange), a PID (in green), and a PID + NN (in black). As seen on Figure A I-9, the controllers with a MPC (in red, blue and purple) give the best closed-loop performance and are able to follow the reference signal.

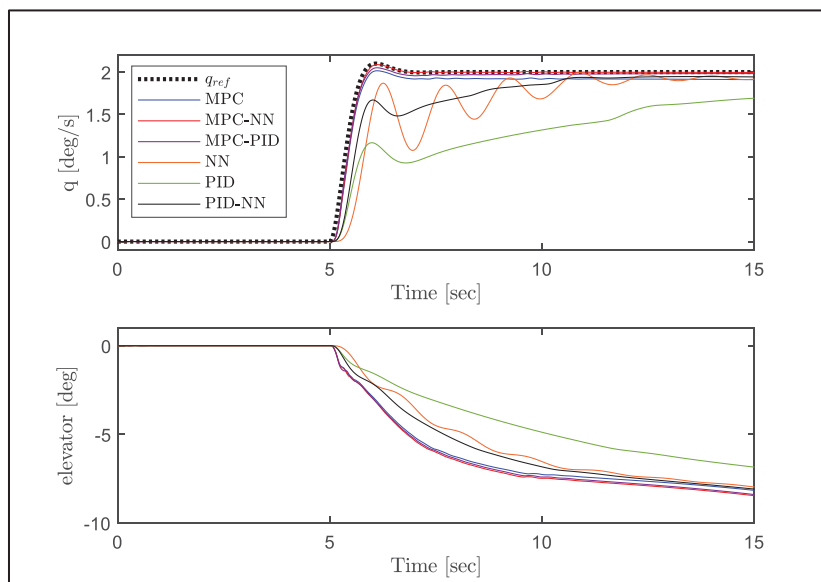


Figure A I-9 Comparison of the proposed MPC - NN controller with different Configurations for One Flight Condition at 35,000 ft and 290 knots for the cruise phase

Moreover, a better look on the steady state error on Figure A I-10 shows that the MPC controller combined with a NN (in red) gives the best closed-loop performance, with practically a 0-tracking error, in comparison with the MPC controller with PID (in purple) and the MPC controller alone. As seen on the elevator response from Figure A I-9, the MPC- NN (in red) gives the additional deflection which compensates for the steady-state error.

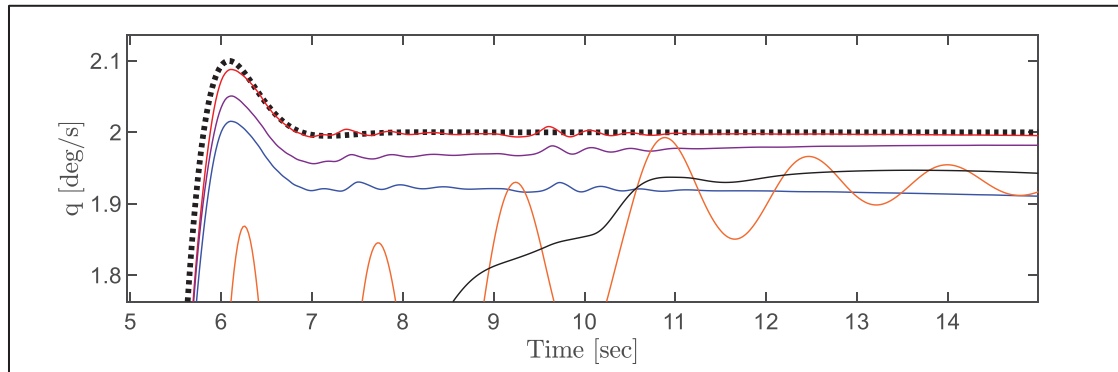


Figure A I-10 Zoom on Figure A I-9. Steady state error

I.5 Discussion of Results

The proposed pitch rate controller was well implemented for the pitch rate control of the Cessna Citation X. The main features of the controller consist in the fact that the control input is computed using the estimates of the local dynamics of the aircraft and does not require any knowledge on the aircraft dynamics. Therefore, a double adaptation was fulfilled. The first adaptation consisted of the combination RLS-MPC controller, for which the local dynamics are estimated using states and control input increments and then used for the optimization, while the second adaptation is fulfilled by the neural network controller.

The main challenge consisted of the selection of the MPC parameters, which are r_ω , N_p and N_c . In fact, these parameters should be set, so that the computation time was short, and a good closed-loop performance was ensured. Finally, the neural network adaptive signal was amplified with a gain K_{NN} , which gave more importance to the adaptive signal and better handled the nonlinearities.

The major take away from this study lies in the fact that all the controller parameters were fixed at specific values. In fact, it only required one configuration of the controller given in Table A I-1 to control the pitch rate q for 63 flight conditions. Therefore, the proposed MPC-NN controller is better than gain scheduling, which requires extensive tuning for only one flight condition.

I.6 Conclusion

A nonlinear adaptive pitch rate controller was successfully designed using Model Predictive Control, online state estimation and neural network adaptive controller. The controller was

successfully validated for 63 flight conditions in the cruise phase. As a result, the proposed control algorithm was perfectly able to track a given reference signal on the pitch rate. One of the major features consisted in the fact that the control input was fully computed using online estimates of the aircraft local dynamics using RLS method, and optimization by minimizing a given cost function. The proposed MPC - NN controller was able to solve the oscillation problem on the elevators for high-speed conditions, which were addressed in previous studies. In fact, the designed controller was able to control a high degree of nonlinearities, specific to the actuator dynamics, without a priori knowledge of the aircraft dynamics. The remaining task (future work) of this study consists of validating the proposed controller for robustness against uncertainties.

ANNEX II

ROBUST PITCH CONTROL OF THE CESSNA CITATION X USING C-STAR ARCHITECTURE AND ADAPTIVE NEURAL NETWORKS

Royo Princy Andrianantara ^a, Francisco Daniel Mancera Coyotl ^a, Georges Ghazi^a, and Ruxandra Botez ^a

^a Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE), System Engineering Department, École de technologie supérieure, 1100 Rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3

Article published in *AIAA SciTech 2024 Forum*, July 2024

<https://doi.org/10.2514/6.2024-4257>

Résumé

Cet article présente une technique permettant de concevoir un contrôleur de tangage robuste à l'aide d'une architecture de contrôle C* et de mettre en œuvre des méthodes adaptatives pour le contrôle longitudinal de l'avion d'affaires Cessna Citation X. Tout d'abord, la conception de la boucle interne vise à contrôler la vitesse de tangage et le facteur de charge, également appelé paramètre C*. Un contrôleur proportionnel-intégral-dérivé (PID) est associé à un contrôleur adaptatif à réseau de neurones. Les gains du PID restent constants pour toutes les conditions de vol, tandis que les hyperparamètres du réseau de neurones sont mis à jour en ligne. Un signal de référence est imposé à la commande C* afin de répondre aux exigences de qualité de maniabilité pour les modes longitudinaux. Deuxièmement, un contrôleur de tangage adaptatif est proposé à l'aide d'un deuxième réseau de neurones. L'algorithme de contrôle global est conçu pour le contrôle en palier et testé pour 63 conditions de croisière dans l'enveloppe de vol du Cessna Citation X. Sa robustesse est testée par rapport à des incertitudes, notamment des rafales de vent, des cisaillements de vent et le modèle de turbulence de Dryden avec différentes intensités. Les résultats montrent que l'algorithme de contrôle proposé est parfaitement capable de contrôler l'avion et de suivre un signal de référence donné en tangage.

Abstract

This paper presents a technique for designing a robust pitch controller using a C* control architecture and implementing adaptive methods for the longitudinal control of the Cessna Citation X business jet aircraft. First, the design of the inner loop aims to control the pitch rate and the load factor, also known as the C-star parameter. A Proportional–Integral–Derivative (PID) controller is combined with a Neural Network Adaptive controller. The gains of the PID remain constant for all flight conditions, while the NN hyperparameters are updated online. A reference signal is imposed on the C-star command in order to meet the handling quality requirements for longitudinal modes. Second, an adaptive pitch controller is proposed using a second neural network. The overall control algorithm is designed for cruise control and tested

for 63 cruise conditions across the Cessna Citation X flight envelope. Its robustness is tested against uncertainties, including wind gust, wind shear and Dryden turbulence model with various intensities. Results show that the proposed control algorithm is perfectly able to control the aircraft and follow a given model reference pitch signal.

II.1 Introduction

Flight control design has always been a challenge for aircraft manufacturers. In fact, flight control laws are essential to make flights possible. Control laws are not deterministic and depend solely on the designer's philosophy and desired performance. However, in order to keep the flight safe and comfortable, flight controllers are subject to certification, namely requirements on performance criteria such as damping and frequencies (Department of Defense, 1997 ; Nelson, 1998).

One of the main challenges for flight controllers is to control the aircraft states while dealing with uncertainties during flight. In fact, an aircraft is always exposed to uncertainties that can hardly affect its stability, and thus, the safety of its passengers. One of the prospective solutions is to design robust controllers, which can adapt to such uncertainties, and therefore provide adaptive inputs to the control surfaces in order to compensate for any failure or change in the aircraft environment.

Today, flight controllers are still designed based on classical and modern theory, such as Proportional-Integral-Derivative (PID) controllers (Nelson, 1998 ; Stevens et al., 2015) and low order controllers such as Linear Quadratic Regulators (LQR) (Nelson, 1998), respectively. One of their main characteristics is their linear property, which aims at controlling a linearized dynamic of the aircraft. Linear control theory relies on time-frequency design and ensures good stability margins (Dorf & Bishop, 2008 ; Ogata & Yang, 2002). However, for a linear controller to be applied to the entire flight envelope, the aircraft flight dynamics must be linearized at given set points, such as Mach number, airspeed and altitude. Gain scheduling is then required, which consists of a table of gains as a function of flight conditions. Therefore, an extensive gain scheduling is required to cover the entire flight envelope, which is time consuming. As an example, for classical PID control, 9 gains are required for only one nominal flight condition in order to control the aircraft in the pitch, roll and yaw axis. To these gains, another set of 9 or more gains could be added when changes occur in the aircraft dynamics or environment.

Several robust methods were later introduced in order to guarantee robustness in the presence of uncertainties, such as H_∞ controller and Linear Quadratic Gaussian (LQG) regulator. Although these two techniques were powerful, they still rely on gain scheduling, and their design requires extensive analysis. There have been attempts to automate the process of identifying these gains by combining them with optimization algorithms. Examples are given by Ghazi and Botez (Ghazi & Botez, 2014, 2015b) which uses Genetic Algorithm and by Boughari et al. (Y. Boughari, Ghazi, et al., 2017b ; Yamina Boughari, Botez, et al., 2017) which uses Differential Evolution algorithm to optimize the identification of the LQR controller gains. Other advanced techniques such as Model Predictive Controllers (MPC) (Eren

et al., 2017 ; Kopf, Bullinger, Giessler, Adden, & Findeisen, 2018 ; Simon, 2014) and Dynamic Inversion (DI) controllers (R. P. Andrianantara et al., 2023a ; Ducard & Geering, 2008 ; Horn, 2019) were also investigated. These two techniques were particularly powerful because they belong to the class of adaptive control. In fact, while the control parameters are fixed for classical controllers, parameters are variable for adaptive controllers and can be tuned with adaptive control laws (Slotine & Li, 1991), allowing adaptive controllers to deal with changing dynamics. However, MPC and DI controllers still rely on precise mathematical models of the aircraft, and a linearization of its dynamics. The main linearization problem is that several nonlinearities are neglected in the design. Thus, any failure that occurs in the control system is often due to these unmodelled nonlinearities (Steffensen et al., 2023) that are not considered in its design.

Since late 1990's, intelligent control methods have brought solutions that deal with these unmodelled uncertainties. These techniques use intelligent features such as neural networks (Ge et al., 2013 ; M. T. Hagan & Demuth, 1999 ; Kim & Calise, 1997), support vector machines (Hashemi & Botez, 2022) or even fuzzy logic controllers (Grigorie, Botez, et al., 2012a ; Hashemi et al., 2020 ; S. Hosseini, Ghazi, et al., 2023a). With the recent advancement of computers, in terms of computing time and memory, it is now possible to implement Machine Learning-based controllers such as Reinforcement Learning (RL) (Bagnell & Schneider, 2001 ; Feith, 2020 ; Heyer et al., 2020), which requires a large amount of training data. However, one issue of ML-based controller consists of offline training, for which the controller should be trained for all possible outcomes. Therefore, the RL-controller may fail if unforeseen uncertainties are not included in the training data.

However, today computers allow fast calculation, thus, the online adaptive control remains applicable (Heyer et al., 2020). In fact, adaptive controllers adapt their hyperparameters with the changing dynamics. They include adaptation laws (Kim & Calise, 1997 ; Rysdyk & Calise, 2005 ; Slotine & Li, 1991) which consist of a cost function on the error and state dynamics. Therefore, in order to output the proper adaptive signals, the designer must identify the proper adaptation laws, which tend to minimize the errors. The designed controller may then follow a given reference signal that respects the requirements given by the handling qualities (Rysdyk & Calise, 2005 ; Saussie, Saydy, & Akhrif, 2003), and then, it ensures the desired performance. Model reference controllers allow particularly to avoid saturations in the control loop, which are often sources of nonlinearities (N. T. Nguyen, 2018).

Adaptive neural network controllers have been experimented on real aircraft and demonstrated their performances on real flight tests (Bosworth & Williams-Hayes, 2007 ; Smith et al., 2010 ; Soares & Burken, 2006 ; Williams, 2004 ; Williams-Hayes, 2005). Nowadays, neural network controllers have been applied so forth in aircraft flight simulation environments (R. P. Andrianantara et al., 2023a ; Feith, 2020 ; Lee & Van Kampen, 2021), and Unmanned Aerial Vehicles (UAV) (Dierks & Jagannathan, 2009 ; Koch, Mancuso, West, et al., 2019) as their safety are still under certification (Cima et al., 2021). However, results have shown their possible application on commercial aircraft on Cessna Citation X flight dynamics (R. P. Andrianantara et al., 2023a).

At our Laboratory of Active Controls, Avionics and Aeroservoelasticity (LARCASE), the research team has demonstrated the design of intelligent controllers, such as Fuzzy Logic (S. Hosseini, Ghazi, et al., 2023a) and neural network (R. P. Andrianantara et al., 2023a ; S. Hosseini, Inga, et al., 2023) controllers applied on the Cessna Citation X flight dynamics for its longitudinal and lateral control. Results have shown that neural networks are suitable for adaptive control, and that no further gain scheduling is required in order to cover the overall flight envelope. This method allows to reduce the number of controllers to a single controller, reducing their complexity.

Neural networks are powerful tools due to their ability of generalization, i.e., they can find nonlinear features in datasets. This ability makes neural networks suitable for system identification (M. Hagan, Demuth, Beale, & De Jesus, 2016), such as complex aircraft flight dynamics (Tondji, Ghazi, & Botez, 2022 ; Zohreh Nejad et al., 2023), engine dynamics (R. P. Andrianantara et al., 2023b ; Ghazi, Tudor, & Botez, 2015 ; Zaag et al., 2019) or even flutter dynamics (Boely et al., 2011). Another solution to encounter the issue of linearization consists of identifying the aircraft dynamics model online. This means that the linearization is performed locally during flight, allowing us to find any change in the aircraft dynamics during flight operation. Several techniques of adaptive filters (Haykin, 2013), such as Extended Kalman Filtering (Puskorius & Feldkamp, 1991 ; Wan & Van Der Merwe, 2000) and Recursive Least Square methods (Haykin, 2013 ; Mahadi et al., 2022) allow to estimate the aircraft state and control matrices during flight. By this way, the estimated information may be fed into the control law for further Dynamic Inversion (R. P. Andrianantara et al., 2023a) or to design MPC controllers (R. P. Andrianantara, Ghazi, et al., 2024).

The objectives of this research are to design, implement, and validate a flight controller in order to control the pitch angle of the Cessna Citation X under different flight conditions, covering the entire flight envelope, and. This research also aims to reduce the number of flight controllers while providing a robust and improved controller design. A single controller was designed that includes a C* control architecture, an Approximative Dynamic Inversion, dual adaptive neural networks, and a PID controller. The resulting controller is capable of tracking a reference signal on the pitch angle. This controller has been tested for its robustness under wind and turbulence conditions covering the entire flight envelope at different altitudes and airspeeds.

The aircraft model and simulation platform of the Cessna Citation X will be described in Section I, while the pitch controller design methodology will be given in Section II, along with a methodology to design a pitch reference model. Finally, Section III will present the simulation results under normal flight conditions and in the presence of winds. Results have shown very good performance, allowing the controller to follow a given pitch reference signal.

II.2 Aircraft Model and Simulation Platform

II.2.1 Cessna Citation X business jet aircraft and its Simulink nonlinear model

The Cessna Citation X is one of the fastest business jet aircraft ever designed. In order to design its flight controller, a simulation platform is always required (Koch, Mancuso, West, et al., 2019 ; Konatala et al., 2021) before implementing the controller onboard. The LARCASE laboratory team has developed a high-fidelity nonlinear Simulink model of the Cessna Citation X for research purposes (Ghazi & Botez, 2015a). The simulation model has been validated using flight tests and accurate data from a Level D Research Flight Simulator of the Cessna Citation X (RAFS) available at the LARCASE laboratory. The Level D is the highest level of qualification given by the Federal Aviation Administration (FAA) for aircraft flight dynamics. The simulation platform includes several complex modules, such as atmospheric module, aerodynamic module, engine and actuator dynamics. It allows to trim the aircraft and to linearize its flight dynamics at a given altitude and airspeed, allowing to design the flight controller for the overall flight envelope of the aircraft. Several flight controllers have already been designed and validated using the simulation platform of the Cessna Citation X (Ghazi & Botez, 2015a), making it a suitable tool for validating the proposed methodology.

II.2.2 Aircraft Nonlinear Dynamics Model

For this study, only the longitudinal flight dynamics were considered. The aircraft longitudinal dynamics model consists of a state space equation including state variations $\dot{\mathbf{x}}$ and an output equation y . If for linear control, the state space equation of longitudinal dynamics is quite explicit (Stevens et al., 2015), its nonlinear representation is quite complex, and can be written as follows (Slotine & Li, 1991):

$$\begin{aligned}\dot{\mathbf{x}} &= \varphi(\mathbf{x}, \eta) \\ y &= h(\mathbf{x})\end{aligned}\tag{A II-1}$$

where $\mathbf{x} = [u \ w \ q \ \theta \ n_z]^T$ is the state vector, composed of the linear speed u , the vertical speed w , the pitch rate q , the pitch angle θ and the normal load factor n_z , respectively, $\eta = \delta_e$ is the command input on the elevators, φ is a function that describes the nonlinear dynamics of the aircraft, y is the output of the model to be controlled, and h is a nonlinear function describing the relationship between the output y , and the state vector $\mathbf{x}$.

II.2.3 C* control architecture

The C* control architecture was first implemented in the flight control systems of the Space Shuttle in the early 1960's, then the French and the British researchers applied this control law to the Concorde (Tischler, 1996). It has fixed architecture and is essentially designed for longitudinal control. It was developed in response to the Pilot Induced Oscillation effect as a consequence of Fly-By-Wire technologies, which worsened flying qualities of the aircraft (Saussié et al., 2009). The parameter C* is defined as a combination of the pitch rate q and the normal load factor n_z as follows (Tischler, 1996):

$$C^* = \Delta n_z + \beta \Delta q\tag{A II-2}$$

II.3.4 Actuator Dynamics

As shown in Figure A II-1, the controlled dynamics include the actuator dynamics, which consists of a second-order nonlinear actuator model, available in the Simulink environment. In fact, the actuator dynamics increases the complexity of the system by introducing high order nonlinearities. In this way, the adaptability of the proposed controller to higher-than-second-order nonlinearity can be evaluated, which increases the generalizability of the proposed methodology.

II.3.1 Pitch attitude control

The diagram illustrates a two-loop control system for aircraft pitch control, divided into an Outer Loop and an Inner Loop.

Outer Loop:

- The reference pitch angle θ_{ref} (labeled ②) is compared with the current pitch angle θ to produce the error signal e_θ .
- The error signal e_θ is processed by a PID controller to generate the control signal v_2 .
- The signal v_2 is summed with the feedback signal v_θ and then passes through an RLS Dynamic Inversion block to produce the reference command C^*cmd .

Inner Loop:

- The reference command C^*cmd is compared with the reference model output C^*_{ref} (labeled ①) to produce the error signal e_{C^*} .
- The error signal e_{C^*} is processed by a PID Controller to generate the control signal v_1 .
- The signal v_1 is summed with the feedback signal v_{δ_e} to produce the control signal $\delta_e cmd$.
- The signal $\delta_e cmd$ is processed by the Actuator Dynamics block to produce the control signal δ_e .
- The signal δ_e is compared with the reference model output δ_e^* (labeled ③) to produce the error signal e_{δ_e} .
- The error signal e_{δ_e} is processed by an NN₁ Adaptive Controller (labeled ①) to produce the control signal δ_e .
- The control signal δ_e is processed by the Aircraft Dynamics block to produce the pitch angle θ , pitch rate $\dot{\theta}$, and pitch acceleration $\ddot{\theta}$.
- The pitch angle θ is compared with the reference pitch angle θ_{ref} to produce the error signal e_θ .
- The pitch rate $\dot{\theta}$ is compared with the reference pitch rate $\dot{\theta}_{ref}$ to produce the error signal $e_{\dot{\theta}}$.
- The pitch acceleration $\ddot{\theta}$ is compared with the reference pitch acceleration $\ddot{\theta}_{ref}$ to produce the error signal $e_{\ddot{\theta}}$.
- The error signals e_θ , $e_{\dot{\theta}}$, and $e_{\ddot{\theta}}$ are processed by an NN₂ Adaptive Controller (labeled ②) to produce the control signal v_θ .
- The control signal v_θ is summed with the feedback signal v_θ to produce the control signal v_2 .

II.3.2 The C^* inner loop controller

The inner loop controller consists of the C^* parameter control. As described in Eq. (A II-2), the C^* parameter is a combination of the load factor Δn_z and the pitch rate variation Δq . Controlling the C^* parameter is particularly important for the longitudinal control at high speeds, since the aircraft is more sensitive to accelerations during high-speed operation.

As shown in Figure A II-1, the inner loop receives a C^* command C_{cmd}^* , which is generated by the pitch controller in the outer loop. The command signal C_{cmd}^* is then filtered through a reference model filter to generate a reference signal C_{ref}^* . The error e_{C^*} on the C^* parameter is then calculated and controlled by a PID controller, whose gains remain constant for all flight conditions. An adaptive signal v_{δ_e} generated by a first neural network NN_1 is added to the baseline PID controller in order to output the command signal δ_e on the elevator to the actuators.

The pitch rate Δq and the load factor Δn_z are then fed back in a loop in order to control the C^* parameter. A washout filter is added to the pitch rate feedback in order to improve the damping on the short period dynamics and cut off the low frequency signals. A first order filter is also added to the load factor feedback in order to reduce the oscillations and filter out high-frequency signals.

The C^* reference model

For most linear control systems, the desired performance and response to control inputs are defined by handling qualities (Department of Defense, 1997). These qualities consist of a set of requirements imposed on the damping and frequency of the system dynamics. Typically, to obtain the handling qualities, it is necessary to obtain a linearized model of the aircraft flight dynamics and then design its linear controllers, for which the time-frequency domain analysis is made possible. However, in the case of nonlinear control, the handling qualities cannot be directly obtained as time-frequency analysis is not possible. One solution is to design a longitudinal controller that follows a desired reference signal on the pitch angle θ . The resulting controller is also known as model reference controller, for which the reference model acquires the desired performance specified by the handling qualities.

In the case of C^* control, the handling qualities requirements are defined for short period dynamics (Saussié et al., 2009 ; Saussie et al., 2003), in the sense that specific frequency and damping are pre-established. The short period dynamics is a fast-oscillating mode which can be approximated by a second order transfer function as follows:

$$\frac{C_{ref}^*}{C_{cmd}^*} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (\text{A II-3})$$

where ζ is the damping, ω_n is the natural frequency, and s is the Laplace operator.

The PID controller

As seen in Figure A II-1, a PID controller, which takes the error e_{C^*} as input, is integrated. Its gains remain constant for all flight conditions, allowing the neural network to fulfill its proper adaptation (Stevens et al., 2015). The output of the PID controller can be expressed as follows:

$$v_1 = \left(K_P + \frac{K_I}{s} + K_D s \right) e_{C^*} \quad (\text{A II-4})$$

where K_P , K_I and K_D are the proportional, integral, and derivative gains respectively.

The Washout filter

The washout filter $W(s)$ is a high-pass filter that rejects signals below the cutoff frequency. It consists of a first-order transfer function defined as follows:

$$W(s) = \frac{s + a}{\tau_W s + b} \quad (\text{A II-5})$$

where a and b are two design parameters, τ_W is the time constant of the filter.

The First Order Filter

A low-pass filter is added to the load factor feedback. It consists of a first order filter, which also act as a noise filter. It allows only the low-frequency signals to be considered and rejects high-frequency disturbances. Its transfer function can be written as follows:

$$N(s) = \frac{K}{\tau_N s + 1} \quad (\text{A II-6})$$

where K is a constant gain, and τ_N is the time constant.

II.3.3 Nonlinear Dynamic Inversion Controller

The dynamic inversion controller consists of generating a command signal η by inverting the system dynamics. While this process is fairly straightforward for linear systems (Stevens et al., 2015), it can be more complex for nonlinear systems (Horn, 2019), because their state space representation is not available. One solution consists of fulfilling the dynamic inversion locally by linearizing the aircraft dynamics. Let's consider an affine form with respect to the command input η of the aircraft longitudinal flight dynamics as follows:

$$\dot{\mathbf{x}} = \varphi(\mathbf{x}, \eta) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\eta \quad (\text{A II-7.a})$$

$$y = \theta = \mathbf{C}\mathbf{x} \quad (\text{A II-7.b})$$

where $\mathbf{x}$ is the state vector, and $\dot{\mathbf{x}}$ are their variations. In addition, $\mathbf{f}$ and $\mathbf{g}$ are nonlinear functions that represent the aircraft longitudinal model, y is the output, corresponding to the pitch angle θ . The matrix $\mathbf{C}$ corresponds to the observation matrix and is equal to $\mathbf{C} = [0 \ 0 \ 0 \ 1 \ 0]$.

As mentioned above, the dynamic inversion consists in inverting the dynamics in order to express the control input η as a function of the states $\mathbf{x}$ and their variations $\dot{\mathbf{x}}$ as follows:

$$\eta = \boldsymbol{\varphi}^{-1}(\dot{\mathbf{x}}, \mathbf{x}) \quad (\text{A II-8})$$

Let's apply the first order Taylor series expansion on the state variations in Eq. (A II-7.1). The expansion is fulfilled around an initial set point $(\mathbf{x}_0, \eta_0)$ and can be written as:

$$\boldsymbol{\varphi}(\mathbf{x}, \eta) = \boldsymbol{\varphi}(\mathbf{x}_0, \eta_0) + \boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}(\eta - \eta_0) \quad (\text{A II-9})$$

where $\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0)$ is the value of $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ at $(\mathbf{x}_0, \eta_0)$, $\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}$ and $\boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta)_{\mathbf{x}_0, \eta_0}$ are the partial derivatives of $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ with respect to the states $\mathbf{x}$, and to the input η , respectively.

The value of $\boldsymbol{\varphi}$ at initial set point $(\mathbf{x}_0, \eta_0)$ and the partial derivatives $\boldsymbol{\varphi}'_{\mathbf{x}}$ and $\boldsymbol{\varphi}'_{\eta}$ in Eq. (A I-4) can be expressed as follows:

$$\boldsymbol{\varphi}(\mathbf{x}_0, \eta_0) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta \quad (\text{A II-10.a})$$

$$\boldsymbol{\varphi}'_{\mathbf{x}}(\mathbf{x}, \eta) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\eta \quad (\text{A II-56.b})$$

$$\boldsymbol{\varphi}'_{\eta}(\mathbf{x}, \eta) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \eta} + \frac{\partial [\mathbf{g}(\mathbf{x})\eta]}{\partial \eta} = \mathbf{g}(\mathbf{x}) \quad (\text{A II-56.c})$$

Thus, rewriting the states and input in terms of their variations $\Delta \mathbf{x} = (\mathbf{x} - \mathbf{x}_0)$ and $\Delta \eta = (\eta - \eta_0)$, and replacing all terms obtained in Eq. (A II-10) in Eq.(A I-4), we obtain Eq.(A II-11):

$$\boldsymbol{\varphi}(\mathbf{x}, \eta) = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta + \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\eta \right)_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A II-11})$$

Let's rewrite the time derivative of the state vector $\mathbf{x}$ in Eq. (A II-7) around an initial condition $(\mathbf{x}_0, \eta_0)$:

$$\dot{\mathbf{x}} = \boldsymbol{\varphi}(\mathbf{x}, \eta) = \dot{\mathbf{x}}_0 + \Delta \dot{\mathbf{x}} \quad (\text{A II-12})$$

As $\boldsymbol{\varphi}(\mathbf{x}, \eta)$ is given by both Eq. (A II-11) and Eq. (A II-12), and $\dot{\mathbf{x}}_0 = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\eta$, then the variation of the state's derivative $\Delta \dot{\mathbf{x}}$ can be expressed as follows:

$$\Delta \dot{\mathbf{x}} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \eta \right)_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A II-13})$$

Considering the following notations:

$$\mathbf{F}_{\mathbf{x}_0, \eta_0} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \eta \right)_{\mathbf{x}_0, \eta_0} ; \mathbf{G}_{\mathbf{x}_0, \eta_0} = \mathbf{g}(\mathbf{x})_{\mathbf{x}_0, \eta_0} \quad (\text{A II-14})$$

Eq. (A II-13) can be rewritten in the following compact form:

$$\Delta \dot{\mathbf{x}} = \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{G}_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A II-15})$$

Recalling from Eq. (A I-2), the variation of the derivative of the output equation may be expressed as follows:

$$\Delta \dot{\mathbf{y}} = \mathbf{C} \Delta \dot{\mathbf{x}} \quad (\text{A II-16})$$

Replacing $\Delta \dot{\mathbf{x}}$ given by Eq. (A II-15) in Eq.(A II-16), we obtain:

$$\Delta \dot{\mathbf{y}} = \mathbf{C} \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x} + \mathbf{C} \mathbf{G}_{\mathbf{x}_0, \eta_0} \Delta \eta \quad (\text{A II-17})$$

By inverting Eq. (A II-17), we can express the input variation at a given initial condition $(\mathbf{x}_0, \eta_0)$ as follows:

$$\Delta \eta = (\mathbf{C} \mathbf{G}_{\mathbf{x}_0, \eta_0})^{-1} (\Delta \dot{\mathbf{y}} - \mathbf{C} \mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta \mathbf{x}) \quad (\text{A II-18})$$

The control problem consists of following a reference signal r . Therefore, the tracking error e_θ may be defined as the difference between the reference signal r , and the output signal y , such as:

$$\begin{aligned} e_\theta &= r - y \\ \Rightarrow \dot{e}_\theta &= \dot{r} - \dot{y} \end{aligned} \quad (\text{A II-19})$$

Considering $\dot{y} \approx \Delta \dot{y}$ for a small sample time, i.e. the local output derivative is equal to its variation, and recalling from Eq. (A II-19) that $\dot{y} = \dot{r} - \dot{e}$, then the variation of the control input $\Delta \eta$ can then be expressed as follows:

$$\Delta\eta = (\mathbf{C}\mathbf{G}_{\mathbf{x}_0, \eta_0})^{-1} (\dot{r} + v_2 - \mathbf{C}\mathbf{F}_{\mathbf{x}_0, \eta_0} \Delta\mathbf{x}) \quad (\text{A II-20})$$

where v_2 is an auxiliary input chosen such as the error derivative satisfies the following dynamics:

$$\dot{e}_\theta = -v_2 \quad (\text{A II-21})$$

The last equation indicates that the error dynamics can be stabilized by choosing an appropriate linear expression for the auxiliary input v_2 .

II.3.4 PID controller

As stated, the selection of the auxiliary input does not require hard requirements on the performance. The only requirements are the system stability and convergence. A simple linear PID controller is sufficient to meet these requirements. However, in order to allow proper control adaptation, the PID signal should not take over the adaptive components, which means that the gains should not be overly tuned and should remain constant for all flight conditions. In this case, we ensure that the control part is fulfilled by the adaptive elements. The output v_2 from the PID can be written in the Laplace form as follows:

$$v_2 = \left(K_P + \frac{K_I}{s} + K_D s \right) e_\theta \quad (\text{A II-22})$$

where K_P , K_I , and K_D are the proportional, integral and derivative gains, respectively.

II.3.5 Online Parameter Estimation

Finding the analytical expressions of $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ and $\mathbf{G}_{\mathbf{x}_0, \eta_0}$ given in Eq. (A II-14) may be a complex task because it involves deriving unknown nonlinear functions. $\mathbf{F}_{\mathbf{x}_0, \eta_0}$ can also be referred to as the Hessian matrix. An alternative consists of estimating them by using approximation methods such as the Recursive Least Square (RLS) method. It consists of an adaptive filter that searches for linear approximation of the aircraft dynamics model at a local level. The RLS method has been used for control purposes on quadcopters, fixed wing aircraft, and even for reinforcement learning (Heyer et al., 2020 ; Konatala et al., 2021). Its principle lies on the fact that the output parameters are computed solely from the state increments $\Delta\mathbf{x}$ and the control input increments $\Delta\eta$. No prior knowledge of the aircraft dynamics is required but only measurements.

Converting the results of Eq. (A II-17) into a discrete form, the following equation is obtained:

$$\Delta y_{k+1} = \mathbf{C}\tilde{\mathbf{F}}_k \Delta\mathbf{x}_k + \mathbf{C}\tilde{\mathbf{G}}_k \Delta\eta_k \quad (\text{A II-23})$$

which can also be rewritten in a more compact matrix form as follows:

$$\Delta y_{k+1} = \mathbf{C}[\Delta \mathbf{x}_k \quad \Delta \eta_k] \begin{bmatrix} \tilde{\mathbf{F}}_k^T \\ \tilde{\mathbf{G}}_k^T \end{bmatrix} \quad (\text{A II-24})$$

In this case, Δy_{k+1} corresponds to the increment of the output variation, and is a function of the current states' variation $\Delta \mathbf{x}_k$ and control input $\Delta \eta_k$. The objective consists of replacing $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{G}}_k$ by their estimates $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$. Therefore, Eq. (A II-24) becomes:

$$\Delta \hat{y}_{k+1} = \mathbf{C}[\Delta \mathbf{x}_k \quad \Delta \eta_k] \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix} \quad (\text{A II-25})$$

Considering the matrix of parameters $\hat{\Theta}_k = \begin{bmatrix} \hat{\mathbf{F}}_k^T \\ \hat{\mathbf{G}}_k^T \end{bmatrix}$ and the state-input vector of previous increments $\mathbf{X}_k = [\Delta \mathbf{x}_k \quad \Delta \eta_k]^T$, the output variation $\Delta \hat{y}_{k+1}$ can be predicted as follows:

$$\Delta \hat{y}_{k+1} = \mathbf{C} \mathbf{X}_k^T \hat{\Theta}_k \quad (\text{A II-26})$$

where $\hat{\Theta}_k$ corresponds to the estimates of Θ_k , $\Delta \hat{y}_{k+1}$ corresponds to the estimates of Δy_{k+1} , and the output matrix $\mathbf{C} = [0 \ 0 \ 0 \ 1 \ 0]$ is defined so that the estimated output $\Delta \hat{y}$ corresponds to the estimated pitch angle $\Delta \hat{\theta}$.

The vector of parameters $\hat{\Theta}$ can be determined online using the RLS method. Let's consider the innovation error matrix ϵ_k as the difference between the actual states vector and its estimate:

$$\epsilon_k = \Delta \mathbf{x}_k - \Delta \hat{\mathbf{x}}_k \quad (\text{A II-27})$$

Then, the vector of parameters $\hat{\Theta}_k$ can be updated numerically (Haykin, 2013 ; Heyer et al., 2020):

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + \frac{\mathbf{L}_{k-1} \mathbf{X}_k}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \epsilon_k \quad (\text{A II-28})$$

where

$$\mathbf{L}_k = \frac{1}{\kappa} \left(\mathbf{L}_{k-1} - \frac{\mathbf{L}_{k-1} \mathbf{X}_k \mathbf{X}_k^T \mathbf{L}_{k-1}}{\kappa + \mathbf{X}_k^T \mathbf{L}_{k-1} \mathbf{X}_k} \right) \quad (\text{A II-29})$$

and where $\mathbf{L}_k$ is the covariance matrix, and κ is the forgetting factor. The two matrices $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ in $\hat{\Theta}_k$ represent the local longitudinal dynamics model of the aircraft. The RLS parameters

to be initialized are the forgetting factor κ , the matrix of covariance $\mathbf{L}_0$, and the matrix of parameter Θ_0 .

II.3.6 Pitch Adaptive Neural Network Controller NN₂

The proposed controller uses two independent adaptive neural networks. As shown in Figure A II-1, the first neural network controller NN₁ in the inner-loop gives an adaptive command on the elevator v_{δ_e} , while the second neural network controller NN₂ is used in the outer-loop and provides the adaptive command signals v_θ to the C* parameter.

Using estimations frequently results in prediction errors in real-world applications [33]. As a result, the actual output variation may be stated as follows:

$$\begin{aligned}\Delta y_{k+1} &= \Delta \hat{y}_{k+1} + \varepsilon \\ \Rightarrow \Delta \hat{y}_{k+1} &= \Delta y_{k+1} - \varepsilon\end{aligned}\tag{A II-30}$$

where $\Delta \hat{y}_{k+1}$ is the predicted output, and ε is the modelling error.

Recalling from Eq. (A II-25), and considering Eq. (A II-30), the control increment $\Delta \eta_k$ at a given time t_k can therefore be expressed as follows:

$$\Delta \eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1}[\Delta y_{k+1} - \mathbf{C}\hat{\mathbf{F}}_k\Delta \mathbf{x} - \varepsilon]\tag{A II-31}$$

Considering the changes in Eq. (A II-19) and (A II-21), the control increment $\Delta \eta_k$ in Eq. (A II-31) becomes:

$$\Delta \eta_k = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1}[\dot{r} + v_2 - \mathbf{C}\hat{\mathbf{F}}_k\Delta \mathbf{x} - \varepsilon]\tag{A II-32}$$

In Eq. (A II-32), the two terms $\mathbf{C}\hat{\mathbf{G}}_k$ and $\mathbf{C}\hat{\mathbf{F}}_k\Delta \mathbf{x}_k$ reflect the local dynamics of the aircraft. Furthermore, it is obvious that they include estimation errors ε , since they are calculated using the RLS approach. In fact, the combination of these two terms suggests that the dynamics may not be correctly inverted, which may lead to a poor control signal. Therefore, to correct the uncertainties and estimation errors for $\hat{\mathbf{G}}_k$ and $\hat{\mathbf{F}}_k$, an adaptive term v_θ is suggested to be added into the control law, such that:

$$\varepsilon = v_\theta\tag{A II-33}$$

In this study, a feedforward neural network was used as an adaptive element to learn the adaptive term v_θ and compensate for the uncertainties of $\hat{\mathbf{G}}_k$ and $\hat{\mathbf{F}}_k$, while ensuring a zero-tracking error on the pitch angle θ .

Among the common neural network architectures, feedforward neural networks have the simplest and most basic structure. This type of neural network is typically used to approximate a mathematical function from a data set in regression problems.

As shown in Figure A II-2a, an artificial neuron consists of a vector of inputs $[x_1 \ x_2 \ \dots \ x_n]$, which are multiplied by a weight matrix $[w_1 \ w_2 \ \dots \ w_n]$. The resulting products are then summed, and fed into an activation function σ , which produces a scalar output value y . The Feedforward Neural Network (FFNN) is a type of neural network, for which the information flows in one direction (forward), as shown in Figure A II-2b. Information enters the network through an input layer, then passes through a series of hidden layers, before being extracted through an output layer. The effectiveness of the FFNN depends on its hyperparameters, i.e., the number of neurons and hidden layers, the type of activation function σ , and the values of the weights $\mathbf{W}$ and $\mathbf{V}$.

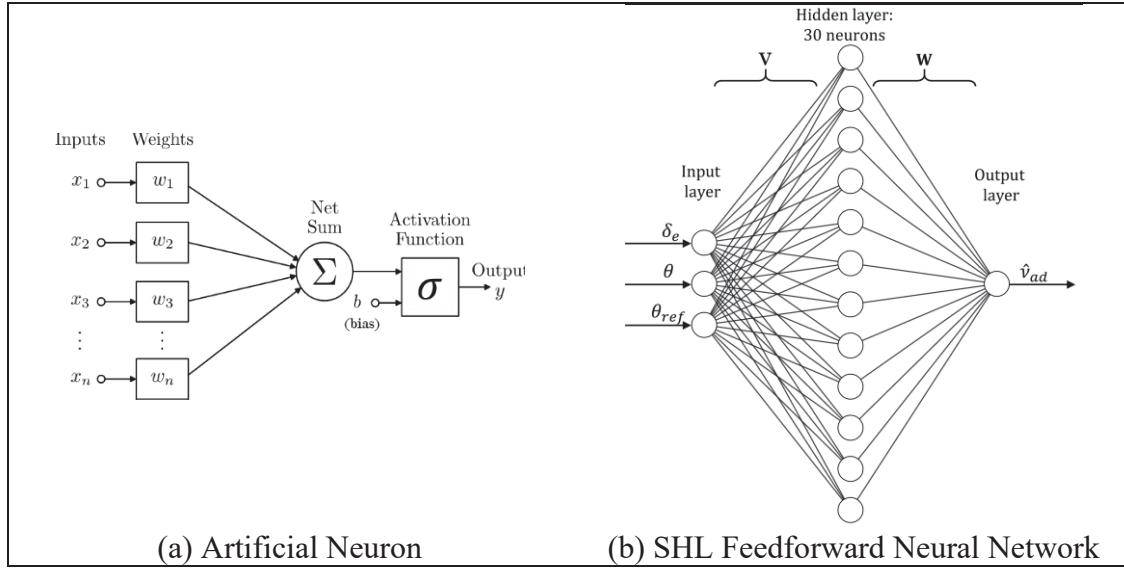


Figure A II-2 Representation of an Artificial Neuron (a) and a SHL Feedforward Neural Network (b)

In this study, a Single Hidden Layer (SHL) FFNN was used for pitch control. Therefore, the resulting output of the FFNN in Figure A II-2b can be expressed as follows (Stevens et al., 2015):

$$\hat{v}_{ad(n \times 1)} = \mathbf{W}_{(N \times n)}^T \sigma_{(N \times N)} \left(\mathbf{V}_{(m \times N)}^T \bar{\mathbf{x}}_{2(m \times 1)} \right) \quad (\text{A II-34})$$

where $\mathbf{W}$ and $\mathbf{V}$ are the weight matrices associated with the hidden and input layers, respectively, σ is the activation function, and $\bar{\mathbf{x}}_2$ is the vector of inputs of the network, defined such as $\bar{\mathbf{x}}_2 = [\delta_e \ \theta \ \theta_{ref}]^T$. The sizes of the weight matrices $\mathbf{W}$ and $\mathbf{V}$ are $N \times n$ and $m \times N$,

respectively, where m is the number of network inputs (i.e., 3), n is the number of network outputs, and N is the number of neurons in the hidden layer.

For each neuron in the hidden layer, a tangent sigmoid is defined as follows (Haykin, 2009):

$$\sigma_i(z) = \frac{1}{1 + \exp(-z)} \quad (\text{A II-35})$$

where i is the position of the neuron in the hidden layer. A linear function was used for all remaining neurons in the input and output layers.

The weights of the neural networks are updated using the backpropagation principle (Kim & Calise, 1997 ; Stevens et al., 2015), based on the following adaptation laws:

$$\begin{aligned} \dot{\mathbf{W}} &= -[(\boldsymbol{\sigma} - \boldsymbol{\sigma}' \mathbf{V}^T \bar{\mathbf{x}}) e_\theta]^T + \lambda \|e_\theta\| \mathbf{W} \boldsymbol{\Gamma}_2 \\ \dot{\mathbf{V}} &= -\boldsymbol{\Gamma}_2 [\bar{\mathbf{x}} e_\theta^T \mathbf{W}^T \boldsymbol{\sigma}' + \lambda \|e_\theta\| \mathbf{V}] \end{aligned} \quad (\text{A II-36})$$

where $\dot{\mathbf{W}}$ and $\dot{\mathbf{V}}$ are the updates of the weight matrices, $\boldsymbol{\Gamma}_2$ is a diagonal matrices of learning rates, $\boldsymbol{\sigma}'$ is the gradient of $\boldsymbol{\sigma}$, λ is a scalar used for the e-modification of the backpropagation algorithm (Johnson & Calise, 2003 ; Stevens et al., 2015) and is equal to 1, and e_θ is the tracking error.

The signal from the neural network is then amplified with a fixed gain K_{NN_θ} , which allows us to control the effect of the adaptive term. Therefore, by use of Eq. (A II-33) and Eq. (A I-41), the adaptation signal v_θ in Figure A II-2 may be expressed as follow:

$$v_{ad} = K_{NN_\theta} \hat{v}_{ad} = K_{NN_\theta} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}_2) = v_\theta \quad (\text{A II-37})$$

The NN is used to take out the most of its adaptive capabilities, its tuning process is quite simple and allows the model to achieve the expected results. It is only necessary to set the number of layers, the number of neurons and the learning rate $\boldsymbol{\Gamma}_2$.

II.3.7 C* Adaptive Neural Network Controller NN₁

The adaptive neural network controller NN₁ in the inner loop has the same structure as the adaptive neural network controller for the pitch control NN₂. Both neural networks are identical in their structure, but work with different learning rates, adaptive laws, and give different command signals according to the control objectives. For the neural network controller on the C* parameter, the adaptation law can be defined as follows:

$$v_{\delta_e} = K_{NN_{\delta_e}} \hat{v}_{ad} = K_{NN_{\delta_e}} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}_1) \quad (\text{A II-38})$$

In this case, the inputs of the neural network $\bar{\mathbf{x}}_1 = [\delta_e \ C^* \ C^*_{ref}]^T$ is composed of the deflection of the elevator, the C^* parameter, and its reference value C^*_{ref} . $\mathbf{W}$ and $\mathbf{V}$ are the weight matrices and have different values than the ones from NN_2 .

As an analogy to the neural network on the pitch control NN_2 , the neural network for the C^* parameter requires the tuning of the learning rate Γ_1 , the number of layers, and the number of neurons. In addition, a single hidden layer is used in this study, reducing the number of parameters to be tuned.

II.3.8 Control laws

As shown in Figure A II-3, the control law on the elevator command $\Delta\delta_e$ can be summarized as follows:

$$\Delta\delta_{e\ cmd} = v_1 + v_{\delta_e} = \left(K_P + \frac{K_I}{s} + K_D s\right) e_{C^*} + K_{NN_{\delta_e}} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}_1) \quad (\text{A II-39})$$

were v_1 and v_{δ_e} are determined from Eq. (A II-4) and (A II-38), respectively.

The pitch controller gives a command signal on the C^* parameter. Therefore, from Eq. (A II-22) and (A II-32), we obtain:

$$C^*_{cmd} = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} [\dot{r} + \left(K_P + \frac{K_I}{s} + K_D s\right) e_\theta - \mathbf{C}\hat{\mathbf{F}}_k \Delta \mathbf{x} + K_{NN_\theta} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}_2)] \quad (\text{A II-40})$$

II.3.9 Handling Qualities Requirements and Reference model

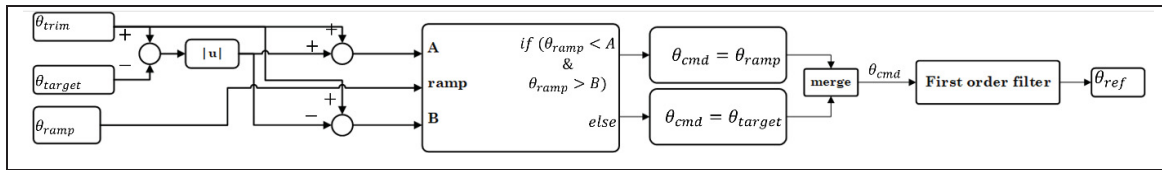


Figure A II-3 Pitch Reference Model Configuration

The model implements a subsystem that generates a pitch reference to the model. As shown in Figure A II-3, this subsystem takes as input the trim value of the pitch angle θ_{trim} in cruise. The value of θ_{trim} is different for each flight condition across the flight envelope. The pitch reference model also considers the pitch target value θ_{target} , and the desired pitch rate θ_{ramp} .

The simulation must start from the trimmed value of the pitch angle θ_{trim} , otherwise it may encounter a non-desired behavior. To solve the problem erratic and aggressive performance, a pitch rate reference is given in the form of a ramp signal. Therefore, the pitch signal between the initial pitch angle θ_{trim} and the target pitch angle θ_{target} is given as a ramp signal θ_{ramp} with a constant pitch rate value. The slope of the ramp is a custom variable that can be set

prior. The direction of the ramp (up or down) follows the sign of the difference between θ_{trim} and θ_{target} . The ramp is positive (increases) when $\theta_{target} - \theta_{trim} > 0$, and negative (decreases) when $\theta_{target} - \theta_{trim} < 0$. The resulting output is the pitch command signal θ_{cmd} .

A first order filter is added in order to smooth the transition between the trim pitch angle θ_{trim} , its ramp value θ_{ramp} , and the target value θ_{target} . This filter allows to avoid abrupt changes in the command signal. The first order transfer function can be written as follows:

$$\frac{\theta_{ref}}{\theta_{cmd}} = \frac{\omega_c}{s + \omega_c} \quad (\text{A II-41})$$

where ω_c is the cut-off frequency and may be tuned by the designer.

The resulting output of the filter is the pitch reference θ_{ref} , which serves as the input to the controller in Figure A II-3. The ramp value indirectly controls the pitch rate because the controller will closely follow the pitch reference signal θ_{ref} . This allows us to maintain the stability margins for the pitch rate. Very good results are obtained during tests when the ramp value is set to 1.2 deg/s or less. It is important to note that the time to reach the target pitch angle θ_{target} depends on the flight conditions, since the initial pitch trim angle θ_{trim} is different for each condition.

II.4 Model Simulation and Validation

This section presents the simulation results for the proposed controller configuration shown in Figure A II-1.

II.4.1 Controller Tuning Parameters

The tuning parameters in the inner and outer loops are presented in Table A II-1, including the values of the PID gains in both loops, the values of the neural network learning rates and other tuning variables presented in the previous sections. The inner loop control parameters were tuned first, before the outer loop parameters were tuned. The following values gave the best results and remained constant for all flight conditions during the cruise phase.

Table A II-1 Design Parameters for the Pitch Controller

Inner loop controller		
Controller element	Parameter	Value
C* reference model , Eq.(A II-3)	ω_n, ξ	1 rad/s, 0.5
PID controller , Eq.(A II-4)	K_p, K_i, K_d	-15, -2, -1
NN₁ controller	Learning rate $\Gamma_1, K_{NN\delta_e}$ Number of layers Number of neurons	0.5, 30 1 30
Pitch rate gain	K_q	12.4
Noise filter , Eq. (A II-6)	K, τ_N	0.125, 0.125 sec
Washout filter , Eq. (A II-5)	a, b, τ_W	1.6, 6.7, 1.6 sec
Outer loop controller		
Controller element	Parameter	Value
First order filter , Eq. (A II-41)	ω_c	2 rad/s
PID controller , Eq.(A II-22)	K_P, K_I, K_D	-1.5, -3, -1.05
NN₂ controller	Learning rate Γ_2 , $K_{NN\theta}, K_e$ Number of layers Number of neurons	1.2, 1.5, 0.1 1 30
RLS Dynamic Inversion controller , Eq.(A II-28), Eq.(A II-29)	$\kappa, \mathbf{L}_0, \hat{\mathbf{F}}_0, \hat{\mathbf{G}}_0$	$1.015, 10^7 \mathbf{I}_{5,5}, \mathbf{I}_{4,4}, -\mathbf{0.1}_{4,1}$

The tuning of each control element follows an iterative process, for which the objective consists of reducing the tracking error e_θ . Specifications on the tuning procedure are presented in the discussion section, especially for tuning the RLS Dynamic Inversion Controller. Simulation and validation results are presented below.

II.4.2 Model Validation for one flight condition

The first simulation was performed for one flight condition at 35,000 ft and a calibrated airspeed of 290 knots, as shown in Figure A II-4. The pitch reference θ_{ref} was set with different ramp signal values during 160 sec in order to validate the accuracy of the model. The indirect control of the pitch rate q was validated for this flight condition validation test as it kept the value of the pitch rate q at the desired value of the ramp signal θ_{ramp} . The simulation was performed for an environment characterized by ideal conditions, i.e., without wind nor turbulence.

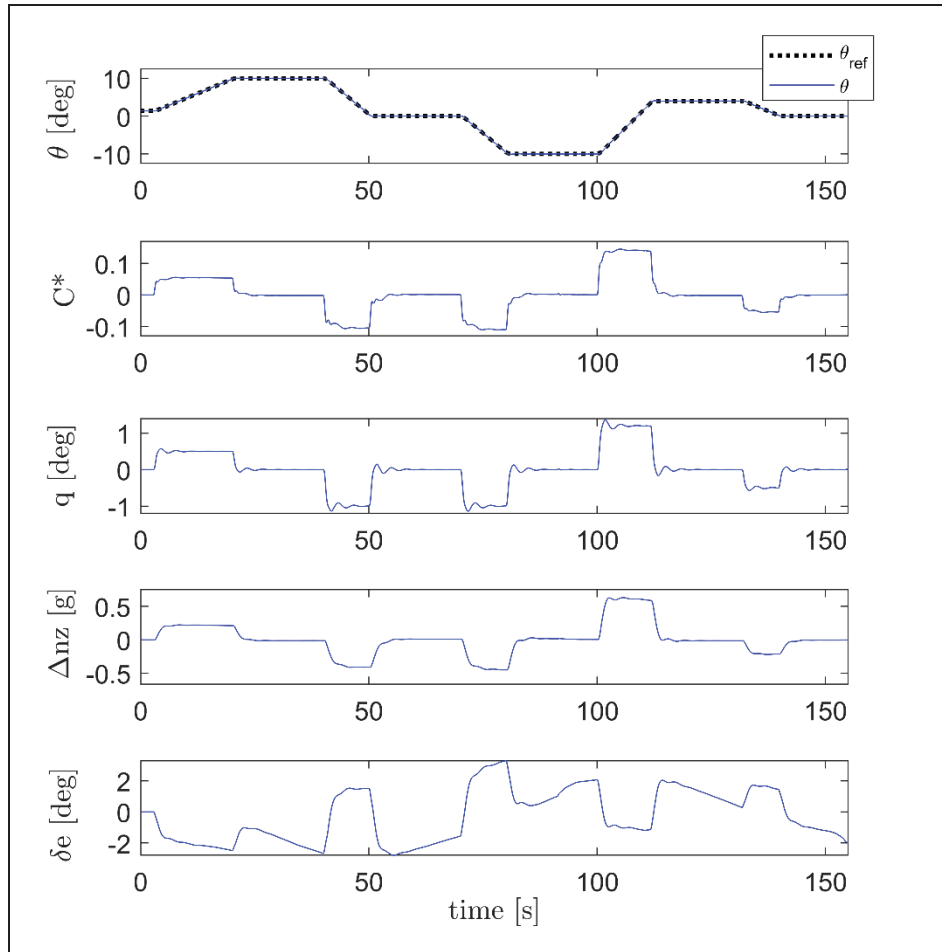


Figure A II-4 State responses on model validation for one flight condition at 35,000 ft and 290 knots

It can be observed in Figure A II-4 that the model follows the pitch angle reference with a high accuracy throughout the simulation as the pitch reference values vary between $\pm 10^\circ$ at different pitch rates. A time window is given allowing the model to stabilize at constant pitch rates. The oscillations on the pitch rate q are highly damped, and they occur only during the transition from one constant pitch angle to another. When the pitch angle reaches a constant value, the pitch rate q and the normal load factor Δn_z reach zero, which is the expected performance. As we can see, the load factor Δn_z also stabilizes at a constant value during the transition between pitch angles. Also, the elevator deflections δ_e are smooth and not saturated. In fact, they remain in a good range of $\pm 2.5^\circ$.

Figure A II-5 shows the variation of the hyperparameters of the adaptive elements during the simulation, i.e., the weights of the neural networks NN_1 and NN_2 , and the estimated matrices $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ from the Recursive Least Square estimation method. As can be seen in Figure A II-5, the weights of the neural networks adapt well during the simulation and provide appropriate

adaptation signals. Also, the values of $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ are well estimated since they do not diverge. We can also notice that any change in the pitch reference leads to sudden changes in the values of the hyperparameters, as well as in the values of $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$. This fact demonstrates the ability of the controller to adapt to any given pitch reference value.

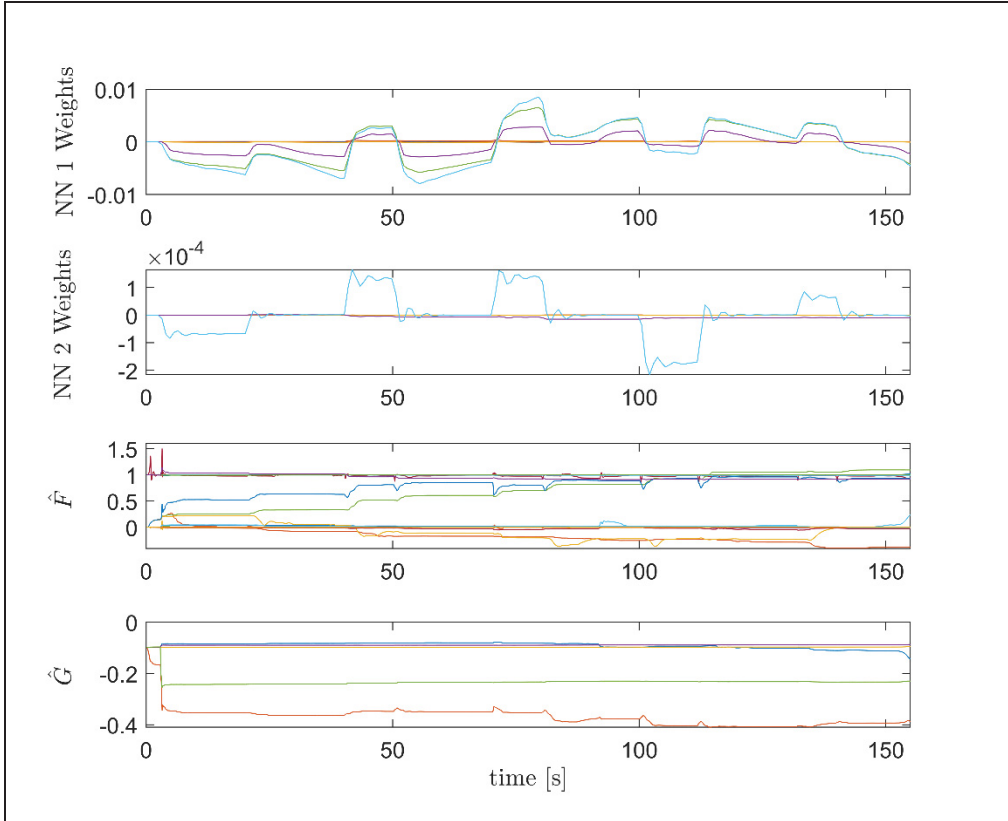


Figure A II-5 Neural Networks hyperparameters and estimated state and control matrices for one simulation at 35,000 ft and 290 knots

II.4.3 Comparison of different controller configurations

Figure A II-6 shows the results from testing different configurations of the controller architecture. During the simulation a pitch reference θ_{ref} of 10° is set, starting from a pitch trim angle $\theta_{trim} = 1.5^\circ$ and under ideal conditions. The flight condition used has an initial altitude of 35,000 ft and a calibrated airspeed of 290 knots.

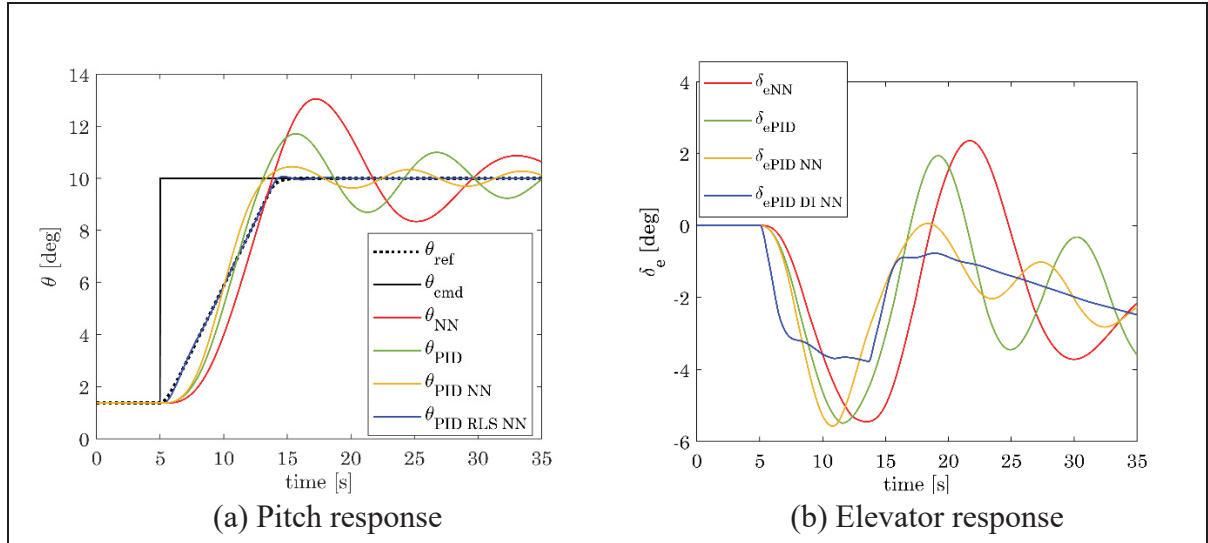


Figure A II-6 Comparison of different controller configurations for one flight condition at 35,000 ft and 290 knots

It can be observed that all configurations converge to the pitch reference θ_{ref} . However, we can also notice that the PID + NN configuration alone is not enough to achieve the desired performance. The best result is obtained when the configuration that combines the three control elements: the PID, the NN, and the dynamic inversion. In this case, the model follows the pitch reference θ_{ref} with high accuracy.

II.4.4 Validation on step signal over the flight envelope

The proposed flight controller was further validated for 63 cruise conditions across the flight envelope of the Cessna Citation X. It should be noted that the simulations were performed with the values given in Table A II-1, and that all tuned parameters remained constant. Therefore, the resulting controller is unique, and no further gain design or modification was required.

The cruise conditions cover the aircraft flight envelope in terms of altitudes from 5 000 ft to 40,000 ft with an increment of 5 000 ft, and calibrated airspeeds from 170 knots to 330 knots with an increment of 20 knots. The first validation on the step signal was performed for all flight conditions. The model used a second order transfer function for the pitch reference model with a damping ratio of 0.8 and a natural frequency of 0.1 rad/s. As shown in Figure A II-7a, the controller is perfectly capable of following the pitch reference signal for all cruise conditions.

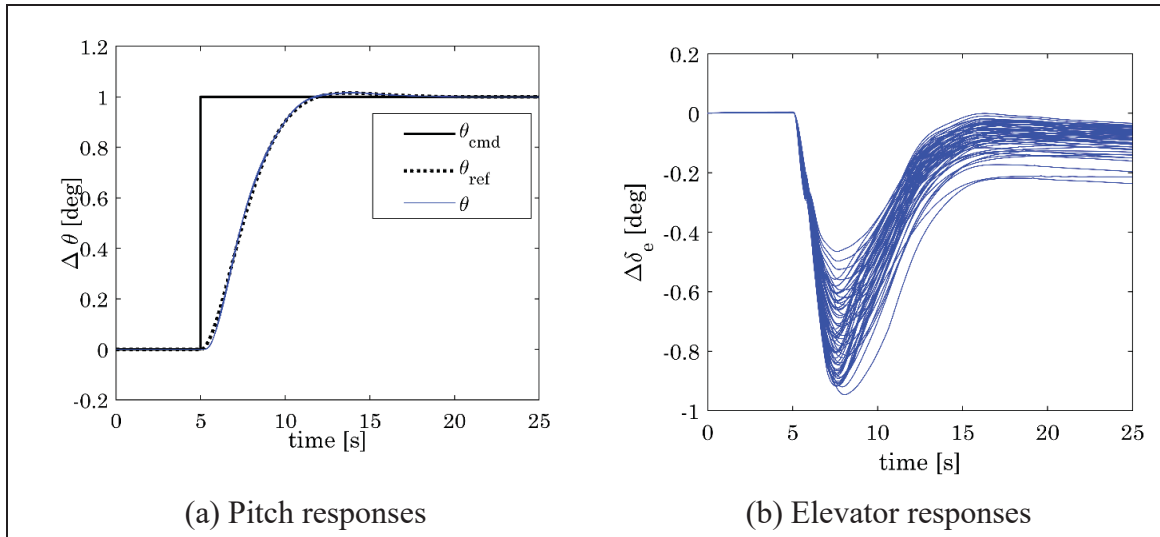


Figure A II-7 Pitch and elevator responses to a step command signal for all flight conditions

II.4.5 Validation for ideal conditions over the flight envelope

In Figure A II-8, two pitch rates were applied, $0.5^\circ/\text{s}$ and $1.2^\circ/\text{s}$, and three different pitch target values were applied. The difference in the time it takes to reach the target pitch value is evident since the pitch rate values are different. There is also a small overshoot is caused by a higher pitch rate. This overshoot is considered acceptable and demonstrates the ability of the controller to correct and converge to the desired pitch reference value. The expected results for the pitch response θ under ideal conditions are obtained because the model reaches the reference value at the desired rate.

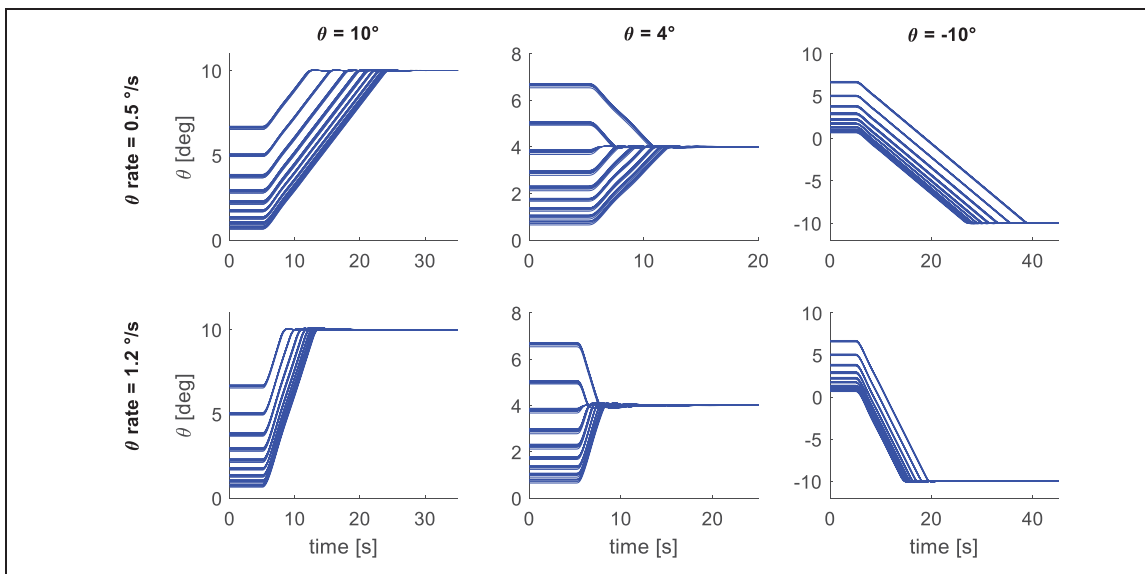


Figure A II-8 Validation for ideal conditions over the flight envelope

When comparing a positive target pitch angle $+10^\circ$ with a negative target pitch angle -10° , it can be noticed that there is no overshoot while reaching the negative target pitch angle. For $\theta_{target} = 4^\circ$ in Figure A II-8, we can see that the controller is able to reach the target pitch θ_{target} starting from any given initial pitch trim angle θ_{trim} .

II.4.6 Aircraft dynamics including uncertainties.

The next validation step consists of simulating the proposed controller under wind gust conditions. The wind gust velocity can be decomposed in the 3 axes in the aircraft body reference, such as (Nelson, 1998):

$$V_W = [u_g \ v_g \ w_g]^T \quad (\text{A II-42})$$

were u_g , v_g and w_g and the horizontal, lateral, and vertical components of the wind speed, respectively. In our case, since we are dealing with longitudinal control, we will only consider the horizontal and vertical components of the wind, u_g and w_g , respectively. Therefore, v_g is set to 0 m/s.

In the aircraft simulation platforms, the wind gust velocities are added directly to the corresponding aircraft velocities in order to simulate the perturbations, which directly affects the aircraft dynamics. Different values for u_g and w_g were applied to validate the controller performance. Extreme wind gust conditions (Frost, Camp, & Wang, 1978) were tested in order to evaluate the overall performance for all flight conditions in cruise phase.

II.4.7 Robustness Validation for Wind Gust

As shown in Figure A II-9, 3 different configurations of wind gusts were applied. The wind gust started at 5 seconds for all flight conditions. A positive sign on u_g is used for tailwind, while a positive sign on w_g means that the wind comes from under the nose of the aircraft.

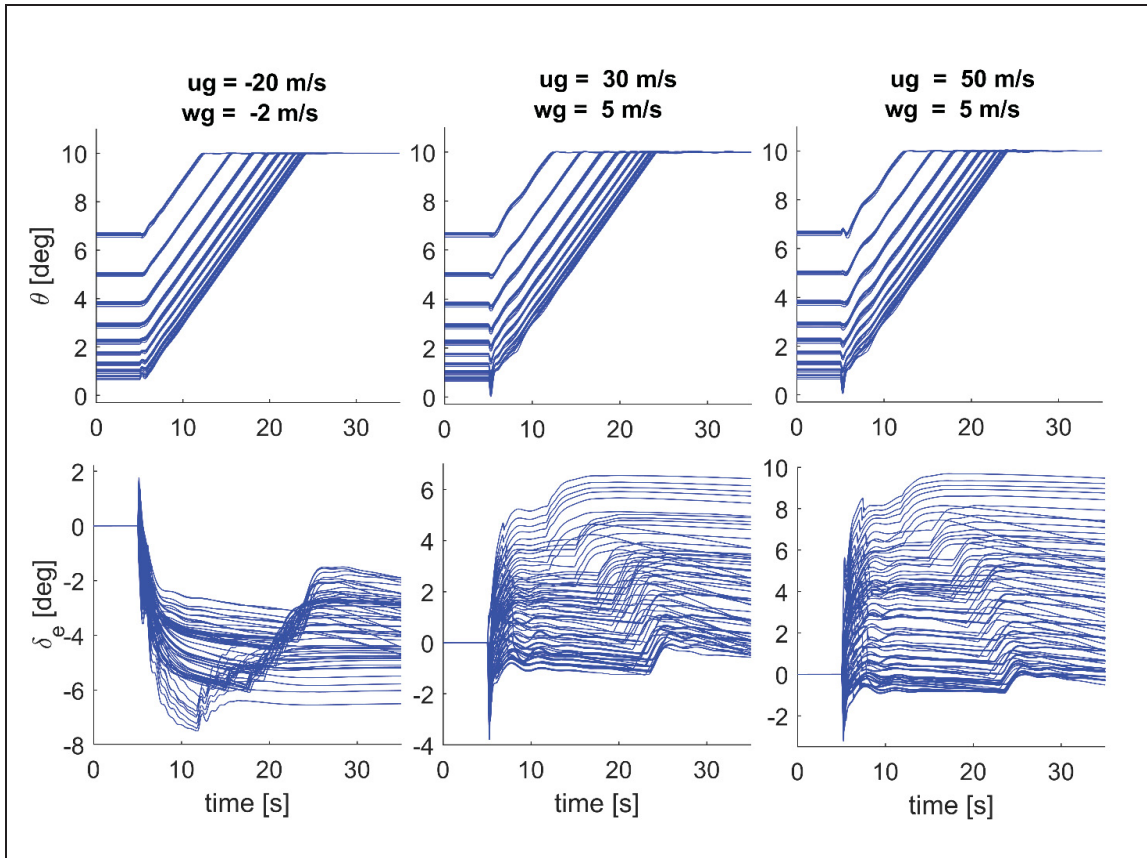


Figure A II-9 Robustness validation for wind gust.

When the wind gust factor is added, it is noticed how much it affects the controller performance. Oscillations occur during the transition from θ_{trim} to θ_{target} . The oscillations are less pronounced when the controller reaches a constant pitch angle value θ_{target} . The results in Figure A II-9 show that the robustness of the controller is good under wind gust conditions. In fact, the controller is able to give an adaptive command signal to the elevators in the presence of wind gusts.

II.4.8 Robustness Validation for wind turbulence

A last set of validations is performed while applying wind shear and turbulence to the aircraft flight dynamics. The Dryden turbulence models (Nelson, 1998) for the longitudinal component with moderate intensities were considered during the simulation. In addition, a wind shear (Nelson, 1998) was applied at 5 sec, with an amplitude of 30 m/s on the horizontal component, 2 m/s on the vertical component and a length scale of 500 m.

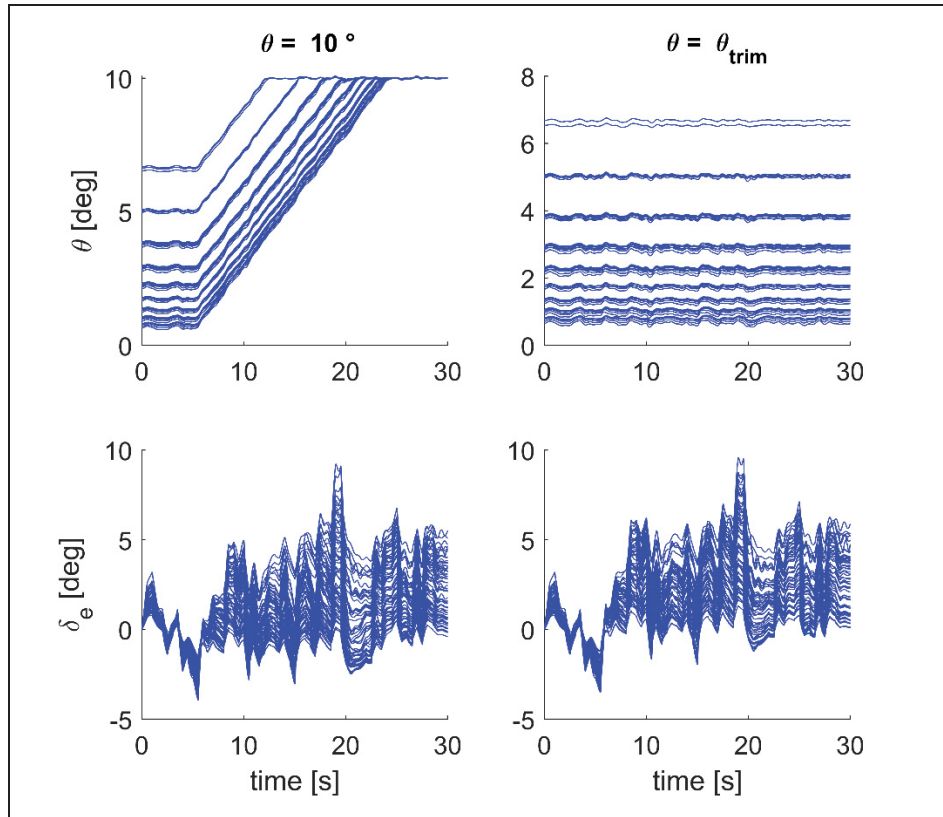


Figure A II-10 Robustness validation for wind turbulence and wind shear for all cruise conditions

Figure A II-10 shows the results of simulations under turbulence and wind shear. In this particular environment, we aim to observe the controller performance for two different target pitch angles θ_{target} . As we can see, the controller is able to follow the pitch angle reference. On the left hand side of Figure A II-10, it can be seen that the controller can reach the desired target pitch angle at $+10^\circ$, while the right hand side of Figure A II-10, it can be seen that it can keep the pitch values at the trimmed conditions despite the presence of turbulences. It should be noticed that the same conditions are applied to both cases, and as we can see, the controller is sending almost the same command signals δ_e on the elevators in both conditions. This aspect means that the proposed controller is robust while trying to reach a desired pitch target θ_{target} under turbulent conditions.

II.5 Discussion of results

Our research aims to address the control limitations and robustness for pitch angle control. The pitch rate control system was successfully implemented on the Cessna Citation X business jet aircraft model. The main benefits of the controller are that no prior knowledge of the aircraft dynamics is needed because the control input is computed using online estimations of the aircraft local dynamics. Consequently, a dual adaptation was fulfilled, first using online estimation and dynamic inversion, and second by using adaptive Neural Networks.

On the contrary to classical control algorithms that use gain scheduling, the control architecture proposed in this study considered constant PID gain values throughout all flight situations. Thus, the PID serves solely as an error selector, and the adaptive components, including the neural networks and dynamic inversion, allow to control the aircraft even in the presence of model uncertainties and nonlinearities, such as wind and turbulence conditions.

The RLS initial conditions have a significant impact on the dynamic inversion. A low forgetting factor κ causes the estimated parameters $\hat{\Theta}$ to oscillate. As a result, the algorithm will tend to forget prior increments. On the other hand, a forgetting factor near to 1 indicates that the RLS keeps most of the previous values in memory. This ability of the RLS algorithm has been demonstrated in previous research (R. P. Andrianantara, Ghazi, et al., 2024 ; R. P. Andrianantara et al., 2023a ; Quintin et al., 2024). Additionally, a low initial value of the covariance matrix $\mathbf{L}_0$ causes the estimated values $\hat{\Theta}$ to converge slowly. One way to solve the problem consists of updating the covariance matrix $\mathbf{L}_0$ each time the error ϵ in Eq. (A II-27) reaches a threshold value. By this way, the values of $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{G}}_k$ would not oscillate, and numerical singularities or instability caused by division by 0 would be avoided.

Before applying the controller on the overall flight envelope, it was required to firstly design and implement it on a small set of 10 flight conditions in order to save time during its tuning. The designed controller has not considered the limitations in the pitch angle values; therefore, higher pitch reference values should be tested. However, the variations of the pitch reference values between $\pm 10^\circ$ were high enough for longitudinal control.

The pitch reference model improved the controller performance, as a desired pitch rate value was set while between two target pitch angles. The pitch rate was tuned by the designer in order to keep its value within the aircraft pitch rate limitations. Therefore, the final controller reached an indirect pitch rate control with its values varying within $\pm 1.2^\circ/\text{s}$.

II.6 Conclusion

An adaptive pitch controller for the Cessna Citation X has been successfully designed, implemented, and validated on the overall flight envelope for all cruise conditions. The controller includes an inner loop architecture which controls the C^* parameter, thus allowing neutral static stability at high-speed conditions. The controller can follow a desired pitch reference signal with a desired pitch rate. It includes 3 main features, such as baseline PID controllers, whose gains remain constant, two adaptive neural networks and a recursive least square dynamic inversion. Therefore, the controller performs two adaptations, firstly with the RLS dynamic inversion, and secondly with the neural network controllers.

Additional filters were added in order to make the controller more robust; however, these filters also remain unchanged for all flight conditions, ensuring that the neural networks and the RLS dynamic inversion fulfill their role for adaptation. The resulting controller is able to provide very good command signals to the elevators in order to follow the pitch angle reference.

The control design for aircraft systems requires an iterative testing approach, especially during validation of different control techniques applied. It is imperative that each control method (PID, NN or Dynamic Inversion) is tested separately and together in order to guarantee the proposed controller's convergence and stability, and to ensure that each method fulfils their design role. The fact that the control input is calculated using the state increments for Dynamic Inversion and does not require knowledge of the aircraft dynamics is one of the main characteristics of the proposed control algorithm. In addition, because of the fact that the actuator dynamics was considered during the design, the proposed control algorithm is suitable to control systems with high degree of nonlinearity.

ANNEX III

FLYING QUALITIES ASSESSMENT FOR NONLINEAR ADAPTIVE CONTROL VALIDATION ON THE CESSNA CITATION X LONGITUDINAL AND LATERAL DYNAMICS

Rojo Princy Andrianantara ^a, Georges Ghazi^a, and Ruxandra Botez ^a

^a Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE), System Engineering Department, École de technologie supérieure, 1100 Rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3

Article published in *AIAA SciTech 2025 Forum*, January 2025

<https://doi.org/10.2514/6.2025-1826>

Résumé

Cet article traite de la validation des contrôleurs adaptatifs non linéaires par l'évaluation des qualités de vol, appliquées spécifiquement à la dynamique de vol du Cessna Citation X. Deux contrôleurs non linéaires distincts ont été conçus pour le contrôle à bas niveau de la dynamique longitudinale et latérale, respectivement. Les deux algorithmes de contrôle comprennent un contrôleur de base proportionnel-dérivé-intégral fixe, un contrôleur d'inversion dynamique non linéaire utilisant l'estimation réursive des moindres carrés et un contrôleur adaptatif à réseau de neurones. Le contrôle longitudinal consiste à contrôler le taux de tangage de l'avion, tandis que le contrôle latéral consiste à contrôler le taux de roulis de l'avion et à assurer la stabilisation du taux de lacet. Les qualités de vol ont été tirées de la norme MIL-STD 1797A, dont les exigences de niveau 1 ont été satisfaites pour la dynamique période courte, le roulis et le roulis hollandais. Les contrôleurs obtenus sont des contrôleurs adaptatifs à modèle de référence (MRAC), ce qui signifie que des signaux de référence spécifiques ont été suivis avec les performances souhaitées en termes de taux de tangage et de taux de roulis. Les contrôleurs de vol ont été testés dans 64 conditions de vol en phase de croisière couvrant l'ensemble de l'enveloppe de vol du Cessna Citation X. Les simulations ont démontré une erreur de suivi minimale, puis l'identification de la fonction de transfert a été effectuée pour chaque simulation afin d'extraire les caractéristiques de performance transitoires. Les résultats ont montré que les qualités de vol de l'avion se situaient dans le niveau 1 pour les contrôleurs adaptatifs longitudinaux et latéraux.

Abstract

This paper discusses the validation of nonlinear adaptive controllers by assessing the flying qualities, specifically applied to the flight dynamics of the Cessna Citation X. Two distinct nonlinear controllers were designed for low-level control of longitudinal and lateral dynamics, respectively. Both control algorithms include a fixed Proportional-Derivative-Integral baseline controller, a Nonlinear Dynamic Inversion controller using Recursive Least Square estimation,

and an adaptive Neural Network controller. The longitudinal control consists of controlling the aircraft pitch rate, while the lateral control task consists of controlling the aircraft roll rate and ensuring the stabilization of the yaw rate. The flying qualities were taken from MIL-STD 1797A for which Level 1 requirements were met for short period, roll and Dutch-roll dynamics. The resulting controllers consist of Model Reference Adaptive Controllers (MRAC), which means that specific reference signals were tracked with desired performances on the pitch and roll rates. The flight controllers were tested for 64 flight conditions in the cruise phase covering the overall flight envelope of the Cessna Citation X. Simulations demonstrated minimal tracking error and then transfer function identification was fulfilled for each simulation to extract the transient performance characteristics. Results have shown that the aircraft flying qualities were lying within the Level 1 for both adaptive longitudinal and lateral controllers.

III.1 Introduction

Flying Qualities (FQ) characterize the maneuvering capabilities of an aircraft. They describe how the aircraft responds to pilot commands during maneuvers or when performing specific tasks (Department of Defense, 1997). Flying qualities are essential for analyzing the aircraft stability and control characteristics. In addition, flying qualities can also be assessed to evaluate the controllers performance (Cook, 2012 ; Saussie et al., 2003).

Flying qualities are typically designed for linearized aircraft dynamics, and thus linear controllers for which time-frequency analysis is possible (Department of Defense, 1997 ; Nelson, 1998). However, no formal method for validating nonlinear adaptive controllers has yet been certified in the aerospace industry (Wilkinson, Lynch, Bharadwaj, & Woodham, 2016). In fact, the main issues and challenge of adaptive elements (S. M. Hosseini, Bematol, et al., 2024a ; Tondji, Ghazi, & Botez, 2023 ; Zohreh Nejad, Aillot, Henry, Ghazi, & Botez, 2024) consist of their stability, convergence, and robustness, which remain uncertain, and therefore they present threats to human life. Several task groups, including NASA (Bhattacharyya, Cofer, Musliner, Mueller, & Engstrom, 2015 ; Soares & Burken, 2006), FAA (Bhattacharyya et al., 2015 ; Wilkinson et al., 2016), and EASA (Cima et al., 2021) are working towards certifying adaptive components applied in aerospace. Standards on the use of adaptive features are already considered and applied to Unmanned Aerial Vehicle (Hakim & Choukri, 2024 ; Koch III, 2019 ; Koch, Mancuso, West, et al., 2019), as UAVs present fewer risks to human life.

In summary, this paper aims to evaluate the aircraft flying qualities of the Cessna Citation X for its longitudinal (S. Hosseini, Ghazi, et al., 2023a ; S. M. Hosseini, Ghazi, et al., 2024a) and lateral flight dynamics (Ghazi & Botez, 2015b ; S. M. Hosseini, Bematol, et al., 2024a) in order to validate the proposed adaptive controllers. This paper does not assess the full spectrum of aircraft flying qualities, but focuses specifically on the short-period dynamics in the longitudinal axis, and Dutch-roll and roll dynamics in the lateral-directional axis (Department of Defense, 1997). In addition, a change in the controller parameters is fulfilled in the lateral-directional axis, which may substantially affect the aircraft flying qualities. Nevertheless, despite these changes, the lateral flying qualities remain within the Level 1 criteria.

This paper provides a detailed description of the Cessna Citation X flight dynamics and its configuration (R. P. Andrianantara, Ghazi, et al., 2024 ; R. P. Andrianantara, Mancera Coyotl, et al., 2024). Then, the flying qualities requirements specific to the Cessna Citation X are established for the cruise phase (Yamina Boughari, Botez, et al., 2017). An overview of the proposed longitudinal and lateral controllers is described, for which the control laws are explicitly given. In the simulation and result section, a summary of the considered cruise flight conditions is given, and simulations are performed for all flight conditions. The proposed control method consists of a Model Reference Adaptive Controller (S. Hosseini, Inga, et al., 2023); therefore, specific performance characteristics are given for the tracking references regarding short-period, Dutch-roll, and roll dynamics.

III.1.1 Cessna Citation X Model and Simulation Platform

The Cessna Citation X is among the fastest business jet aircraft. It has a maximum range of 3 091 nm, a ceiling altitude of 51,000 ft, and a maximum operating Mach number of 0.92. The Cessna Citation X is powered by two Rolls-Royce AEC3007C-1 turbofan engines with a maximum thrust of 6 442 lbf. The aircraft can carry 12 passengers and 3 crew members.

In order to develop the proposed flight controllers, a high-fidelity Matlab/Simulink simulation platform was used (Ghazi & Botez, 2015a), and is available at the LARCASE laboratory. This Simulink platform has been validated with a Level D Research Flight Simulator (RAFS) of the Cessna Citation X business jet. The Level D is the highest qualification level given by the Federal Aviation Administration. The nonlinear simulation platform implements complex physics-based components, such as atmosphere module, aerodynamic and engine modules, actuator dynamics, wind and turbulence models, forces and moment equations. The simulation platform has been used for the validation of several controllers (R. P. Andrianantara et al., 2023a ; R. P. Andrianantara, Mancera Coyotl, et al., 2024 ; Y. Boughari, Ghazi, et al., 2017a ; Yamina Boughari, Botez, et al., 2017).

The actuators are modeled using second-order nonlinear dynamics, which allows to increase the system's nonlinearity and complexity. By this way, it is possible to improve the generalizability of the proposed controllers on different control tasks with high degree of nonlinearity.

III.1.2 Flying qualities requirements

According to the aircraft classification given in the Flying Qualities standards (Department of Defense, 1997), the Cessna Citation X is categorized as a Class II mid-sized aircraft, for which the weight ranges between 5 000 lb and 30,000 lb. The proposed longitudinal and lateral controllers were designed for cruise operations, which correspond to Category B flight. The following Table A III-1 establishes the desired Level 1 requirements for short-period, Dutch-roll and roll flying qualities (Department of Defense, 1997).

Table A III-1 Level 1 Flying Qualities requirements for longitudinal and lateral-directional dynamics and category B flight phase

Longitudinal flying qualities	
Short period frequency, rad/s	$0.3 \leq \omega_{sp} \leq 2$
Control Anticipation Parameter	$0.085 \leq CAP \leq 3.6$
Settling time at 5%, sec	$T_s < 3$
Dropback	$-0.2 \leq \frac{D_{rb}}{q_{ss}} \leq 0.5$
Short-period time constant, sec	$0.1 \leq T_{\theta_2} \leq 1.5$
Delay allowed, sec	0.1
Lateral-directional flying qualities	
Roll time constant	$\tau_p < 1.4$ sec
Dutch-roll minimum damping,	$\min \xi_{dr} = 0.08$
Dutch-roll minimum frequency, rad/s	$\min \omega_{dr} = 0.4$ rad/s
Minimum $\xi_{dr}\omega_{dr}$	$\min \xi_{dr}\omega_{dr} = 0.15$
Delay allowed, sec	0.1

III.2 Controller design methodology

III.2.1 Longitudinal pitch rate controller

Figure A III-1 presents the designed adaptive algorithm for the pitch rate control of the Cessna Citation X. It includes a PID controller with fixed gain, an Approximate Dynamic Inversion controller using a Recursive Least Square estimation method, and an adaptive neural network controller (R. P. Andrianantara et al., 2023a ; R. P. Andrianantara, Mancera Coyotl, et al., 2024).

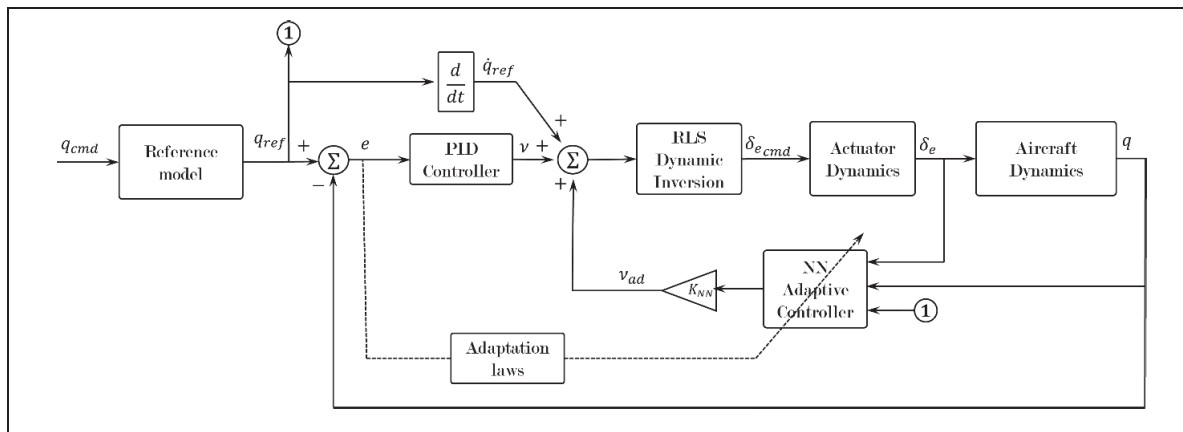


Figure A III-1 Designed pitch rate controller including PID, RLS Dynamic Inversion and NN controller

Adaptive Control law

The overall command law sent to the elevator is given in the following equation (its mathematical development is provided in (R. P. Andrianantara et al., 2023a)):

$$\Delta\delta_{e_k} = (\mathbf{C}\hat{\mathbf{G}}_k)^{-1} \left[\dot{r} + \left(K_P + \frac{K_I}{s} + K_D s \right) e - \mathbf{C}\hat{\mathbf{F}}_k \Delta\mathbf{x}_k - K_{NN} \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}) \right] \quad (\text{A III-1})$$

where e is the pitch rate error, $\mathbf{C} = [0 \ 0 \ 1 \ 0]$ is the output matrix corresponding to the pitch rate, $\dot{r}$ is the derivative of the reference signal, $\Delta\mathbf{x}_k$ is the longitudinal states increment vector at sample time k , σ is the tangent-sigmoid function of the neural network.

The control law sends the variation of the command input signal $\Delta\delta_{e_k}$ to the elevators. The PID controller includes a proportional gain K_P , an integral gain K_I , and a derivative gain K_D , whose values remain fixed for all simulations. The approximate dynamic inversion consists of estimating the local state space matrix $\hat{\mathbf{F}}_k$ and the local control matrix $\hat{\mathbf{G}}_k$ using the Recursive Least Square algorithm (Haykin, 2013). The neural network component is used to compensate for any estimation errors from the dynamic inversion. The weight matrices $\mathbf{W}$ and $\mathbf{V}$ of the neural network are dynamically updated using custom adaptive laws based on backpropagation algorithm (Stevens et al., 2015). A Single hidden layer neural network is enough for adaptation purpose.

Reference model

The aircraft longitudinal motion may be divided into two motions: a fast dynamic mode, also referred to as the short-period mode, and a slow dynamic mode, known as the phugoid mode. For low-level control, specifically for pitch rate control, only the short-period dynamics will be taken into account.

A reference model for the pitch rate command using the short-period approximation (Nelson, 1998 ; Stevens et al., 2015) and based on the following second-degree transfer function :

$$\frac{q_{ref}}{q_{cmd}} = \frac{\omega_{sp}^2 (1 + T_{\theta_2} s)}{s^2 + 2\xi_{sp}\omega_{sp}s + \omega_{sp}^2} \quad (\text{A III-2})$$

The short-period mode is an oscillatory motion with a specific natural frequency ω_{sp} and damping ratio ξ_{sp} . The zero of the transfer function in eq. (A III-2) is defined as $-\frac{1}{T_{\theta_2}}$, where T_{θ_2} is the short-period time constant. The requirements for their values are given in Table A III-1.

III.2.2 Lateral roll rate controller and yaw damper

Figure A III-2 shows the designed adaptive algorithm roll rate control and the yaw rate stabilization for the Cessna Citation X. The control algorithm includes a PID controller with fixed gains, an Approximate Dynamic Inversion controller using RLS estimation, and an adaptive neural network controller.

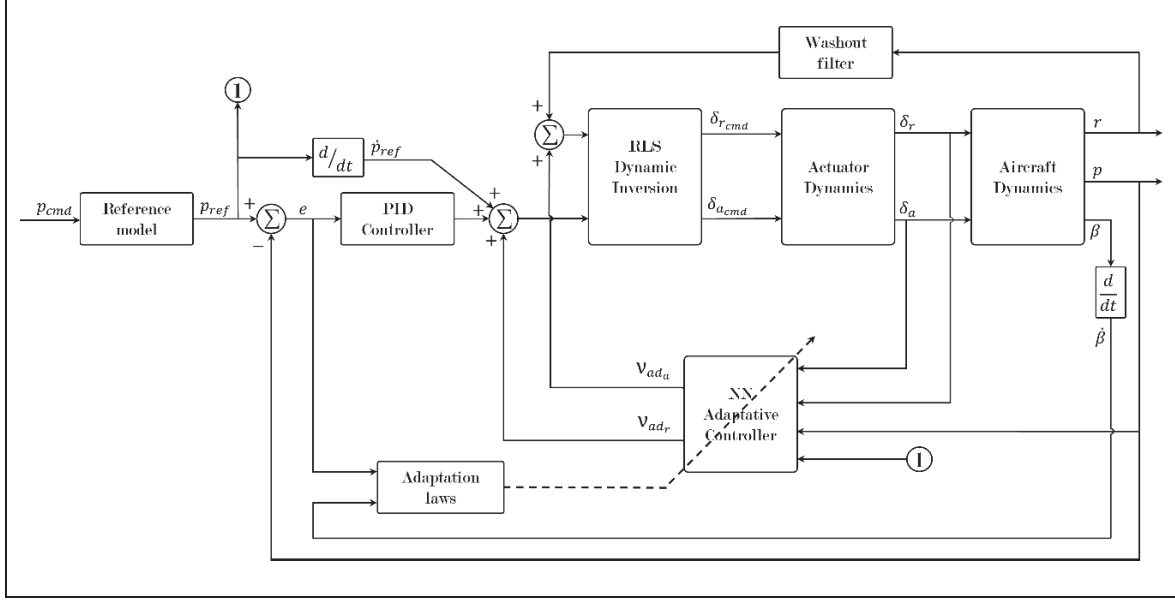


Figure A III-2 Designed roll rate controller and yaw damper including PID, RLS Dynamic Inversion and NN controller

Adaptive control law

The control principle is the same as for the pitch rate controller presented in Section III.2.III.2.1.0. The overall command law sent to the ailerons $\Delta\delta_{a_k}$ and the rudders $\Delta\delta_{r_k}$ are given in the following equations (their mathematical development can be found in (Quintin et al., 2024)) :

$$\begin{aligned} \Delta\delta_{a_k} &= (\mathbf{C}_1 \hat{\mathbf{G}}_{1k})^{-1} \left[\dot{p}_{ref} + \left(K_P + \frac{K_I}{s} + K_D s \right) e - \mathbf{C}_1 \hat{\mathbf{F}}_k \Delta \mathbf{x}_k - \mathbf{C}_1 \hat{\mathbf{G}}_{2k} \Delta\delta_{r_{k-1}} \right. \\ &\quad \left. - \mathbf{W}_a^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}) \right] \\ \Delta\delta_{r_k} &= (\mathbf{C}_2 \hat{\mathbf{G}}_{2k})^{-1} \left[\frac{K_{WF} \cdot \tau_S}{\tau_S + 1} r - \mathbf{C}_2 \hat{\mathbf{F}}_k \Delta \mathbf{x}_k - \mathbf{C}_2 \hat{\mathbf{G}}_{1k} \Delta\delta_{a_{k-1}} - \mathbf{W}_r^T \sigma(\mathbf{V}^T \bar{\mathbf{x}}) \right] \end{aligned} \quad (\text{A III-3})$$

where e is the roll rate error, $\mathbf{C}_1 = [0 \ 1 \ 0 \ 0 \ 0]$ is the output matrix for the roll rate, while $\mathbf{C}_2 = [0 \ 0 \ 1 \ 0 \ 0]$ is the output matrix for the yaw rate; $\dot{p}_{ref}$ is the derivative of the roll reference signal, $\Delta \mathbf{x}_k$ is the lateral states increment vector at sample time k , σ is the tangent-sigmoid

function of the neural network. The approximate dynamic inversion consists of estimating the local state matrix $\hat{\mathbf{F}}_k$ and the local control matrices $\hat{\mathbf{G}}_{1k}$ and $\hat{\mathbf{G}}_{2k}$ using the RLS algorithm. The neural network is used to compensate for estimation errors. The output weight matrix $\mathbf{W}_a$ of the adaptive command on the ailerons, and the output weight matrix $\mathbf{W}_r$ of the adaptive command on the rudders are dynamically updated using custom adaptive laws based on backpropagation (Quintin et al., 2024). $\mathbf{W}_a$ is updated using the tracking error on the roll rate, $\mathbf{W}_r$ is updated using the sideslip angle derivative, while the input weight matrix $\mathbf{V}$ is updated using both roll rate error and sideslip derivative.

Reference model

The aircraft lateral motion may be divided into three different motions: a fast dynamic mode, referred to as the roll mode; a slow dynamic mode, known as the spiral mode; and an oscillatory mode, termed the Dutch-roll mode. For low-level control, specifically for roll rate control, only the roll and Dutch-roll dynamics are taken into account. Therefore, a reference model is set as follows for roll rate command:

$$\frac{p_{ref}}{p_{cmd}} = \frac{\omega_{dr}^2}{s^2 + 2 \xi_{dr} \omega_{dr} s + \omega_{dr}^2} \frac{1}{\tau s + 1} \quad (\text{A III-4})$$

The aircraft roll mode may be approximated using a first order transfer function $\frac{1}{\tau s + 1}$, where τ is the roll time constant. The Dutch-roll mode may be approximated using a second-order transfer function with a specific damping ratio ξ_{dr} and a specific frequency ω_{dr} . The requirements on their specific values are given in Table A III-1.

III.3 Simulation and validation

III.3.1 Cruise trimmed conditions

The longitudinal and lateral controllers were designed for the cruise phase. To this aim, the Cessna Citation X was trimmed for 64 cruise flight conditions, covering its entire flight envelope. The proposed control algorithms were designed to adaptively control the aircraft at trimmed conditions, without the need to tune their parameters. The aircraft was trimmed at given calibrated airspeed CAS and altitude h . The overall flight conditions are shown on Figure A III-3.

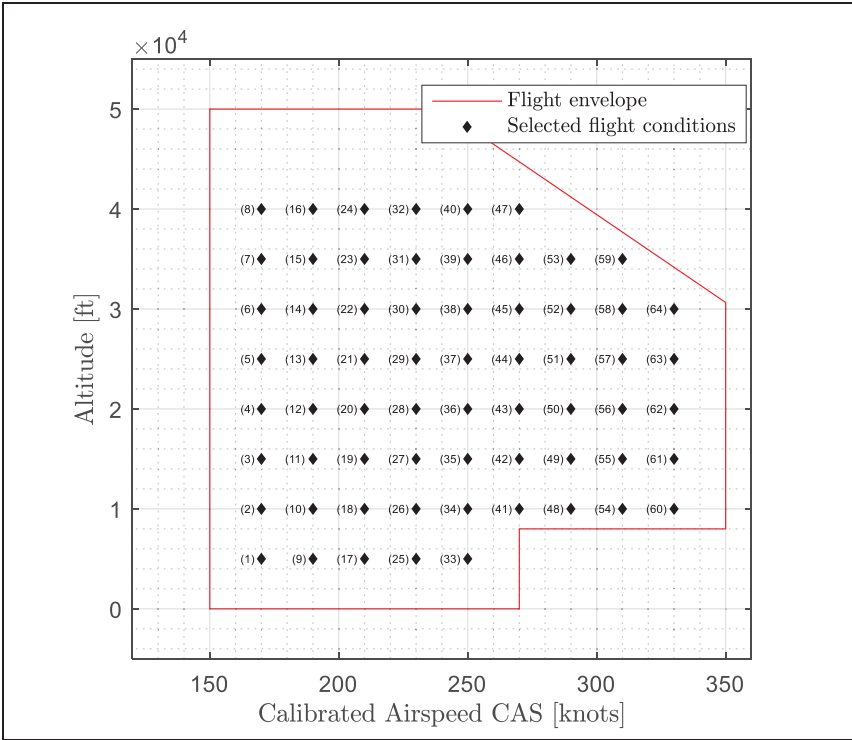


Figure A III-3 Selected 64 cruise flight conditions

The altitude was selected from 5 000 ft to 40,000 ft with an increment of 5 000 ft, and the CAS was selected from 170 kts to 330 kts with an increment of 20 kts. A fixed center-of-gravity (CoG) of 25%, and a weight of 33,000 lb were set for all flight conditions.

III.3.2 Longitudinal controller simulation results

Simulation

The pitch rate controller was tested for 64 cruise conditions. The desired performances of the reference pitch rate signal are given in Table A III-2. These performances were set based on the flying qualities requirements specified in Table A III-1. The simulations were performed for 3 seconds, and the performance characteristics are summarized in Table A III-2.

Table A III-2 Reference signal and pitch rate controller performances

	Damping	Frequency, rad/s	Rising time at 90%, sec	Settling time at $\pm 5\%$ q_{ss} , sec	Maximum pitch rate	Overshoot, %
Variable	ξ_{sp}	ω_{sp}	T_r	T_s	q_{max}	—
Reference model signal q_{ref}	0.69	4	0.21	1.22	1.21	21
Pitch rate signal q	0.52 – 0.81	5.71 – 6.97	0.25 – 0.29	1 – 1.85	1.06 – 1.28	6 – 28

As shown in Figure A III-4, the controller was able to successfully track the pitch rate reference signal after the settling time with a relative error lesser than 0.1% during steady state. The high error in the transient dynamics is due to the adaptation time required by the controller. However, that relative error is less than 0.5% in the transient behavior. In addition, it can be seen that the elevator deflections did not reach saturation, and the transient behavior for all flight conditions is close to the reference signal. The identified performances in Table A III-2 are close to those of the reference signal. Small differences are noticeable, such as a small delay in comparison to the reference signal. It can be noticed that the overshoot is different for each flight condition. Nonetheless, the overall behavior of the pitch rate remains consistent with the reference signal.

A transfer function identification was performed to determine the resulting damping and frequency for each flight condition, thereby facilitating the evaluation of the Flying Qualities. For this purpose, the Matlab function *tfest* for system identification was applied using the nonlinear least square algorithm. A delay of 0.07 sec was allowed, which complies with the Level 1 flying qualities (see Table A III-1) while enabling achieve a curve fitting success rate higher than 98%.

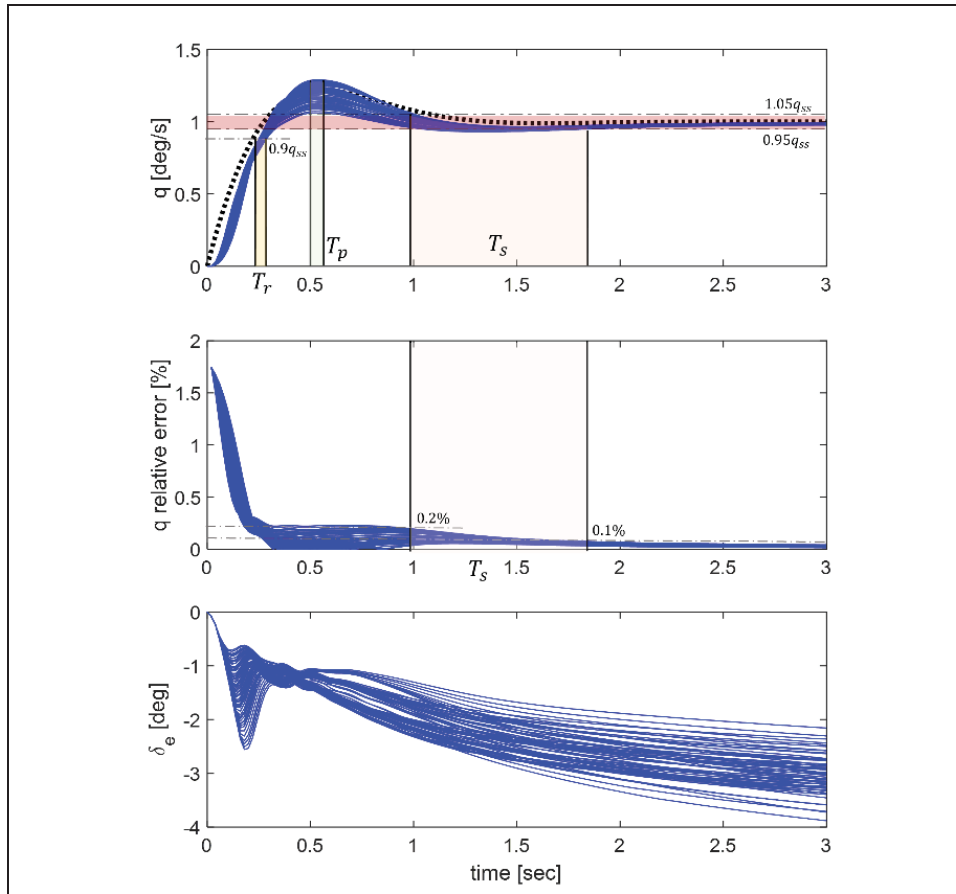


Figure A III-4 Pitch rate, relative error and elevator deflection responses for all cruise conditions.

Flying quality validation

From curve fitting, the pitch rate controller overall performances were extracted, namely, the damping ξ_{sp} , the frequency ω_{sp} and the short period time constant T_{θ_2} . The short period requirements are delimited with the damping and frequency boundaries, which were taken from MIL-STD-1797A for Level 1 and Level 2 flying qualities (FQ) in Figure A III-5.

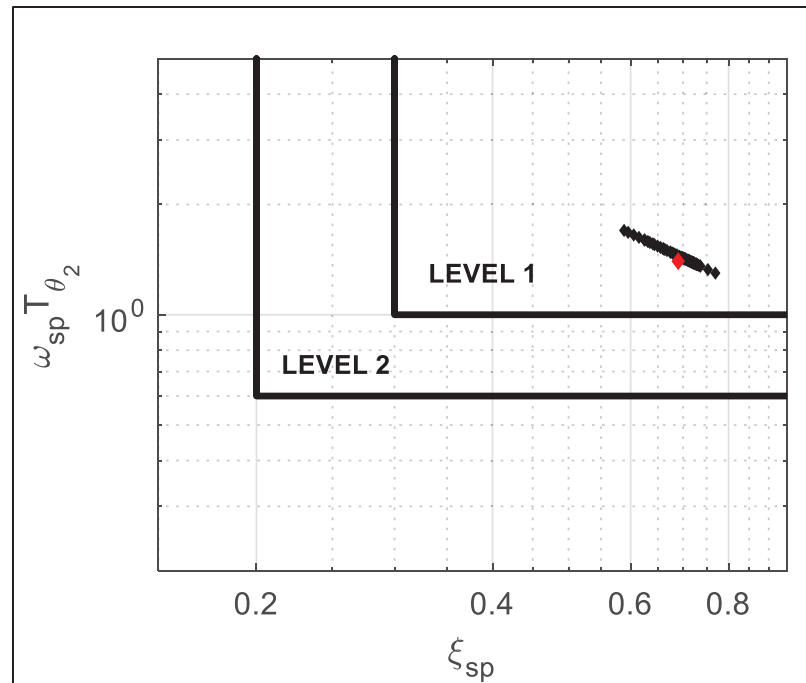


Figure A III-5 Short Period Flying Quality assessment for pitch rate controller (black dot), reference model performance (red dot), and FQ limits (black line)

Additionally, the Control Anticipation Parameter (CAP) was also evaluated. In fact, many studies have demonstrated that traditional requirements on damping and frequency alone are not sufficient for the evaluation of the short-period performance, because they do not account for the effects of the numerator component. Consequently, the impact of the zero, given by $-1/T_{\theta_2}$, was introduced in the calculation of the CAP, as proposed by Bihrlé (Bihrlé, 1966). Practically, the CAP effectively reflects the pilot perceptions in response to normal and pitch acceleration. The approximation of the CAP parameter can be expressed as follows:

$$CAP \approx \frac{\omega_{sp}^2}{\frac{V_0}{g} \frac{1}{T_{\theta_2}}} \quad (\text{A III-5})$$

where V_0 is the aircraft true airspeed, and g is the gravitational acceleration. The boundaries for the CAP corresponding to Level 1 and Level 2 Flying Qualities are delimited in Figure A III-6 (Department of Defense, 1997). As can be seen in this figure, all data points fall within the Level 1 region. Security margins are clearly set as the points are located far from the Level 1 bounds.

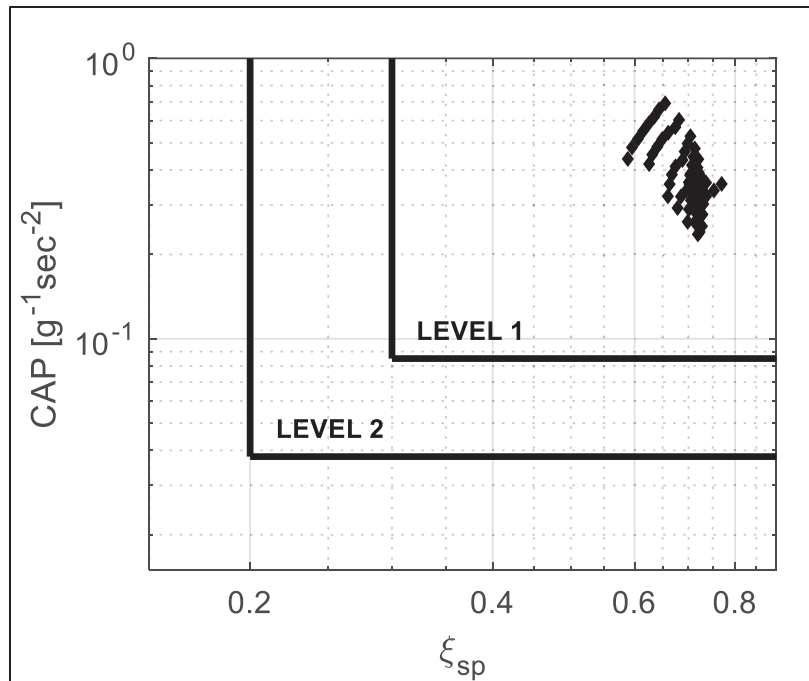


Figure A III-6 CAP assessment for pitch rate controller

III.3.3 Lateral controller simulation

Simulation

The roll rate controller was tested for 64 cruise conditions. The desired performance of the reference roll rate signal was set in Table A III-3. The desired performances were set following the Level 1 flying qualities requirements specified in Table A III-1. The simulations were performed for 5 seconds as seen in Figure A III-7a, and the performance characteristics were summed in Table A III-3. The roll rate control parameters were then changed. This ensured that the controller was robust enough in the presence of external disturbances such as wind and turbulence.

Table A III-3 Reference signal, ideal roll rate controller and reconfigured controller performances on unit signal tracking

	Damping	Frequency, rad/s	Roll time constant, sec	Rising time at 90%, sec	Settling time at $\pm 5\% p_{ss}$, sec	Maximum roll rate	Overshoot, %
	ξ_{dr}	ω_{dr}	τ	T_r	T_s	p_{max}	—
Reference model signal p_{ref}	0.5	3	0.1	0.82	1.87	1.15	15.5
Roll rate signal p (a)	0.44 – 0.51	2.9 – 3.18	0.05 – 0.14	0.82 – 0.88	1.9 – 2.14	1.18	15 – 18.2
Reconfigured controller (b)	0.23 – 0.58	3.54 – 4.06	0.21 – 0.64	0.95 – 1.48	1.75 – 3.2	1.12	0 – 12

As similar to the pitch rate controller, a transfer function identification was performed in order to identify the resulting damping and frequency for each flight condition and then conclude on the lateral Flying Qualities. A delay of 0.04 sec was allowed, which still respects the Level 1 flying qualities (see Table A III-1) and at the same time ensures a curve fitting higher than 97% success rate. As seen in Table A III-3, the performances of the controller in Figure A III-7a are close to these of the reference model. The roll rate relative error is less than 4.5% after settling time.

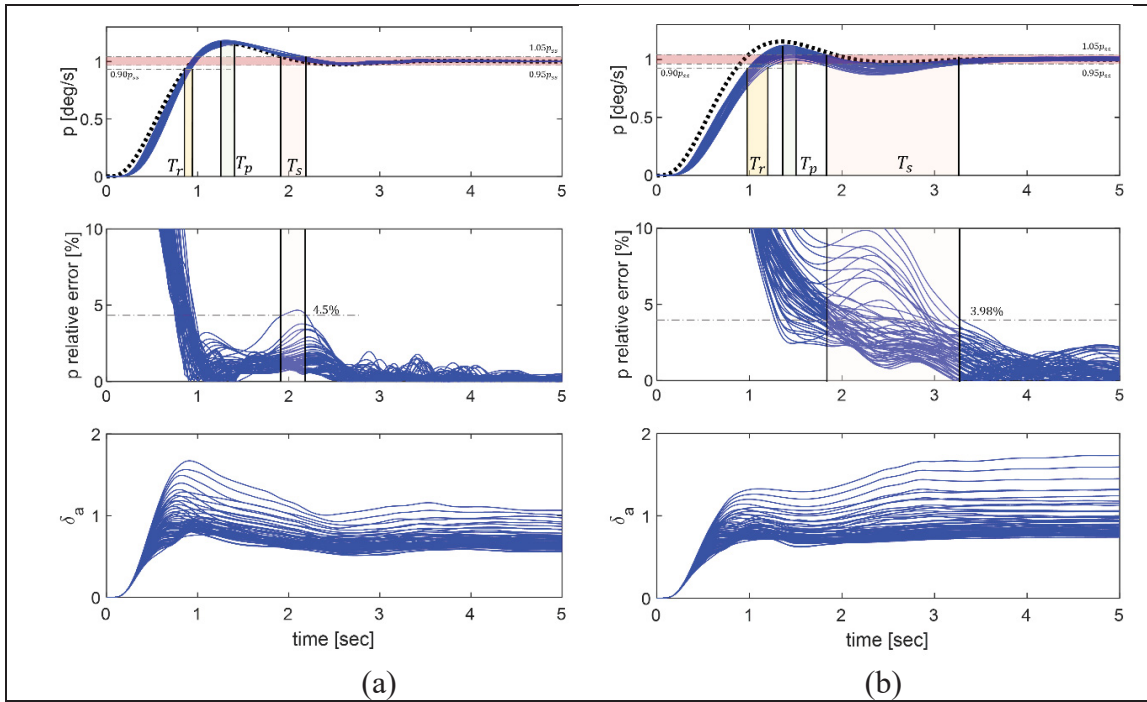


Figure A III-7 Roll rate, relative error and aileron deflection responses for all cruise conditions, for ideal controller (a), for reconfigured controller (b)

Figure A III-8 shows the controller response on the yaw rate and the sideslip angle derivative. As we can see, the yaw rate is linear, therefore damped, and the sideslip derivative is also tending to 0. When the controller parameter is changed, the damping is less effective, however, the controller is still able to reduce the effect of the sideslip derivative and keep a damped yaw rate.

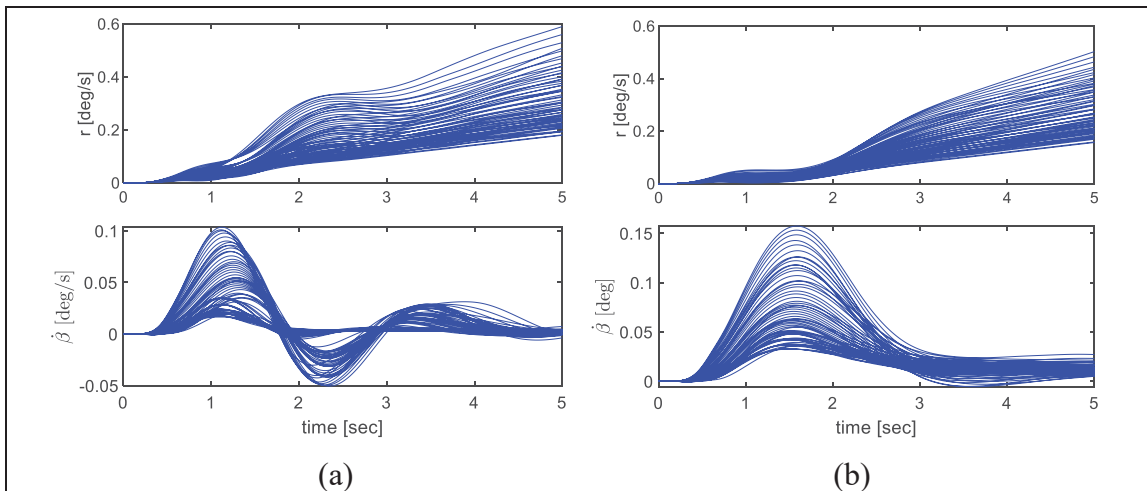


Figure A III-8 Yaw rate and sideslip derivative responses for all cruise conditions, for ideal controller (a), for reconfigured controller (b)

Flying quality validation

As for the longitudinal controller, the overall roll rate controller performances were assessed using curve fitting. The extracted performances were namely the Dutch-roll damping ξ_{dr} and frequency ω_{dr} and the roll time constant τ , and thus, for all flight conditions. The roll and Dutch-roll requirements are delimited with the damping and frequency boundaries, which were taken from MIL-STD-1797A (Department of Defense, 1997) for Level 1 and Level 2 flying qualities (FQ) in Figure A III-9.

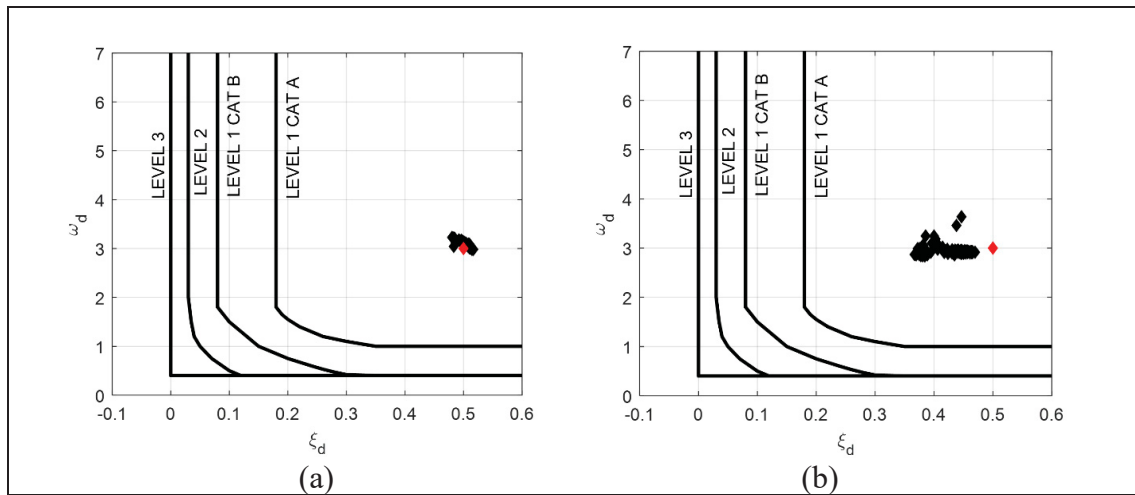


Figure A III-9 Dutch roll Flying Quality Assessment for ideal controller (a), for reconfigured controller (b)

As seen in Figure A III-9a, the controller is perfectly able to catch the reference signal performances in terms of damping and frequency. Even though the neural network parameters were reconfigured in Figure A III-9b, in order to include the uncertainties dynamics, the resulting Dutch-roll damping and frequency still remained within the Level 1. The roll time constant τ is also less than 1.4 sec for all flight conditions, which respects the Level 1 requirement in Table A III-1.

III.4 Discussions

In this paper, adaptive controllers for longitudinal pitch rate control and lateral roll rate control, as well as a yaw damper were validated using flying qualities. As a result, the controllers were able to meet the Level 1 Flying Qualities given by MIL-STD 1997A.

The assessment of flying qualities in order to validate the designed adaptive controllers is reviewed. It is critical that stability, convergence, and robustness criteria are met by all controller components to qualify for Flying Qualities assessment. In cases where these criteria are not met, as demonstrated with the lateral control in this study, reconfiguration is required to ensure compliance with Level 1 flying qualities.

Hypothetical margins may be established when evaluating flight performance. The designer must ensure that the performance of the controllers remains within the Level 1 requirements for both longitudinal and lateral control. These margins would then be defined as the gap between the actual performance of the controllers and their defined limits.

In addition, the designer must verify that the transient response of the controllers closely matches that of the reference signal. Only then should a transfer function identification be performed in order to describe the actual performance of the controllers. The primary goal of the adaptive control signal, which is to track a quadrature signal rather than a reference model signal, may require further tuning.

III.5 Conclusion

Flying Qualities were assessed for the validation of the proposed longitudinal and lateral controllers applied on the Cessna Citation X flight dynamics. The adaptive controllers include PID controllers, Approximate Dynamic Inversion controllers and Neural Network controllers. The flight controllers were tested for 64 flight conditions in the cruise phase, which covered the overall flight envelope of the Cessna Citation X. Simulation were run with low tracking error, then transfer function identification was fulfilled for each simulation in order to extract the controller performance.

Results have shown that the designed control algorithms are able to track the given reference signals on the pitch and the roll rates, while damping the yaw rate. The tracking errors are sufficiently small for the pitch rate tracking, and less than 5% for the roll rate tracking. The flying qualities have shown that the proposed controllers are stable and remain within Level 1 requirements for short period and Dutch-roll flying qualities. The roll time constant is also less than 1.4 sec for both ideal and retuned roll rate controllers.

LIST OF BIBLIOGRAPHICAL REFERENCES

- Agogino, A., Brat, G., He, Y., Hulse, D., Lipkis, R., Pressburger, T., ... others. (2024). *Recommendations on evidence and process for certification of learning-enabled components in aerospace systems* (Rapport No. NASA NTRS 20240006865). Moffett Field, CA, USA : National Aeronautics and Space Administration, Ames Research Center. Repéré à <https://ntrs.nasa.gov/citations/20240006865>
- Al-Gabalawy, M. A. (2019). Machine Learning For Aircraft Control. *Control Systems*, 11(08), 27.
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2021). Aircraft Engine Performance Model Identification using Artificial Neural Networks. Dans *AIAA Propulsion and Energy 2021 Forum* (p. 3247).
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2023a). Neural network adaptive controller with approximate dynamic inversion for pitch control of the Cessna Citation X. Dans *AIAA AVIATION 2023 Forum* (p. 3798). San Diego, CA, USA.
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2023b). Performance Model Identification of the General Electric CF34-8C5B1 Turbofan Using Neural Networks. *Journal of Aerospace Information Systems*, 1-18.
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2024). Model Predictive Controller with Adaptive Neural Networks and Online State Estimation for Pitch Rate Control of the Cessna Citation X. Dans *AIAA SCITECH 2024 Forum* (p. 0118). Orlando, FL, USA.
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2025). Flying Qualities Assessment for Nonlinear Adaptive Control Validation on the Cessna Citation X Longitudinal and Lateral Dynamics. Dans *AIAA SCITECH 2025 Forum* (p. 1826). Orlando, FL, USA.
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2026). Nonlinear adaptive longitudinal controller and flight qualities validation for a business aircraft. *The Aeronautical Journal*, 1-27. <https://doi.org/10.1017/aer.2026.10132>
- Andrianantara, R. P., Mancera Coyotl, F. D., Ghazi, G., & Botez, R. M. (2024). Robust Pitch Control of the Cessna Citation X Using C-Star Architecture and Adaptive Neural Networks. Dans *AIAA AVIATION FORUM AND ASCEND 2024* (p. 4257). Las Vegas, NV, USA.
- Andrianantara, Rojo Princy, Ghazi, G., & Botez, R. M. (2021). General Electric CF34-8C5B1 Turbofan Parameters Estimation using Artificial Neural Networks and Flight Tests.
- Bagnell, J. A., & Schneider, J. G. (2001). Autonomous helicopter control using reinforcement learning policy search methods. Dans *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)* (Vol. 2, pp. 1615-1620). IEEE.
- Bhattacharyya, S., Cofer, D., Musliner, D., Mueller, J., & Engstrom, E. (2015). Certification considerations for adaptive systems. Dans *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 270-279). IEEE.
- Bihrlé, W. (1966). A Handling Qualities Theory for Precise Flight-Path Control. *AFFDL-TR-65-198*.

- Boely, N., & 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>
- 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.
- Bonnans, J.-F., Gilbert, J. C., Lemaréchal, C., & Sagastizábal, C. A. (2006). *Numerical optimization: theoretical and practical aspects*. (S.I.) : Springer Science & Business Media.
- Bosworth, J., & Williams-Hayes, P. (2007). Flight test results from the nf-15b intelligent flight control system (ifcs) project with adaptation to a simulated stabilator failure. Dans *AIAA Infotech@ Aerospace 2007 Conference and Exhibit* (p. 2818).
- Boughari, Y., Ghazi, G., Botez, R. M., & Theel, F. (2017a). New Methodology for Optimal Flight Control Using Differential Evolution Algorithms Applied on the Cessna Citation X Business Aircraft—Part 1: Design and Optimization. *INCAS Bulletin*, 9(1), 31-44. <https://doi.org/10.13111/2066-8201.2017.9.2.3>
- Boughari, Y., Ghazi, G., Botez, R. M., & Theel, F. (2017b). New Methodology for Optimal Flight Control Using Differential Evolution Algorithms Applied on the Cessna Citation X Business Aircraft—Part 2: Validation on Aircraft Research Flight Level D Simulator. *INCAS Bulletin*, 9(1), 45-56. <https://doi.org/10.13111/2066-8201.2017.9.2.4>
- Boughari, Yamina, Botez, R. M., Ghazi, G., & Theel, F. (2017). Flight Control Clearance of the Cessna Citation X Using Evolutionary Algorithms. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 231(3), 510-532.
- Boughari, Yamina, Ghazi, G., & Botez, R. M. (2017). Optimal Control New Methodologies Validation On the Research Aircraft Flight Simulator of the Cessna Citation X Business Aircraft. *Internationa Journal of Contemporary Energy*, 3(1).
- Camacho, E. F., & Bordons, C. (2007). *Model Predictive control*. London : Springer London. <https://doi.org/10.1007/978-0-85729-398-5>
- Campbell, S., Nguyen, N., Kaneshige, J., & Krishnakumar, K. (2009). Parameter estimation for a hybrid adaptive flight controller. Dans *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference* (p. 1803).
- Cima, G., Cluzeau, J.-M., Henriquel, X., Semde, H., Soudain, G., Vaubourg, S., ... Whittington, I. (2021). Concepts of Design Assurance for Neural Networks (CoDANN) II. *Public Report Extract Version 1.0*, 113.
- Cook, M. V. (2012). *Flight dynamics principles: a linear systems approach to aircraft stability and control*. (S.I.) : Butterworth-Heinemann.
- Dai, P., Feng, D., Zhao, J., Cui, J., & Wang, C. (2022). Asymmetric integral barrier Lyapunov function-based dynamic surface control of a state-constrained morphing waverider with anti-saturation compensator. *Aerospace Science and Technology*, 131, 107975. <https://doi.org/10.1016/j.ast.2022.107975>
- Department of Defense. (1997). *Flying qualities of piloted aircraft* (Rapport No. MIL-STD-1797A). Washington, DC, USA.
- Dierks, T., & Jagannathan, S. (2009). Output feedback control of a quadrotor UAV using neural networks. *IEEE transactions on neural networks*, 21(1), 50-66.

- Doff-Sotta, M., Cannon, M., & Bacic, M. (2025). Data-Driven Robust Model Predictive Control of Tiltwing Vertical Takeoff and Landing Aircraft. *Journal of Guidance, Control, and Dynamics*, 48(1), 203-211.
- Dorf, R. C., & Bishop, R. H. (2008). *Modern control systems*. (S.l.) : Pearson Prentice Hall.
- Ducard, G., & Geering, H. P. (2008). Stability analysis of a dynamic inversion based pitch rate controller for an unmanned aircraft. Dans *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 360-366). IEEE.
- Efremov, A. V., Mbikayi, Z., & Efremov, E. V. (2021). Comparative study of different algorithms for a flight control system design and the potentiality of their integration with a sidestick. *Aerospace*, 8(10), 290.
- Emami, S. A., Castaldi, P., & Banazadeh, A. (2022). Neural network-based flight control systems: Present and future. *Annual Reviews in Control*, 53, 97-137.
- Enns, D., Bugajski, D., Hendrick, R., & Stein, G. (1994). Dynamic inversion: an evolving methodology for flight control design. *International Journal of control*, 59(1), 71-91.
- Eren, U., Prach, A., Koçer, B. B., Raković, S. V., Kayacan, E., & Açıkmeşe, B. (2017). Model Predictive Control in Aerospace Systems: Current State and Opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7), 1541-1566. <https://doi.org/10.2514/1.G002507>
- European Union Aviation Safety Agency. (2024). *Artificial Intelligence Concept Paper — Issue 2: Guidance for Level 1 & 2 Machine-Learning Applications*. Cologne, Germany : EASA. Repéré à <https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-concept-paper-issue-2>
- Federal Aviation Administration. (2009). *Human Factors Guide for Aviation Maintenance*. Washington, DC, USA : Federal Aviation Administration. Repéré à file:///mnt/data/HF_Guide.pdf
- Federal Aviation Administration. (2024). *Roadmap for Artificial Intelligence Safety Assurance*. Washington, DC, USA : Federal Aviation Administration. Repéré à https://www.faa.gov/aircraft/air_cert/step/roadmap_for_AI_safety_assurance
- Feith, R. (2020). Safe Reinforcement Learning in Flight Control: Introduction to Safe Incremental Dual Heuristic Programming.
- Ferrier, Y., Nguyen, N. T., Ting, E., Chaparro, D., Wang, X., de Visser, C. C., & Chu, Q. P. (2018). Active gust load alleviation of high-aspect ratio flexible wing aircraft. Dans *2018 AIAA Guidance, Navigation, and Control Conference* (p. 0620). Kissimmee, FL, USA.
- Franklin, G. F., Powell, J. D., Emami-Naeini, A., & Powell, J. D. (2002). *Feedback control of dynamic systems* (Vol. 4). (S.l.) : Prentice hall Upper Saddle River, NJ.
- Frost, W., Camp, D. W., & Wang, S. (1978). *Wind shear modeling for aircraft hazard definition*.
- Gao, C., & Liu, H. H. (2017). Dubins path-based dynamic soaring trajectory planning and tracking control in a gradient wind field. *Optimal Control Applications and Methods*, 38(2), 147-166.
- Ge, S. S., Hang, C. C., Lee, T. H., & Zhang, T. (2013). *Stable adaptive neural network control* (Vol. 13). London, UK : Springer Science & Business Media.

- Geronel, R., Botez, R., & Bueno, D. (2023). Dynamic responses due to the Dryden gust of an autonomous quadrotor UAV carrying a payload. *The Aeronautical Journal*, 127(1307), 116-138.
- Ghazi, G., & Botez, R. (2014). New robust control analysis methodology for Lynx helicopter and Cessna Citation X aircraft using Guardian Maps, Genetic Algorithms and LQR theories combinations. Dans *AHS 70th Annual Forum, Montreal, Quebec, Canada*.
- Ghazi, G., & Botez, R. (2015a). Development of a High-Fidelity Simulation Model for a Research Environment. Dans *SAE 2015 AeroTech Congress & Exhibition*. Seattle, WA, USA. <https://doi.org/10.4271/2015-01-2569>
- Ghazi, G., & Botez, R. M. (2015b). Lateral controller design for the Cessna citation X with handling qualities and robustness requirements. Dans *62nd CASI Aeronautics Conference and AGM*,. Montreal, Quebec, Canada.
- Ghazi, G., Tudor, M., & Botez, R. (2015). Identification of a Cessna Citation X aero-propulsive model in climb regime from flight tests. Dans *International Conference on Air Transport INAIR*.
- Goodwin, G. C., Graebe, S. F., Salgado, M. E., & others. (2001). *Control system design* (Vol. 240). (S.l.) : Prentice Hall New Jersey.
- Grigorie, T. L., Botez, R. M., Popov, A. V., Mamou, M., & Mébarki, Y. (2012a). A hybrid fuzzy logic proportional-integral-derivative and conventional on-off controller for morphing wing actuation using shape memory alloy Part 1: Morphing system mechanisms and controller architecture design. *The Aeronautical Journal*, 116(1179), 433-449. <https://doi.org/10.1017/S0001924000006977>
- Grigorie, T. L., Botez, R. M., Popov, A. V., Mamou, M., & Mébarki, Y. (2012b). A hybrid fuzzy logic proportional-integral-derivative and conventional on-off controller for morphing wing actuation using shape memory alloy Part 2: Controller implementation and validation. *The Aeronautical Journal*, 116(1179), 451-465. <https://doi.org/10.1017/S0001924000006989>
- Grigorie, T. L., Popov, A. V., Botez, R. M., Mamou, M., & Mébarki, Y. (2012a). On-off and proportional-integral controller for a morphing wing. Part 1: Actuation mechanism and control design. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(2), 131-145. <https://doi.org/10.1177/0954410011408226>
- Grigorie, T. L., Popov, A. V., Botez, R. M., Mamou, M., & Mébarki, Y. (2012b). On-off and proportional-integral controller for a morphing wing. Part 2: Control validation – numerical simulations and experimental tests. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(2), 146-162. <https://doi.org/10.1177/0954410011408271>
- Grondman, F., Looye, G., Kuchar, R. O., Chu, Q. P., & Van Kampen, E.-J. (2018). Design and flight testing of incremental nonlinear dynamic inversion-based control laws for a passenger aircraft. Dans *2018 AIAA Guidance, Navigation, and Control Conference* (p. 0385).
- Hagan, M., Demuth, H., Beale, M., & De Jesus, O. (2016). *Neural Network Design*, 2nd. Ed., Lexington, KY.

- Hagan, M. T., & Demuth, H. B. (1999). Neural networks for control. Dans *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)* (Vol. 3, pp. 1642-1656 vol.3). <https://doi.org/10.1109/ACC.1999.786109>
- Hakim, M., & Choukri, S. (2024). Froude Similarity and Flying Qualities Assessment in the Design of a Low-Speed BWB UAV. *International Journal of Aeronautical and Space Sciences*, 25(1), 46-60.
- Harris, J., Elliott, C. M., & Tallant, G. S. (2022). L1 adaptive nonlinear dynamic inversion control for the innovative control effectors aircraft. Dans *AIAA SCITECH 2022 Forum* (p. 0791).
- Hashemi, S. M., & Botez, R. M. (2022). Support Vector Regression Application for the Flight Dynamics New Modelling of the UAS-S4. Dans *AIAA SCITECH 2022 Forum* (p. 2576).
- Hashemi, S. M., Botez, R. M., & Grigorie, L. T. (2020). Adaptive Fuzzy Control of Chaotic Flapping relied upon Lyapunov-based Tuning Laws. Dans *AIAA Aviation 2020 Forum. VIRTUAL EVENT: American Institute of Aeronautics and Astronautics*. <https://doi.org/10.2514/6.2020-3193>
- Haykin, S. (2009). *Neural networks and learning machines*, 3/E. (S.l.) : Pearson Education India.
- Haykin, S. (2013). *Adaptive Filter Theory: International Edition*, 5/E. (S.l.) : Pearson.
- Heyer, S., Kroezen, D., & Van Kampen, E.-J. (2020). Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft. Dans *AIAA Scitech 2020 Forum* (p. 1844).
- Hodgkinson, J. (1999). *Aircraft Handling Qualities*. Reston, VA : American Institute of Aeronautics and Astronautics.
- Horn, J. F. (2019). Non-linear dynamic inversion control design for rotorcraft. *Aerospace*, 6(3), 38.
- Hosseini, S., Ghazi, G., & Botez, R. M. (2023a). Application of Type One Adaptive Fuzzy Sliding Mode Control System for the Longitudinal Motion of the Cessna Citation X. Dans *AIAA AVIATION 2023 Forum* (p. 3801).
- Hosseini, S., Ghazi, G., & Botez, R. M. (2023b). Design of a Type Two Fuzzy-based system to Control the Pitch Rate of the Cessna Citation X. Dans *AIAA AVIATION 2023 Forum* (p. 3802).
- Hosseini, S., Inga, C., Ghazi, G., & Botez, R. M. (2023). Model-referenced Adaptive Flight Controller based on Recurrent Neural Network for the Longitudinal Motion of Cessna Citation X. Dans *AIAA AVIATION 2023 Forum* (p. 3797).
- Hosseini, S. M., Bematol, I., Ghazi, G., & Botez, R. M. (2024a). A novel particle swarm optimization based fuzzy super-twisting sliding mode control system for the lateral motion of Cessna Citation X. Dans *AIAA AVIATION FORUM AND ASCEND 2024* (p. 4256).
- Hosseini, S. M., Bematol, I., Ghazi, G., & Botez, R. M. (2024b). Enhanced Fuzzy-Based Super-Twisting Sliding-Mode Control System for the Cessna Citation X Lateral Motion. *Aerospace*, 11(7). <https://doi.org/10.3390/aerospace11070549>
- Hosseini, S. M., Ghazi, G., & Botez, R. M. (2024a). New type-2-fuzzy-logic-based control system for the Cessna Citation X. *Journal of Aerospace Information Systems*, 21(10), 846-864.

- Hosseini, S. M., Ghazi, G., & Botez, R. M. (2024b). Novel controller methodology for the Cessna Citation X under turbulence during cruise. *Journal of Aerospace Information Systems*, 21(11), 892-905.
- Hosseini, S. M., Ghazi, G., & Botez, R. M. (2026). New fuzzy-RNN autopilot system for the Cessna Citation X aircraft with a fuzzy transition algorithm. *The Aeronautical Journal*, 1-31. <https://doi.org/10.1017/aer.2025.10114>
- Hsu, M., & Liu, H. H. (2021). Design of a nonlinear hierarchical adaptive controller for a novel tilt-rotor VTOL AquaUAV. Dans *AIAA Scitech 2021 Forum* (p. 1283).
- Hussein, M., Zhang, Y., & Liu, Z. (2024). Adaptive non-singular fast terminal sliding mode control for car-like vehicles with faded neighborhood information and actuator faults. *Journal of Intelligent & Robotic Systems*, 110(2), 1-14.
- ICAO. (2024). *Safety Report*. Repéré à <https://www.icao.int/safety-reports>
- Johnson, E. N., & Calise, A. J. (2003). Limited authority adaptive flight control for reusable launch vehicles. *Journal of Guidance, Control, and Dynamics*, 26(6), 906-913.
- Kammegne, M. J. T., Botez, R. M., Grigorie, L. T., Mamou, M., & Mébarki, Y. (2017). Proportional fuzzy feed-forward architecture control validation by wind tunnel tests of a morphing wing. *Chinese Journal of Aeronautics*, 30(2), 561-576.
- Khalil, A., & Fezans, N. (2021). Gust load alleviation for flexible aircraft using discrete-time preview control. *The Aeronautical Journal*, 125(1284), 341-364.
- Khalil, I. S., Doyle, J., & Glover, K. (1996). *Robust and optimal control*. (S.l.) : prentice hall, new jersey.
- Khan, S., Grigorie, T. L., Botez, R. M., Mamou, M., & Mébarki, Y. (2020). Novel morphing wing actuator control-based Particle Swarm Optimisation. *The Aeronautical Journal*, 124(1271), 55-75.
- Kim, B. S., & Calise, A. J. (1997). Nonlinear Flight Control Using Neural Networks. *Journal of Guidance, Control, and Dynamics*, 20(1), 26-33. <https://doi.org/10.2514/2.4029>
- Koch III, W. F. (2019). *Flight controller synthesis via deep reinforcement learning* (PhD Thesis). Boston University.
- Koch, W., Mancuso, R., & Bestavros, A. (2019). Neuroflight: Next generation flight control firmware. *arXiv preprint arXiv:1901.06553*.
- Koch, W., Mancuso, R., West, R., & Bestavros, A. (2019). Reinforcement learning for UAV attitude control. *ACM Transactions on Cyber-Physical Systems*, 3(2), 1-21.
- Konatala, R., Milz, D., Weiser, C., Looye, G., & van Kampen, E. (2024). Flight Testing Reinforcement-Learning-Based Online Adaptive Flight Control Laws on CS-25-Class Aircraft. *Journal of Guidance, Control, and Dynamics*, 47(11), 2460-2467.
- Konatala, R., Van Kampen, E.-J., & Looye, G. (2021). Reinforcement Learning based Online Adaptive Flight Control for the Cessna Citation II (PH-LAB) Aircraft. Dans *AIAA Scitech 2021 Forum* (p. 0883).
- Kopf, M., Bullinger, E., Giessler, H.-G., Adden, S., & Findeisen, R. (2018). Model predictive control for aircraft load alleviation: Opportunities and challenges. Dans *2018 Annual American Control Conference (ACC)* (pp. 2417-2424). IEEE.
- Kouba, G., Botez, R., & Boely, N. (2009). Identification of F/A-18 model from flight tests using the fuzzy logic method. Dans *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. Orlando, Florida : American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/6.2009-1483>

- Kouba, G., Botez, R. M., & Boely, N. (2010). Fuzzy logic method use in F/A-18 aircraft model identification. *Journal of aircraft*, 47(1), 10-17.
- Kouvaritakis, B., & Cannon, M. (2016). *Model Predictive Control*. Cham : Springer International Publishing. <https://doi.org/10.1007/978-3-319-24853-0>
- Lane, S. H., & Stengel, R. F. (1988). Flight control design using non-linear inverse dynamics. *Automatica*, 24(4), 471-483.
- Lavretsky, E., & Wise, K. A. (2024). Robust adaptive control. Dans *Robust and adaptive control: With aerospace applications* (pp. 469-506). (S.l.) : Springer.
- Lee, J. H., & Van Kampen, E.-J. (2021). Online reinforcement learning for fixed-wing aircraft longitudinal control. Dans *AIAA Scitech 2021 Forum* (p. 0392).
- Lewis, F. L., & Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, 9(3), 32-50. <https://doi.org/10.1109/MCAS.2009.933854>
- Lewis, F. L., Yeşildirek, A., & Liu, K. (1995). Neural net robot controller: Structure and stability proofs. *Journal of Intelligent and Robotic Systems*, 12, 277-299.
- Li, X., Li, L., Gao, J., He, X., Chen, J., Deng, L., & He, J. (2015). Recurrent reinforcement learning: a hybrid approach. *arXiv preprint arXiv:1509.03044*.
- Li, Y., Liu, K., Wen, C.-Y., Liu, X., Zhang, W., & Zheng, Y. (2025). Fast fixed-time incremental backstepping fault-tolerant control for aircraft with asymmetric wing damage. *Aerospace Science and Technology*, 164, 110405.
- Liu, S., Lyu, W., Zhang, Q., Yang, C., & Whidborne, J. F. (2025). Neural-Network-Based Incremental Backstepping Sliding Mode Control for Flying-Wing Aircraft. *Journal of Guidance, Control, and Dynamics*, 48(3), 600-614.
- Lungu, M., & Lungu, R. (2017). Landing auto-pilots for aircraft motion in longitudinal plane using adaptive control laws based on neural networks and dynamic inversion. *Asian Journal of Control*, 19(1), 302-315.
- Lungu, R., Lungu, M., & Efrim, C. (2021). Adaptive control of DGMSCMG using dynamic inversion and neural networks. *Advances in Space Research*, 68(8), 3478-3494.
- Ma, W., Liu, S., & Huang, W. (2025). Robust prescribed-time observer-based sliding mode control: Theoretical design and flight control applications. *ISA Transactions*, 167, 1446-1457. <https://doi.org/10.1016/j.isatra.2025.10.014>
- Mahadi, M., Ballal, T., Moinuddin, M., & Al-Saggaf, U. M. (2022). A Recursive Least-Squares with a Time-Varying Regularization Parameter. *Applied Sciences*, 12(4), 2077.
- McLaughlin, R., & Perhinschi, M. G. (2022). Partitioned Immunity-based Approach for Detection and Identification of Sensor and Actuator Faults on a UAV. Dans *AIAA SCITECH 2022 Forum* (p. 2080).
- Mohseni, N., & Bernstein, D. S. (2022). Recursive least squares with variable-rate forgetting based on the F-test. Dans *2022 American Control Conference (ACC)* (pp. 3937-3942). IEEE.
- Moncayo, H., Perhinschi, M., Wilburn, B., Wilburn, J., & Karas, O. (2012). Extended nonlinear dynamic inversion control laws for unmanned air vehicles. Dans *AIAA Guidance, Navigation, and Control Conference* (p. 4675). Minneapolis, MN, USA.
- Morari, M., & Zafiriou, E. (1989). *Robust process control*. (S.l.) : Morari.

- Mosbah, A. B., Botez, R. M., Dao, T. M., Guezguez, M. S., & Zaag, M. (2016). A neural network controller new methodology for the ATR-42 morphing wing actuation. *INCAS Bulletin*, 8(2), 59.
- Narendra, K., & Annaswamy, A. (1987). A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic control*, 32(2), 134-145.
- Nelson, R. C. (1998). *Flight stability and automatic control* (Vol. 2). (S.l.) : WCB/McGraw Hill New York.
- Nguyen, N. (2011). Hybrid adaptive flight control with model inversion adaptation. Dans *Advances in Flight Control Systems* (Vol. 18, p. 19). (S.l.) : InTech.
- Nguyen, N., & Krishnakumar, K. (2007). Hybrid damage adaptive flight control with model inversion adaptation. *IFAC Proceedings Volumes*, 40(7), 657-662.
- Nguyen, N. T. (2018). Model-reference adaptive control. Dans *Model-Reference Adaptive Control* (pp. 83-123). (S.l.) : Springer.
- O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., & Chung, S.-J. (2022). Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66), eabm6597.
- Ogata, K., & Yang, Y. (2002). *Modern control engineering* (Vol. 4). (S.l.) : Prentice hall India.
- Park, O., Shin, H.-S., Lee, H.-I., & Antonios, T. (2021). Optimal and Adaptive Control Design Using Recursive Least Square with a New Exponential Forgetting Factor. Dans *International Conference on Robot Intelligence Technology and Applications* (pp. 116-128). Springer.
- Pedro, J., & Meyer, N. (2007). Neural network-based dynamic inversion controller for fighter aircrafts. *IFAC Proceedings Volumes*, 40(12), 946-951.
- Popov, Andrei V, Grigorie, L. T., Botez, R., Mamou, M., & Mebarki, Y. (2010). Closed-loop control validation of a morphing wing using wind tunnel tests. *Journal of Aircraft*, 47(4), 1309-1317.
- Popov, Andrei Vladimir, Labib, M., Fays, J., & Botez, R. M. (2008). Closed-loop control simulations on a morphing wing. *Journal of Aircraft*, 45(5), 1794-1803.
- Puskorius, G. V., & Feldkamp, L. A. (1991). Decoupled extended Kalman filter training of feedforward layered networks. Dans *IJCNN-91-Seattle International Joint Conference on Neural Networks* (Vol. 1, pp. 771-777). IEEE.
- Quintin, E., Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2024). Neural Network Adaptive Controller with Approximate Dynamic Inversion for the Cessna Citation X Lateral Control. Dans *AIAA Scitech 2024 Forum* (p. 0261). Orlando, FL,USA.
- Reynolds, O. R., Pachter, M., & Houppis, C. (1996). Full envelope flight control system design using quantitative feedback theory. *Journal of Guidance, control, and Dynamics*, 19(1), 23-29.
- Rysdyk, R., & Calise, A. J. (2005). Robust nonlinear adaptive flight control for consistent handling qualities. *IEEE Transactions on Control Systems Technology*, 13(6), 896-910.
- Santoso, F., Garratt, M. A., & Anavatti, S. G. (2017). State-of-the-art intelligent flight control systems in unmanned aerial vehicles. *IEEE Transactions on Automation Science and Engineering*, 15(2), 613-627.
- Saussié, D., Akhrif, O., Bérard, C., & Saydy, L. (2009). Longitudinal flight control synthesis with guardian maps. Dans *AIAA Guidance, Navigation, and Control Conference* (p. 5988).

- Saussie, D., Saydy, L., & Akhrif, O. (2003). Flight control design with robustness and handling quality requirements. Dans *CCECE 2003 - Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology* (Cat. No.03CH37436) (Vol. 3, pp. 1749-1752 vol.3). <https://doi.org/10.1109/CCECE.2003.1226248>
- Schumann, J., & Liu, Y. (Éds). (2010). *Applications of Neural Networks in High Assurance Systems* (Vol. 268). Berlin, Heidelberg : Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-10690-3>
- Simon, D. (2014). *Model Predictive Control in Flight Control Design : Stability and Reference Tracking*. (S.l.) : Linköping University Electronic Press. <https://doi.org/10.3384/lic.diva-103742>
- Slotine, J.-J. E., & Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs, N.J : Prentice Hall.
- Smith, T., Barhorst, J., & Urnes, J. M. (2010). Design and flight test of an intelligent flight control system. Dans *Applications of Neural Networks in High Assurance Systems* (pp. 57-76). (S.l.) : Springer.
- Snell, S., & Stout, P. (1996). Robust control of angle of attack using dynamic inversion combined with quantitative feedback theory. Dans *Guidance, Navigation, and Control Conference*. San Diego, CA, U.S.A. : American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/6.1996-3783>
- Soares, F., & Burken, J. (2006). A Flight Test Demonstration of On-line Neural Network Applications in Advanced Aircraft Flight Control System. Dans *2006 International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)* (pp. 136-136). IEEE.
- Steffensen, R., Steinert, A., & Smeur, E. J. (2023). Nonlinear Dynamic Inversion with Actuator Dynamics: An Incremental Control Perspective. *Journal of Guidance, Control, and Dynamics*, 46(4), 709-717. <https://doi.org/10.2514/1.G007079>
- Stevens, B. L., Lewis, F. L., & Johnson, E. N. (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. (S.l.) : John Wiley & Sons.
- Sun, B., & van Kampen, E.-J. (2020). Incremental model-based global dual heuristic programming with explicit analytical calculations applied to flight control. *Engineering Applications of Artificial Intelligence*, 89, 103425.
- Sun, B., & van Kampen, E.-J. (2021). Reinforcement-Learning-Based Adaptive Optimal Flight Control with Output Feedback and Input Constraints. *Journal of Guidance, Control, and Dynamics*, 44(9), 1685-1691. <https://doi.org/10.2514/1.G005715>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. (S.l.) : MIT press.
- Szymanski, M., Ghazi, G., & Botez, R. M. (2025). Enhanced Neural Network Model for Aircraft Flight Dynamics Prediction From Flight Test Data-Business Aircraft Application. Dans *AIAA SCITECH 2025 Forum* (p. 2228).
- Tischler, M. B. (1996). *Advances in aircraft flight control*. (S.l.) : CRC Press.
- Tondji, Y., Ghazi, G., & Botez, R. M. (2022). CRJ 700 Aerodynamic Coefficients Identification in Dynamic Stall Conditions using Neural Networks. Dans *AIAA SCITECH 2022 Forum* (p. 2577).

- Tondji, Y., Ghazi, G., & Botez, R. M. (2023). CRJ 700 Longitudinal Aerodynamic Coefficients Identification using Support Vector Machine. Dans *AIAA AVIATION 2023 Forum* (p. 4073).
- Tondji, Y., Wade, M., Ghazi, G., & Mihaela Botez, R. (2022). 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>
- van Kampen, E.-J., Chu, Q. P., & Mulder, J. A. (2006). Continuous Adaptive Critic Flight Control Aided with Approximated Plant Dynamics. Dans *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Keystone, Colorado : American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/6.2006-6429>
- Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. Dans *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)* (pp. 153-158). Ieee.
- Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB®*. London : Springer London. <https://doi.org/10.1007/978-1-84882-331-0>
- Wiegmann, D. A., & Shappell, S. (2001). *A human error analysis of commercial aviation accidents using the human factors analysis and classification system*. Oklahoma City, OK, USA : U.S. Department of Transportation. Federal Aviation Administration.
- Wilkinson, C., Lynch, J., Bharadwaj, R., & Woodham, K. (2016). Verification of adaptive systems. *Federal Aviation Administration, DOT/FAA/TC-16/4*.
- Williams, P. (2004). Selected Flight Test Results for Online Learning Neural Network-Based Flight Control System. Dans *AIAA 1st Intelligent Systems Technical Conference*. Chicago, Illinois : American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/6.2004-6283>
- Williams-Hayes, P. (2005). Flight test implementation of a second generation intelligent flight control system. Dans *Infotech@ Aerospace* (p. 6995). (S.l.) : (s.n.).
- Wunderlich, T. F., Dähne, S., Reimer, L., & Schuster, A. (2022). Global aero-structural design optimization of composite wings with active manoeuvre load alleviation. *CEAS Aeronautical Journal*, 13(3), 639-662.
- Xiaoqian, T., Feicheng, Z., Zhengbing, T., & Hongying, W. (2022). Nonlinear Extended Kalman Filter for Attitude Estimation of the Fixed-Wing UAV. *International Journal of Optics*, 2022(1), 7883851.
- Xu, J., Du, T., Foshey, M., Li, B., Zhu, B., Schulz, A., & Matusik, W. (2019). Learning to fly: computational controller design for hybrid UAVs with reinforcement learning. *ACM Transactions on Graphics*, 38(4), 1-12. <https://doi.org/10.1145/3306346.3322940>
- Yu, Z., Zhang, Y., Jiang, B., & Su, C.-Y. (2023). Distributed FTCC of Multi-UAVs Under Actuator Fault and Input Saturation. Dans *Fault-Tolerant Cooperative Control of Unmanned Aerial Vehicles* (pp. 51-76). (S.l.) : Springer.
- Yue, F., Zonghua, S., Liaoni, W., Yongshun, W., Bin, X., Weng Khuen, H., & Yancheng, Y. (2023). Nonlinear adaptive flight control system: Performance enhancement and validation. *Chinese Journal of Aeronautics*, 36(4), 354-365.

- 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. Dans *AIAA Modeling and Simulation Technologies Conference* (p. 1941).
- Zaag, M., Botez, R. M., & Wong, T. (2019). Cessna Citation X engine model identification using neural networks and extended great deluge algorithms. *INCAS Bulletin*, 11(2), 195-207.
- Zames, G. (1966). On the input-output stability of time-varying nonlinear feedback systems Part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, 11(2), 228-238. <https://doi.org/10.1109/TAC.1966.1098316>
- Zhang, Q., & Liu, H. H. (2018). Aerodynamic model-based robust adaptive control for close formation flight. *Aerospace Science and Technology*, 79, 5-16.
- Zhang, T., Kahn, G., Levine, S., & Abbeel, P. (2016). Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. Dans *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 528-535). Stockholm, Sweden : IEEE. <https://doi.org/10.1109/ICRA.2016.7487175>
- Zhou, Y., Ho, H. W., & Chu, Q. (2021). Extended incremental nonlinear dynamic inversion for optical flow control of micro air vehicles. *Aerospace Science and Technology*, 116, 106889.
- Zhou, Y., van Kampen, E.-J., & Chu, Q. P. (2018). Incremental model based online dual heuristic programming for nonlinear adaptive control. *Control Engineering Practice*, 73, 13-25. <https://doi.org/10.1016/j.conengprac.2017.12.011>
- Zohreh Nejad, E. E., Aillot, P., Henry, C., Ghazi, G., & Botez, R. M. (2024). Cessna Citation X Ground Speed Control using Deep Reinforcement Learning. Dans *AIAA SCITECH 2024 Forum* (p. 2152).
- Zohreh Nejad, E. E., Ghazi, G., & Botez, R. M. (2023). Modeling the Longitudinal Dynamics of the Cessna Citation X using Neural Network Methodology. Dans *AIAA SCITECH 2023 Forum* (p. 2191).