

Novel Trajectory Prediction and Flight Dynamics Modelling and Control based on Robust Artificial Intelligence Algorithms for the UAS-S4

by

Seyed Mohammad HASHEMI

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, MARCH 7TH, 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

© Copyright reserved by Seyed Mohammad Hashemi 2022



© Copyright reserved

It is forbidden to reproduce, save or share the content of this document either in whole or in parts. The reader who wishes to print or save this document on any media must first get the permission of the author.

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED
BY THE FOLLOWING BOARD OF EXAMINERS

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

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

Mr. Guy Gauthier, Member of the jury
System Engineering Department at École de technologie supérieure

Mr. Adrian Hiliuta, External Evaluator
CMC Electronic

Mr. Jeremy Laliberte, External Evaluator
Mechanical and Aerospace Engineering at Carleton University

THIS THESIS WAS PRESENTED AND DEFENDED
IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC
ON MARCH 4TH, 2022
AT ÉCOLE DE TECHNOLOGIE SUPÉRIEUR

ACKNOWLEDGMENT

I am extremely grateful to my esteemed supervisor, Prof. Ruxandra Mihaela Botez for giving me this opportunity to finalize my PhD thesis under her supervision. I would like to offer my special thanks to her for providing an excellent collection of equipment at the Research Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticity (LARCASE), that supported me in the validation of my research studies. I would like to give my sincere appreciation for her constant support, constructive comments, and contributions to our published articles. In addition, I am especially grateful to her for the constant care and support that she has offered when I was suffering from a health problem far from my home country.

I would like to thank to Prof. Teodor Lucian Grigorie for the meetings and discussions regarding intelligent controllers, and for his contributions to our conference and journal papers in 2019. I would like to extend my thanks to Prof. Philippe Bocher and Guy Gauthier for their insightful comments and suggestions on the proposal that helped me to modify the fundamentals of this research study.

Special thanks are due to the Natural Sciences and Engineering Research Council of Canada (NSERC) for the Canada Research Chair Tier 1 in Aircraft Modeling and Simulation Technologies funds. I would also like to thank Mrs. Odette Lacasse and Mr. Oscar Carranza for their support at ETS, as well as to Hydra Technologies' team members Mr. Carlos Ruiz, Mr. Eduardo Yakin, and Mr. Alvaro Gutierrez Prado in Mexico. I wish to express my appreciation to the Canada Foundation for Innovation CFI, the Ministère de l'Économie et de l'Innovation and Hydra Technologies for their support in the acquisition of the UAS-S4 at the LARCASE.

I am deeply grateful to Prof. Georges Ghazi for his help, suggestions, and advice at the LARCASE. It was a wonderful experience for me to be the Co-Chair at the American Institute of Aeronautics and Astronautics (AIAA) Sci-Tech conference session. I would like to extend my special thanks to Dr. Maxime Alex Junior Kuitche for sharing his knowledge on the UAS-

S4. I am very grateful to Mrs. Laurie McLaughlin for her comments and suggestions on this thesis writing.

I would like to thank my colleagues for their kind friendship and collaboration: Mohammad, Musavir, Simon, Mohamed, Marine, Manuel, Radu, Yvan, Shehryar, Hugo and others.

Finally, I must express my deep gratitude to my family for their continued help, support, encouragement, and motivation that kept me on target to complete my PhD program.

NOUVELLE PRÉDICTION DE TRAJECTOIRE ET MODÉLISATION ET CONTRÔLE DE LA DYNAMIQUE DE VOL BASÉS SUR DES ALGORITHMES D'INTELLIGENCE ARTIFICIELLE ROBUSTES POUR L'UAS-S4

Seyed Mohammad HASHEMI

RÉSUMÉ

La gestion du trafic aérien (ATM) et l'évitement des collisions aériennes (ACA) sont les problèmes les plus importants dans le transport aérien. Une prédiction de trajectoire d'aéronef (ATP) précise, un modèle de dynamique de vol (FDM) précis et un contrôle de dynamique de vol (FDC) efficace sont les exigences fondamentales pour les opérations basées sur la trajectoire, telles que l'ATM et l'ACA. L'objectif de cette thèse est de concevoir et de développer ces trois exigences fondamentales mentionnées ci-dessus pour les opérations critiques basées sur les trajectoires. Pour chaque exigence fondamentale, une étude de recherche approfondie a été menée pour atteindre ses objectifs relatifs.

La première étude s'est concentrée sur la précision de la prédiction de trajectoire d'aéronef (ATP). Cette étude a commencé par la formulation d'un ATP en tant que problème de régression de séries chronologiques. Ensuite, six modèles de réseaux de neurones basés sur les données ont été conçus et affinés pour produire des ATP précis. Leurs architectures étaient basées sur la régression logistique (LR), la régression vectorielle de support (SVR), le réseau neuronal profond (DNN), le réseau neuronal convolutif (CNN), le réseau neuronal récurrent (RNN) et la mémoire à long et à court terme (LSTM). Les six modèles ATP ont été évalués en termes de précision de leur prédiction et leur super-efficacité a été confirmée. Malgré leurs excellentes performances, nous avons pu générer des échantillons contradictoires pour les induire en erreur, ce qui présente un problème de sécurité concernant les modèles ATP basés sur le réseau neuronal. Par conséquent, un algorithme de défense a été conçu en se basant sur une stratégie de recyclage contradictoire. Les nouveaux modèles ATP robustes basés sur l'apprentissage ont montré d'excellentes performances contre les attaques adverses tout en exécutant leurs tâches ATP avec précision.

La deuxième étude portait sur la conception d'un contrôleur de dynamique de vol (FDC) efficace. S'appuyant sur le modèle de dynamique de vol programmé linéaire local UAS-S4 (LLS-FDM) disponible au laboratoire de recherche sur les commandes actives, l'avionique et l'aéroservoélasticité (LARCASE), un FDM non linéaire a été conçu en se basant sur l'approche de la logique floue Takagi-Sugeno (TS). Simultanément, le modèle de référence souhaité a été déterminé, puis stabilisé à l'aide d'un régulateur quadratique linéaire (LQR). En ce qui concerne le FDM de référence, un FDC "basé sur un modèle" a été conçu pour le FDM UAS-S4, qui a très bien fonctionné en se basant sur des erreurs de suivi. Ensuite, un contrôleur de logique floue (FLC) contenant des gains adaptatifs robustes a été conçu afin de prendre en compte les non-linéarités et les incertitudes dues à la fuzzification et aux perturbations externes. Les

VIII

résultats ont confirmé que le contrôleur à logique floue adaptative robuste pouvait stabiliser la dynamique de vol et suivre avec précision les variables d'état du modèle de référence.

La troisième étude a été menée pour la conception d'une méthode précise de modélisation de la dynamique de vol (FDM). Des FDM précis permettent aux ingénieurs de concevoir des FDC basés sur des modèles très efficaces. Des essais en vol ont été effectués sur l'UAS-S4 Ehecatl (au LARCASE) et ensuite 216 FDM locaux ont été obtenus en utilisant le modèle interne de dynamique de vol à planification linéaire locale (LLS-FDM) conçu pour gérer 216 conditions de vol. Le LLS-FDM a été encore augmenté en utilisant les méthodologies d'interpolation et d'extrapolation des k plus proches voisins. S'appuyant sur ces données augmentées, la régression vectorielle de support (SVR) a été utilisée comme algorithme de référence pour la régression LLS-FDM. Le SVR formé pouvait prédire le FDM de l'UAS-S4 pour l'ensemble du domaine de vol. Un diagramme Root Locus a été utilisé pour valider le SVR-FDM UAS-S4 en évaluant la proximité des valeurs propres prédites avec leurs valeurs d'origine. La précision de la prédiction SVR a été étudiée pour différentes conditions de vol, le nombre de voisins et une gamme de fonctions du noyau. Malgré les excellentes performances du SVR formé, le FDM était vulnérable aux échantillons contradictoires. Par conséquent, Adversarial Retraining Defense (ARD) a été développée en s'appuyant sur des FDM contradictoires, qui ont été créés via la méthode de signe de gradient rapide adaptée (AFGSM) pour concevoir un Robust-SVR FDM. Le Robust-SVR FDM a très bien fonctionné lors d'attaques contradictoires tout en offrant de meilleures propriétés dans le domaine temporel pour la stabilisation des variables d'état que le LLS-FDM.

Mots-clés : prédiction de trajectoire d'aéronef, réseau de neurones, attaques adverses, contrôle de la dynamique de vol, logique floue, gains adaptatifs robustes, stabilité de Lyapunov, modèle de dynamique de vol, augmentation des données, régression vectorielle de support.

NOVEL TRAJECTORY PREDICTION AND FLIGHT DYNAMICS MODELLING AND CONTROL BASED ON ROBUST ARTIFICIAL INTELLIGENCE ALGORITHMS FOR THE UAS-S4

Seyed Mohammad HASHEMI

ABSTRACT

Aerial Traffic Management (ATM) and Aerial Collision Avoidance (ACA) are the most important issues in aviation transportation. Accurate Aircraft Trajectory Prediction (ATP), precise Flight Dynamics Model (FDM), and efficient Flight Dynamics Control (FDC) are the main fundamental requirements, for trajectory-based operations, such as ATM and ACA. The aim of this thesis is to design, and further develop these above-mentioned three fundamental requirements for critical trajectory-based operations. For each fundamental requirement, a thorough research study was conducted to meet its related objectives.

The first study focused on accurate Aircraft Trajectory Prediction (ATP). This study began with formulating an ATP as a time series regression problem. Next, six data-driven Neural Network models were designed and fine-tuned to produce accurate ATPs. Their architectures were based on Logistic Regression (LR), Support Vector Regression (SVR), Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long-Short Term Memory (LSTM). The six ATP models were evaluated in terms of prediction accuracy, and their super-efficiency was confirmed. Despite their excellent performance, we could generate adversarial samples to mislead them. This issue presents a security concern regarding Neural Network-based ATP models. Therefore, a defense algorithm was designed based on an adversarial retraining strategy. The new, robust learning-based ATP models showed excellent performance against adversarial attacks while still performing their ATP tasks accurately.

The second study addressed the design of an efficient Flight Dynamics Controller (FDC). Relying on the available UAS-S4 Local Linear Scheduled Flight Dynamics Model (LLS-FDM) at the Laboratory of Research in Active Controls, Avionics and Aeroservoelasticity (LARCASE), a nonlinear FDM was designed based on the Takagi-Sugeno (TS) Fuzzy Logic approach. Simultaneously, the desired reference model was determined, and then stabilized using a Linear Quadratic Regulator (LQR). Regarding the reference FDM, a “model-based” FDC was designed for the UAS-S4 FDM, which worked very well based on tracking errors. Next, a Fuzzy Logic Controller (FLC) containing robust adaptive gains was designed in order to consider the nonlinearities and uncertainties due to fuzzification and external disturbances. The results confirmed that the robust adaptive fuzzy logic controller could stabilize the flight dynamics and accurately track the reference model state variables.

The third study was conducted for the design of an accurate Flight Dynamics Modelling (FDM) method. Accurate FDMs allow engineers to design highly efficient model-based FDCs. Flight tests were conducted on the UAS-S4 Ehecaltl (at the LARCASE), and 216 local FDMs were

obtained by using the in-house Local Linear Scheduling Flight Dynamics Model (LLS-FDM) designed to handle 216 flight conditions. The LLS-FDM was further augmented using the k-nearest neighbor interpolation and extrapolation methodologies. Relying on this augmented data, Support Vector Regression (SVR) was used as a benchmarking algorithm for the LLS-FDM regression. The trained SVR could predict the UAS-S4 FDM for the entire flight envelope. A Root Locus diagram was utilized to validate the UAS-S4 SVR-FDM by evaluating the predicted eigenvalues' closeness to their original values. The SVR prediction accuracy was studied for different flight conditions, number of neighbors, and a range of kernel functions. Despite the excellent performance of the trained SVR, the FDM was vulnerable to adversarial samples. Hence, an Adversarial Retraining Defense (ARD) was developed by relying on adversarial FDMs, that were created via the Adapted Fast Gradient Sign Method (AFGSM) to design a Robust-SVR FDM. The Robust-SVR FDM worked very well under adversarial attacks while providing better time domain properties for state variable stabilization than the LLS-FDM.

Keywords: Aircraft Trajectory Prediction, Neural Network, Adversarial Attacks, Flight Dynamics Control, Fuzzy Logic, Robust Adaptive Gains, Lyapunov Stability, Flight Dynamics Model, Data Augmentation, Support Vector Regression.

TABLE OF CONTENTS

	Page
INTRODUCTION.....	1
CHAPTER 1 LITERATURE REVIEW.....	7
1.1 Trajectory Prediction (TP)	7
1.1.1 Deterministic TP	7
1.1.2 Probabilistic TP.....	8
1.1.3 Deterministic-Probabilistic TP.....	9
1.2 Flight Dynamics Control (FDC)	11
1.2.1 Classical Flight Dynamics Control	11
1.2.2 Optimal Flight Dynamics Control	11
1.2.3 Adaptive Flight Dynamics Control.....	12
1.2.4 Robust Flight Dynamics Control	13
1.2.5 Artificial Intelligence Flight Dynamics Control.....	13
1.3 Flight Dynamics Model (FDM) Improvement.....	14
1.3.1 Nonlinear Flight Dynamics Model	14
1.3.2 Linearized FDM.....	14
1.3.3 Local Linear Scheduled FDM.....	15
1.3.4 Regressed Scheduled Local Linear FDMs.....	15
CHAPTER 2 RESEARCH OBJECTIVES, METHODOLOGIES, AND CONTRIBUTIONS	17
2.1 Objectives, Methodologies, and Contributions.....	17
2.1.1 First Study – The Aircraft Trajectory Prediction (TP) Problem.....	17
2.1.1.1 Objective 1	17
2.1.1.2 Methodology 1	17
2.1.1.3 Contributions.....	17
2.1.2 Second Study – The Design of a Flight Dynamics Controller (FDC)	18
2.1.2.1 Objective 2	18
2.1.2.2 Methodology 2.....	18
2.1.2.3 Contribution 2	19
2.1.3 Third Study – The UAS-S4 Flight Dynamics Model (FDM) Improvement.....	19
2.1.3.1 Objective 3	19
2.1.3.2 Methodology 3	20
2.1.3.3 Contribution 3	20
CHAPTER 3 NEW RELIABILITY STUDIES OF DATA-DRIVEN AIRCRAFT TRAJECTORY PREDICTION.....	23
3.1 Introduction	25
3.2 Related Works on Trajectory-Based Operations.....	27
3.2.1 Collision Avoidance.....	28

3.2.2	Data-Driven Trajectory Prediction	30
3.3	Building Data-Driven Predictors.....	31
3.3.1	Logistic Regression (LR).....	31
3.3.2	Support Vector Regression (SVR).....	33
3.3.3	Deep Neural Network (DNN).....	33
3.3.4	Convolutional Neural Network (CNN).....	36
3.3.5	Recurrent CNN (RNN).....	38
3.3.6	Long Short-Term Memory (LSTM)	40
3.4	Numerical Results	42
3.4.1	Dataset.....	42
3.4.2	Measuring the Resiliency of Models	42
3.4.3	Adversarial Retraining.....	52
3.5	Conclusions.....	53
CHAPTER 4	LYAPUNOV-BASED ROBUST ADAPTIVE CONFIGURATION OF THE UAS-S4 FLIGHT DYNAMICS FUZZY CONTROLLER	55
4.1	Introduction.....	56
4.2	UAS-S4 Flight Dynamics Modeling.....	59
4.2.1	UAS-S4 Linear Local Models	62
4.2.2	UAS-S4 Fuzzy Logic Model	64
4.3	Flight Dynamics Control.....	66
4.3.1	Reference Model.....	66
4.3.2	Fuzzy Logic Controller (FLC).....	67
4.3.3	T-S Fuzzy Logic Controller.....	68
4.3.4	Adaptive T-S Fuzzy Logic Controller	69
4.3.5	Robust Adaptive T-S Fuzzy Logic Controller	72
4.4	Results.....	78
4.5	Conclusion.....	94
CHAPTER 5	A NOVEL FLIGHT DYNAMICS MODELING USING ROBUST SUPPORT VECTOR REGRESSION AGAINST ADVERSARIAL ATTACKS.....	97
5.1	Introduction	99
5.2	The UAS-S4 Flight Dynamics Model (FDM).....	103
5.2.1	Local Linear Scheduled Flight Dynamics Model (LLS-FDM)	103
5.2.2	Data Augmentation.....	105
5.2.3	Support Vector Regression (SVR) of a Flight Dynamics Model	107
5.3	Adversarial Attacks.....	108
5.3.1	Adapted Fast Gradient Sign Method (AFGSM).....	110
5.3.2	Defense against Adversarial Attacks	110
5.4	Results.....	111
5.4.1	The UAS-S4 FDM Root-Locus Diagram	111
5.4.2	Regression effectiveness under the operation of a unique LQR.....	117
5.4.3	Robustness against Adversarial Attacks	120

5.5 Conclusions125

CHAPTER 6 DISCUSSION OF THE RESULTS.....127

CONCLUSION131

RECOMMENDATIONS133

LIST OF REFERENCES135

LIST OF TABLES

	Page
Table 0.1 Number of Pilot-Reported Near Midair Collisions (NMAC) by degree of hazard.....	1
Table 3.1 Mean values of ϵ and δ for training samples of the TFMS dataset.....	44
Table 3.2 Fooling rate and prediction confidence of the models.....	45
Table 3.3 Transferability of adversarial samples from one model to another model.....	45
Table 3.4 Performance comparison of logistic regression (LR) on local intrinsic dimensionality (LID) scores. The solver for this LR classifier is “liblinear”.....	47
Table 3.5 Transferability of adversarial samples crafted by the adapted fast.....	52
Table 3.6 Performance comparison of data-driven models by adversarial retraining.....	53
Table 4.1 UAS-S4 specifications (geometrical and flight data).....	60
Table 4.2 Sum of Absolute Tracking Errors (<i>time = 40 sec</i> and <i>sampling time = 0.01 sec</i>) while the controlled UAS-S4 state variables are tracking the reference model states.....	92
Table 4.3 Sum of Absolute Tracking Errors (<i>time = 40 sec</i> and <i>sampling time = 0.01 sec</i>) while the controlled UAS-S4 state variables are tracking the reference model state variables in the presence of various uncertainties.....	93
Table 4.4 Sum of Absolute Tracking Errors (<i>time = 40 sec</i> and <i>sampling time = 0.01 sec</i>) the controlled UAS-S4 state variables are tracking the reference model state variables in the presence of uncertainties for different adaptation weight values.....	94
Table 5.1 The UAS-S4 geometrical and flight data Specification.....	100
Table 5.2 The UAS-S4 FDM prediction accuracy using the SVR.....	115
Table 5.3 The UAS-S4 FDM prediction accuracy using SVR with.....	116
Table 5.4 The UAS-S4 FDM prediction accuracy while.....	116
Table 5.5 The controlled step response properties of the developed and.....	119
Table 5.6 The UAS-S4 pitch angle step response time-domain characteristics.....	122

Table 5.7 The UAS-S4 roll angle step response time-domain characteristics.....	123
Table 5.8 The UAS-S4 yaw angle step response time-domain characteristics.....	124
Table 5.9 The SVR and Robust-SVR step response improvements compared	125

LIST OF FIGURES

	Page
Figure 3.1 Encounter scenario	27
Figure 3.2 Proposed deep neural network (DNN) architecture for the aircraft trajectory regression problem (ATRP).....	35
Figure 3.3 Proposed convolutional neural network (CNN) architecture for the ATRP	37
Figure 3.4 Proposed recurrent CNN (RNN) architecture for the ATRP.....	39
Figure 3.5 Proposed long short-term memory (LSTM) architecture for the ATRP	41
Figure 3.6 The LID score comparisons for different random samples chosen from the training set.....	50
Figure 4.1 Hydra Technologies UAS-S4 Ehecatl.....	60
Figure 4.2 The followed procedure to control the UAS-S4 flight dynamics.....	61
Figure 4.3 The fuzzy logic controller utilized for the UAS-S4 flight dynamics	68
Figure 4.4 The designed robust adaptive T-S fuzzy logic	77
Figure 4.5 the RAFL controller performance in terms of longitudinal and lateral state variables stabilization.....	82
Figure 4.6 The RAFL controller performance in terms of convergence error.....	85
Figure 4.7 RAFLC performance in terms of pitch angle and pitch rate tracking in the absence of uncertainties	86
Figure 4.8 The RAFL controller performance in terms of the reference model pitch angle tracking in the presence of uncertainties caused by unknown controller's parameters.....	88
Figure 4.9 The RAFL controller performance in presence of external disturbances and modeling errors	89
Figure 4.10 Comparing the AFLC with the Robust AFLC (RAFLC) in terms of reference model tracking for different	90
Figure 5.1 Hydra Technologies UAS-S4 Ehecatl.....	100
Figure 5.2 The procedure to obtain the UAS-S4 FDM for its entire flight envelope.....	102

Figure 5.3 Augmenting local FDMs using an interpolation methodology based on the k-nearest neighbors105

Figure 5.4 Augmenting local FDMs using an extrapolation methodology based on the k-nearest neighbors106

Figure 5.5 Architecture of the Trained Support Vector Regression (SVR) for the UAS-S4 Flight Dynamics Model (FDM)108

Figure 5.6 Root Locus diagram of the UAS-S4 longitudinal flight dynamics model under its elevator angle actuation112

Figure 5.7 Root Locus diagram of the UAS-S4 lateral flight dynamics model under aileron angle actuation113

Figure 5.8 Root Locus diagram of the UAS-S4 lateral flight dynamics model under rudder angle actuation.....114

Figure 5.9 FDM design based on “Local Linear Scheduling” and “SVR” approaches.....117

Figure 5.10 The controlled pitch angle step responses of the developed (SVR) and the initial (LLS) UAS-S4 FDMs.....119

Figure 5.11 The effect of an imposed adversarial attack on the UAS-S4 longitudinal FDM prediction.....120

Figure 5.12 The UAS-S4 pitch angle step response using a LQR controller when different types of FDMs are used while an adversarial attack is imposed.....121

Figure 5.13 The UAS-S4 roll angle step responses using the LQR controller when different types of FDMs are used while an adversarial attack is imposed.....122

Figure 5.14 The UAS-S4 yaw angle step response with an LQR controller when different types of FDMs are used while an adversarial attack is imposed.....124

LIST OF ABBREVIATIONS

ACA	Aerial Collision Avoidance
AFGSM	Adapted Fast Gradient Sign Method
AFL	Adaptive Fuzzy Logic
AI	Artificial Intelligence
AIAA	American Institute of Aeronautics and Astronautics
ASIAS	Aviation Safety Information Analysis and Sharing
ATM	Air Traffic Management
ATMCA	Air Traffic Management and Collision Avoidance
ATP	Aircraft Trajectory Prediction
ATRP	Aircraft Trajectory Regression Problem
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DP	Dynamic Programming
DT	Decision Tree
FAA	Federal Aviation Administration
FDC	Flight Dynamics Control
FDM	Flight Dynamics Model
FIS	Fuzzy Interface System
FLC	Fuzzy Logic Model
FLM	Fuzzy Logic Control
GA	Genetic Algorithm

HMM	Hidden Markov Model
LARCASE	Laboratory of Applied Research in Active Controls, Avionics and Aeroservoelasticity
LID	Local Intrinsic Dimensionality
LLS	Local Linear Scheduled
LQG	Linear-Quadratic-Gaussian
LQR	Linear Quadratic Regulator
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MPC	Model Predictive Control
MLP	Multilayer Perceptron
NDE	Nonlinear Differential Equations
NN	Neural Network
NMAC	Near Midair Collision Avoidance
PID	Proportional-Integral-Derivative
PSO	Particle swarm optimization
RAFLC	Robust Adaptive Fuzzy Logic Control
RAFS	Research Aircraft Flight Simulators
RNN	Recurrent Neural Network
SATE	Sum of Absolute Tracking Error
SMC	Sliding Mode Control

SVM	Support Vector Machine
SVR	Support Vector Regression
TCAS	Traffic Collision Avoidance System
TFMS	Traffic Flow Management System
TP	Trajectory Prediction
TS	Takagi-Sugeno
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle

LIST OF SYMBOLS

Symbols

A_{lon}	Longitudinal flight dynamics state Matrix of the UAS-S4
A_{lat}	Lateral flight dynamics state matrix of the UAS-S4
B_{lon}	Longitudinal flight dynamics control matrix of the UAS-S4
B_{lat}	Lateral flight dynamics control matrix of the UAS-S4
C_{lon}	Longitudinal flight dynamics output matrix of the UAS-S4
C_{lat}	Lateral flight dynamics output matrix of the UAS-S4
D_{lon}	Longitudinal flight dynamics feedforward matrix of the UAS-S4
D_{lat}	Lateral flight dynamics feedforward matrix of the UAS-S4
A_i, B_i	State and control matrices of the UAS-S4 Takagi-Sugeno fuzzy model
A_r, B_r	State and control matrices of the reference model
$d(X, t)$	Bounded external disturbance
E	Tracking error
G_u, H_u, M_u	state matrix dimensional stability derivatives related to axial velocity
G_w, H_w, M_w	state matrix dimensional stability derivatives related to vertical velocity
M_q	state matrix dimensional stability derivatives related to Pitch rate
$G_\delta, H_\delta, M_\delta$	Longitudinal control matrix dimensional stability derivatives

K	Linear Quadratic Regulator (LQR) gain
$K_{i_{1 \times n}}, Z_i$	Takagi-Sugeno fuzzy logic controller gains
$k_{i_{1 \times n}}, z_i$	Desired fuzzy logic controller gains based on the reference model
p	Roll rate
q	Pitch rate
Q, R	Linear Quadratic Regulator cost function weights
r	Reference input
η	Yaw rate
u	Axial velocity
v	Side velocity
V	Lyapunov function
w	Vertical velocity
x	Original sample (in Chapter 3 and 5) / State variables in fuzzy rules (Chapter 4)
X	Given input to the data-driven algorithms (in Chapter 3) / The UAS-S4 state variables vector (in chapter 4 and 5)
\mathbb{X}	Support Vector Regression input
\tilde{x}	Adversarial sample
X_r	Reference model state variables vector

Y	Given output to the data-driven algorithms (in Chapter 3) / The UAS-S4 output (in chapter 5)
\mathbb{Y}	Support Vector Regression output
Y_v, L_v, N_v	state matrix dimensional stability derivatives related to side velocity
Y_p, L_p, N_p	state matrix dimensional stability derivatives related to pitch rate
Y_η, L_η, N_η	State matrix dimensional stability derivatives related to yaw rate
$Y_\delta, L_\delta, N_\delta$	Lateral control matrix dimensional stability derivatives
Z_j	Original local flight dynamics model
\bar{Z}_j	New embedding local flight dynamics model
Z_k	Computed centroid flight dynamics model
Greek letters	
Θ	Support Vector Regression weighting vector
θ	Data-driven models weighting vector (in chapter 3) / Pitch angle in chapter (4 and 5)
φ	Roll angle
δ	Control input vector
δ_a	Aileron angle
δ_e	Elevator angle
δ_r	Rudder angle

ϕ_i	Fuzzy rule activator
Γ_j^i	Membership function's collected grades in fuzzy subsystems
γ	Threshold for surpassing the regression boundary
γ_1, γ_2	Constant weights of the Lyapunov function
$\epsilon_{A_i}, \epsilon_{B_i}$	UAS-S4 model uncertainties
λ_{ext}	Extrapolation tuning factor
λ_{int}	Interpolation tuning factor
ϵ	Support Vector Regression decision boundary
ε	Injected perturbation to the SVR output
σ	Utilized output value for generating adversarial sample
μ	Decision boundary toward generating adversarial sample
Θ	Support Vector Regression weighting vector

INTRODUCTION

Air Traffic Management and Control (ATMC) with the special aim of Aerial Collision Avoidance (ACA) is the most important issue in aerial transportation. Aerial collisions are the type of accidents that occur when at least two aircraft come into an unsafe mutual zone and damage each other. An aerial collision can lead to loss of handling and even destruction (Cockcroft & Lameijer, 2003).

0.1 Brief History

The first catastrophic aerial collision occurred in France in 1912, killing both French Army pilots in a midair crash. This collision inspired researchers to define anti-collision safety requirements (A. Cook, 2007). A catastrophic aircraft collision over the Grand Canyon which led to the death of all 128 passengers and crew in 1950 gave a strong motivation to researchers to design a reliable safety procedure to avoid aerial collisions.

It took almost two decades to design the first prototype of the Aerial Collision Avoidance (ACA) system, which was first utilized in 1970. The broad range of alarms and unnecessary resolution advisories due to its simple, conservative approach motivated researchers to improve this first ACA system prototype (J. Kuchar & Drumm, 2007). A large number of studies have been undertaken to design a more reliable and efficient collision avoidance systems (Albaker & Rahim, 2009).

The importance of Air Traffic Management and Collision Avoidance (ATMCA) system improvement is reinforced by the information presented in Table 0.1 (U.S. Department of Transportation, 2021), showing the number of pilot-reported Near Midair Collisions (NMAC).

Table 0.1 Number of Pilot-Reported Near Midair Collisions (NMAC) by Degree of Hazard

Year	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Critical	14	12	15	27	29	56	67	20	19	23
Potential	37	45	52	91	95	179	208	131	55	66
No hazard	27	17	19	27	33	43	38	19	13	18
NMAC	31	17	23	42	31	74	90	38	32	24

These NMACs were reported by pilots of general aviation, air carriers, and other aircraft utilized for public-use operations. This information was shared by the Federal Aviation Administration (FAA) Aviation Safety Information Analysis and Sharing (ASIAS) system (U.S. Department of Transportation, 2021). Table 0.1 lists the NMACs by the degree of hazard; all occurred despite the use of modern anti-collision systems.

Based on the ASIAS information, despite all modifications and improvements, the number of NMAC reports has remained high. This situation indicates that major research contributions are needed for the ATMCA system improvement. In this way, the fundamental requirements and algorithms that ATMCA relies and functions based on them must be improved.

0.2 Problem Statement

For problem statement, firstly, it is needed to explain the “Encounter” situation. Then novel algorithms for solving encounter scenarios should be determined in order to avoid aerial collisions.

0.2.1 Encounter Situation

In the Encounter situation it is assumed that a flight zone containing different aircraft that are equipped with the transponders and receivers for establishing communications with each other. The "Owner" aircraft knows its future trajectory for a specific time, while receiving information related to the altitude, heading, velocity, and position of the other “Intruder” aircraft. The collision avoidance algorithm defines a virtual border that includes three zones around the "Owner" aircraft.

The external zone is called the “Caution Area” and is located where the ACA system gives an alert to make the owner aircraft aware of an intruder aircraft. The middle zone is called the “Warning Area”, where the ACA system must perform its critical task. In this zone, the owner aircraft adopts a new safe path given by a resolution algorithm, relying on its flight dynamics controller to avoid potential collisions. The core zone is called the “Collision Area” in which the distance between the owner and the intruder aircraft is not long enough and a collision

would most likely occur (Henely, 2017). At this point, it is needed to be aware of the fundamental requirements for detecting conflicts and then solving the encounter situations.

0.2.2 Air Traffic Management and Collision Avoidance Requirements

Successful requirements in the “Caution Area” and in the “Warning Area” can guarantee aerial collision avoidance. The first requirement is a reliable Trajectory Prediction (TP) algorithm that is the fundamental tool for the “Caution Area”, and it plays a key role toward early conflict detection. The second requirement is an efficient Flight Dynamics Control (FDC) algorithm that is the essential tool in the “Warning Area” for performing trajectory resolution accurately. The third requirement is a precise Flight Dynamics Model (FDM), that is the most important element for designing a model-based flight dynamics controller. The improvement of these requirements can significantly increase the reliability and efficiency of the ATMCA systems.

0.3 Solutions

Many causes can increase the probabilities of aerial collisions. Parametric and non-parametric uncertainties, human errors, communication failures, hacks and security attacks are the most common reasons that can lead to errors in navigation, and therefore, in aircraft deviation from its desired flight trajectory (Kochenderfer, Holland, & Chryssanthacopoulos, 2012).

The key requirements for improving the ATMCA systems reliability and efficiency can be classified in three categories. The first essential step towards an anti-collision algorithm improvement is to design a highly accurate Trajectory Prediction (TP). Accurate TPs not only provide early conflict detection, but they can also reduce unnecessary alarms and resolutions.

Once a conflict has been detected (by the TP), a new safe path must be followed by the aircraft in order to escape from the detected conflict. Accurate tracking of the new safe path can guarantee collision avoidance. An efficient Flight Dynamics Controller (FDC) is needed to achieve this task, which is the second requirement.

Basically, Flight Dynamics Controllers that function by relying on excellent Flight Dynamics Models can provide evidently excellent performance (Xu & Shi, 2015). Therefore, the design

of an accurate Flight Dynamics Model can significantly improve a flight dynamics controller's effectiveness.

Given their inter-dependency, the design of accurate Trajectory Prediction (TP) algorithms, Flight Dynamics Controllers (FDCs), and aircraft Flight Dynamics Models (FDMs) can meet ATMCA requirements with high reliability.

0.4 Thesis Organization

Following the above introduction, this thesis is composed of six Chapters. Chapter 1 gives a bibliographical literature review in fields of trajectory prediction, as well as flight dynamics modeling and control. Trajectory Prediction (TP) strategies from traditional to modern were investigated. Traditional model-based TP strategies were reviewed, including kinetic, point-mass, kinematic, and receding horizon. Modern approaches from probabilistic to data-driven were analyzed to determine which would offer the best performance. Model-based Flight Dynamics Control (FDC) algorithms were investigated, including their classical, optimal, adaptive, robust, and intelligent approaches. Methodologies to obtain accurate Flight Dynamics Model (FDM) were also described at the end of Chapter 1, as they are essential to the FDC performance.

Chapter 2 expresses the three studies composed of their new objectives, new methodologies, and original contributions of this research.

In the first study, the design of a reliable and accurate trajectory prediction algorithm was determined as the 1st objective, followed by the novel proposed methodology in order to meet that objective. At its end, the novelties and contributions were highlighted.

In the second study, the design of a flight dynamics controller, that is robust in the presence of uncertainties was determined as the second objective, followed by the new methodology and its contributions.

The third study concerned a procedure for obtaining an excellent secured FDM, that was represented as the third objective. The relevant new methodology and its original contributions were presented.

Chapters 3-5 contain the journal articles that satisfy the first to the third studies' objectives, respectively. Innovative data-driven models designed for accurate trajectory prediction based on deep neural networks (Chapter 3) fulfilled the first objective. The second objective was met in Chapter 4, with the new design methodology of a robust adaptive fuzzy flight dynamics controller. In Chapter 5, a novel flight dynamics modeling approach based on secured Support Vector Regression against adversarial attacks, satisfied the third objective.

Chapter 6 gives a general discussion based on the results in presented in Chapters 3-5. After Chapter 6, conclusion that obtained from the carried-out research is presented. Recommendations for future work and research areas needed to investigate are presented with the aim of further development of this integrated approach with the aim to increase the performance of trajectory-based operation systems.

CHAPTER 1

Literature Review

This chapter represents the literature review on the fundamental requirements for improving aerial trajectory-based operations, such as aerial traffic management and collision avoidance. Research studies on Trajectory Prediction (TP), Flight Dynamics Control (FDC), and precise Flight Dynamics Modelling (FDM) approaches were investigated separately and are reviewed below in three sections.

1.1 Trajectory Prediction (TP)

Basically, Trajectory Prediction (TP) can be discussed from three perspectives: deterministic, probabilistic, and probabilistic-deterministic. These types of approaches have been utilized for a variety of trajectory-based operations (Georgiou et al., 2018).

1.1.1 Deterministic TP

Deterministic trajectory prediction refers to the techniques in which future paths are fully determined using their initial and parameter values. "Lookup Table" is a simple deterministic TP method that can be easily implemented based on a performance database. Several studies were carried out at the LARCASE (Laboratory of Research in Active Controls, Avionics and Aeroelasticity) to perform flight trajectory prediction using Lookup Tables (Ghazi, Botez, & Tudor, 2015; Ghazi, Tudor, & Botez, 2015; A. Murrieta-Mendoza & R. M. Botez, 2015). While the algorithms based on Lookup Tables developed in these works could predict and optimize the flight trajectories very well, their performance decreased when heading angle turns and environmental uncertainties were introduced.

There exist other deterministic TP methods that are based on a "Kinetic" model (Fossen, 2011). In this approach, the future flight trajectory can be predicted via mathematical equations containing an aircraft's forces and moments (Nuic, Poles, & Mouillet, 2010). (Delahaye, Puechmorel, Tsiotras, & Féron, 2014) completed a survey study on using Kinetic models for TP that revealed their satisfactory performance for the Air Traffic Management and Control

(ATMC). However, their efficiency decreased due to the computational complexity of the Kinetic model approach in real-time operations (Ghasemi Hamed, Gianazza, Serrurier, & Durand, 2013; Herty & Visconti, 2018; Karr, Vivona, Woods, & Wing, 2017). To overcome that problem of calculations complexity, a simplified version was developed, called the “Point-Mass” model, for fast-time simulations (Weitz, 2015). This method has been useful for a wide range of TP applications (Ghasemi Hamed et al., 2013), but due to the varying mass and center of gravity of aircraft, the accuracy of TPs produced with a Point-Mass model was not used as much as expected due to safety issues.

The use of “Kinematic” model is another method for trajectory prediction (Anderson, Vasudevan, & Johnson-Roberson, 2021). This method does not rely upon the knowledge of an aircraft’s forces and moments, and it was implemented to provide statistical trajectory prediction (Y. Lin, Zhang, & Liu, 2018). However, predicted trajectories using Kinematic models were not precise enough due to the lack of information regarding system dynamics.

Among the deterministic approaches, the most successful one was then developed using dynamics and kinematic models. This approach was further developed into an optimal technique called "receding horizon planning " (Bellingham, Richards, & How, 2002). This technique was used for Model Predictive Control (MPC) (Camacho & Alba, 2013), as well as for trajectory prediction (Ayhan & Samet, 2016), path planning (Mousavi, Heshmati, & Moshiri, 2013), and collision avoidance (Chaloulos, Hokayem, & Lygeros, 2010).

While MPC approaches offer good performance in light traffic zones, their performance decreases dramatically in crowded zones (Ekaputri & Syaichu-Rohman, 2013), where the algorithm must solve a complex Dynamic Programming (DP) problem while respecting a wide range of constraints (Plancher, Manchester, & Kuindersma, 2017). Hence, probabilistic approaches have been developed to overcome this challenge.

1.1.2 Probabilistic TP

Many research studies were conducted on trajectory prediction based on probabilistic approaches (Weiyi Liu & Hwang, 2011; Wiest, Höffken, Kreßel, & Dietmayer, 2012).

Evolutionary algorithms quickly became some of the utilized probabilistic methodologies used for trajectory prediction and path planning (Nikolos, Valavanis, Tsourveloudis, & Kostaras, 2003). Among these algorithms, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and hybrid PSO-GA have offered good performance. Particle Swarm Optimization (PSO) was used to generate 3D geometrical position based on B-spline curves (Foo, Knutzon, Kalivarapu, Oliver, & Winer, 2009). Genetic Algorithms (GAs) were utilized for greedy regression and path prediction in highly convex scenarios (Cobano, Conde, Alejo, & Ollero, 2011).

The GA execution time is higher than that of the PSO approach, while the PSO is poor in local search compared to the GA (Kachitvichyanukul, 2012). In an attempt to profit from the benefits of both GA and PSO approaches, a combined PSO-GA algorithm was proposed by (Duan, Luo, Shi, & Ma, 2013). Therefore, the hybrid PSO-GA algorithm was designed for the UAVs formations while avoiding collisions based on their predicted trajectories. Their results showed that the hybrid PSO-GA could outperform individual PSO and GA algorithms in terms of trajectory prediction accuracy. With respect to their contributions, these evolutionary algorithms incorporated only local nearby consistencies in their optimization process, and they generated to-the-point location regardless of the entire trajectory (Al-Salami, 2009).

1.1.3 Deterministic-Probabilistic TP

With the goal of exploiting the advantages of both deterministic and probabilistic techniques, Neural Networks (NNs) have provided many hybrid estimation algorithms (H. Wang et al., 2017). Learning algorithms from simple Multi-Layer Perceptron (MLP) to advanced Deep Neural Networks (DNNs) successfully aggregate the deterministic-probabilistic advantages (Nanjundappan, 2016). In this hybrid approach, deterministic datasets and architectures that rely on probabilistic activation functions are expected to learn who to predict future trajectories.

Machine Learning (ML) algorithms have been broadly applied for trajectory prediction, and they have offered excellent performances when enough training dataset were provided. Hidden

Markov Model (HMM) (Fine, Singer, & Tishby, 1998), Decision Tree (DT) (Song & Ying, 2015), Deep Neural Network (DNN) (Weibo Liu et al., 2017) are known as the most powerful algorithms for TP time series regression. These algorithms are investigated in the following paragraphs.

A long-term path prediction algorithm was successfully implemented by use of a trained hidden Markov model (HMM) (Ayhan & Samet, 2016). In addition to the aircraft's 3D position, environmental data was considered as the fourth dimension of the training dataset to solve the weather uncertainties. Unfortunately, HMMs are expensive in terms of memory and computing time, as they require solving Dynamic Programming (DP) in order to perform optimal path regression.

Random Forest, a decision tree-based algorithm, was utilized for aircraft trajectory prediction and showed good performance due to its tendency to the high variance and low bias. (Hernández, Magaña, & Berna, 2018). Worth to mention, such variance-bias trade-off results in model complexity, and decision tree-based algorithms may have instability problems, and therefore become unstable, as small changes in their input data may lead to huge changes in their optimal structure.

Among the Deep Neural Network (DNN) approaches, long-term 4D trajectory prediction was performed using a deep generative neural network while considering environmental uncertainties (Y. Liu & Hansen, 2018). In (Wu, Chen, Sun, Zheng, & Wang, 2017), Recurrent Neural Network (RNN) was designed to estimate aircraft trajectories by relying on sequences of their geometrical data. High improvements in the geometrical position predictions accuracy were found in results. By conceding a sequence of geometrical data, a Long Short-Term Memory (LSTM) trajectory prediction algorithm was designed and, proven to outperform its predecessor (Park, Kim, Kang, Chung, & Choi, 2018).

Although a survey study by Guan et al., (2016) revealed that DNNs could outperform all other Machine Learning approaches, their sensitivity to adversarial attacks was their major weakness (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2017). Designing robust DNNs against

adversarial attacks (Chakraborty, Alam, Dey, Chattopadhyay, & Mukhopadhyay, 2018) would thus satisfy security issues while benefiting from their probabilistic-deterministic advantages.

1.2 Flight Dynamics Control (FDC)

After conflict detection relying on the predicted trajectory, in order to track a new safe trajectory, an appropriate Flight Dynamics Control (FDC) algorithm is needed. Basically, the problem of Flight Dynamics Control (FDC) for trajectory prediction (TP) can be categorized into five classes: Classical, Optimal, Adaptive, Robust, and Artificial Intelligence. (Karnopp, Margolis, & Rosenberg, 2012). These approaches are discussed in the next sub-section.

1.2.1 Classical Flight Dynamics Control

In the classical control theory, the Proportional Integrated Derivative (PID) controller is known as the generic standard controller for industrial purposes (Amoroso, Liverani, Francia, & Ceruti, 2021). This controller works by relying on its feedback signal; its aim is to reduce the error during stabilization and tracking by adjusting the control signal.

The PID is used for a wide range of UAVs (Unmanned Aerial Vehicles) as a flight dynamics controller (Kada & Ghazzawi, 2011), and is also employed for the flight dynamics stabilization of our UAS-S4 at LARCASE (M. A. J. Kuitche & Botez, 2019). While a PID controller can stabilize flight dynamics without needs of complex calculations to obtain the control signal, this stabilization requires significant control signal efforts. Moreover, it cannot handle well enough the parametric and non-parametric uncertainties well. These lacunae underscore the need for control approaches that can stabilize aircraft flight dynamics with the least control efforts, and further reinforce the need to design an optimal flight dynamics controller.

1.2.2 Optimal Flight Dynamics Control

Optimal control is a mathematical process that computes the control and state variables values of a dynamic system while minimizing a specific cost function (Lewis, Vrabie, & Syrmos, 2012). This cost function considers the energy consumed by the generation of the control signals required to approaches state variables to specific setpoint.

The Linear Quadratic Regulator (LQR) is known as the most user-friendly optimal approach and it has been used for many FDC cases. The LQR is able to respect a wide range of determined constraints, and it showed very good flight dynamics control efficiency when it was utilized for our UAS-S4 (Yañez-Badillo, Kuitche, & Botez, 2020). The LQR was then modified into the Linear Quadratic Gaussian (LQG) controller in order to solve linear time-varying FDC problems (Chrif & Kadda, 2014). However, in such cases, where function based on the state variables estimation, the LQR performance degrades, and worsens with increasing distance from the equilibrium point (Yit, Rajendran, & Wee, 2016). This issue can be solved through the Adaptive control approach (Åström & Wittenmark, 2013).

1.2.3 Adaptive Flight Dynamics Control

Adaptive Flight Dynamics Control (FDC) is based on a control strategy that tunes its gains in order to deal with varying parameters, model imperfection, and uncertainties (Landau, Lozano, M'Saad, & Karimi, 2011). This method can solve the problem with our designed UAS-S4 Linear Quadratic Regulator (LQR) as described in the previous subsection (1.2.2).

Linear and non-linear adaptive FDC strategies have been utilized for flight dynamics stabilization and tracking (Landau et al., 2011). The proposed linear adaptive control could handle small range of time-varying parameters. Hence, in (Xian, Wang, & Yang, 2019), a nonlinear adaptive controller was designed for a wide range time-varying parameters. In this way, they validated their proposed flight dynamics controller of the UAV that was transporting a varying payload.

Another study addressed a model-based adaptive controller that has shown great performance when it worked by relying on flight dynamics models (Whitehead & Bieniawski, 2010). But the Adaptive FDC was impacted by overreactions to small changes in varying parameters, and uncertainties. This is an important issue which can be solved by using a robust control approach (Ioannou & Sun, 2012).

1.2.4 Robust Flight Dynamics Control

Robust control is known as the best static approach for explicitly dealing with uncertainties. It can stabilize flight dynamics model while disturbances and modelling errors are taken into consideration (Chabir, Boukhniher, Bouteraa, Chaibet, & Ghommam, 2016).

A robust controller that can perform its control task while removing the adverse effects of external disturbances, such as wind shear, gust, and turbulence was designed and evaluated by (Mokhtari & Cherki, 2015). The robust control approach was utilized in (McEneaney & Fitzpatrick, 2002) for the UAV control, where it was designed to operate based on time-varying parameters and without access to high-order dynamics.

While the robust FDC approach could stabilize state variables regardless of parameter information, stability was only guaranteed for a limited range of uncertainties. A combination of robust and adaptive control approaches can thus offer much better performance compared to each individual control approach.

1.2.5 Artificial Intelligence Flight Dynamics Control

Fuzzy Logic is a fundamental Artificial Intelligence (AI) approach (De Silva, 2018) for critical control systems, such as those used in UAVs. It can efficiently solve mathematical complexities while dealing with nonlinearities and uncertainties (J. Lin, Zhou, Lu, Wang, & Yi, 2020). The use of fuzzy logic in FDC makes it possible to extract accurate inputs from the approximate ones by following an intuitive converting process (Babaei, Mortazavi, & Moradi, 2011).

The impressive advantages of the fuzzy logic approach were illustrated when it was employed for the UAV navigation and flight dynamics control with the aim of specific path tracking (Doitsidis, Valavanis, Tsourveloudis, & Kontitsis, 2004). Takagi-Sugeno models (Hušek & Narenathreyas, 2016) have provided a methodology that made possible the use of Fuzzy Logic Control (FLC) by relying on simplified FDMs. Despite its considerable abilities, its robustness was not ideal due to the imperfect fuzzy model produced after the fuzzification process. Therefore, a robust control strategy must be embedded inside the fuzzy logic controller.

Regarding the Flight Dynamics Control (FDC) investigation a model-based robust adaptive fuzzy approach can be the most proper control approach. This model-based controller can function very accurately if a precise flight dynamics model is provided. Therefore, in the following section strategies for Flight Dynamics Model (FDM) improvement are investigated.

1.3 Flight Dynamics Model (FDM) Improvement

The Flight Dynamics Controllers (FDCs) investigated in the previous section (1.2) rely on the UAV Flight Dynamics Model (FDM). Given the direct relationship between the UAV FDM accuracy and its model-based FDC performance, improving the UAV-FDM results in increasing the FDC efficiency in the closed-loop architecture (Hjalmarsson, Gevers, & De Bruyne, 1996).

1.3.1 Nonlinear Flight Dynamics Model

Basically, the UAV Flight Dynamics Model must be represented through its nonlinear differential equations in order to be the most accurate possible (Lone & Cooke, 2014). Nonlinear FDMs are extremely useful for flexible-structure aircraft (Palacios, Murua, & Cook, 2010; Shearer & Cesnik, 2007). On the other hand, Nonlinear Differential Equations (NDEs) increase the computational complexity for a model-based flight dynamics controller, and they can degrade the controller's time-domain properties in real-time operations (Stengel, 2015; Zúñiga, Souza, & Góes, 2020). In contrast, fixed structure aircraft can provide acceptable maneuvers using a linearized FDM; this linear FDM approach has been broadly used for fixed-wing UAVs (M. V. Cook, 2012).

1.3.2 Linearized FDM

Aircraft system nonlinear equations were determined, and their state equations were further linearized in (Duke, Antoniewicz, & Krambeer, 1988). Linearization about a specific equilibrium point allowed state and control matrices dimensional stability derivatives to provide a simplified flight dynamics model for the corresponding controller.

Linear FDMs have been used successfully for a variety of model-based FDCs. These FDCs include Proportional Integrated Derivative (PID) control (M. Liu, Egan, & Santoso, 2015), Adaptive control (Zhen, Tao, Xu, & Song, 2019), optimal Model Predictive Control (MPC) (Iskandarani, Givigi, Fusina, & Beaulieu, 2014), and Sliding Mode Control (SMC) (Bouadi, Bouchoucha, & Tadjine, 2007), which could all successfully perform the FDM stabilization task. The common drawback of these studies is their performance reduction with changing the flight conditions such as altitude, speed, and mass (Gaonkar & Peters, 1986) the need for FDMs that consider the effects of flight condition changes.

1.3.3 Local Linear Scheduled FDM

It is possible to estimate a nonlinear FDM using several local linear FDMs. This approach was first introduced in (Slotine & Li, 1991), where local linear FDMs corresponding to specific flight conditions were scheduled, and local controllers were designed. This flight dynamics control approach is known as “gain-scheduled control” (Shamma & Athans, 1990).

Gain-scheduled control was utilized at the LARCASE to obtain the UAS-S4 flight dynamics model. This Local Linear Scheduled Flight Dynamics Model (LLS-FDM) was then used for the optimal model-based controller (Yañez-Badillo et al., 2020). LLS-FDM performance reduces with increasing flight dynamics’ distances from their equilibrium points, highlighting the need of a regression algorithm to improve the flight dynamics model in flight conditions that become distant from equilibrium points.

1.3.4 Regressed Scheduled Local Linear FDMs

Fuzzy Logic methodologies can estimate a nonlinear FDM composed of the soft association of local linear models (Cheng, Rees, Cao, & Feng, 1996). For a smooth regression among local models, fine expert-designed membership functions are considered (Civanlar & Trussell, 1986). This approach can reduce the computational complexity in real-time operations (Albertos, Sala, & Olivares, 1998). The Takagi-Sugeno (TS) fuzzy logic approach could provide a nonlinear FDM using several linear state-space representations of local FDMs

(Mehran, 2008). The excellent accuracy of this approach is limited by the fuzzy logic rules for the design of the FDC.

An accurate nonlinear FDM can be obtained by relying on local linear FDMs if an appropriate regression algorithm is used. There are many regression algorithms, using simple interpolations to advanced Neural Network regression models (Chatterjee & Hadi, 2013). With the exponential growth of data acquisition equipment, data-driven regression algorithms can be utilized to improve Local Linear Scheduled Flight Dynamics Model (LLS-FDM) accuracy.

The weaknesses of various approaches to trajectory prediction, flight dynamics control, and flight dynamics model improvement were detailed in sections 1.1, 1.2, and 1.3, respectively. Chapter 2 introduces the research objectives and the approaches deployed to meet the research objectives, and then it highlights the original contributions of this research study.

CHAPTER 2

Research Objectives, Methodologies, and Contributions

2.1 Objectives, Methodologies, and Contributions

This research was conducted in three separate novel studies. Each study was organized based on its objectives, its methodologies, and its original contributions.

2.1.1 First Study – The Aircraft Trajectory Prediction (TP) Problem

2.1.1.1 Objective 1

This goal of this research study was to formulate the flight trajectory prediction as a regression problem. Neural-network models, from conventional to deep data-driven approaches, were utilized to solve the aircraft trajectory regression problem. The trajectory prediction models were designed to respect security issues. In other words, the prediction models should be robust against adversarial samples in case of adversarial attacks.

2.1.1.2 Methodology 1

To fulfill these aims, firstly, aircraft Trajectory Prediction (TP) was formulated as a time-series regression problem. Next, six data-driven models, Logistic Regression (LR), Support Vector Regression (SVR), Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM), were designed to perform accurate regressions. Adversarial samples were then generated through an Adapted Fast Gradient Sign Method (AFGSM) with the aim to evaluate the robustness of these designed TP models. The TP models were then retrained based on the Adversarial Retraining method so they could be robust against adversarial attacks.

2.1.1.3 Contributions

The original contributions of this study are summarized below:

- 1- The design of six original data-driven models for aircraft Trajectory Prediction.
- 2- The design of an original Adapted Fast Gradient Sign Method (AFGSM) to generate adversarial samples for time series regression problems.
- 3- The characterization of possible security challenges by imposing adversarial samples into TP models.
- 4- The design of a defense algorithm based on the Adversarial Retraining method to make the TP models robust against adversarial attacks.

The results from this first study confirmed the realization of its objectives and made original contributions. It was published in the Aerospace Journal.

Article 1: Hashemi, S. M., Botez, R. M., & Grigorie, T. L. (2020). New reliability studies of data-driven aircraft trajectory prediction. *Aerospace*, 7(10), 145.
DOI:10.3390/aerospace7100145

2.1.2 Second Study – The Design of a Flight Dynamics Controller (FDC)

2.1.2.1 Objective 2

The objective of the second study was the design of an intelligent model-based Flight Dynamics Controller (FDC). The designed controller should stabilize flight dynamics while considering aircraft system nonlinearities, imperfections. The FDC needs also to be enough robust in the presence of uncertainties and external disturbances such as wind sheer, gust, and turbulence. The effectiveness of this intelligent FDC was validated while it was implemented on the UAS-S4.

2.1.2.2 Methodology 2

In order to achieve the above objective, the UAS-S4 Local Linear Scheduled FDM (LLS-FDM) was considered for 9 altitudes, 5 masses, and 4 speeds. Next, the nonlinear FDM containing its time-varying mass was linearized around two equilibrium points. The nonlinear FDM was then represented with the Takagi-Sugeno (TS) Fuzzy Logic Model (FLM) by relying

on the soft association of local linear models. To design a model-based FDC design, an optimal reference model was determined and then stabilized by the Linear Quadratic Regulator (LQR) procedure to measure the reference model tracking error errors. Next, a Fuzzy Logic Controller (FLC) was designed, such that its adaptive gains could handle FDM nonlinearities. The FLC adaptive gains were thus robust enough in order to remove the adverse effects of imperfect fuzzified FDM and external disturbances.

2.1.2.3 Contribution 2

The novelties related to the second study are explicitly pointed out as the following:

- 1- The design of a customized Takagi-Sugeno Fuzzy Logic FDM for LARCASE's UAS-S4.
- 2- The redesign of adaptive gains inside the Fuzzy Logic FDC, thereby making the controller robust against external disturbances.
- 3- The proposal and application of a general Theorem (followed by its Lyapunov stability proof) with the aim to generalize the designed FDC for a variety of UAVs.

The novel contributions from the results of the second study were fully described in the second article, which will be soon published in the Aeronautical Journal.

Article 2: Hashemi, S., & Botez, R. (2022). Lyapunov-based Robust Adaptive Configuration of the UAS-S4 Flight Dynamics Fuzzy Controller. The Aeronautical Journal, 1-23. doi:10.1017/aer.2022.2.

2.1.3 Third Study – The UAS-S4 Flight Dynamics Model (FDM) Improvement

2.1.3.1 Objective 3

Improving the UAS-S4 FDM accuracy under marginal flight conditions is the third research objective. Marginal flight condition refers to the state between two trimmed flight dynamics models regarding scheduled altitude, mass, and speed. The Local Linear Scheduled LLS-FDM at LARCASE did not accurately represent the flight dynamics behaviour under marginal flight

conditions. In fact, our super-efficient Robust Adaptive Fuzzy Logic (RAFL) controller relied on the Fuzzy Logic Model which applied membership functions on the UAS-S4 LLS-FDM. That is why other model-based control approaches (Such as LQR) could not offer excellent performance under marginal flight conditions using LLS-FDM as discussed in Literature Review. In this way, a regression algorithm must be employed to accurately estimate an FDM in marginal flight conditions, and it can be useful for any model-based FDC approaches such as the LQR that has optimal purposes.

2.1.3.2 Methodology 3

To achieve the objective of FDM high accuracy under marginal conditions, local FDMs related to a sort of flight conditions were used to design a Local Linear Scheduling Flight Dynamics Model (LLS-FDM) for the UAS-S4 Ehecatl. The initial flight envelope that contained these local FDMs was then augmented based on interpolation and extrapolation strategies. In accordance with the augmented dataset, the Support Vector Machine (SVM) methodology was utilized as a benchmarking regression algorithm. Although the SVR could give excellent performance for the FDM regression and prediction, it remained sensitive against adversarial FDM samples. Hence, it was secured against adversarial attacks by means of an adversarial retraining defense algorithm. The robust SVR provided secured and precise FDMs for the UAS-S4 for its entire flight envelope.

2.1.3.3 Contribution 3

The novel achievements of the third study are summarized below:

- 1- A Data Augmentation method based on the k-nearest neighbours was developed in order to enlarge the UAS-S4 FDM dataset for the designed SVR algorithm.
- 2- A UAS-S4 FDM was regressed using the Support Vector Machine (SVM) methodology.
- 3- The UAS-S4 FDC was hacked by generating local adversarial FDM samples, and by imposing them into the UAS-S4 SVR-FDM.
- 4- An analysis of the impact of kernel function types on the FDM robustness against adversarial attacks was completed.

- 5- A defense algorithm was designed to make the UAS-S4 SVR-FDM robust in case of adversarial attacks.

The methodologies and results of this study related to the design of the SVR-FDM were presented (as an invited paper) and published at the American Institute of Aeronautics and Astronautics (AIAA), Sci-Tech 2022 Forum.

Conference Paper: Hashemi, S. M., & Botez, R. M. (2022). Support Vector Regression Application for the Flight Dynamics New Modelling of the UAS-S4. In *AIAA SCITECH 2022 Forum* (p. 2576).

The complete methodologies and results of this work related the robust-SVR against adversarial attacks were written in a paper submitted for publication in the SAE International Journal of Aerospace, Special Issue on Unmanned Aircraft Systems (UAS) and Autonomy.

Article 3: Hashemi, S. M., & Botez, R. M., A Novel Flight Dynamics Modeling using Robust Support Vector Regression against Adversarial Attacks.

This article was submitted for publication in *SAE International Journal of Aerospace*, On November 23, 2021

CHAPTER 3

New Reliability Studies of Data-Driven Aircraft Trajectory Prediction

Seyed Mohammad Hashemi ^a, Ruxandra Mihaela Botez ^b, and Lucian Teodor Grigorie ^c

^{a,b} Department of System Engineering, École de Technologie Supérieure, Université du Québec, 1100 Notre Dame West, Montréal, QC, H3C 1K3, Canada

^c Military Technical Academy Ferdinand I, 39-49 George Cosbuc, Bucharest, Romania

This article was published in the *Aerospace*, Vol. 7, No. 10, pp. 145, October 2020
DOI:10.3390/aerospace7100145

Résumé

Deux facteurs principaux, incluant la précision de la régression et la robustesse des attaques adverses, de six modèles de prédiction de trajectoire sont mesurés dans cet article à l'aide de l'ensemble de données publiques du système de gestion des flux de trafic (TFMS) des trajectoires des aéronefs à voilure fixe sur un itinéraire spécifique fourni par la Federal Aviation Administration. Six régresseurs basés sur les données ayant leurs architectures souhaitées, de l'apprentissage en profondeur conventionnel de base à l'apprentissage en profondeur avancé, sont explorés en termes de précision et de fiabilité de leurs trajectoires prédites. La principale contribution de l'article est que l'existence d'échantillons contradictoires a été caractérisée pour un problème de trajectoire d'avion, qui est redéfini comme une tâche de régression dans cet article. En d'autres termes, même si les algorithmes basés sur les données soient actuellement les meilleurs régresseurs, il est montré qu'ils peuvent être attaqués par des échantillons contradictoires. Les échantillons contradictoires sont similaires aux échantillons d'entraînement; cependant, ils peuvent causer des régresseurs finement entraînés à effectuer des prédictions incorrectes, ce qui pose un problème de sécurité pour les algorithmes de prédiction de trajectoire basés sur l'apprentissage. Il est montré que même si les algorithmes basés sur l'apprentissage en profondeur (par exemple, la mémoire à long et à court terme

(LSTM)) aient une précision de régression plus élevée par rapport aux classificateurs conventionnels (par exemple, la régression à vecteur de support (SVR), ils sont plus sensibles aux états spécialement conçus, qui peuvent être soigneusement manipulés pour rediriger leurs états prédits vers des états incorrects. Ce fait pose un réel problème de sécurité pour les aéronefs, car les attaques adverses peuvent entraîner des collisions intentionnelles et délibérément conçues de systèmes intégrés pouvant inclure tout type de prédicteur de trajectoire basé sur l'apprentissage.

Abstract

Two main factors, including regression accuracy and adversarial attack robustness, of six trajectory prediction models are measured in this paper using the traffic flow management system (TFMS) public dataset of fixed-wing aircraft trajectories in a specific route provided by the Federal Aviation Administration. Six data-driven regressors with their desired architectures, from basic conventional to advanced deep learning, are explored in terms of the accuracy and reliability of their predicted trajectories. The main contribution of the paper is that the existence of adversarial samples was characterized for an aircraft trajectory problem, which is recast as a regression task in this paper. In other words, although data-driven algorithms are currently the best regressors, it is shown that they can be attacked by adversarial samples. Adversarial samples are similar to training samples; however, they can cause finely trained regressors to make incorrect predictions, which poses a security concern for learning-based trajectory prediction algorithms. It is shown that although deep-learning-based algorithms (e.g., long short-term memory (LSTM)) have higher regression accuracy with respect to conventional classifiers (e.g., support vector regression (SVR)), they are more sensitive to crafted states, which can be carefully manipulated even to redirect their predicted states towards incorrect states. This fact poses a real security issue for aircraft as adversarial attacks can result in intentional and purposely designed collisions of built-in systems that can include any type of learning-based trajectory predictor.

3.1 Introduction

Avionics transportation standards and policies established by official agencies require all aviation companies to respect the approved safety protocols. These standards have been developed to ensure safe aircraft transportation, especially for modern automatic flights. Huge investments have been made in the United States over the last decades by the Federal Aviation Administration (FAA) into “The Next-Generation of Aerial Transportation” project, with the aim of increasing the safety and reliability of flights (Adesina, Adagunodo, Dong, & Qian, 2019).

Safety protocols are required for air traffic control, safe path definition, and collision avoidance, which determine conditions in which aircraft are allowed to fly, while safety policies reduce the chance of collisions. In this way, aircraft trajectory prediction (ATP) can be considered as an excellent tool for achieving safe aerial transportation. This prediction method may be used at different times, including for short-term and long-term predictions. Long-term prediction is useful for air traffic control, fuel consumption optimization, and logistics operations while short-term prediction is useful for conflict detection. The predicted trajectories may be utilized by ground computer units as part of an air traffic control system (ATCS) or by computer units of the aerial collision avoidance system (ACAS) in the cockpit.

Many aerial control tasks are processed by avionics systems. Such tasks might include aircraft trajectory optimization (Murrieta-Mendoza, Romain, & Botez, 2016) and its application into flight management systems (A. Murrieta-Mendoza & R. Botez, 2015), which aim to reduce operational costs (A. Murrieta-Mendoza & R. M. Botez, 2015), fuel consumption, and adverse environmental side effects (Dancila, Beulze, & Botez, 2016). A variety of algorithms, such as genetic algorithm (GA) (Patrón & Botez, 2015), particle swarm optimization (PSO) (Murrieta-Mendoza, Ruiz, Kessaci, & Botez, 2017), ant colony (Murrieta-Mendoza, Hamy, & Botez, 2017), bee colony (Murrieta-Mendoza, Botez, & Bunel, 2018), beam search (Murrieta-Mendoza, Ternisien, Beuze, & Botez, 2018), and harmony search (Ruby & Botez, 2016) have been employed to solve aircraft trajectory optimization problems. However, the main aim of avionics control systems is aerial collision avoidance (Nolan, 2011), where ATP contributes to solving encounter scenarios efficiently. This paper is focused on ATP accuracy and

reliability; the accuracy of predicted trajectories was assessed using the error rate in the test phase, and the reliability of ATP neural network models was evaluated based on the fooling rate for the adversarial attack. Evaluating the ACAS performance analysis based on data-driven trajectory predictors was not the aim of this article.

Generally, the predicted trajectory for each moving aircraft, produced by an algorithm, consists of a sequence of position states in the Cartesian space with their respective displacements from other sequences needed to prevent aircraft from colliding with each other. In cases when a safe zone constraint related to the predicted paths is violated, real-time adjustment is required from the prediction system in order to rearrange the aircraft position states (Gardner et al., 2016). In this type of setup, the computational complexity of the predictors is a key factor in providing a rapid and practical solution (Kochenderfer et al., 2012) as delays in aircraft equipped with aircraft trajectory prediction (ATP) systems can result in costly and mainly dangerous collisions. In this paper, a novel algorithm is proposed for real-time and accurate ATP in order to meet the high standards of a reliable control system.

Among all the algorithms developed for ATP, neural networks, especially deep learning approaches, have shown the most accurate performance if enough training data are provided. Many public trajectory datasets that are available online can be used for this aim. Deep learning (DL) models trained for path prediction purposes significantly outperform any other data-driven algorithms based on comparisons of runtime, from regression correctness to computational complexity. Unfortunately, recent studies have uncovered the vulnerability of all data-driven models, whereby some input samples can be purposely manipulated to mislead them (Jin, Li, Xu, Wang, & Tang, 2020). These fake samples are known as adversarial samples and, unfortunately, detection of fake sample intrusion is presently an ongoing problem for the machine learning community. In this paper, the existence and impacts of these samples are characterized in relation to ATP for both conventional regressors and cutting-edge deep learning models.

The organization of this paper is as follows. The common approaches developed for air vehicle trajectory prediction are reviewed in the following section. Brief explanations of data-driven predictors are provided in the Section 3.3. Section 3.4 is dedicated to our experimental results

and to a deep analysis of adversarial attacks on a variety of trained models. Finally, the related ongoing problems are elaborated for future works. For instance, we can characterize the existence of adversarial attacks for any learning-based algorithm while there is no certain systematic defense. Moreover, unfortunately, studies show that these adversarial samples are transferable from one model to another, even if they have been manipulated for other algorithms.

3.2 Related Works on Trajectory-Based Operations

Generally, trajectory-based operations are categorized as either short- or long-term predictions, whereby each prediction type has its own advantages relevant to the corresponding task. Figure 3.1 depicts the general setup of an aircraft encounter scenario, which could be visualized in short- and/or long-term prediction frameworks.

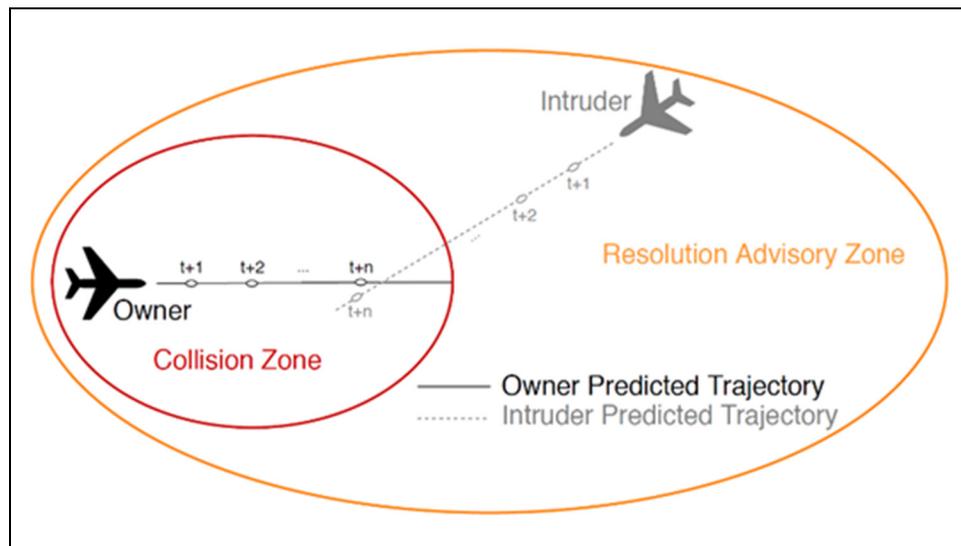


Figure 3.1 Encounter scenario

Although encounter scenarios, such as the one shown in Figure 3.1, have been solved using Traffic Collision Avoidance System (TCAS) without future trajectories, the collision avoidance task can be performed more optimally by relying on predicted trajectories. In fact, it is known that the TCAS modifies the owner's future trajectory if an intruder enters into the owner's resolution advisory zone. Moreover, the TCAS design is based on the current aircraft

position and on its conservative virtual unsafe zone. Hence, many false alarms and unnecessary resolution advisory events may occur during the flight. In this way, the collision zone can be reduced using an accurately predicted trajectory (position of aircraft in the n th step), which leads to avoiding unnecessary trajectory modification. Therefore, the design of a reliable and precise trajectory prediction algorithm is needed (Munoz, Narkawicz, & Chamberlain, 2013).

There is a large volume of research targeting these frameworks. Since the trajectory prediction could be formulated as a regression problem, these researches could be mainly employed for improving regression performance (J. K. Kuchar & Yang, 2000). In all these frameworks, the encounter scenario is defined based on owner and intruder attitudes (Munoz et al., 2013). The encounter scenario may occur due to the pilot mistake (A Ceruti, Bombardi, & Piancastelli, 2016), lack of visibility (Saggiani et al., 2007), actuator failure (Z. Yu et al., 2020), or loss of communications (Z.-q. Yu, Liu, Zhang, Qu, & Su, 2019). When an intruder arrives in the vicinity of the owner's neighborhood, after intruder detection (Alessandro Ceruti, Curatolo, Bevilacqua, & Marzocca, 2015), the ACAS resolution advisories commands should be transferred to the fixed-wing aircraft control system, which is supposed to deflect control surfaces with the aim to modify future trajectory. The control system of the owner aircraft (that is flying in a specific route) updates its subsequent trajectory with respect to the built-in regression model, while sensory radar information is being provided simultaneously (Julian, Kochenderfer, & Owen, 2019). Finally, the safety control system takes proper actions in order to avoid a possible collision (Guo, Yu, & Zhang, 2020).

3.2.1 Collision Avoidance

Model predictive control is an algorithm designed for trajectory prediction and path planning (Benavides, Kaneshige, Sharma, Panda, & Steglinski, 2014; Sahawneh & Beard, 2014). This strategy is used to model both the dynamics and the kinematics of a moving vehicle in order to predict the most appropriate trajectory to be followed. In contrast to this deterministic approach, a stochastic method is proposed in (Jilkov, Ledet, & Li, 2018), which implements the assembly of the multiple models to be tuned via optimization techniques. Since real-time optimization for prediction, even for a single model, is very costly, an adaptive control model that runs quadratic programming optimizers is developed in (Pereida & Schoellig, 2018).

In (M. Wang, Luo, & Walter, 2016), a nonlinear model predictive setup was proposed in an effort to solve multi-convex obstacles. A linear optimization algorithm was designed for the own aircraft model to avoid collisions with other aircraft models. A multiagent control policy for handling complex encounter scenarios is discussed in (Dai, Cao, Xia, & Gao, 2017). Since the agents were distributed, and the agreement of each agent was needed, the optimization problem was expensive. Instead, their optimization procedure generated more accurate position states that were followed by the aircraft.

Given that nonlinearities and uncertainties are involved in all these optimization problems, research efforts have been employed to develop evolutionary algorithms for fine path regression. Particle swarm optimization (PSO) has been adapted to generate 3D position states illustrated by B-spline curves (Foo et al., 2009). Genetic algorithm (GA) is a greedy-based evolutionary procedure that has been utilized for greedy regression in highly convex scenarios (Cobano et al., 2011). This algorithm incorporates only local nearby consistencies in its optimization routine with the aim of generating to-the-point states regardless of the entire path. In order to fully take advantage of the benefits of GA and PSO approaches, their combination has been proposed in (Duan et al., 2013), and it was demonstrated that the combined GA–PSO algorithm was able to outperform GA and PSO individually.

In addition to these greedy approaches, data-driven algorithms have been developed for trajectory predictions by rectifying the trajectories' local state shortcomings. For example, a neural-network-based clustering approach that implements an unsupervised learning process is discussed in (Barratt, Kochenderfer, & Boyd, 2018). In some research studies, deep neural networks have been utilized for two aims, firstly for safe zone clustering and, secondly, for correct position prediction of an aircraft over time (Andersson, Wzorek, & Doherty, 2017; Pham, Tran, Alam, Duong, & Delahaye, 2019). Deep reinforcement learning approaches have also been embedded into this setup, and promising results have been reported thus far (X. Wang et al., 2019).

3.2.2 Data-Driven Trajectory Prediction

A long-term aircraft trajectory is predicted using a trained hidden Markov model (HMM) (Ayhan & Samet, 2016) using 3D positional and, in addition, environmental data, which are considered as the fourth dimension of the dataset needed to consider weather uncertainties. That work divides the whole path into small patches of 3D cubes and then predicts the future trajectory under real flight conditions. Similarly, a long-term four-dimensional (4D) aircraft trajectory has been predicted using a deep generative neural network architecture modeled in the presence of uncertainties, such as wind, convective weather, and temperature (Y. Liu & Hansen, 2018).

In (Wu et al., 2017), aircraft trajectory prediction is considered as a flight sequence estimation problem. That work proposes a recurrent neural network for trajectory prediction. The results reveal noticeable improvements in state predictions. Following this idea, a long short-term memory (LSTM) algorithm has been developed which outperformed its predecessor (Park et al., 2018). Although a comparative study conducted in (Guan, Lv, Sun, & Liu, 2016) showed that deep learning algorithms outperform all other machine learning approaches, a variety of their models were implemented to further investigate their prediction capability, as well as their vulnerability to adversarial attacks.

Overall, the trajectory may be predicted using conventional approaches (e.g., model predictive control (MPC)) based on aircraft dynamics models or modern data-driven techniques (e.g., deep neural network (DNN)) that rely on large amounts of recorded data. Studies have shown that modern data-driven techniques outperform conventional approaches if enough training data is available and security issues are respected. It is known that in conventional approaches, uncertainties backpropagate through the prediction horizon, and errors increase dramatically. Hence, the data-driven algorithms were adopted for trajectory prediction, and the ATP task was performed regardless of aircraft dynamics models, which is a remarkable advantage of data-driven predictors. A carefully tuned and real-time predictable path is therefore required for each aircraft. Since data-driven algorithms have been used for path prediction, they have been found to be not completely fault-tolerant, and they may create security issues for aviation transportation systems. In the following section, some of our benchmarking algorithms and

datasets are explained. Then, we will explain how the adversarial samples can be generalized to models being trained using standard aircraft trajectory datasets.

3.3 Building Data-Driven Predictors

Data-driven predictors have shown great performance in all regression tasks, which is also shown in our present study. Therefore, several different learning-based algorithms are explored for solving the aircraft trajectory regression problem (ATRP). The benchmarking algorithms that we will propose range from conventional (e.g., logistic regression) to state-of-the-art (e.g., convolutional neural network). The performance of these algorithms is totally dependent on the characteristics of the given dataset and on its sample distributions, in which sampling distribution is defined as a probability distribution of a statistic that is derived from a considered population. Since there is no practical approach to define the best regression algorithm for our dataset, conducting experiments on all of them to determine the most proper one is needed. Although, nowadays, deep learning-based approaches (such as CNN, LSTM) are the best performing algorithms, there is no guarantee of outperforming conventional algorithms, such as support vector regression (SVR). Due to these reasons, six regression algorithms have been included in our study. These regression algorithms are logistic regression (LR), support vector regression (SVR), deep neural network (DNN), convolutional neural network (CNN), recurrent CNN (RNN) and, finally, long short-term memory (LSTM). Our motivation for utilizing all these algorithms is to measure and compare the strength, generalizability, and robustness of these models. Brief explanations are provided for each of these algorithms in the following subsections.

3.3.1 Logistic Regression (LR)

Logistic regression has the potential to fit its results to the training data if the uniformity of the given dataset is standard and without fluctuations. Since our experimental dataset is “evenly” distributed over time, it does not contain noticeable fluctuations and, thus, LR can learn from the finely tuned mentioned dataset finely and make accurate predictions (Ter Braak & Looman, 1986).

By assuming that the given inputs and outputs to the algorithm are $X = [x_1 \ x_2 \ \dots \ x_n]^T$ and $Y = [y_1 \ y_2 \ \dots \ y_n]^T$, respectively, Equation (3.1) is considered for LR model learning (Grimm & Yarnold, 1995):

$$Y = \frac{1}{\exp(\theta_0 - \theta_1 x_1 - \theta_2 x_2 - \dots - \theta_n x_n)} \quad (3.1)$$

where θ is the weight vector that could be obtained during training by optimization using a relevant cost function $J(\theta)$. Conventionally, the cost function is defined in Equation (3.2) (Friedman, Hastie, & Tibshirani, 2001).

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m [-y_i \log(h_\theta(x_i)) - (1 - y_i) \log(1 - h_\theta(x_i))] \quad (3.2)$$

where the number of samples is denoted by m in the training set, and $h_\theta(X)$, known as the hypothesis, is defined in Equation (3.3) (Allison, 2012).

$$h_\theta(X) = \frac{1}{\exp(-\theta X^T)} \quad (3.3)$$

where $\theta \in \{\theta_i\}$. One of the crucial observations is that the logistic function θ considered in the above equations increases the risk of saturation during the training phase; the regularization term is added, as shown in Equation (3.4), to rectify this problem (Friedman et al., 2001):

$$\frac{\lambda}{2m} \sum_{j=1}^n (\theta_j)^2 \quad (3.4)$$

where λ the regularization term that binds the cost function given by Equation (3.3) to more parameters shown in Equation (3.4) in order to improve the model's precision. λ should be manually tuned with respect to the training statistics. Training statistics refer to weight vectors obtained while running an iterative process for learning, in which their fine tuning increases the chance of obtaining better weight vectors. The addition of this term to Equation (3.2) contributes to avoiding overfitting of the dataset and the need to memorize samples. We trained

this LR model on some standard aircraft trajectory datasets and fine-tuned its hyperparameters. The basic problem of this regressor is its generalization to complicated patterns, which could be challenging for the LR model to learn. Hence, support vectors are used to capture data distribution better than the LR, even in cases when the training data are not linearly separable.

3.3.2 Support Vector Regression (SVR)

This conventional regressor is based on the well-known principle of support vector machines, which is capable of learning from high-dimensional spaces. The concept supported by the SVR is the mapping of training data from the Euclidean space to another higher dimension space by using the “kernel trick”, then the learning of the decision boundaries. There are many kernel functions that could implement this mapping, such as homogeneous/inhomogeneous polynomials, tanh, Gaussian, and others. Different experiments were performed by us on these kernels in order to determine and adopt the best ones. The optimization process employed for our SVR model is given in Equation (3.5) (D.-R. Chen, Wu, Ying, & Zhou, 2004):

$$\min \frac{1}{2} \|\theta\|^2 \quad s. t. \begin{cases} \theta_i x_i + b - y_i \leq \epsilon \\ y_i - \theta_i x_i - b \leq \epsilon \end{cases} \quad (3.5)$$

where ϵ denotes the decision boundary precision which should be tuned carefully. The performance of the SVR model as well as the performance of any other learning-based algorithm is totally dependent on the type of regression task and on the dataset used for training. Moreover, SVR learns from a mapped subspace, which could be very challenging. To address this potential problem, some other algorithms that can learn from raw samples are used. The state-of-the-art of these algorithms will be reviewed in the following subsections.

3.3.3 Deep Neural Network (DNN)

Neural network algorithms have been implemented for many regression tasks. It has been shown that multilayer perceptrons (MLPs) can produce accurate models for any regression problem if enough training samples are provided (Rocha, Cortez, & Neves, 2007). With the advancement of deep neural networks, many interesting architectures have been introduced, outperforming MLPs. These algorithms learn from raw data, and can be used to solve time

series problems (Qiu, Zhang, Ren, Suganthan, & Amaratunga, 2014), such as aircraft trajectory prediction.

Unlike conventional data-driven models, modern DNNs learn from training sample distributions with any dimensionality; sometimes, dimension conversion has to be conducted with respect to the complexity of the regression task. This fact means that learning-based algorithms can be categorized into feature-based (conventional algorithms such as SVR, LR) and raw inputs (modern deep learning algorithms, such as CNN, LSTM). The latter category does not need to be provided by handcrafted feature vectors, but they need more training samples than conventional algorithms. Otherwise, their performance may decrease. It is important to have a large enough dataset for training deep learning algorithms. When there is no access to a large dataset, transferring of dimensions can be applied to enhance sample distributions in order to improve the algorithm performance. To some extent, DNNs can be sensitive to the volume of the training set, and their performance may degrade if the training dataset is not large enough. To rectify this issue, several data augmentation algorithms have been proposed (Van Dyk & Meng, 2001).

Similarly to MLPs, input, hidden, and output layers are the main components of DNNs. New proposed architectures for DNNs include very dense hidden layers with a massive number of filters. AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), GoogLeNet (Szegedy et al., 2015), and ResNet (He, Zhang, Ren, & Sun, 2016) are some of the modern architectures proposed for DNNs.

Cutting-edge DNN architectures consist of very deep hidden layers, but they also take advantage of modern blocks in their hidden layers, such as dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), rectified nonlinear activation functions (Glorot, Bordes, & Bengio, 2011), and optimized cost functions with momentum and adaptive learning rates (Zeiler, 2012). “Dropout” is a regularization technique for training a neural network. It randomly freezes some weight vectors in the training process and avoids updating them to the end of the ongoing epoch, which boosts the training performance especially for very dense CNNs. Rectified nonlinear activation function is a discrete activation function

including two linear functions. Mathematically, $ReLU(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$. It has been demonstrated that it outperforms the traditional sigmoid function in neural network training. Momentum and adaptive learning rate tune the training cost function with slightly perturbation weight vectors toward the maximum variations direction.

The abovementioned DNN architectures have been developed for complex computer vision applications, and they are not fully compatible with the aircraft trajectory regression problem (ATRP). Therefore, we propose our DNN architecture, adapted to our dataset as shown in Figure 3.2.

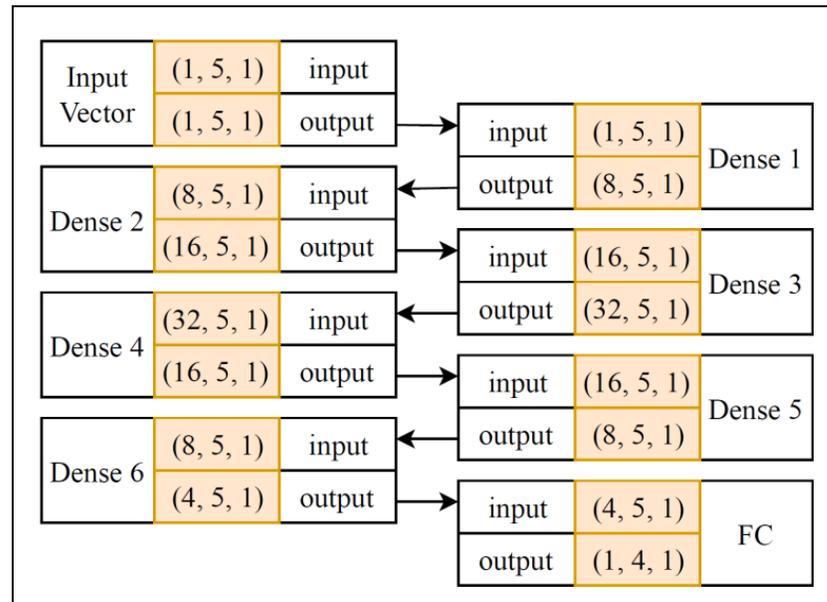


Figure 3.2 Proposed deep neural network (DNN) architecture for the aircraft trajectory regression problem (ATRP).

This figure shows the architecture of our proposed DNN. There are three types of blocks, namely the input vector (input layer), dense (hidden layer), and fully connected (FC). The dense and FC layers are the same, but the latter is not connected to any other layers after it. The highlighted parts of each block specify its input and output dimensions as well as the number of trained filters (weight vectors per layer) shown in triplet of (number of filters, input dimensions, output dimensions). For instance, by considering the input of Dense 2, triplet of

(8,5,1) means that there are eight filters in this layer, and the dimension of the input vector is 5×1 . Moreover, the connection between layers is shown with oriented arrows.

The input layer in this architecture is a $1 \times 5 \times 1$ tuple consisting of one filter. Filter dimension is defined based on the input dataset which consists of five recorded measured parameters represented as [latitude, longitude, altitude, velocity, time] $_{5 \times 1}^T$. “Hidden layers” are shown as five dense layers that are fully connected to the next layer in order to produce outputs as [latitude, longitude, altitude, time] $_{4 \times 1}^T$. Except for the latter layer, all other layers include batch normalization (Ioffe & Szegedy, 2015) with normal distribution, dropout with a 0.5 ratio, and rectified linear unit (*ReLU*) activation function as a nonlinearity.

Unfortunately, there is no deterministic approach for designing DNN architecture that is obtained after running several exploratory experiments and after achieving the desired DNN performance in terms of regression accuracy or error rate. In order to avoid overfitting our model to the training set, “early stopping” (Prechelt, 1998) was used to achieve the highest regression accuracy while keeping it still generalizable.

Although DNN filters are capable of learning very complex sample distributions, incorporating convolution layers can noticeably improve model performance. These layers could be added with/without dense hidden layers, and they could be revealed by running experiments on the given dataset. In the next subsection, our desired CNN architecture for the ATRP is presented.

3.3.4 Convolutional Neural Network (CNN)

Assuming that a random function $g_i(\theta)$ and an input sample $x_i \in \{X\}$ are given, then their convolution, $g_i \circledast x_i$ for $i \in \{1, 2, 3, \dots, n\}$ will give a “convolution filter” if Equation (3.6) approaches to zero (Kullback & Leibler, 1951):

$$d_{KL}(d_g \parallel d_x) = - \sum_{i=1}^m d_g \log \left(\frac{d_g}{d_x} \right) = 0 \quad (3.6)$$

where d_{KL} denotes the Kullback–Leibler divergence of filter distributions d_g and input distributions d_x .

CNNs use the backpropagation technique for their cost function optimization, and the filter size totally depends on the dataset features. For the ATRP, $g_i(\theta)$ with dimensions of 5×1 is suggested, as well as our DNN filter size. Our proposed CNN architecture, adapted to the ATRP upon conducting various experiments, is shown in Figure 3.3, which has only one difference with respect to Figure 3.2. In this architecture, there are convolution layers followed by a max-pooling operation which interpolates the dimensions of the outputs to half of it for each dense layer. This operation reduces potential noise in the input vectors.

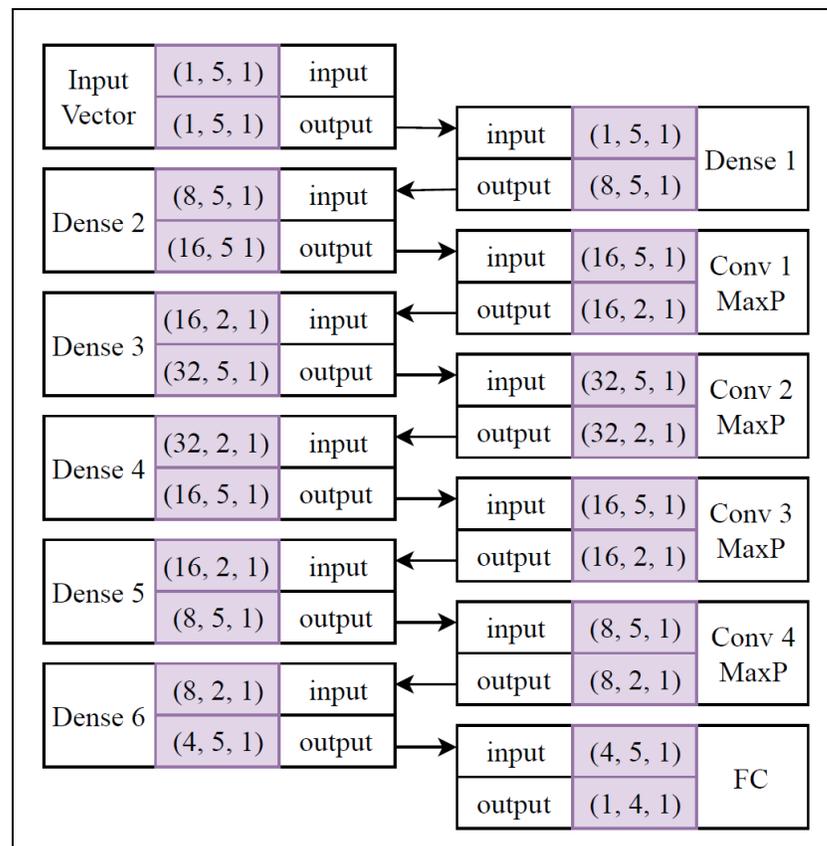


Figure 3.3 Proposed convolutional neural network (CNN) architecture for the ATRP

The initialization scheme, batch normalization, regularization, and dropout ratio have been set to be the same as the ones of our DNN architecture, with one exception: the inclusion of a max-pooling (MaxP) operation for dimension reduction and noise removal purposes. Max-pooling is a post-processing operation that usually comes after convolution layer operation, which shrinks the input dimensions by half. It has been shown that these operations reduce potential noise in the input vectors. The ratio of MaxP is chosen to be 0.5 in order to reduce the input sequence by half, which is a default value for all the deep learning packages.

Although this proposed CNN architecture outperforms the aforementioned DNN, it is still not appropriate for our regression task. Therefore, this architecture is extended to include some recurrent blocks, thus aiming to improve the characterization of input sample distributions.

3.3.5 Recurrent CNN (RNN)

Recurrent neural networks (RNN) are versions of CNNs/RNNs developed for complex input streams (input data distributed over time characterized by strong dependency between consecutive vectors) as their current states are dependent upon their previous and subsequent states. In fact, multiple feedbacks among the layers are needed to maintain the dependency of distributions. For this type of dataset, a recurrent neural network (RNN) may outperform common CNN. RNN implements transitions between consecutive input vectors, which are distributed over time using connected states. Each state is similar to a hidden layer in a typical CNN. Connection between states could be bijective from one state to another, which is called “feedback”. RNN states and feedbacks can extract the input vector dependence on time.

Since samples in our dataset are distributed over time, a suitable RNN architecture is proposed for the ATRP, as shown in Figure 3.4. The interpretation of this architecture is as same as the one shown as Figure 3.3 with one difference. In this architecture, feedbacks from one state (layer) to another state have been shown by dotted arrows. For example, gradient information which has been computed for states of Dense 5 will be transferred to states of Dense 3, and affect its weight vectors. This setup tracks temporal dependency in a sequence of input streams.

Normally feedback from the last convolutional layer to the first dense layer provides the least gradient vanishing (Conv 4 to Dense1). Consecutive feedbacks in the last layers are normally executed in order to avoid gradient saturation (Dense 6 to Dense 4, and Dense 5 to Dense 3).

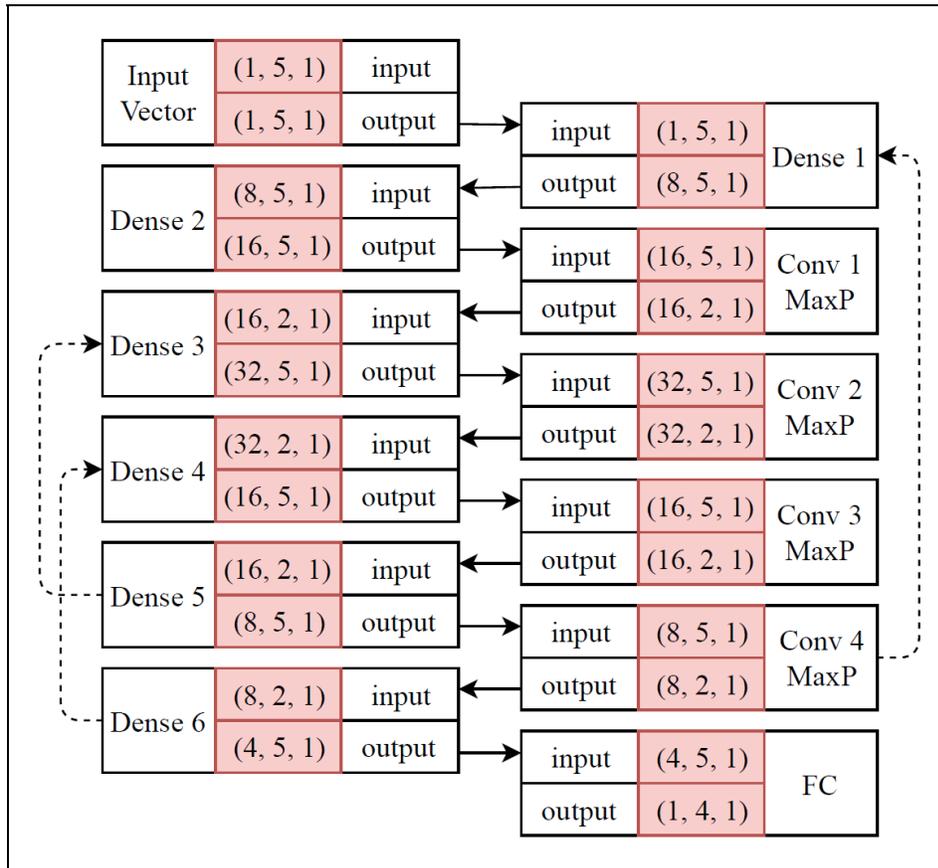


Figure 3.4 Proposed recurrent CNN (RNN) architecture for the ATRP

This architecture is similar to our designed CNN-based architecture. Hidden layers have been empirically designed to achieve competitive performance with our CNN. The activation function used for this RNN is “tanh with a random bias vector”. Kernel, bias, and recurrent initializers have been set to a truncated normal distribution of samples with $\mu = 0.5$ and $std = 0.5$. The constraints applied to the recurrent blocks are max-norm, while there are no constraints defined for kernel or bias. As explained earlier, the dropout is a regularization scheme which randomly freezes some weight vectors from their updates. The dropout ratio identifies the probability of randomly selection of neurons. When this dropout ratio is set to

0.5, then it means that there is a 50% chance for every neuron to be frozen in each epoch. In this paper, a dropout ratio of 0.5 was considered for all the layers. There are three recurrent blocks in our proposed RNN.

Since there is no optimal approach in order to automatize this process, different feedbacks have been tested for hidden layers. While designing the architecture for our RNN, consecutive feedbacks (from one state to another) were discovered from one hidden layer to another in addition to gradient saturation to memorize their dependencies among samples, in which the generalizability of the model was negatively affected. In other words, the chance of overtraining of RNN is very high, which highly depends on the number of feedbacks among hidden layers. If this setup is not tuned properly, the gradient information might be saturated, and the training track would be lost. Therefore, a feedback was set up from the sixth dense layer to the fourth and from the fifth to the third dense layer.

Since the dimensionality of our input training data is low, 5×1 , a connection is set from the fourth convolution layer to the first dense layer in order to rectify the gradient vanishing problem. “Gradient vanishing” refers to any operation which may give a “zero” value to the gradient information. If gradient vectors vanish, then no weight vector can be updated. As shown in our previous networks, the last layer of the RNN’s final mapping is fully connected to the output.

Although RNNs are very much qualified for time-distributed feature learning, some short-term dependencies of the input vector among its measured states may be lost within training. Short-term dependency can be expressed by the relation between velocity and acceleration, or velocity and displacement. Therefore, a long short-term memory (LSTM) algorithm is implemented to solve the problem discussed here.

3.3.6 Long Short-Term Memory (LSTM)

Currently, the use of LSTM algorithms represents the cutting-edge data-driven approach for classification tasks as they are conveniently generalizable for regression problems. LSTMs incorporate three major gates: input, output, and forget (Hochreiter & Schmidhuber, 1997).

The LSTM relying on cooperation of these blocks can temporarily remember some information about previously input vectors. The forget gate is used for tracking similar patterns over time/sequence. The schematic of our LSTM adapted to our regression task is depicted in Figure 3.5.

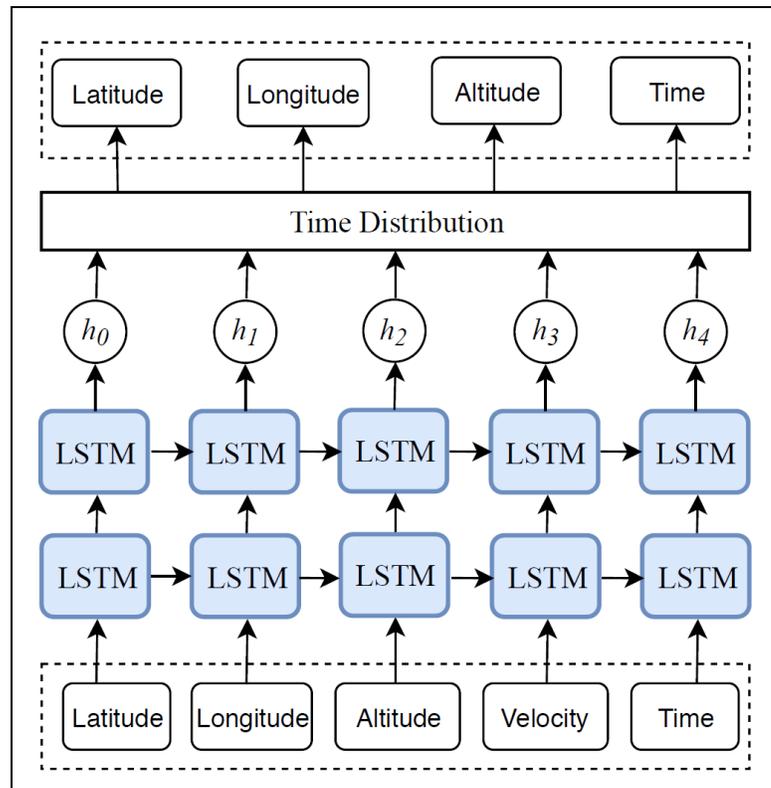


Figure 3.5 Proposed long short-term memory (LSTM) architecture for the ATRP

Dimensions of our LSTM input are 5×1 for a given number of five measured parameters corresponding to the latitude, longitude, altitude, speed, and time for input vector. The dimensions of our LSTM output are 4×1 , and include four predicted parameters corresponding to the latitude, longitude, altitude, and time for output vector. Gates have been represented by blue blocks followed by hyperbolic operations of h . This module tunes the timing of output vectors derived from the states.

As seen in Figure 3.5, the conducted LSTM architecture includes the input unit followed by a hyperbolic tangent (\tanh) activation function. For recurrent activation, a “hard sigmoid”

(between LSTM blocks) is used, through the linear activation function works fine as well. Kernel, recurrent, and bias initializers are chosen to be “random uniform functions” with “min” and “max” values equal to 0.05. For all these modules, L_2 similarity metric (Euclidean distance) is embedded with regularization constant $\lambda = 0.01$. For simplicity of calculations, we have not taken any kernel, bias, or recurrent constraints into account. Moreover, the dropout ratio for all the layers was tentatively set to 0.5.

Basically, sensitivity analysis is useful for LSTMs aimed at classification tasks. Hence, for this regression problem no sensitivity analysis was conducted for this LSTM architecture. We have adopted this network due to its very good regression accuracy with respect to the accuracy of previous deep networks.

3.4 Numerical Results

In this section, a brief explanation is provided regarding the dataset used and utilization procedure, and then the prediction results of several models are discussed.

3.4.1 Dataset

The benchmarking dataset for conducting the proposed research is the traffic flow management system publicly available online for educational use. Each record of this dataset contains latitude, longitude, altitude, velocity, and time obtained from 1676 flights (Aqib et al., 2019).

3.4.2 Measuring the Resiliency of Models

As stated earlier, all six employed algorithms were trained with the maximum generalizability possible for our dataset. We implemented a 10-fold cross validation (Wiens, Dale, Boyce, & Kershaw, 2008) for all these models and then controlled their comprehensiveness by using an early stopping technique. Experiments were further performed with the aim to determine the extent to which these models could resist given perturbations and random noise.

We assume that the trained model was built, including post-activation operations, on the training set of $x_i \in \{X\}$. The following optimization problem was further solved:

$$\min_{\epsilon} \|x - \tilde{x}\| \leq \epsilon \quad s. t. \quad f(x_i) \neq f(\tilde{x}) \quad (3.7)$$

In general, this optimization problem is known as an ‘‘adversarial attack’’ (Goodfellow, Shlens, & Szegedy, 2014), which results in producing samples similar to the original samples but which might lead to mistakes in the model and, thus, its correction is needed.

Although classification and regression tasks are similar to each other, Equation (3.7) should be updated to non-label values for regression problems. In fact, unlike classification, there is no label for input vectors in the regression task. Therefore, to justify adversarial optimization problem we would need to replace the ‘‘label’’ with a ‘‘threshold’’ and solve for it. To achieve the minimum perturbation of ϵ , the following optimization statement is suggested:

$$\min_{\epsilon} \|x - \tilde{x}\| \leq \epsilon \quad s. t. \quad \min_{\delta} \|f(x_i) - f(\tilde{x})\| \geq \delta. \quad (3.8)$$

Optimizing for ϵ and δ generates a series of samples that are remarkably similar to the legitimate inputs, but they are totally different to their associated outputs. In other words, after running the optimization inequality as defined in Equation (3.8), the manipulated input, \tilde{x} , is similar to the given legitimate input x although their associated output vectors are not similar. This optimization problem could be developed to include certain conditions, namely by redirecting the $f(\tilde{x})$ towards a predefined or random value, which can identify a targeted attack. This condition could add overhead to our abovementioned optimization problem and, therefore, we do not analyze it in the current paper. In our future studies, we will study possible approaches for the defense of our developed prediction models against adversarial attacks.

Having access to the training set, parameters, and hyperparameters of the trained model constitute a white-box attack, although it would still be possible to attack even without them. Both white- and black-box attacks are explained next.

The architectures and training setups of all six models were all the same in this paper, as explained earlier. For the training data with columns of latitude, longitude, altitude, time, and speed, the models were finely trained to predict their future states (latitude, longitude, altitude,

time). The given input sample x_i was randomly perturbed while keeping it close to its associated original value by using an L_2 similarity metrics.

There is no generic approach to define the exact values for these hyperparameters. We empirically obtained these values and they can be changed following the adversary's suggestions. Here, the initial values assigned to ϵ and δ are 0.01 and 100, respectively. Table 3.1 summarizes the values of ϵ and δ achieved for all models trained on the traffic flow management system (TFMS) public dataset of aircraft trajectories.

Table 3.1 Mean values of ϵ and δ for training samples of the TFMS dataset

	LR	SVR	DNN	CNN	RNN	LSTM
ϵ	0.0103	0.0174	0.0139	0.0165	0.0237	0.0142
δ	59	126	207	67	106	92

Table 3.1 compares ϵ and δ values found by use of six benchmarking regression algorithms. Basically, adoption of smaller values for ϵ results in higher similarity between generated adversarial samples and their associated legitimate samples. Additionally, adoption higher values for δ leads to higher discrepancies between the ground-truth and the predicted outputs. Ground-truth is defined for supervised learning methods in order to measure the accuracy of the training set. Among these models, higher values for δ were achieved using DNN, which means this model yields higher variation in its predictions for legitimate inputs.

We generated adversarial samples for all the records of the dataset and we tested them by using of all the trained models. Interestingly, by applying these samples, all models predicted incorrectly.

Table 3.2 lists the fooling rates of all six models with their prediction confidence scores. This table compares fooling rates of six victim models against adversarial attacks that were generated by FGSM algorithm.

Table 3.2 Fooling rate and prediction confidence of the models

	LR	SVR	DNN	CNN	RNN	LSTM
Fooling rate	100	100	100	100	100	100
Prediction confidence score	0.784	0.843	0.732	0.879	0.910	0.881

Unfortunately, all these models were completely vulnerable against adversarial samples. The results shown in Table 3.2 clearly restate a security concern regarding the robustness of the data-driven models, including the conventional and advanced deep learning architectures. Scaled values of prediction confidence reveal the weakness of each model in terms of its prediction. The main difference between these algorithms is their prediction confidence. Apparently, RNN predicted wrongly with the highest confidence.

Another important concern is the transferability of the generated fake samples from one model to another. To evaluate this situation, adversarial samples were crafted for each model, and were feed-forwarded to another model. The results of this experiment are shown in Table 3.3. This table statistically explains the transferability property of adversarial samples.

Table 3.3 Transferability of adversarial samples from one model to another model

	LR	SVR	DNN	CNN	RNN	LSTM
LR	100	78.36	84.14	91.23	89.66	91.17
SVR	81.23	100	84.17	95.07	84.56	89.59
DNN	90.07	89.23	100	95.81	97.33	94.46
CNN	86.75	88.71	91.63	100	91.55	93.57
RNN	97.29	94.58	90.67	95.58	100	98.26
LSTM	79.16	81.92	89.99	93.52	88.37	100

This table depicts the transferability of adversarial samples from one victim model to another. Reported percentage values are averaged among all 10 folds, which is equivalent to say that the given dataset was divided into 10 equal-size segments versus time and, thus, each one of them was considered a test segment. Finally, the average of accuracy was we computed for

these segments. The most transferable adversarial samples for each model are shown in Table 3.3 in bold characters. For instance, 81.23% of total crafted adversarial samples for SVR are successfully transferable to the LR model.

Although LSTM is more advanced than the RNN, it is more vulnerable to transferred adversarial attacks. Equation (3.8) is further explored for a better understanding of crafted samples. A first impression could be that adversarial samples are “noises”. To accept or reject this impression, we need to run experiments to determine if ϵ and δ constitute “noise” (or not).

To answer the abovementioned question, we utilized the local intrinsic dimensionality (LID) score (Ma et al., 2018). This score differentiates “noisy samples” from “crafted adversarial samples”. Assuming that $d_i(x)$ refers to the distance from legitimate sample x_i to its nearest neighbors, $d_k(x)$, then the maximum of the neighbor distances can be found in which k is the number of neighbor samples. Therefore, the LID score can be computed as shown in Equation (3.9).

$$LID(x) = - \left(\frac{1}{k} \sum_{i=1}^k \log \frac{d_i(x)}{d_k(x)} \right)^{-1} \quad (3.9)$$

Around 15% of the training set and generated random noisy samples were randomly selected using Gaussian distribution with 10 different values of $\mu \in [-1, 1]$ and $\sigma \in [-0.75, 0.75]$. For fairness comparison, we repeated this generation 10 times and exported all the generated noisy samples into the original dataset by building a new directory to include both noisy and legitimate samples. We also generated new adversarial samples for every record in the original training set and further exported them into the adversarial category. Eventually, a logistic regression algorithm is trained for two considered classes in order to classify legitimate and adversarial samples. Table 3.4 summarizes the details of this binary classification.

Table 3.4 primarily compares the accuracy of LR on the LID scores as well its setups for training. For example, the first row of this table shows that LR without cross validation has 86.36% and 84.27% accuracy in training and test, respectively. These accuracies have been achieved at the 120th iteration with L_2 regularization penalty and with a prediction tolerance

(error) of $1e^{-5}$. Training has been executed using four CPU core (jobs) without weight normalization (false fitting intercept). The inverse of the regularization strength (C) for this model is set to 0.002.

Table 3.4 Performance comparison of logistic regression (LR) on local intrinsic dimensionality (LID) scores. The solver for this LR classifier is “liblinear”

	Max Iteration	Training Accuracy (%)	Test Accuracy (%)	Penalty	Tolerance	Fitting Intercept	# Jobs	C
Without cross-validation	120	86.36	84.27	L_2	$1e^{-5}$	False	4	0.002
5-fold cross-validation	100	91.23	87.75	L_2	$1e^{-5}$	False	4	0.001
10-fold cross-validation	95	92.13	86.49	L_1	$1e^{-6}$	True	8	0.003
15-fold cross-validation	85	92.67	86.18	L_1	$1e^{-6}$	True	8	0.002

As shown in Table 3.4, the LR is favorably used for the binary classes of the LID scores, and it supports our previous hypothesis (can adversarial samples be interpreted as noisy samples or not?) regarding the fundamental difference between noisy and adversarial samples. For a very good characterization of the distribution values of the original, noisy, and adversarial samples, we plotted their LID scores in Cartesian space. Please note that LID is a score given to every input. Figure 3.6 visually shows distribution of LID scores for triplet of original, noisy, and adversarial samples.

Figure 3.6 shows the LID score comparisons for random samples chosen from the training set. As this figure indicates, original and noisy samples lie in the same LID subspace, which denotes their structural similarity. Conversely, adversarial samples are located in a separated upper subspace different from the original and noisy sets. To demonstrate that these LID scores were also statistically different, we trained an LR in order to classify LID scores of original,

noisy, and adversarial samples. Obviously, higher values of accuracy of the trained LR mean better classification for LIDs.

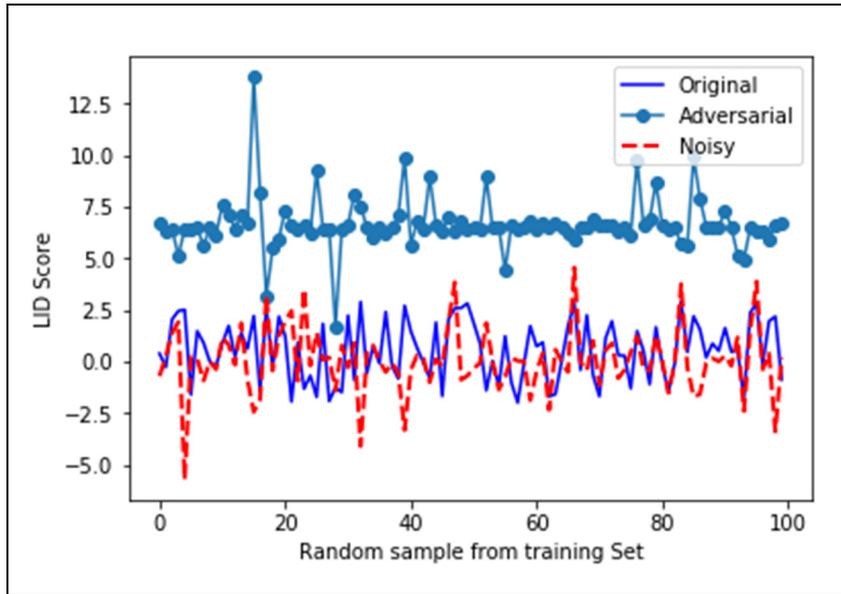


Figure 3.6 (a) The LID score for the 1st random set

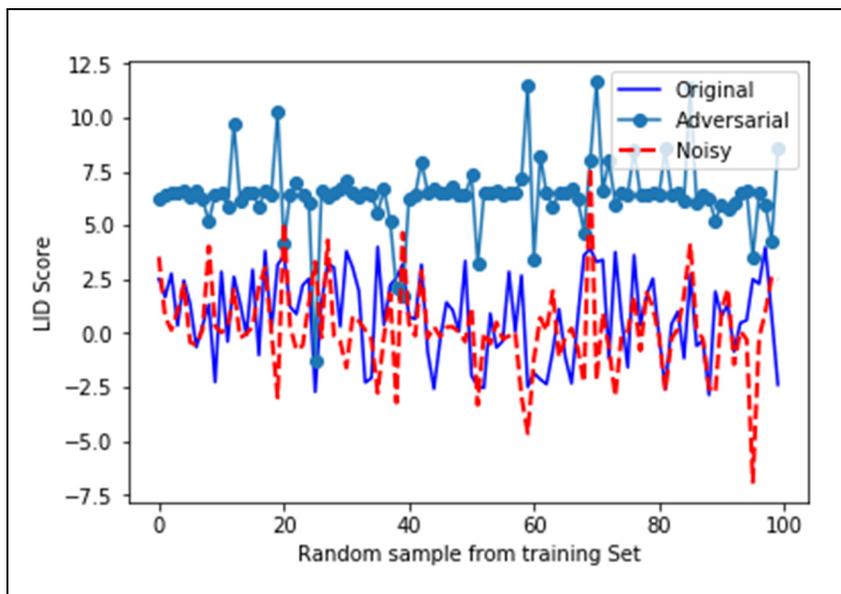


Figure 3.6 (b) The LID score for the 2nd random set

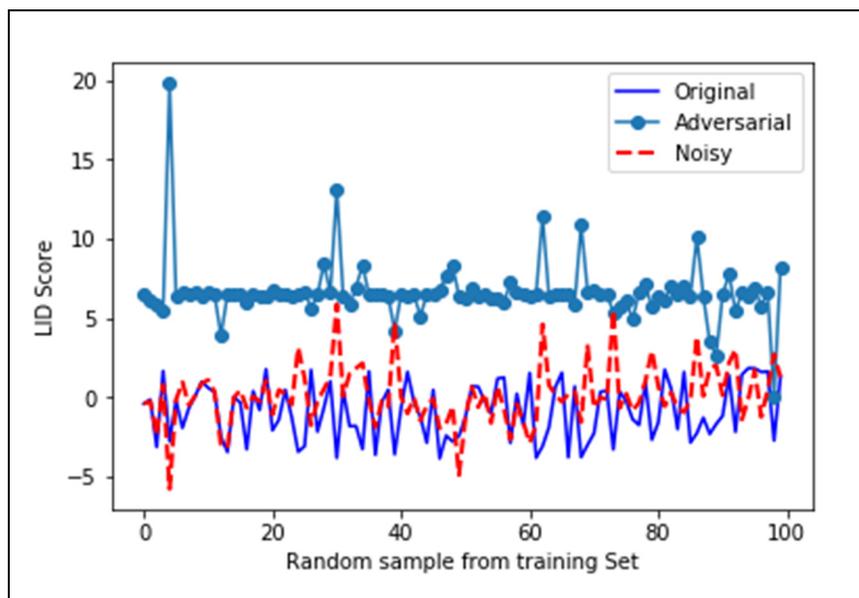


Figure 3.6 (c) The LID score for the 3rd random set

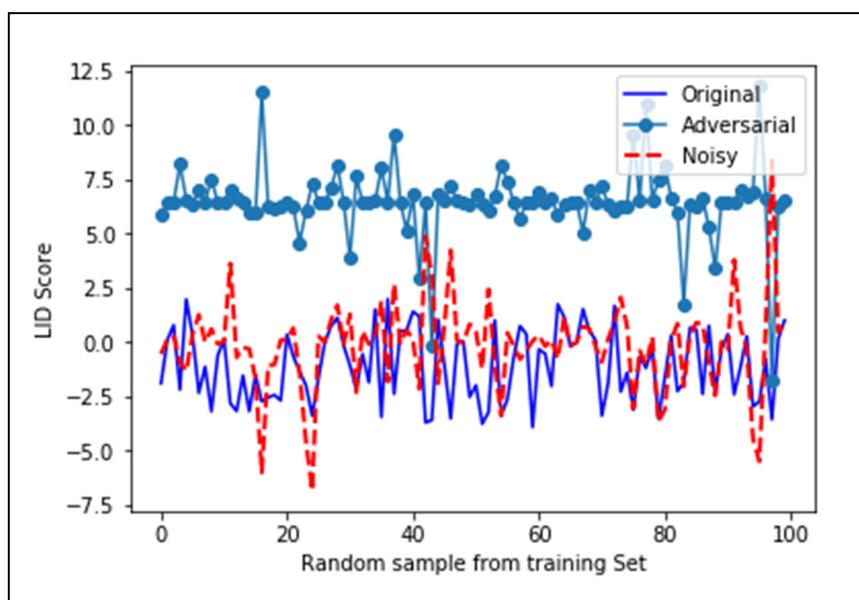


Figure 3.6 (d) The LID score for the 4th random set

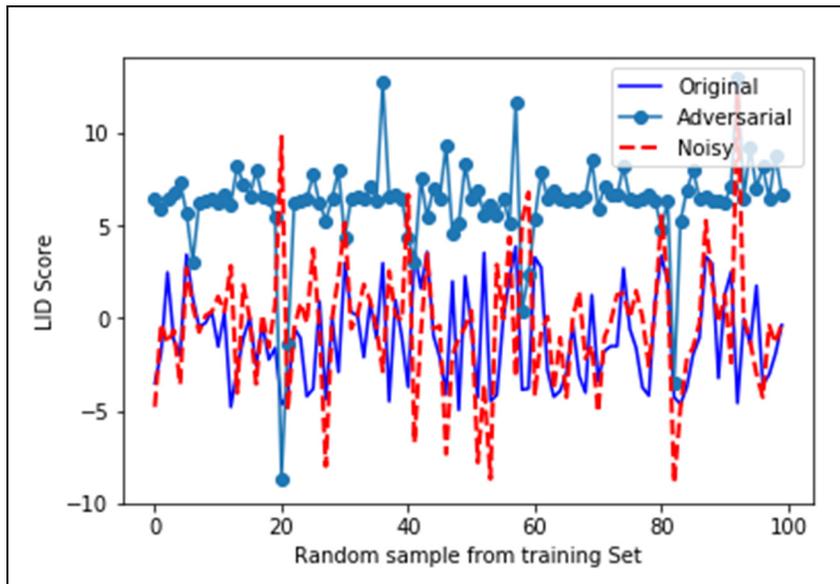


Figure 3.6 (e) The LID score for the 5th random set

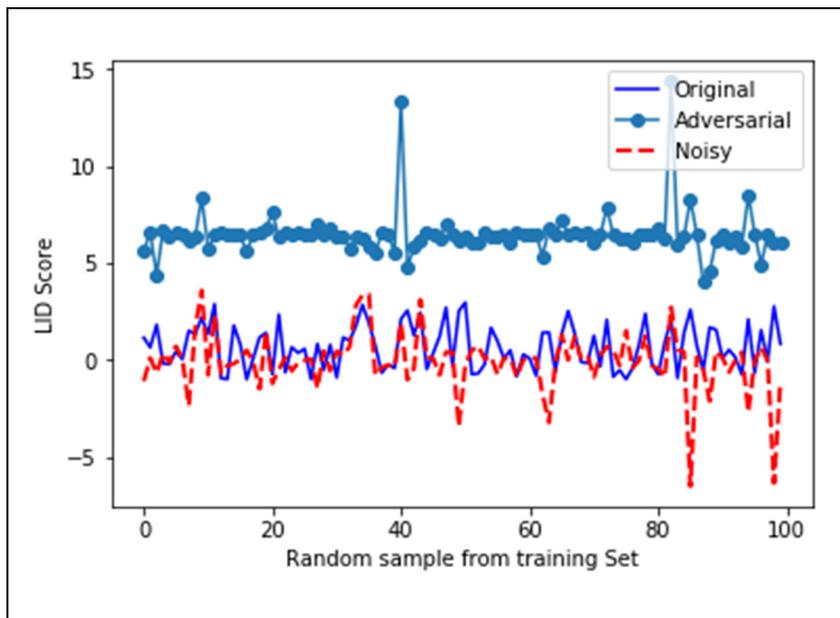


Figure 3.6 (f) The LID score for the 6th random set

Figure 3.6 The LID score comparisons for different random samples chosen from the training set

We summarize the details of the LR in Table 3.4 as well as other training information. Overall, Table 3.4 statistically proves that LIDs for adversarial samples are far from original and noisy samples, and Figure 3.6 shows this difference visually.

Generating adversarial samples with respect to the intrinsic characteristics of the given dataset could be very costly in terms of optimization overhead. In other words, Equation (3.8) does not always show a complex optimization task and could be a non-polynomial problem. These problems cannot be solved by polynomial functions approximation (of any degree). Therefore, Equation (3.8) could be replaced by a faster operation, namely, by taking advantage of gradient information backpropagated through the network during its training. Generating adversarial samples relying on gradient information was first introduced in the computer vision community, and was called “fast gradient sign method” (FGSM) (Goodfellow et al., 2014). We will adapt this attack for our regression task.

The FGSM is categorized as a white-box and non-targeted adversarial attack, mainly for architectures trained by backpropagation, and requires the model gradient information. For a given input x_i , the FGSM crafts adversarial sample \tilde{x} , as defined in Equation (3.10):

$$\hat{x} = x + \epsilon \times \text{sign}(\nabla_x J(\theta, x, l)) \quad (3.10)$$

where J is the cost function of the model, and ϵ is a float scalar to be defined by a local search. Since the FGSM attack was introduced for classification purposes, we needed to update the label index of l to a bounded value by providing a “supremum” and an “infimum”. Therefore, Equation (3.10) should be written under the following form (Goodfellow et al., 2014):

$$\hat{x} = x + \epsilon \times \text{sign}(\nabla_x J(\theta, x, v)) \quad (3.11)$$

where $v \notin [a - \lambda, a + \lambda]$ is an output value, and a is the actual value as defined in the training set. Our adapted version of the FGSM (AFGSM) requires its optimization for both ϵ and λ .

In our next experiment, we generated adversarial samples using the AFGSM for our proposed DNN, CNN, RNN, and LSTM architectures. We also studied the transferability property of crafted samples, as shown in Table 3.5.

Table 3.5 Transferability of adversarial samples crafted by the adapted fast gradient sign method (AFGSM). The highest values are in bold characters

	LR	SVR	DNN	CNN	RNN	LSTM
DNN	78.25	86.94	100	91.25	89.36	90.71
CNN	85.13	84.58	92.47	100	91.55	92.05
RNN	90.96	92.37	88.24	93.37	100	91.08
LSTM	91.45	90.33	89.69	89.99	92.28	100

Table 3.5 compares the transferability of adversarial samples using our proposed AFGSM algorithm. For instance, the first element in Table 3.5 suggests that 78.25% of total crafted adversarial samples are successfully transferable from DNN to the LR model. As shown in Table 3.5, all the models are vulnerable to our version of FGSM attack.

Not surprisingly, generated adversarial samples using the AFGSM for DNN and CNN are the most transferable samples to each other and are shown in bold characters (91.25, 92.47). Moreover, AFGSM-generated adversarial samples for RNN architecture are the samples most transferable to the CNN model (93.37). One hypothesis could be that this is related to their same utilized convolution layers, regardless of their filters shape, sizes, or order.

3.4.3 Adversarial Retraining

One potential defense against the threat of adversarial attack would be to train models by use of a combination of legitimate and adversarial samples. In other words, both original and crafted adversarial samples could be fed with the correct labels to the model within it training, with the aim of avoiding being misled during the testing time. Equation (3.12) shows our proposed retraining policy:

$$\tilde{J}(x_i, y_i, \theta) = cJ(x_i, y_i, \theta) + (1 - c)J(\tilde{x}_i, y_i, \theta) \quad (3.12)$$

where c is a constant value set to 0.25, 0.5 and 0.75 for our dataset. Table 3.6 presents the performance of the retraining policy for 3 different c values.

Table 3.6 Performance comparison of data-driven models by adversarial retraining

	c	LR	SVR	DNN	CNN	RNN	LSTM
Fooling rate (%)	0.25	86.45	77.36	80.35	81.06	78.37	79.08
	0.50	83.25	76.28	79.47	81.69	79.84	74.41
	0.75	84.27	75.79	80.23	81.97	80.56	73.19
Regression accuracy (%)	0.25	66.16	56.33	61.74	57.19	59.67	61.54
	0.50	64.87	57.13	64.24	58.36	58.14	60.79
	0.75	68.97	52.87	61.58	60.76	59.42	59.45

3.5 Conclusions

In this paper, the accuracy of data-driven regressors was investigated for conventional (LR and SVR) and state-of-the-art (DNN, CNN, RNN, and LSTM) algorithms for aircraft trajectory prediction by use of the traffic flow management system (TFMS) of aircraft trajectories. Although the results testify the higher performance of the modern algorithms in terms of regression accuracy, they also show the lowest resiliency against crafted adversarial attacks. We implemented FGSM and AFGSM adversarial attacks for all the trained models, and measured their fooling rates. Interestingly, conventional classifiers showed a higher robustness to adversarial attacks compared to the advanced deep neural networks.

As a pro-active approach for improving the robustness of the models, we adversarially trained all of them, which also increased their error rates. This increased error rate poses a security issue for learning-based regressors, especially since adversarial samples are transferable from any learned model to another model, as already shown. For our future work, a data-driven regression algorithm will be developed that will give a reasonable tradeoff between regression accuracy and fooling rate.

CHAPTER 4

Lyapunov-based Robust Adaptive Configuration of the UAS-S4 Flight Dynamics Fuzzy Controller

Seyed Mohammad Hashemi ^a, Ruxandra Mihaela Botez ^b

^{a,b} Department of System Engineering, École de Technologie Supérieure, Université du Québec, 1100 Notre Dame West, Montréal, QC, H3C 1K3, Canada

This article was published in *The Aeronautical Journal*, pp. 1-23, February 2022.
DOI:10.1017/aer.2022.2

Résumé

Parallèlement à la demande croissante de véhicules aériens sans pilote (UAVs) pour la surveillance et la reconnaissance, des contrôleurs avancés sont nécessaires pour ces systèmes critiques. Cet article propose la conception d'un contrôleur de dynamique de vol qui prend en compte diverses incertitudes pour l'UAS-S4 Ehécatl d'Hydra Technologies. Afin d'être réaliste, en plus des non-linéarités de la dynamique de vol, trois sources principales d'incertitudes sont considérées, comme celles causées par des paramètres de contrôleur inconnus, des erreurs de modélisation et des perturbations externes. Un contrôleur de logique floue adaptatif robuste est conçu, en charge de la dynamique de vol non linéaire en présence d'une variété d'incertitudes. La dynamique de vol non linéaire est modélisée en se basant sur la méthode Takagi-Sugeno par l'association souple de modèles linéaires locaux. Étant donné que ce contrôleur est basé sur un modèle, un modèle de référence optimal est défini, qui est stabilisé par la procédure du régulateur quadratique linéaire. Un contrôleur à logique floue est alors conçu pour le modèle non linéaire. Enfin, dans le but de gérer les incertitudes, les gains du contrôleur flou sont reconfigurés, et sont ajustés continuellement par des lois adaptatives robustes basées sur la théorie du Lyapunov. Les performances du contrôleur à logique floue adaptative robuste UAS-

S4 sont évaluées en termes de stabilisation de la dynamique latérale et longitudinale du vol, et du suivi des variables d'état du modèle de référence pour diverses incertitudes.

Abstract

In tandem with the fast-growing demand for Unmanned Aerial Vehicles UAVs for surveillance and reconnaissance, advanced controllers for these critical systems are needed. This paper proposes a flight dynamics controller design that considers various uncertainties for the Hydra Technologies UAS-S4 Ehécatl. In order to be realistic, in addition to flight dynamics nonlinearities, three main sources of uncertainties are considered, as those caused by unknown controller's parameters, modeling errors, and external disturbances. A robust adaptive fuzzy logic controller is designed, in charge of nonlinear flight dynamics in presence of a variety of uncertainties. The nonlinear flight dynamics is modeled based on the Takagi-Sugeno method relying on the soft association of local linear models. Since this controller is model-based, an optimal reference model is defined, which is stabilized by the Linear Quadratic Regulator procedure. A fuzzy logic controller is then designed for the nonlinear model. Lastly, with the aim to handle the uncertainties, the gains of the fuzzy controller are reconfigured, and are continuously adjusted by Lyapunov-based robust adaptive laws. The performance of the UAS-S4 robust adaptive fuzzy logic controller is evaluated in terms of lateral and longitudinal flight dynamics stabilization, and the reference model state variables tracking under various uncertainties.

4.1 Introduction

Unmanned Aerial Vehicles (UAVs) are remotely controlled aircraft designed to perform specific tasks. Due to the fast-growing demand for UAVs aimed at a variety of applications, the design of UAVs has remained a dynamic research field (Cir, 2011). In most cases, UAVs have been produced for military and disaster relief purposes, as well as for surveillance and reconnaissance (Watts, Ambrosia, & Hinkley, 2012). The UAS-S4 Ehecattl is such an UAV, designed and manufactured by the Hydra Technologies company in Mexico to perform military and civilian surveillance (M. A. J. Kuitche & Botez, 2019).

Critical UAV systems are equipped with accurate flight dynamics controllers (H. Chen, Wang, & Li, 2009). Designing an efficient controller requires an accurate flight dynamics model (Tuzcu, Marzocca, Cestino, Romeo, & Frulla, 2007). In fact, the access to the flight dynamics model enhances our ability to evaluate the controller performance in the early phases of the UAV development instead of relying mainly on flight test phases, which dramatically improves flight safety while reducing both costs and time (X. Yu, Guo, Zhang, & Jiang, 2021). The present work seeks to design a fully functional controller for the UAS-S4 based on its flight dynamics model. In this context, the model refers to the mathematical representation of the UAS-S4 flight dynamics system, which is basically used for its better understanding, prediction, and control.

Since fixed-wing UAS-S4s have the minimum number of required control surfaces, only a few actuators should provide a safe and reliable flight. While utilizing a fewer number of actuators results in a simpler UAV flight dynamics model, flight stability may become more affected in the presence of uncertainties (Borello, Cestino, & Frulla, 2010). These uncertainties may be external disturbances (due to environmental conditions (Ghommam, Saad, Mnif, & Zhu, 2020)), unknown controller parameters (affected by actuator and sensor imperfections (Cao & Hovakimyan, 2007)), and model imperfections (due to model approximation and to experimental errors (Bucolo, Buscarino, Famoso, Fortuna, & Frasca, 2019)). Additionally, following changes in flight conditions, the flight dynamics behave nonlinearly (Q. Wang & Stengel, 2004). In order to provide stable flight, it is essential to obtain an accurate mathematical flight dynamics model for the UAS-S4, and then to design an efficient controller that can consider flight dynamics nonlinearities and uncertainties.

Basically, any UAV flight dynamics model depends on its geometrical data, aerodynamic performance estimation, onboard actuators and sensors model, controller model, signal processing, and environmental functioning conditions (Stengel, 2015). By conducting flight tests, the model parameters can be determined. The interpretation of UAV propulsion and actuation systems in terms of its mass and inertia are the essential requirements for obtaining an accurate UAV model. To that end, both linear and nonlinear representations of aircraft models are shown in (Etkin & Reid, 1959). When obtaining an accurate flight dynamics model

is possible, a model-based controller will be highly successful in performing the intended tasks (Brosilow & Joseph, 2002). Thanks to the equipment available at our LARCASE (The Active Control, Avionics and Aeroservoelasticity Research Laboratory), including the UAS-S4, the Price-Paidoussis subsonic blow down wind tunnel, and the tow Research Aircraft Flight Simulators (RAFS) level-D for the [R]-too and Cessna Citation X (R. Botez, 2018), the accurate modeling of UAS-S4 flight dynamics is possible. Thus, the “model-based” control approach can be used to design the desired UAS-S4 controller.

From the classical control theory aspect, the PID approach is known as the generic and standard industrial control law (Amoroso et al., 2021). Basically, this controller operates via the feedback mechanism with the objective of reducing the stabilization and tracking error by modifying its signal. Although the PID technique can stabilize UAS-S4 flight dynamics without needing complex calculations for tuning the corresponding controller gains (M. A. J. Kuitche & Botez, 2019), performing the stabilization tasks requires major control signal efforts. The need of such a controller that gives the desired output while considering a cost function led us at our LARCASE to investigate the LQR approach. The LQR methodology controls the state variables by using an optimal state-feedback law computed while minimizing a fine-tuned energy-like cost function (Minchala-Avila, Garza-Castañón, Vargas-Martínez, & Zhang, 2015). This method showed high efficiency when it was utilized for our UAS-S4 flight dynamics control (Yañez-Badillo et al., 2020). However, state variables estimation decreases the LQR’s effectiveness, which worsens with increasing distance from the equilibrium point (Hashemi, Menhaj, & Amani, 2006). With respect to the designed PID and LQR controllers for our UAS-S4, we need to design an efficient flight dynamics controller that can solve challenges including, parametric and nonparametric uncertainties while flight dynamics behaves nonlinearly.

A control strategy is expected to be designed, such that it could work very well despite uncertainties (Lungu, Lungu, & Grigorie, 2013). These issues led us to choose the Fuzzy Logic Control (FLC) method, which has proven its ability to handle nonlinearities in a broad range of operation (Grigorie & Botez, 2011). Fuzzy Logic can provide a nonlinear model constructed by the soft association of several local linear models, while reduces computational complexity

for the controller in real time operations. Then, a Fuzzy Logic Controller can be designed based on the provided Fuzzy Logic Model (FLM). Where, the classical feedback control technique aimed at flight dynamics stabilization and tracking can be employed to control each local model. The designed Fuzzy Logic Controller (FLC) can be developed into the Adaptive FLC that can solve uncertainties due to unknown controller parameters (affected by actuator and sensor imperfections (Cho, Seo, & Lee, 2007)).

The objective of this article is the adaptive fuzzy methodology reconfiguration aimed at UAVs flight dynamics control for a wide range of uncertainties that may be caused by unknown controller's parameters. The novelty of this study is to modify the adaptive laws in order to make them robust against external disturbances (e.g., wind turbulence, wind shear, wind gust) or model imperfections (dues to fuzzification and defuzzification process errors) which were not considered in (Cho et al., 2007). Moreover, a general Theorem, followed by its stability proof is given to be useful for flight dynamics control of a variety of UAVs.

This paper is arranged in five sections. Following Section 4.1 on "Introduction", the UAS-S4 flight dynamics model and its fuzzy logic representation are stated in Section 4.2. The fuzzy controller developed for the desired reference model is described, and then its robustness and adaptive aspects are developed in Section 4.3. Section 4.4 presents the simulation results and their numerical analysis. The research contributions and achievements are summarized in Section 4.5, and they are followed by an outline of proposed future works.

4.2 UAS-S4 Flight Dynamics Modeling

The first essential step towards the design of an efficient controller for a flight dynamics system is the calculation of an appropriate model that accurately expresses the system dynamics behavior. In this way, the UAS-S4 is considered as the case study which is equipped with elevators, ailerons and rudders that are controlling its loads through the pitch, roll, and yaw axes. Figure 4.1 shows Hydra Technologies UAS-S4 Ehecatl, and Table 4.1 lists its specifications.



Figure 4.1 Hydra Technologies UAS-S4 Ehecattl

Table 4.1 UAS-S4 specifications (geometrical and flight data)

Specifications	Values
Wingspan	4.2 m
Wing area	2.3 m ²
Total length	2.5 m
Mean aerodynamic chord	0.57 m
Empty weight	50 kg
Maximum take-off weight	80 kg
Loitering airspeed	35 knots
Maximum speed	135 knots
Service ceiling	15000 ft
Operational range	120 km

For an UAV that flies in a broad operating range, a vast number of internal and external factors could affect its flight dynamics behavior. To obtain an accurate model for the design of an efficient controller, the flight dynamics data was mapped in a Mach-altitude flight envelope. A scheduled model was provided to represent the flight envelope containing nodes associated to the flight dynamics trim models. For each node, the flight dynamics model nonlinearities and parametric uncertainties were reduced.

The model in charge of each node was mathematically represented using differential equations with respect to the time-varying mass, and then linear models were designed around several equilibrium points. The controller was allocated to all nodes, in which a time-varying mass existed. Figure 4.2 depicts step-by-step procedure followed to reach the research objective.

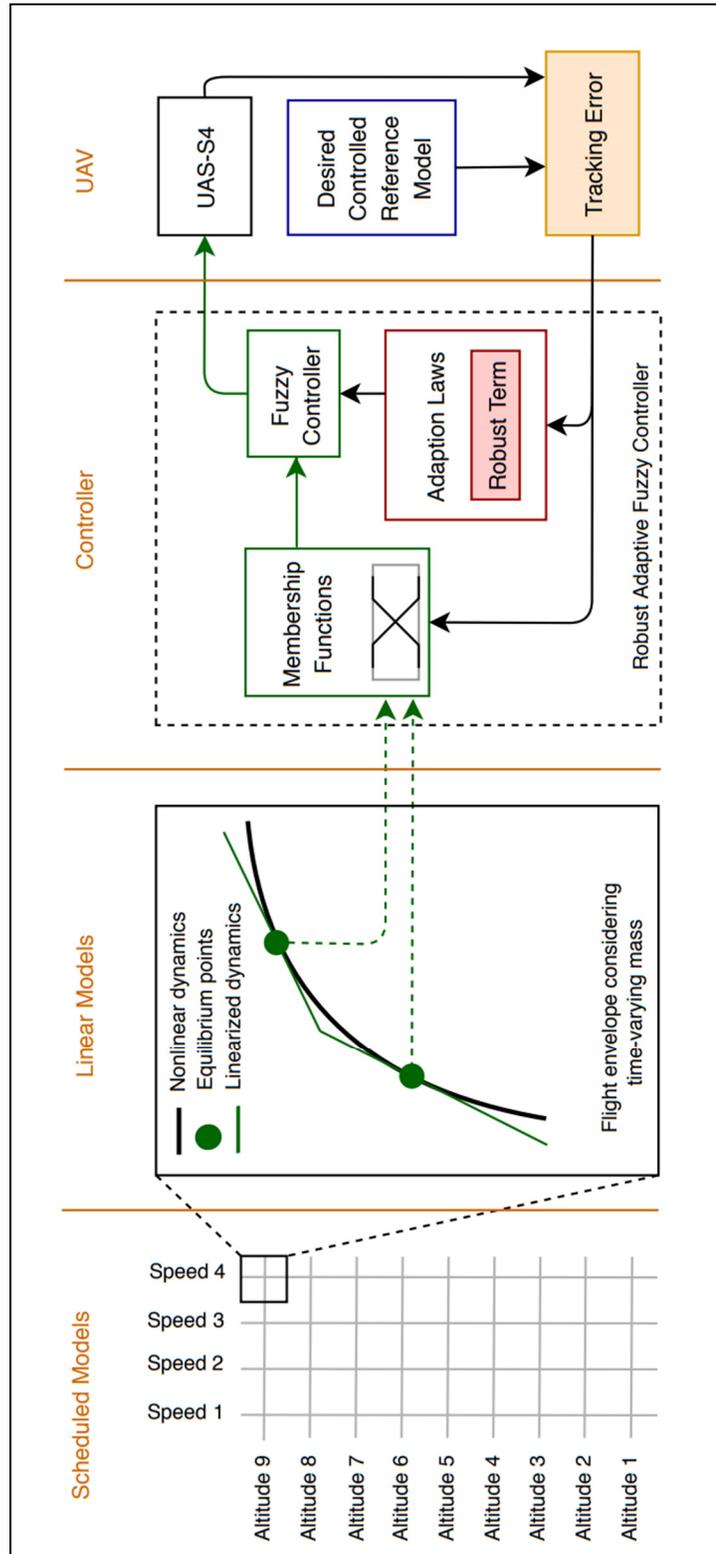


Figure 4.2 The followed procedure to control the UAS-S4 flight dynamics

Firstly, the flight envelope schedules the UAS-S4 flight dynamics model for 9 altitudes and 4 speeds. Then, the nonlinear model considering time-varying mass is linearized around several equilibrium points. Linearized models apply to the defined membership functions in order to obtain the UAS-S4 Fuzzy Logic Model (FLM). Next, the Fuzzy Logic Controller (FLC) computes the control signal based on the provided UAS-S4 FLM. Finally, the desired and controlled flight dynamics values are compared, and the error is used by the adaptation block for adjusting FLC gains. The UAS-S4 flight dynamics model and its controller design are explained in details in the following sections.

4.2.1 UAS-S4 Linear Local Models

By considering the aircraft differential equations of motion (Caughey, 2011), UAS-S4 state variables can be linearly modeled about its several equilibrium points. The UAS control problem can be solved for both its lateral and longitudinal motions. In this paper, the state variables of the UAS-S4 both lateral and longitudinal motions are controlled.

The state variables of the longitudinal flight dynamics are represented by $X_{lon} = [u \ w \ q \ \theta]^T$, with the axial velocity u , vertical velocity w , pitch rate q , and pitch angle θ while the control input is $\delta_{lon} = [\delta_e \ \delta_T]^T$. Even though the control vector is formed by the elevator deflection δ_e and thrust δ_T , the former plays the key role for the pitch control. The lateral flight dynamics state variables represented by $X_{lat} = [v \ p \ \eta \ \varphi]^T$, with the side velocity v , roll rate p , yaw rate η , and roll angle φ . Based on the aileron and ruder deflections, $\delta_{lat} = [\delta_a \ \delta_r]^T$ is in charge of lateral controls input.

Knowing that the linearized state-space representation of the model around an equilibrium point is (Nelson, 1998):

$$\dot{X}(t) = A X(t) + B \delta(t) \quad (4.1)$$

where the longitudinal state-space matrices are:

$$\begin{aligned}
A_{lon} &= \begin{bmatrix} G_u & G_w & 0 & -g \cos \theta_0 \\ H_u & H_w & u_0 & -g \sin \theta_0 \\ M_u + M_{\dot{w}}H_u & M_w + M_{\dot{w}}H_w & M_q + u_0H_{\dot{w}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\
B_{lon} &= \begin{bmatrix} G_{\delta_e} & G_{\delta_T} \\ H_{\delta_e} & H_{\delta_T} \\ M_{\delta_e} + M_{\dot{w}}H_{\delta_e} & M_{\delta_T} + M_{\dot{w}}H_{\delta_T} \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{4.2}$$

and the lateral state-space matrices are:

$$A_{lat} = \begin{bmatrix} Y_v & Y_p & -(u_0 - Y_\eta) & g \cos \theta_0 \\ L_v & L_p & L_\eta & 0 \\ N_v & N_p & N_\eta & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad B_{lat} = \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \tag{4.3}$$

where $G_u, G_w, H_u, H_w, M_u, M_w, M_q$ are the UAS-S4 longitudinal state matrix dimensional stability derivatives, and $G_\delta, H_\delta, M_\delta$ are its longitudinal control matrix dimensional stability derivatives. In addition, $Y_v, Y_p, Y_\eta, L_v, L_p, L_\eta, N_v, N_p, N_\eta$ are the UAS-S4 lateral state matrix dimensional stability derivatives, and $Y_\delta, L_\delta, N_\delta$ are its lateral control matrix dimensional stability derivatives.

In order to obtain the UAS-S4 state-space matrices' elements, it is needed to compute the dimensional aerodynamic coefficients and their derivatives. While several research projects on aircraft modeling have been conducted at the LARCASE (Bardela & Botez, 2017; R. M. Botez et al., 2015; Rodriguez & Botez, 2013), the most comprehensive study on the UAS-S4 modeling was detailed in (M. A. J. Kuitche & Botez, 2019). The UAS-S4 model was obtained at the LARCASE using four sub-models representatives of ‘‘aerodynamics’’, ‘‘actuator’’, ‘‘propulsion’’, and ‘‘mass and inertia’’.

The first sub-model (aerodynamics) was set up according to the Fderivatives in-house code; this code was based on new aerodynamics methodologies added to DATCOM (Anton, Botez, & Popescu, 2011). The second sub-model (propulsion) was built using a two-stroke engine integration model relying on the operation of an internal combustion engine (Otto Cycle (M. A. J. Kuitche, Botez, Viso, Maunand, & Moyao, 2020)), and on the propeller analysis (Blade

Element Theory (Romeo, Cestino, Pacino, Borello, & Correa, 2012)). Raymer and DATCOM techniques were used to implement the third sub-model (mass and inertia (Tondji & Botez, 2017)). Finally, the fourth sub-model (a control surface actuation system) was designed using the servomotors' characteristics, and the final UAS-S4 model was obtained by the sub-models integration (M. A. J. Kuitche & Botez, 2019).

In this way, the UAS-S4 flight dynamics related to both longitudinal and lateral motions was represented using several linear state-space models. Each state-space model expresses the linearized state variables about a specific equilibrium point corresponding to a certain range of altitudes and speeds. However, by increasing the operational range about an equilibrium point, the modeling error due to the linearization also increases. In order to enhance the models' accuracy, several equilibrium points can be considered, and consequently, several local linear models can be better fitted into the actual flight dynamics model. Therefore, a fuzzy logic approach is utilized for the UAS-S4 modeling.

4.2.2 UAS-S4 Fuzzy Logic Model

Basically, an aircraft nonlinear Flight Dynamics Model (FDM) can be represented through its affine system formulation (Z. Lin, 2002) by the equation $\dot{X} = F(X) + G(X) \delta$, where the control input vector δ is adjusting the state vector variables X using $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$ functions, that are unknown. A simple nonlinear FDM was found to be more efficient than a complex nonlinear system for the design of a model-based controller, which was our main objective. The higher efficiency of the simple nonlinear FDM was due to its reduced computational complexity, while providing fast control signal calculations in real-time operations (Ying, 1998). Therefore, the fuzzy logic approach was chosen, as it provided this procedure for approximating affine nonlinear systems (Zeng, Keane, & Wang, 2006).

Fuzzy logic offers the type of models that can be used to support the impression of partial truths, where the truth concept may range between “completely true” and “entirely false” (Zadeh, 1988). Fuzzy logic provides a tool for assembling several local linear models, relying on membership functions, with the objective of approximating a nonlinear model. The Takagi-

Sugeno Fuzzy Logic modeling method is known as a practical and user-friendly technique for modeling real physical systems (Takagi & Sugeno, 1985), and was chosen in this study.

The Takagi-Sugeno Fuzzy Logic Model (T-S FLM) consists of a set of models that have been locally linearized about their equilibrium points. Based on the expert-defined fuzzy rules in Equation (4.4), the association of local models can approximate the actual nonlinear continuous-time flight dynamics model. According to the T-S procedure for generating rules, the i^{th} rule of the fuzzy model is defined as the following (Takagi & Sugeno, 1985).

$$\text{Rule}^i: \begin{cases} \text{if } x_1 \text{ is } \Gamma_1^i \text{ and } \dots \text{ and } x_n \text{ is } \Gamma_n^i \\ \text{then } \dot{X}(t) = A_i X(t) + B_i \delta(t) \\ \text{where } i = 1, \dots, j \end{cases} \quad (4.4)$$

where the state variables vector $X(t) \in R^n$ is controlled by the input $\delta(t) \in R$ for a j number of defined rules. The state-space matrices for the UAS-S4 model should then be converted into their controllable Canonical form, as shown in Equation (4.5).

$$A_{i_{n \times n}} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ a_i^n & a_i^{n-1} & \dots & a_i^2 & a_i^1 \end{bmatrix} \quad (4.5)$$

The fuzzy logic model representation based on the first-order models relying on j rules is (Takagi & Sugeno, 1985):

$$\dot{X}(t) = \frac{\sum_{i=1}^j \phi_i(t) (A_i X(t) + B_i \delta(t))}{\sum_{i=1}^j \phi_i(t)} \quad (4.6)$$

It should be mentioned that $\phi_i(t) = \prod_{h=1}^n \Gamma_h^i(X(t))$ activates the i^{th} rule by considering the collected grades $\Gamma_h^i(X(t))$ that are associated with the membership of $X(t)$ in Γ_h^i . An appropriate algorithm is further designed for flight dynamics control by utilizing the fuzzy model presented in this section.

4.3 Flight Dynamics Control

Having effective control over a UAV's flight dynamics would allow efficient flights in terms of their costs and safety. This section first defines the desired reference model for the chosen model-based control strategy by utilizing the LQR controller that performed very well under ideal conditions for the UAS-S4. The controlled model specifications (using the LQR methodology) about the equilibrium point are considered as the reference specifications for the controlled model using the Robust Adaptive Fuzzy Logic Control (RAFLC) methodology.

4.3.1 Reference Model

Basically, a reference model should define the desirable response of the controlled system to the input command. The design of the reference model is one of the basic aspects of an adaptive control strategy. In addition to offering performance index values (whether for frequency-domain or time-domain characteristics), the reference model should also satisfy its constraints, such as its relative degree and order.

According to the above-mentioned concerns regarding the reference model definition, the desired reference model specifications are determined using the Linear Quadratic Regulator (LQR) procedure applied around the equilibrium point. An LQR controls the state variables using an optimal state-feedback law, that is computed while minimizing a fine-tuned cost function (Boughari & Botez, 2012). The design of an LQR is based on the linear state-space model representation, as given in Equation (4.1). The LQR algorithm calculates the control signal while minimizing the following energy-based cost function:

$$J = \frac{1}{2} \int_0^{\infty} X^T(t) Q X(t) + \delta^T(t) R \delta(t) dt \quad (4.7)$$

where Q and R are the weight matrices (positive-semi-definite or positive-definite), that clarify the importance of cost function related to the state vector and the control vector, respectively. Consequently, the LQR control law is:

$$\delta(t) = -K X(t) \quad (4.8)$$

Following the state feedback gain K and state variables vector X values, the LQR procedure stabilizes the flight dynamics of the closed-loop model with respect to the state-space variables using Equation (4.9):

$$\dot{X}(t) = (A - BK) X(t) + BK \delta(t) \quad (4.9)$$

The feedback gain K is computed by:

$$K = R^{-1} B^T \mathcal{P} \quad (4.10)$$

where matrix \mathcal{P} is obtained by solving the following algebraic Riccati equation:

$$A^T \mathcal{P} + \mathcal{P} A + Q - \mathcal{P} B R^{-1} B^T \mathcal{P} = 0 \quad (4.11)$$

Next, the control block of the UAS-S4 model needs to be designed by taking into account the controlled reference model. The Fuzzy Logic Control (FLC) approach is employed in order to solve the challenge of model nonlinearities, as well as to outperform linear controllers.

4.3.2 Fuzzy Logic Controller (FLC)

Over the past two decades, the use of fuzzy logic for systems control has been developed for a variety of industrial applications. In most comparison studies, the FLC outperforms classical controllers in solving the challenges of nonlinearities, mathematical complexities, and in uncertainties removal (Babaei et al., 2011; J. Lin et al., 2020; Radhakrishnan & Swarup, 2020). In fact, FLC allowed obtaining accurate inputs from approximate inputs through an intuitive converting process (Babaei et al., 2011).

Basically, the FLC implementation is done in three fundamental steps: fuzzification, fuzzy interface, and defuzzification (Mehrjerdi, Saad, & Ghommam, 2010). The fuzzification block converts crisp data into fuzzy data using proper membership functions. The prepared data is then fed to the Fuzzy Inference System (FIS), which processes the fuzzy data and performs the control tasks according to the *IF-THEN* rules. Finally, the computed fuzzy control signal is converted into its real signal values through the defuzzification block. The FLC signal is applied to the UAS-S4 flight dynamics which is modeled using FLM. This control signal is

computed as function of the error (the difference between the measured and the desired flight dynamics values). Figure 4.3 shows the concept of FLC utilized in the closed-loop architecture in charge of the UAS-S4 flight dynamics.

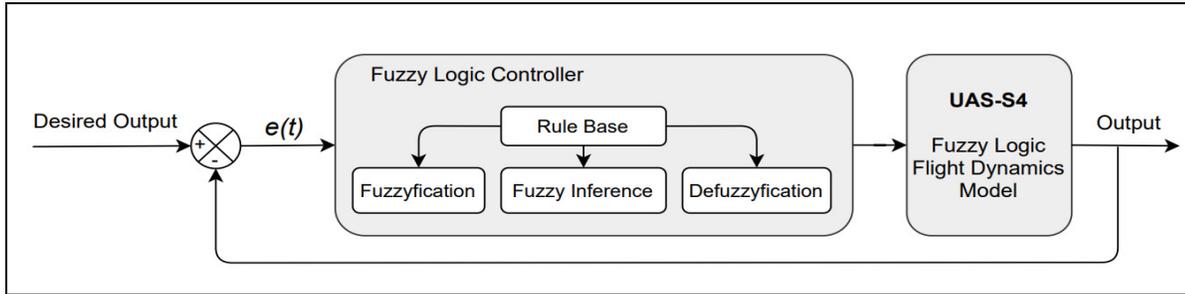


Figure 4.3 The fuzzy logic controller utilized for the UAS-S4 flight dynamics

Regarding the Takagi-Sugeno Fuzzy Logic Model (T-S FLM) described in subsection 4.2.2, the UAS-S4 FLM should be controlled by use of a compatible FLC. Hence, the T-S Fuzzy Logic Controller (T-S FLC) is needed to be designed.

4.3.3 T-S Fuzzy Logic Controller

Takagi-Sugeno Fuzzy Logic Control (T-S FLC) method can manage nonlinearities and time-varying parameters while avoiding control algorithm complexity (Tseng, Chen, & Uang, 2001). T-S FLC proved its efficiency on nonlinear systems in terms of state variables regulation and reference model tracking (Kamalasadan & Ghandakly, 2007). The T-S FLC is structured based on the classical feedback compensator theory (Doyle, Francis, & Tannenbaum, 2013), that is established for each local model. The rule-based control law can be mathematically represented by Equation (4.12) (Takagi & Sugeno, 1985).

$$\text{Rule}^i: \begin{cases} \text{if } x_1 \text{ is } \hat{r}_1^i \text{ and } \dots \text{ and } x_n \text{ is } \hat{r}_n^i \\ \text{then } \delta(t) = -K_i X(t) + Z_i r(t) \\ \text{where } i = 1, \dots, j \end{cases} \quad (4.12)$$

where the state variables are controlled by $\delta(t)$, and rely on the reference signal $r(t)$ and adjustable gains denoted by $K_{i_{1 \times n}}$ and $Z_{i_{1 \times 1}}$.

The T-S FLC output is given by Equation (4.13):

$$\delta(t) = \frac{\sum_{i=1}^j \phi_i(t) (-K_i X(t) + Z_i r(t))}{\sum_{i=1}^j \phi_i(t)} \quad (4.13)$$

By considering $\phi_i(t) = \prod_{h=1}^n \hat{f}_h^i(X(t))$, which activates the i^{th} rule of the fuzzy controller based on the collected grades $\hat{f}_h^i(X(t))$ associated with the membership of $X(t)$ in \hat{f}_h^i . With the aim of obtaining a zero-value tracking error, $\phi_i(t) = \phi_i(t) |Z_i^{-1}|$ should be determined in order to formulate Lyapunov function for the system to become asymptotically stable; when the gain of the reference signal value was 1, the controller could fire the proper rule with the same collected grade in the fuzzy model.

The T-S control law can be reproduced, as shown in Equation (4.14):

$$\sum_{i=1}^j \phi_i(t) Z_i^{-1} \delta(t) - \sum_{i=1}^j \phi_i(t) Z_i^{-1} (-K_i X(t) + Z_i r(t)) = 0 \quad (4.14)$$

Even though the FLC handles nonlinearities, it is affected by the adverse effects of parameters uncertainties. Since the concept of adjustable gains is supposed to overcome these problems, the modified Adaptive Fuzzy Logic Controller (AFLC) is employed, as it relies on adjustable gains. Additionally, we consider the other two main sources of uncertainties, namely "external disturbance", and "model imperfection". The robust adaptive configuration of the T-S FLC is our solution.

4.3.4 Adaptive T-S Fuzzy Logic Controller

In Control Systems Engineering, uncertainty is an issue that may appear due to a variety of reasons, and it can adversely affect controller performance. Uncertainty presence may reduce controller robustness, and may lead to systems dynamics instabilities. Therefore, an algorithm should control the nonlinear flight dynamics model while remaining efficient in the presence of uncertainties. To fulfill this objective, a reference model is defined by applying the T-S FLC. Then, the errors are measured by subtracting the UAS-S4 state variables values from the reference model's state variables values (Cho et al., 2007). Finally, using a Lyapunov function (which relies on the measured error) for guarantying the flight dynamics asymptotic stability,

the adaptation laws for gain tuning are calculated. Equation (4.15) defines the reference model containing the desired state variables, as follows:

$$\dot{X}_r(t) = A_r X_r(t) + B_r r(t) \quad (4.15)$$

If $k_{i_{1 \times n}}$ and $z_{i_{1 \times 1}}$ are assumed to be the gains of the desired compensator corresponding to each fuzzy rule, which can regulate the closed-loop response, such that the UAS-S4 state variables exactly follow the reference model state variables, then $A_r = A_i - B_i k_i$ and $B_r = B_i z_i$ need to be satisfied. By rearranging these last formulations as $A_i = A_r + B_i k_i$ and $B_r = B_i z_i$, and then, by substituting them into Equation (4.6), the aircraft's T-S fuzzy logic representation using the reference model is given in Equation (4.16).

$$\dot{X}(t) = \left(A_r + \frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1} k_i}{\sum_{i=1}^j \phi_i(t)} \right) X(t) + \left(\frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1}}{\sum_{i=1}^j \phi_i(t)} \right) \delta(t) \quad (4.16)$$

The error is defined as $E(t) = X(t) - X_r(t)$. This error is further obtained by subtracting Equation (4.16) from Equation (4.15). Therefore, the next Equation (4.17) represents this error.

$$\begin{aligned} \dot{E}_j(t) &= A_r E_j(t) + \left(\frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1} k_i}{\sum_{i=1}^j \phi_i(t)} \right) X(t) \\ &+ \left(\frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1}}{\sum_{i=1}^j \phi_i(t)} \right) \delta(t) - \left(\frac{\sum_{i=1}^j \phi_i(t) B_r}{\sum_{i=1}^j \phi_i(t)} \right) r(t) \end{aligned} \quad (4.17)$$

By replying Equation (4.14) into Equation (4.17), the error can be obtained using next Equation (4.18):

$$\begin{aligned} \dot{E}_j(t) &= A_r E_j(t) + \left(\frac{\sum_{i=1}^j \phi_i(t) B_r (k_i z_i^{-1} - K_i Z_i^{-1})}{\sum_{i=1}^j \phi_i(t)} \right) X(t) \\ &+ \left(\frac{\sum_{i=1}^j \phi_i(t) B_r (z_i^{-1} - Z_i^{-1})}{\sum_{i=1}^j \phi_i(t)} \right) \delta(t) \end{aligned} \quad (4.18)$$

In order to converge the error to zero, the following Lyapunov function for the stabilization analysis and reference signal tracking was employed:

$$V = E_j^T P E_j + \sum_{i=1}^j \left(\frac{1}{\gamma_1} (k_i - K_i)^T |z_i^{-1}| (k_i - K_i) + \frac{1}{\gamma_2} (z_i - Z_i)^T |z_i^{-1}| (z_i - Z_i) \right) \quad (4.19)$$

where $P = P^T > 0$ is positive-definite matrices and A_r stability assumption is guaranteed by use of $A_{r_i}^T P + P A_{r_i} < -Q_i$ for all matrices $Q_i = Q_i^T > 0$. In addition, γ_1 and γ_2 are positive constant parameters that are used to finely tune the gains. The gains of the fuzzy controller in Equation (4.13) can be adjusted via the following adaptation laws (based on FLC gains and their derivatives), obtained by solving Equation (4.19) (Cho et al., 2007).

$$\dot{K}_i = \gamma_1 \text{sign}(z_i) \frac{\phi_i B_r^T P E_j X^T}{\sum_{i=1}^j \phi_i}, \dot{Z}_i = -\gamma_2 \text{sign}(z_i) \frac{\phi_i B_r^T P E_j (\delta + K_i X)}{Z_i \sum_{i=1}^j \phi_i} \quad (4.20)$$

The stability theorem of adaptive gains is given in (Cho et al., 2007). Uncertainties dues to the unknown controller's parameters could affect the adaptive gain Z_i , and may approach it to zero value. Since adaptive gain Z_i appears in the denominator of Equation (4.20), in order to guarantee the model stability, the adaptation laws should be modified in cases when the denominator approaches to zero. Therefore, the modified tuning law for an adaptive fuzzy controller is represented in Equation (4.21) (S. M. Hashemi, R. M. Botez, & L. T. Grigorie, 2020a):

$$\dot{Z}_i = \begin{cases} w_i, & \text{if } |Z_i| > Z_{i_0} \quad \text{or} \quad Z_i = Z_{i_0} \text{ and } w_i \text{ sign}(Z_i) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.21)$$

where $w_i = -\gamma_2 \text{sign}(z_i) \frac{B_r^T P E_j (\delta + K_i X)}{Z_i \sum_{i=1}^j \phi_i}$

With respect to the stability proof given in (Cho et al., 2007), By assuming a uniformly-bounded reference input while analyzing the stable reference model, the control law $(K_i, Z_i, \hat{\phi}_i)$ and tracking error E were guaranteed bounded for all j fuzzy logic rules. The convergence of the reference model was ensured, such that $\lim_{t \rightarrow \infty} E_j(t) = 0$, as the tracking error E converges to zero. This assumption is clarified in the mathematical proof of the general theorem

formulated for the designed robust adaptive fuzzy logic laws after Equation (4.26). Although the presented Adaptive Fuzzy Logic Controller (AFLC) can control nonlinear flight dynamics in the presence of uncertainties, that are due to unknown controller's parameters, it remains sensitive against other sources of uncertainties. Model imperfection and external disturbances are the two main causes of uncertainties that adversely affect controller performance, and both of them can be solved using robust control theory.

4.3.5 Robust Adaptive T-S Fuzzy Logic Controller

Robust control is a static approach that deals explicitly with uncertain parameters and disturbances. In other words, it is utilized to guarantee stability and to obtain robust performance while taking into account disturbances and modeling errors (both of which are assumed to be “bounded” (Chabir et al., 2016)).

The uncertainties due to external disturbances, such as wind shear, gust, and turbulence can be considered mathematically as bounded functions $d(X, t)$, in which $D_{n \times 1} = [0 \ 0 \ \dots \ 0 \ 1]^T$.

$$\dot{X}(t) = \frac{\sum_{i=1}^j \phi_i(t) (A_i X(t) + B_i \delta(t))}{\sum_{i=1}^j \phi_i(t)} + Dd(X, t) \quad (4.22)$$

Additionally, even if an aircraft is modeled by a “skilled expert”, relying on “perfect aircraft data”, uncertainties in modeling may be due to other causes:

- Time-varying parameters, where a fixed controller can not always stabilize its state variables.
- Ignoring high-order dynamics for the nominal model simplification.
- Nonlinearities, where systems contain nonlinear dynamics, and models are represented approximately (such as our aircraft nonlinear dynamics, which is approximated using Fuzzy Logic modeling).

Eventually, uncertainties associated with modeling errors of system dynamics can be added mathematically into the state-space matrices of a T-S fuzzy model, as shown in Equation (4.23):

$$\dot{X}(t) = \frac{\sum_{i=1}^j \phi_i(t) \left([A_i + \epsilon_{A_i}] X(t) + [B_i + \epsilon_{B_i}] \delta(t) \right)}{\sum_{i=1}^j \phi_i(t)} + Dd(X, t) \quad (4.23)$$

where the errors are bounded, such as $\|\epsilon_{A_i}\|_\infty < \epsilon$ and $\|\epsilon_{B_i}\|_\infty < \epsilon$ for $i = 1, \dots, j$.

Equation (4.23) can be written under the following form:

$$\begin{aligned} \dot{X}(t) = & \left(A_r + \frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1} k_i}{\sum_{i=1}^j \phi_i(t)} \right) X(t) + \left(\frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1}}{\sum_{i=1}^j \phi_i(t)} \right) \delta(t) \\ & + \epsilon(X, \delta) + Dd(X, t) \end{aligned} \quad (4.24)$$

and then by considering the uncertainties defined as $f(\epsilon, d) = \epsilon(X, \delta) + Dd(X, t)$ the model would be:

$$\begin{aligned} \dot{X}(t) = & \left(A_r + \frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1} k_i}{\sum_{i=1}^j \phi_i(t)} \right) X(t) + \left(\frac{\sum_{i=1}^j \phi_i(t) B_r z_i^{-1}}{\sum_{i=1}^j \phi_i(t)} \right) \delta(t) \\ & + f(\epsilon, d) \end{aligned} \quad (4.25)$$

Following a procedure for computing adaptive gains similar to the ones used in the previous subsection, and based on a robust control strategy (Dullerud & Paganini, 2013), the modified adaptation laws are:

$$\begin{aligned} \dot{K}_i &= \gamma_1 \left[\text{sign}(z_i) \frac{\phi_i B_r^T P E_j X^T}{\sum_{i=1}^j \phi_i} - \vartheta K_i \|E_j\| \right], \\ \dot{Z}_i &= \begin{cases} w_i, & \text{if } |Z_i| > Z_{i0} \text{ or } Z_i = Z_{i0} \text{ and } w_i \text{ sign}(Z_i) < 0 \\ 0, & \text{otherwise} \end{cases} \\ w_i &= -\gamma_2 \left[\text{sign}(z_i) \frac{B_r^T P E_j (\delta + K_i X)}{Z_i \sum_{i=1}^j \phi_i} - \vartheta Z_i \|E_j\| \right] \end{aligned} \quad (4.26)$$

Theorem: Considering a UAV flight dynamics model represented by Equation (4.25); its desired reference flight dynamics model is given in Equation (4.15) (which respects $A_{r_i}^T P + P A_{r_i} < -Q_i$ Inequality), in which the control function is represented by Equation (4.13), which is tuned by the robust adaptive laws, shown in Equation (4.26). By assuming uniformly-bounded reference input and stable reference model, signals corresponding to the control law $(K_i, Z_i, \dot{\phi}_i)$ and E_j are guaranteed to be bounded for all fuzzy rules. The reference model

tracking convergence is ensured, so that $\lim_{t \rightarrow \infty} E_j(t) = 0$, as the tracking error E converges to zero.

Proof: The stability analysis is done based on the designed adaptive laws, as seen in Equations (4.20) and (4.21), by use of the Lyapunov function, described in Equation (4.19). The conditions $|Z_i| > Z_{i_0}$ or $Z_i = Z_{i_0}$ and $w_i \text{sign}(Z_i) < 0$ were considered, and $\dot{V} = -E_j^T Q E_j$ was obtained. In the condition expressed by $Z_i = Z_{i_0}$, when the Lyapunov function is represented with Equation (4.19), its time derivative is given by:

$$\dot{V} = -E_j^T Q E_j + 2E_j^T P \frac{\sum_{i=1}^j \phi_i(t) B_r (z_i^{-1} - Z_0^{-1})(K_i X + \delta)}{\sum_{i=1}^j \phi_i(t)} \quad (4.27)$$

As $|z_i| > Z_0$, so that $(z_i^{-1} - Z_0^{-1})\text{sign}(z_i^{-1}) < 0$, therefore:

$$E_j^T P \frac{\sum_{i=1}^j \phi_i(t) B_r (z_i^{-1} - Z_0^{-1})(K_i X + \delta)}{\sum_{i=1}^j \phi_i(t)} < 0 \quad (4.28)$$

which means that $\dot{V} < 0$. Hence, for both conditions shown in Equation (4.21):

$$\dot{Z}_i = \begin{cases} w_i, & \text{if } |Z_i| > Z_{i_0} \text{ or } Z_i = Z_{i_0} \text{ and } w_i \text{sign}(Z_i) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.21)$$

$$\text{where } w_i = -\gamma_2 \text{sign}(z_i) \frac{B_r^T P E_j (\delta + K_i X)}{Z_i \sum_{i=1}^j \phi_i}$$

Following Equation (4.28) one obtains:

$$\dot{V} > -E_j^T Q E_j \quad (4.29)$$

Therefore:

$$\int_0^{\infty} E_j^T E_j \leq \frac{V(0) - V(\infty)}{\lambda_{\min}(Q)} \quad (4.30)$$

while relying on the Barbalat's lemma, $\lim_{t \rightarrow \infty} E_j(t) = 0$.

Then, by considering that the adaptive laws contain a robust term represented by Equation (4.26) in the conditions expressed by $|Z_i| > Z_{i_0}$ or $Z_i = Z_{i_0}$ and $w_i \text{sign}(Z_i) < 0$, the Lyapunov function is expressed by Equation (4.19), therefore, its time derivative becomes:

$$\begin{aligned}
\dot{V} \leq & -\lambda_{\min}(Q)\|E_j\|^2 + 3\lambda_{\max}(P)\varepsilon\|E_j\|^2 + \lambda_{\max}(P)\varepsilon\|X_r\|^2 \\
& + 2\lambda_{\max}(P)\varepsilon \sum_{i=1}^j \|K_i\|\|E_j\|^2 \\
& + 2\lambda_{\max}(P)\varepsilon\|E_j\| \sum_{i=1}^j [|Z_i r| + \|K_i\|\|E_j\|] \\
& - \vartheta \sum_{i=1}^j (k_i - K_i)^T |z_i^{-1}| (k_i - K_i) \\
& - \vartheta \sum_{i=1}^j |z_i^{-1}| (z_i - Z_i)^2 \\
& + \vartheta \sum_{i=1}^j \left\| (k_i - K_i) |z_i^{-1}| - K_i \|E_j\| \right\|^2 \\
& + \vartheta \sum_{i=1}^j \left\| (z_i - Z_i) |z_i^{-1}| - Z_i \|E_j\| \right\|^2
\end{aligned} \tag{4.31}$$

We can determine Q , so that $6\lambda_{\max}(P)\varepsilon < \lambda_{\min}(Q)$. Therefore:

$$\begin{aligned}
\dot{V} \leq & -\frac{1}{2}\lambda_{\min}(Q)\|E_j\|^2 + \lambda_{\max}(P)\varepsilon\|X_r\|^2 \\
& + 2\lambda_{\max}(P)\varepsilon \sum_{i=1}^j \|K_i\|\|E_j\|^2 \\
& + 2\lambda_{\max}(P)\varepsilon\|E_j\| \sum_{i=1}^j [|Z_i r| + \|K_i\|\|E_j\|] \\
& - \vartheta \sum_{i=1}^j (k_i - K_i)^T |z_i^{-1}| (k_i - K_i) \\
& - \vartheta \sum_{i=1}^j |z_i^{-1}| (z_i - Z_i)^2 \\
& + \vartheta \sum_{i=1}^j \left\| (k_i - K_i) |z_i^{-1}| - K_i \|E_j\| \right\|^2 \\
& + \vartheta \sum_{i=1}^j \left\| (z_i - Z_i) |z_i^{-1}| - Z_i \|E_j\| \right\|^2
\end{aligned} \tag{4.32}$$

If we consider that:

$$\dot{V} \leq -\alpha V + \beta \quad (4.33)$$

Where:

$$\alpha = \frac{\min\left\{\frac{1}{2}\lambda_{\min}(Q), \vartheta\right\}}{\max\{\lambda_{\min}(Q), \gamma_1^{-1}, \gamma_2^{-1}\}} \quad (4.34)$$

And:

$$\begin{aligned} \beta = & \lambda_{\max}(P)\varepsilon\|X_r\|^2 + 2\lambda_{\max}(P)\varepsilon \sum_{i=1}^j \|K_i\|\|E_j\|^2 \\ & + 2\lambda_{\max}(P)\varepsilon\|E_j\| \sum_{i=1}^j [|Z_i r| + \|K_i\|\|E_j\|] \\ & + \vartheta \sum_{i=1}^j \left\| (k_i - K_i)|z_i^{-1}| - K_i\|E_j\| \right\|^2 \\ & + \vartheta \sum_{i=1}^j \left\| (z_i - Z_i)|z_i^{-1}| - Z_i\|E_j\| \right\|^2 \end{aligned} \quad (4.35)$$

Therefore, $V \leq \frac{\beta}{\alpha}$ causes exponentially convergence of the Lyapunov function, and feasible stable region in order to guarantee the flight dynamics stability is:

$$\mathbb{O} = \left\{x \mid \frac{\beta}{\alpha} < V\right\} \quad (4.36)$$

In other words, adaptive gains guarantee the flight dynamics stability, as long as the amount of bounded uncertainties respect the threshold (the border of Equation (4.36) as the feasible stable region). Additionally, the leakage factor ϑ in the robust term should be carefully tuned based on a trade-off; a larger value for ϑ improves the controller robustness, while a smaller value provides more accurate reference model state variables tracking (Blažič, Matko, & Škrjanc, 2010). The mechanism of our designed T-S-based robust adaptive fuzzy logic controller (RAFLC) block diagram is depicted in Figure 4.4.

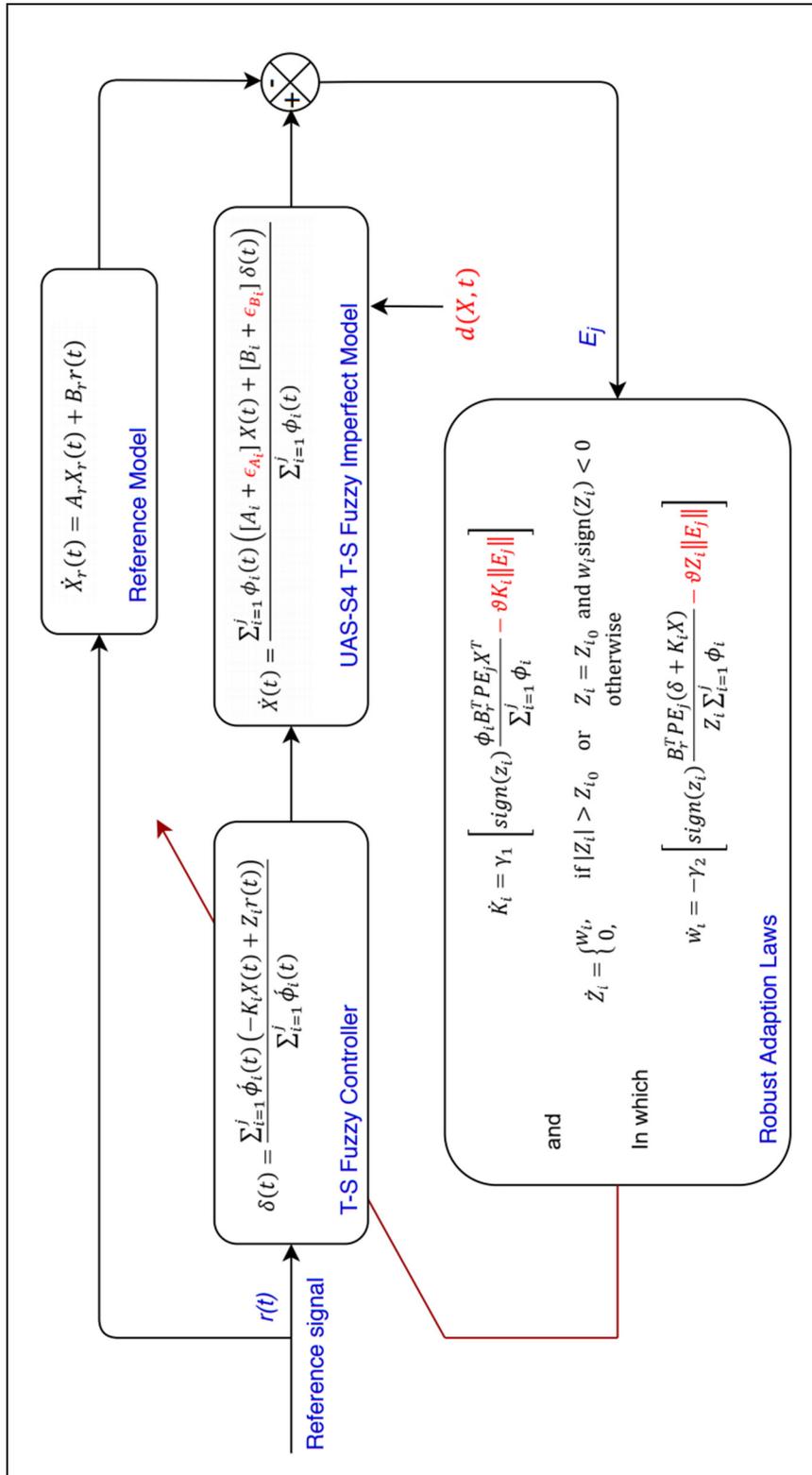


Figure 4.4 The designed robust adaptive T-S fuzzy logic controller (RAFLC) mechanism

4.4 Results

The effectiveness of the designed RAFL controller is evaluated in terms of UAS-S4 state variables stabilization and reference model state variables tracking. The efficiency of adaptation laws can be assessed by the convergence of the reference model's state variables tracking error. The designed RAFL controller was utilized for all trim conditions and showed very good servo-accuracy performance. The numerical results corresponding to several trim conditions were utilized to demonstrate the controller's functioning in details. By assuming that the aircraft is in the trim condition at the *speed* = 45 m/s, *altitude* = 6,100 m, and its mass is varying in time between 53 kg and 55 kg, the trim for the local models of the UAS-S4 are obtained through the following two Fuzzy Logic rules:

Rule 1: if E is positive then $\dot{X}(t) = A_1X(t) + B_1\delta(t)$

Rule 2: if E is negative then $\dot{X}(t) = A_2X(t) + B_2\delta(t)$

Knowing that, the UAS-S4 Fuzzy model was designed using 216 local FDMs. If a reference model was not employed, we had to calculate the membership functions using state variables. But, since we firstly designed the desired reference model, and the UAS-S4 FDM was supposed to track the reference model, we utilized the tracking error for calculating membership functions (which were used for all trim conditions). Hence, we defined the membership functions such that:

$$MF_1 = \begin{cases} 0, & E_j < -0.1 \\ 0.5 + 5E_j & \\ 1, & E_j > +0.1 \end{cases}, \quad MF_2 = \begin{cases} 1, & E_j < -0.1 \\ 0.5 - 5E_j & \\ 0, & E_j > +0.1 \end{cases}$$

The corresponding longitudinal and lateral state-space matrices are:

$$A_{1lon} = \begin{bmatrix} -0.0726 & 0.2346 & -0.9547 & -9.7830 \\ -0.3729 & -4.5992 & 43.3325 & -0.2240 \\ -0.1308 & -1.3599 & 0.4664 & -0.0118 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_{1lon} = \begin{bmatrix} -0.0133 \\ 0.0631 \\ -0.1525 \\ 0 \end{bmatrix}$$

$$A_{2lon} = \begin{bmatrix} -0.0640 & 0.2434 & -1.0870 & -9.7844 \\ -0.3616 & -4.2617 & 43.8266 & -0.2514 \\ -0.1369 & -1.2685 & 0.4455 & -0.0126 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{2lon} = \begin{bmatrix} -0.0124 \\ 0.0592 \\ -0.1454 \\ 0 \end{bmatrix}$$

$$A_{1lat} = \begin{bmatrix} -0.2423 & 0.2954 & -50.3286 & 9.7613 \\ -0.0619 & -12.8788 & 0.8274 & 0 \\ 0.0870 & -0.2368 & -0.1602 & 0 \\ 0 & 1 & 0.0060 & 0 \end{bmatrix}, B_{1lat} = \begin{bmatrix} 0 & 0.0386 \\ 0.6512 & 0.0074 \\ -0.0078 & -0.1628 \\ 0 & 0 \end{bmatrix}$$

$$A_{2lat} = \begin{bmatrix} -0.2473 & 0.0629 & -56.0717 & 9.7615 \\ -0.0594 & -14.2328 & 0.8345 & 0 \\ 0.0955 & -0.1886 & -0.1748 & 0 \\ 0 & 1 & 0.0013 & 0 \end{bmatrix}, B_{2lat} = \begin{bmatrix} 0 & 0.0440 \\ 0.8058 & 0.0091 \\ -0.0081 & -0.1993 \\ 0 & 0 \end{bmatrix}$$

and the reference model state-space matrices for longitudinal and lateral flight dynamics are expressed by:

$$A_{r lon} = \begin{bmatrix} -0.07073 & 0.2392 & -0.9704 & -9.9760 \\ -0.3818 & -4.621 & 43.41 & 0.6919 \\ -0.1093 & -1.307 & 0.2867 & -2.225 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{r lon} = \begin{bmatrix} 0.17770 \\ -0.843 \\ 2.037 \\ 0 \end{bmatrix}$$

$$A_{r lat} = \begin{bmatrix} -0.2423 & 0.2992 & -50.3243 & 9.8117 \\ -0.0655 & -13.227 & 0.5562 & -4.6161 \\ 0.0869 & -0.2488 & -0.1751 & -0.1573 \\ 0 & 1 & 0.0060 & 0 \end{bmatrix}, B_{r lat} = \begin{bmatrix} 0 & 0.2777 \\ 4.6072 & 0.0532 \\ -0.055 & -1.171 \\ 0 & 0 \end{bmatrix}$$

To analyze the designed controller effectiveness, the convergence of state variables (flight dynamics) for the reference model and controlled UAS-S4 model are evaluated during the flight dynamics stabilization. Regarding the initial state variables vectors $X_0 = [0 \ 0 \ 0 \ 0.1]^T$ and $X_{r_0} = [0 \ 0 \ 0 \ 0.08]^T$, Figure 4.5 depicts the Robust Adaptive Fuzzy Logic Controller (RAFLC) performance in terms of pitch angle, pitch rate, roll angle, and yaw rate stabilization while tracking those of the reference model, with respect to the control surfaces angles deflection limits ($-20 < \delta_e < 15$, $-40 < \delta_a < 40$, and $-30 < \delta_r < 30$).

For the longitudinal flight dynamics study, Figure 4.5(a) and Figure 4.5(c) show that the RAFL controller can stabilize the UAS-S4 pitch angle and the pitch rate, respectively. Figure 4.5(b) shows the elevator deflection during the pitch angle stabilization. For the lateral flight

dynamics study, Figure 4.5(d) and Figure 4.5(f) illustrate the UAS-S4 roll angle and yaw rate regulation, respectively. Figure 4.5(e) shows the aileron deflection during the roll angle stabilization.

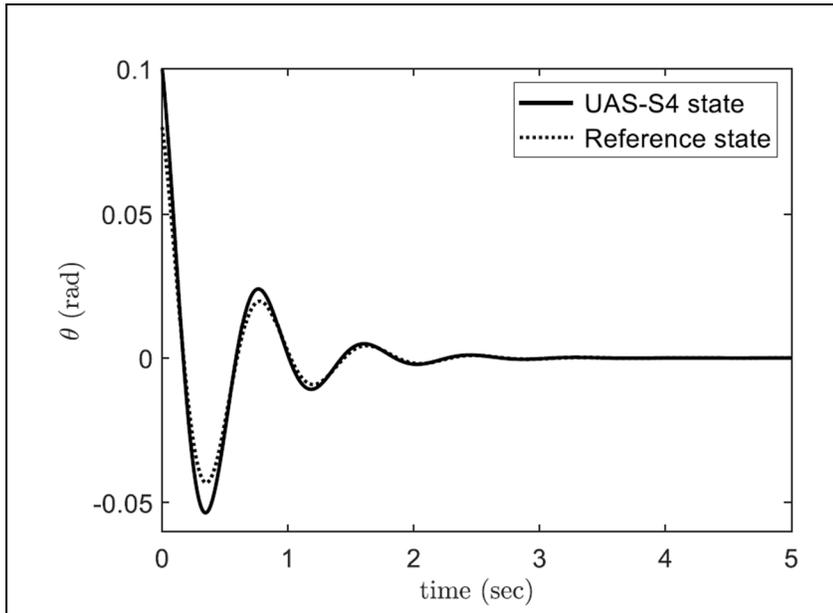


Fig 4.5(a) Pitch angle stabilization of the controlled UAS-S4 with respect to the reference model

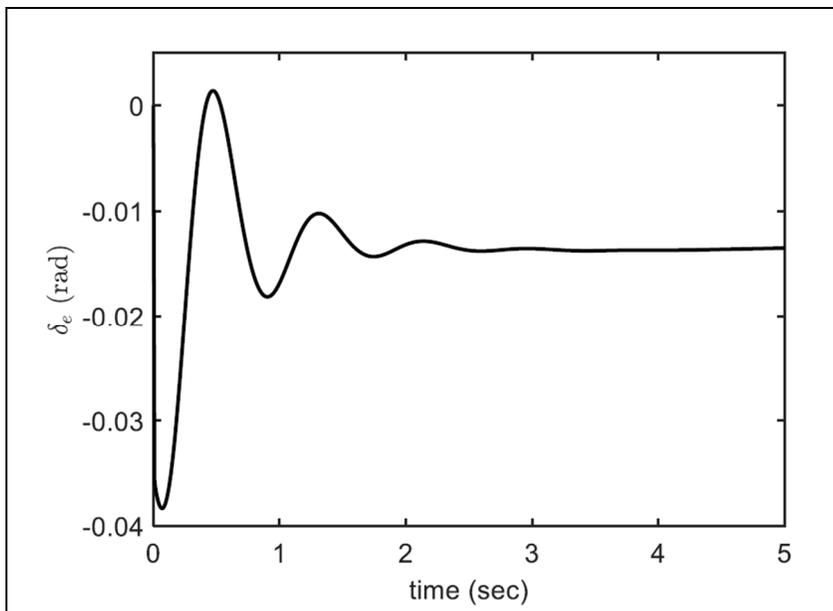


Fig 4.5(b) The UAS-S4 elevator angle deflection time variation in order to control the pitch angle

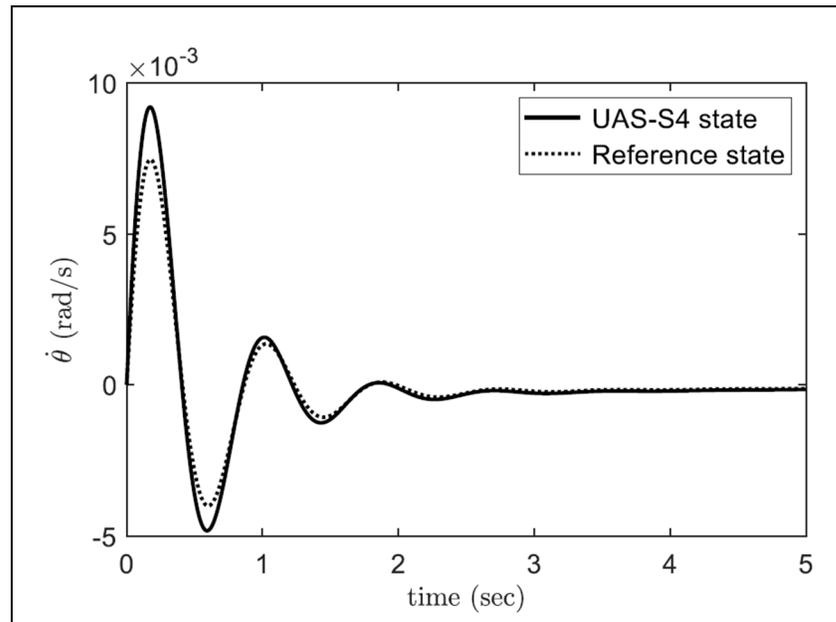


Fig 4.5(c) Pitch rate stabilization of the controlled UAS-S4 with respect to the reference model

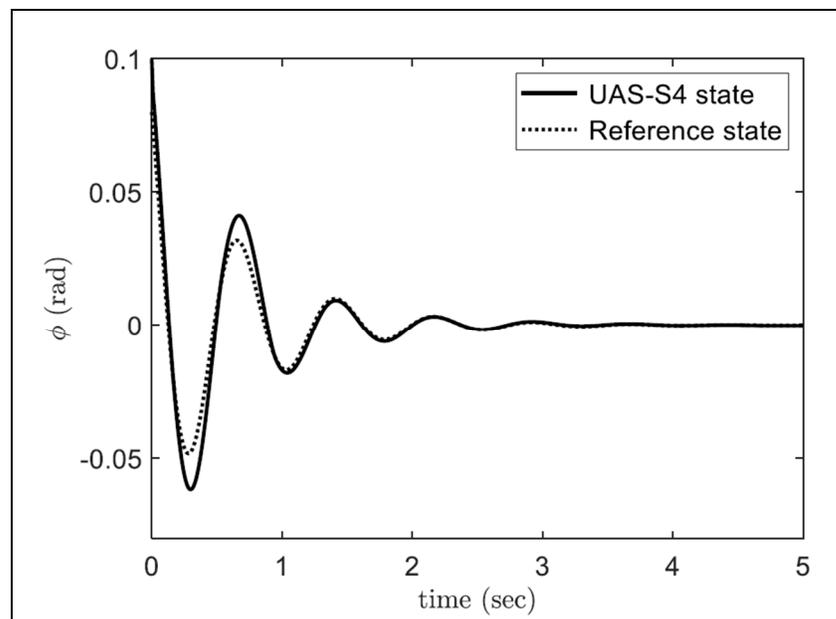


Fig 4.5(d) Roll angle stabilization of the controlled UAS-S4 with respect to the reference model

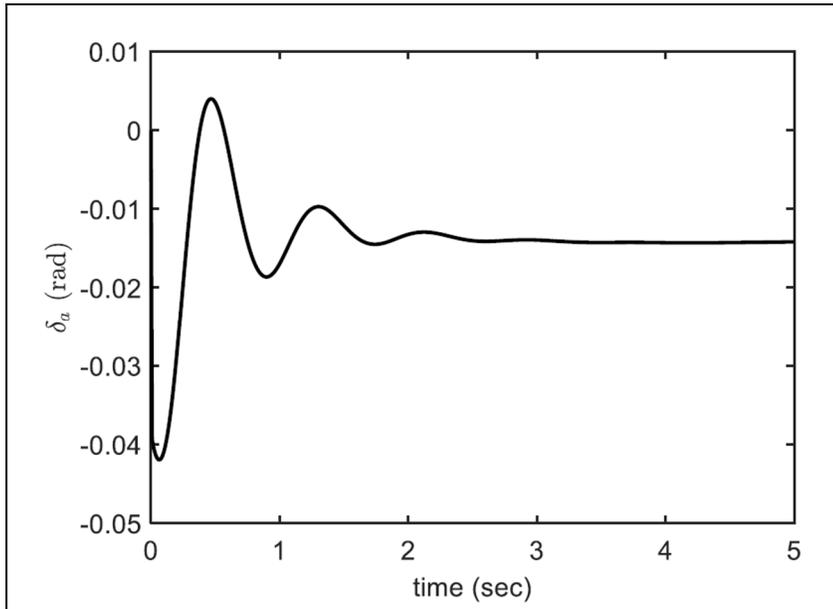


Fig 4.5(e) The UAS-S4 aileron angle deflection time variation in order to control the roll angle

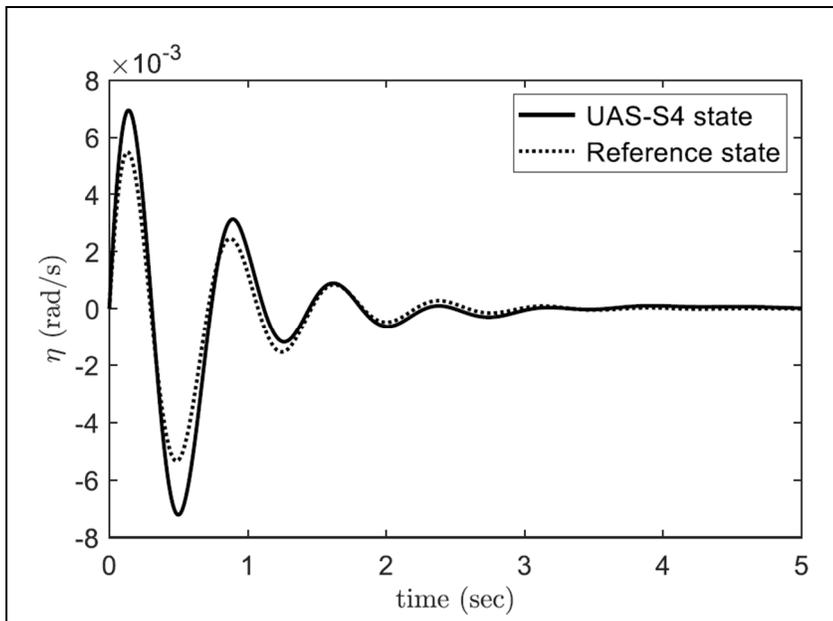


Fig 4.5(f) Yaw rate stabilization of the controlled UAS-S4 with respect to the reference model

Figure 4.5 the RAFL controller performance in terms of longitudinal and lateral state variables stabilization

State variables stabilization using the RAFL control mechanism was performed very well, while the UAS-S4 state variables track the reference model's state variables as well. Tracking the reference model state variables including pitch angle, pitch rate, roll angle, and yaw rate are shown in figure 4.6 during stabilization.

Figure 4.6(a) indicates the pitch angle and pitch rate convergence during stabilization. Figure 4.6(b) depicts the convergence error, that is expressed as the difference between the controlled UAS-S4 pitch angle and its reference pitch angle. Figure 4.6(c) shows the roll angle and yaw rate convergence towards stabilization. The convergence error obtained between the reference model and the controlled UAS-S4 model for the roll angle is shown in Figure 4.6(d).

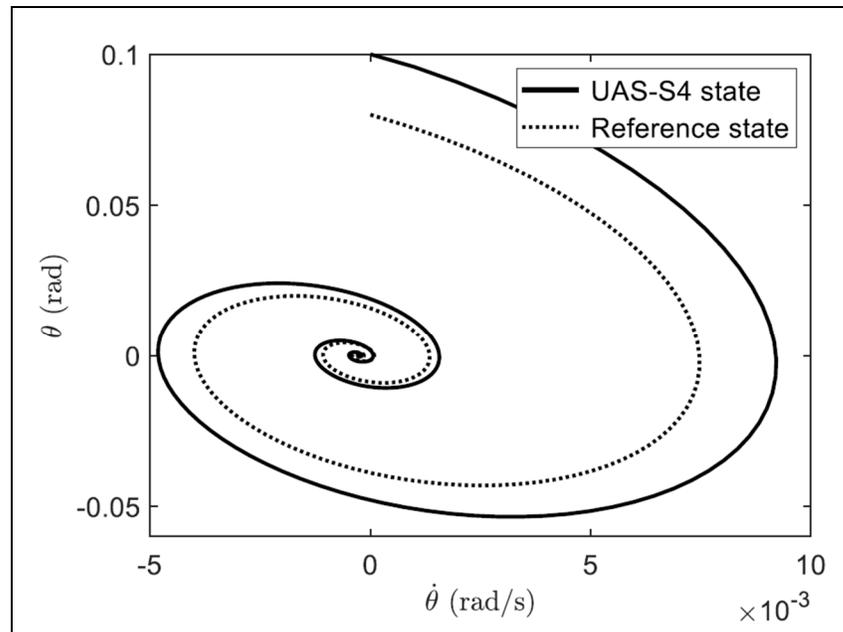


Fig 4.6(a) Convergence of longitudinal state variables (including pitch angle and pitch rate) while the UAS-S4 is tracking the reference model with the aim of stabilizing its state variables.

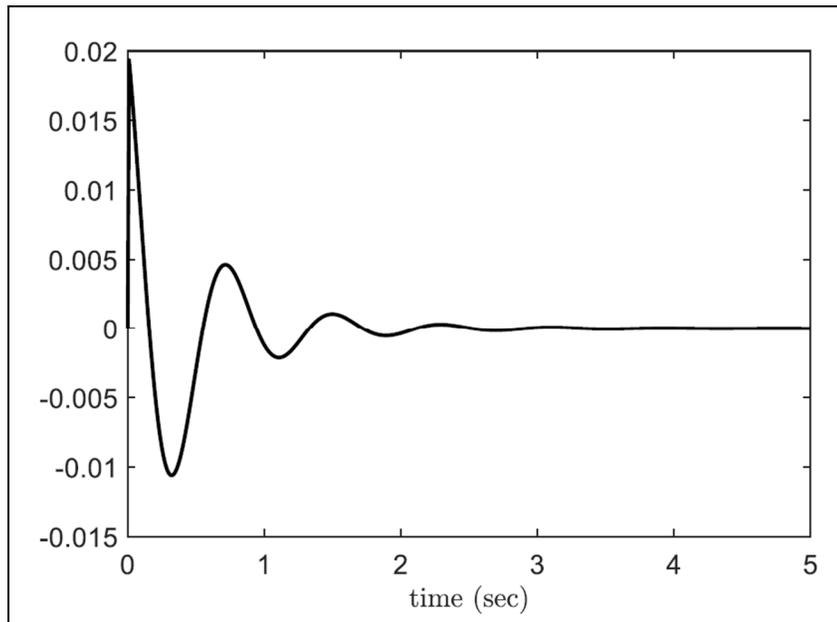


Fig 4.6(b) Convergence error between the reference model and the controlled UAS-S4 for the pitch angle

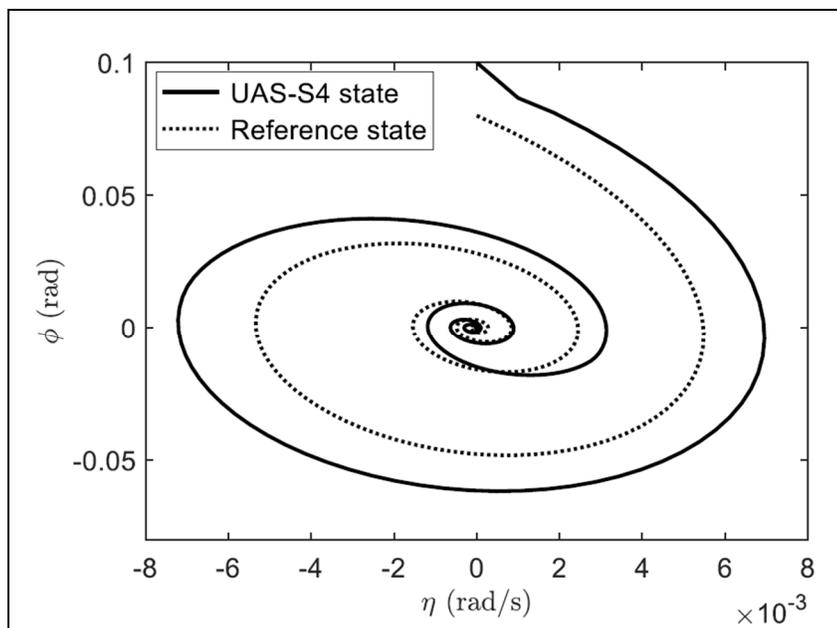


Fig 4.6(c) Convergence of lateral state variables (including roll angle and yaw rate) while the UAS-S4 is tracking the reference model with the aim of stabilizing its state variables.

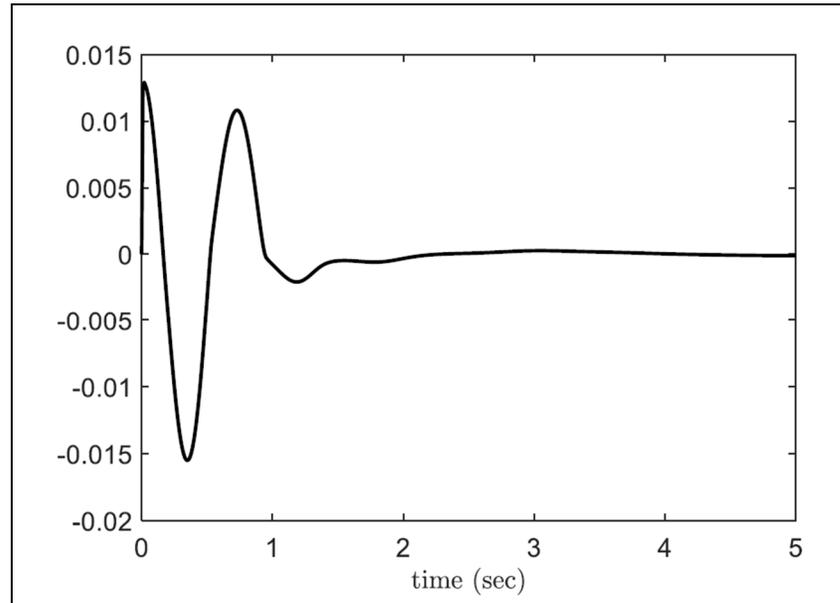


Fig 4.6(d) Convergence error between the reference model and the controlled UAS-S4 for the roll angle

Figure 4.6 The RAFL controller performance in terms of convergence error

It should be noted that this work is a part of an ongoing research project to design a novel aerial collision avoidance system. This project will predict the future trajectory of an aircraft, and if a conflict will be detected, then the system will provide a new safe trajectory for the aircraft to follow (S. M. Hashemi, R. M. Botez, & T. L. Grigorie, 2020b). According to the Traffic Collision Avoidance System (TCAS) criteria (Munoz et al., 2013), our UAS-S4 has to change its altitude using its elevator in order to avoid collisions. Hence, we analyzed our RAFL controller performance in terms of reference pitch angle tracking. With this aim, soft time-varying bounded signals are considered as the controller reference input in order to evaluate the controller's performance. For evaluating this model-based RAFL controller, the tracking error (the error obtained when the controlled UAS-S4 state variables track the reference model state variables) is considered as the "performance index". In this approach, a valid bounded reference input excites both the UAS-S4 and its reference model state variables, and the tracking error should converge to zero. Assuming the reference state as $\theta_r = 1.7 \cos 0.5t$, and initial condition given as $X_0 = [0 \ 0 \ 0 \ 0.2]^T$ and $X_{r_0} = [0 \ 0 \ 0 \ 0.18]^T$, the RAFL controller performance is shown in Figure 4.7.

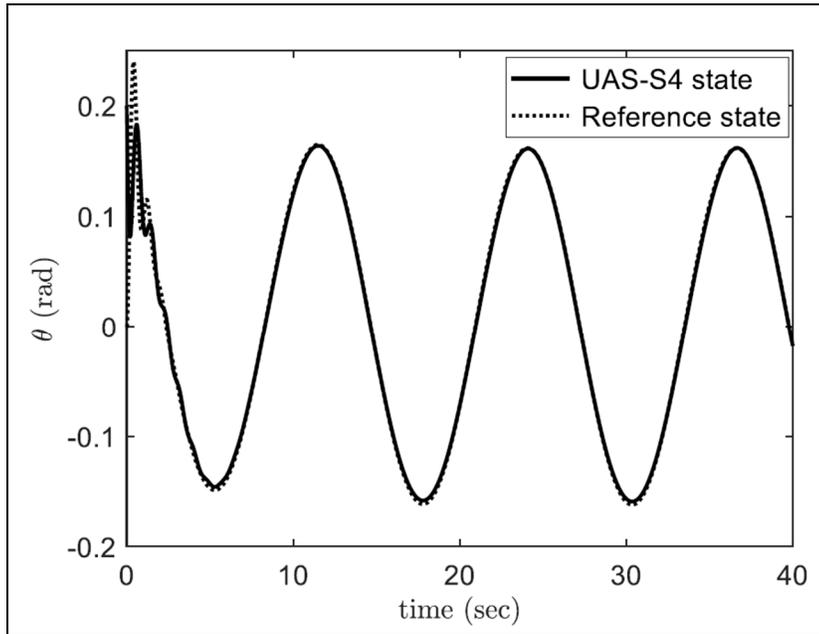


Fig 4.7(a) The UAS-S4 state variable is tracking the reference model pitch angle

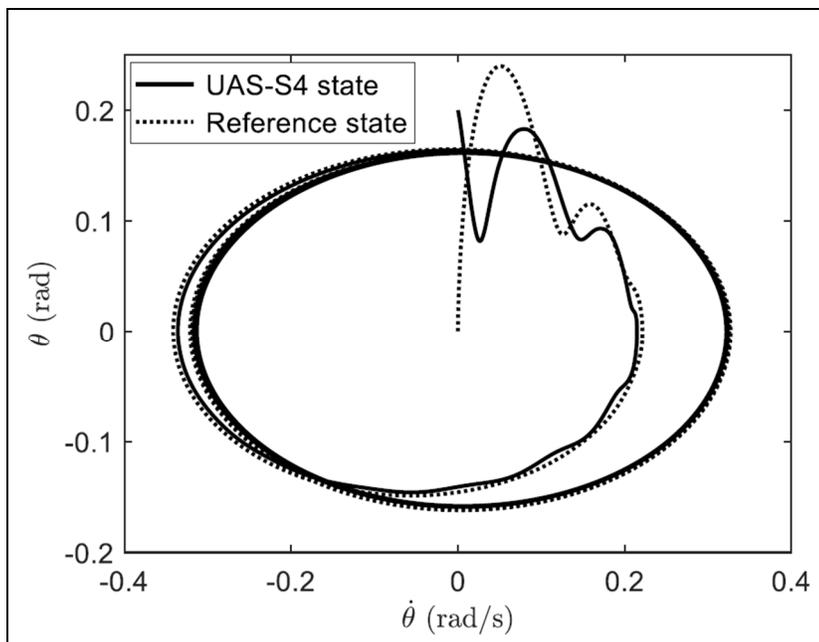


Fig 4.7(b) Concurrent longitudinal reference model state variables tracking

Figure 4.7 RAFLC performance in terms of pitch angle and pitch rate tracking in the absence of uncertainties

where the RAFL controller task is to control the UAS-S4 state variables, such that they track the reference model state variables accurately. As seen in Figures 4.7(a) and 4.7(b), the designed controller for UAS-S4 is able to perform its task in terms of reference model state variables tracking.

The adaptive gains effectiveness is well identified when uncertainties are considered. We therefore incorporated the uncertainties due to the unknown controller's parameters ($f = 0.02 \cos t$), and the controller performance was quantified in terms of pitch angle as state variable. Figure 4.8 shows the controller effectiveness in terms of the reference model state variables tracking by the controlled UAS-S4 state variables (Figure 4.8(a)), and tracking errors (Figure 4.8(b)). Figure 4.8(a) shows that, even though the RAFL controller efficiency was slightly degraded in terms of integrated tracking error (especially at the extremums), the RAFL controller could still handle the unknown controller's parameters uncertainties. Its performance is very good accordingly the reference model's tracking error, as shown in Figure 4.8(b).

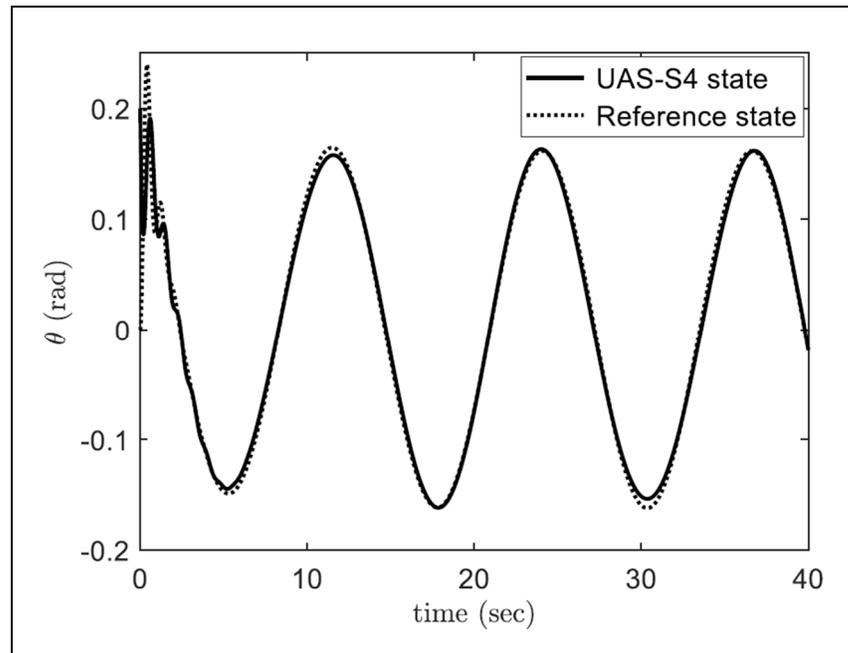


Fig 4.8(a) The reference model pitch angle tracking by controlled UAS-S4 state variables

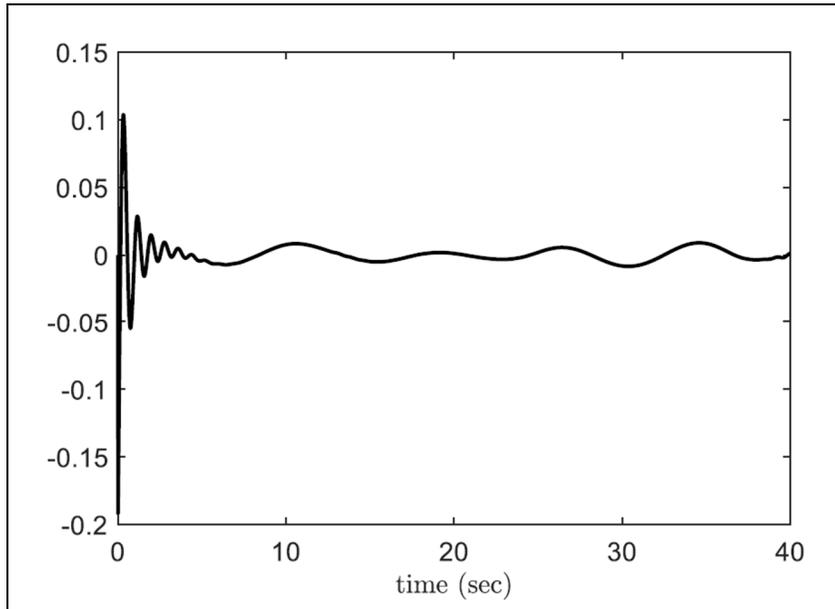


Fig 4.8(b) The tracking error convergence for pitch angle

Figure 4.8 The RAFL controller performance in terms of the reference model pitch angle tracking in the presence of uncertainties caused by unknown controller's parameters

In addition to the uncertainties related to the controller, model external disturbances and modeling errors are other sources of uncertainties that the controller is designed to remove their adverse effects. Figure 4.9 shows the efficiency of the controller under all considered.

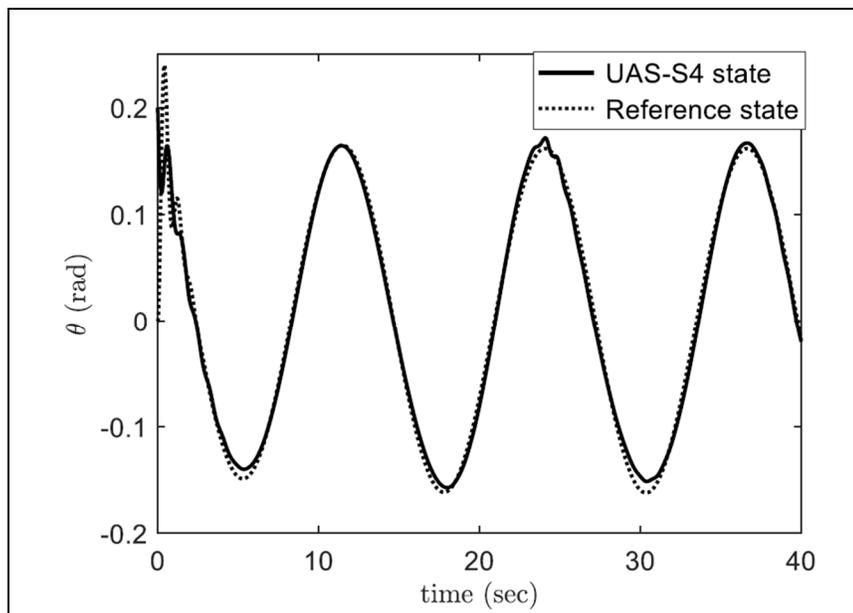


Fig 4.9(a) The reference model pitch angle tracking by controlled UAS-S4 state variable

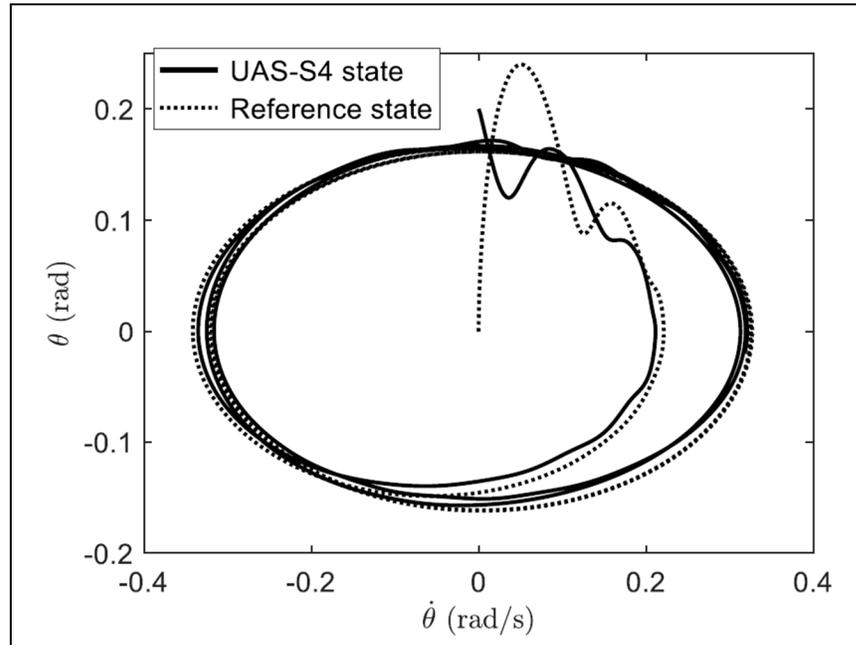


Fig 4.9(b) Controlled UAS-S4 state variables track the reference model state variables

Figure 4.9 The RAFL controller performance in presence of external disturbances and modeling errors

As shown in Figure 4.9, the controlled UAS-S4 state variables (pitch angle and pitch rate) followed the reference model state variables quite accurately. Although the controller performance slightly decreased compared to the case of controller's uncertainty (Figure 4.8), especially at the extremums, the robust terms $\vartheta K_i \|E_j\|$ and $\vartheta Z_i \|E_j\|$ in the adaptation laws could handle all uncertainties due to the controller parameters, such as model external disturbances and modeling errors.

The next challenge is the controller robustness threshold required to respect a feasible region for guarantying UAS-S4 stability. Figure 4.10 displays a visual representation of the uncertainties surpassing the feasible region. This figure shows three separate time varying regions. The first region (0 – 20 sec) shows the quality of the reference model state variables tracking (by controlled UAS-S4 state variables) when it is not affected by uncertainties.

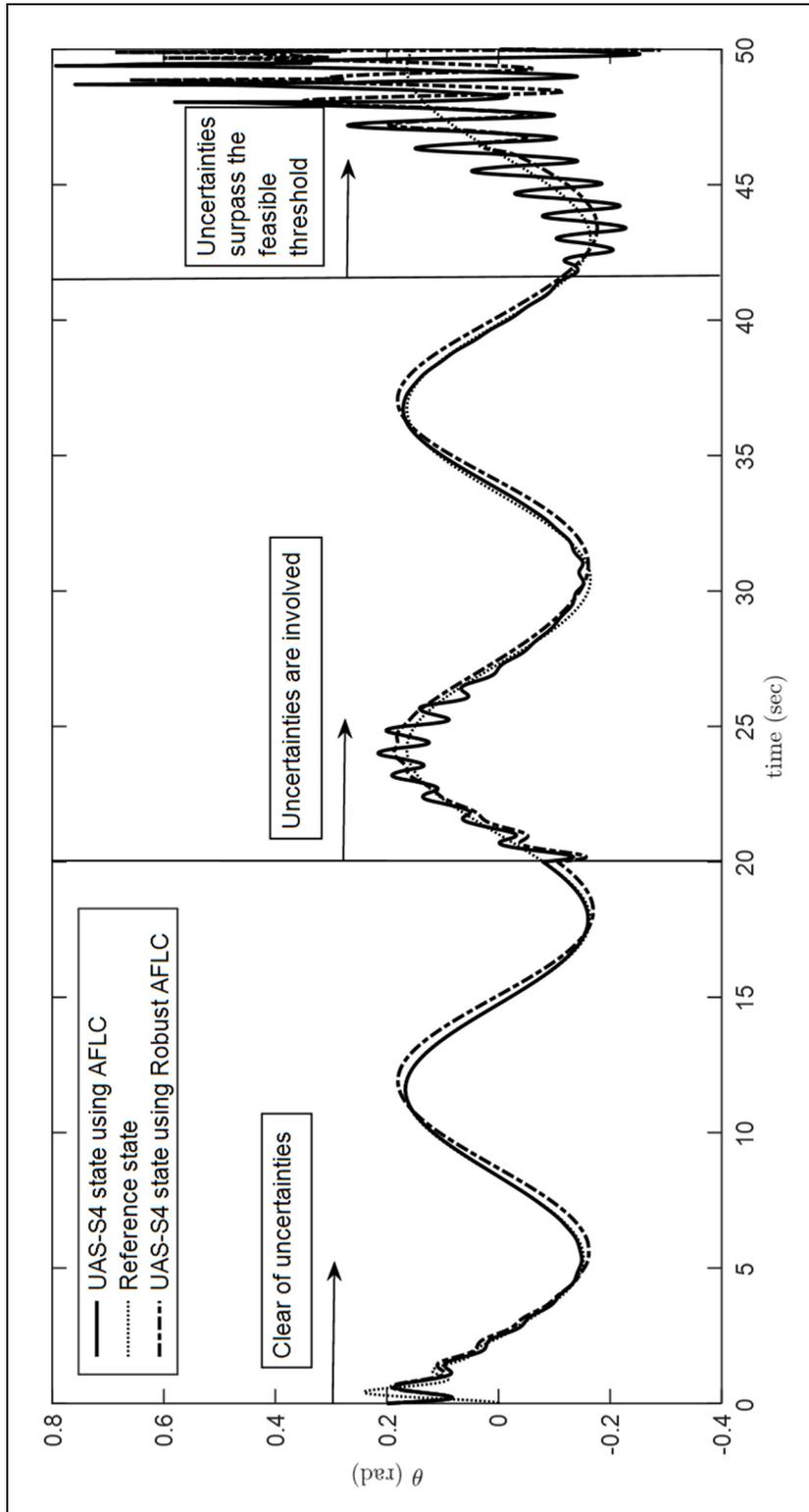


Figure 4.10 Comparing the AFLC with the Robust AFLC (RAFLC) in terms of reference model tracking for different uncertainties situations (from none to unbounded)

The second region (20 – 42 sec) depicts the controller performance when uncertainties (unknown controller's parameters, external disturbances and modeling errors) are considered in which $f(\epsilon, d) = 0.05 \cos 0.9t + 0.01 \cos 7t$, and indicates that the controller does manage the bounded uncertainties after a short initial adjustment period. The third region (42 – 50 sec) illustrates the state variables trajectories when the uncertainties surpass their boundaries, and therefore, the controller can not guarantee the stability and convergence of the UAS-S4 state variables, as the uncertainties moved the state variables outside the feasible region.

For comparison purposes, the RAFLC and AFLC approaches were chosen in the flight dynamics control algorithm. As seen on Figure 4.10, when uncertainties were due to the external disturbances and model imperfection, the RAFLC approach could track the reference state variable with less fluctuations than the AFLC approach. The average time delays for the RAFLC and AFLC approaches were 0.3 sec and 0.01 sec, respectively. In real-time operations, these average time delays are acceptable. Therefore, it can be concluded that the RAFLC outperformed the AFLC, and has provided a stabler flight in presence of uncertainties.

In addition to the above approaches for controller performance evaluation, the controller effectiveness can be assessed based on the tracking error value. In this approach, the differences between the controlled UAS-S4 and its reference model state variables are measured; they are further considered for evaluation the RAFL controller performance. In details, by considering the sampling time (0.01 seconds), the Sum of Absolute Tracking Errors (SATE) while the controlled UAS-S4 state variables are tracking the reference model state variables (during 40 seconds) characterizes the “performance index”.

The SATEs for two types of reference models in three trim conditions are represented on Tables 4.2 - 4.4. Concretely, each individual reference model was stabilized using the LQR procedures by determining both proper weighting matrices (Q and R). The stabilized reference model by assuming $Q = 1$ and $R = 1$ is named “moderate”, and the stabilized reference model by assuming $Q = 50$ and $R = 1$ is named “rigorous”.

Table 4.2 Sum of Absolute Tracking Errors (*time = 40 sec* and *sampling time = 0.01 sec*) while the controlled UAS-S4 state variables are tracking the reference model states

Flight condition	Considered uncertainties	Reference model	Sum of Absolute Tracking Errors (SATE)	
			Pitch angle (<i>rad</i>)	Pitch rate (<i>rad/s</i>)
Altitude = 6,100 <i>m</i> Speed = 45 <i>m/s</i> Mass = 53 – 55 <i>kg</i>	without uncertainty	Moderate	14.55	13.32
		Rigorous	15.16	13.81
	unknown controller's parameters	Moderate	25.62	24.53
		Rigorous	27.03	25.99
	unknown controller's parameters, external disturbances, and model imperfection	Moderate	33.56	30.85
		Rigorous	40.52	33.57
Altitude = 3000 <i>m</i> Speed = 39 <i>m/s</i> Mass = 65 – 67 <i>kg</i>	without uncertainty	Moderate	14.71	13.45
		Rigorous	15.33	13.94
	unknown controller's parameters	Moderate	25.82	24.79
		Rigorous	27.27	26.31
	unknown controller's parameters, external disturbances, and model imperfection	Moderate	33.92	31.26
		Rigorous	40.88	33.98
Altitude = 100 <i>m</i> Speed = 26 <i>m/s</i> Mass = 75–77 <i>kg</i>	without uncertainty	Moderate	14.84	13.66
		Rigorous	15.46	14.05
	unknown controller's parameters	Moderate	25.98	24.95
		Rigorous	27.46	26.36
	unknown controller's parameters, external disturbances, and model imperfection	Moderate	34.09	31.44
		Rigorous	41.34	34.34

According to the recorded tracking error for both pitch angle and pitch rate, it can be inferred that there is a proportional relationship between the SATE value and the reference model rigorousness. When the reference model is tuned such that, it strictly concerns fast time-domain response, tracking the reference model state variables becomes more difficult for RAFL controlled UAS-S4, and consequently, the tracking accuracy decreases.

Another observation is that although the robust adaptive fuzzy controller can guide the UAS-S4 state variables to track very well the reference model state variables, its accuracy degrades when large uncertainties occur. This inference is obtained from Table 4.3 that lists the SATEs for different uncertainties $f(\epsilon, d)$ in three flight conditions, and for two types of reference models.

Table 4.3 Sum of Absolute Tracking Errors ($time = 40\ sec$ and $sampling\ time = 0.01\ sec$) while the controlled UAS-S4 state variables are tracking the reference model state variables in the presence of various uncertainties

Flight condition	Uncertainties $f(\epsilon, d)$	Reference model	Sum of Absolute Tracking Error (SATE)	
			Pitch angle (rad)	Pitch rate (rad/s)
Altitude = 6,100 m Speed = 45 m/s Mass = 53 – 55 kg	0.05 sin 0.9t + 0.01 cos 7t	Moderate	26.67	25.62
		Rigorous	28.34	27.34
	0.07 sin 0.9t + 0.01 cos 7t	Moderate	28.35	26.46
		Rigorous	30.21	28.11
Altitude = 3000 m Speed = 39 m/s Mass = 65 – 67 kg	0.05 sin 0.9t + 0.01 cos 7t	Moderate	26.96	25.89
		Rigorous	28.61	27.63
	0.07 sin 0.9t + 0.01 cos 7t	Moderate	28.66	26.75
		Rigorous	30.49	28.52
Altitude = 100 m Speed = 26 m/s Mass = 75–77 kg	0.05 sin 0.9t + 0.01 cos 7t	Moderate	27.08	26.01
		Rigorous	28.75	27.73
	0.07 sin 0.9t + 0.01 cos 7t	Moderate	28.72	26.84
		Rigorous	30.60	28.51

For instance, in the second flight condition (Altitude = 3000 m, Speed = 39 m/s, Mass = 65 – 67 kg), by considering “rigorous” reference model ($Q = 50$ and $R = 1$), the pitch rate SATE is $27.63\ rad/s$ for smaller uncertainties (0.05 sin 0.9t), and the pitch rate SATE is $28.52\ rad/s$ for larger uncertainties (0.07 sin 0.9t).

Finally, according to the RAFLC architecture, adaptation weights are assigned to the adaptive laws in order to regulate the RAFL controller gain. The SATE for different adaptation weights values are listed in Table 4.4.

Table 4.4 Sum of Absolute Tracking Errors ($time = 40\ sec$ and $sampling\ time = 0.01\ sec$) the controlled UAS-S4 state variables are tracking the reference model state variables in the presence of uncertainties for different adaptation weight values

Flight condition	Adaptation weights	Reference model	Sum of Absolute Tracking Error (SATE)	
			Pitch angle (rad)	Pitch rate (rad/s)
Altitude = 6,100 m Speed = 45 m/s Mass = 53 – 55 kg	$\gamma_1 = \gamma_2 = 0.001$	Moderate	26.67	25.62
		Rigorous	28.34	27.34
	$\gamma_1 = \gamma_2 = 0.0001$	Moderate	25.96	24.91
		Rigorous	27.68	26.69
Altitude = 3000 m Speed = 39 m/s Mass = 65 – 67 kg	$\gamma_1 = \gamma_2 = 0.001$	Moderate	26.96	25.89
		Rigorous	28.61	27.63
	$\gamma_1 = \gamma_2 = 0.0001$	Moderate	26.25	25.22
		Rigorous	27.93	26.95
Altitude = 100 m Speed = 26 m/s Mass = 75 – 77 kg	$\gamma_1 = \gamma_2 = 0.001$	Moderate	27.08	26.01
		Rigorous	28.75	27.73
	$\gamma_1 = \gamma_2 = 0.0001$	Moderate	26.35	25.33
		Rigorous	28.04	26.98

Table 4.4 shows that small values for the weights of adaptation laws result in lower SATE. For instance, in the first flight condition (Altitude = 6,100 m, Speed = 45 m/s, Mass = 53 – 55 kg), by considering “Moderate” reference model ($Q = 1$ and $R = 1$), the pitch angle SATE is 26.67 rad if the adaptation weights are small ($\gamma_1 = \gamma_2 = 0.0001$) if the adaptation weights are large ($\gamma_1 = \gamma_2 = 0.001$), and pitch angle SATE is 25.96 rad if the adaptation weights are small ($\gamma_1 = \gamma_2 = 0.0001$). However, these weights must be carefully tuned, as if they would be too-small, they could cause the UAS-S4 state variables to drift outside the feasible region.

4.5 Conclusion

A Robust Adaptive Fuzzy Logic (RAFL) flight dynamics controller was designed for Hydra Technologies UAS-S4 Ehecattl. The UAS-S4 was mathematically modeled using the Takagi-Sugeno fuzzy logic method to design its corresponding controller. Adaptive gains were assigned to the fuzzy controller to ensure that it could perform very well despite uncertainties. For the adaptive control mechanism, a reference model was defined, which was stabilized

through the LQR method. The numerical results show that there is an inverse relationship between the reference model rigorousness and the RAFL controller performance. When the controlled UAS-S4 state variables track the reference model state variables, the tracking errors increase if the reference model strictly determines ideal time-domain response properties, such as rise-time or settling-time. The tuneable controller gains were adjusted utilizing Lyapunov-based adaptation laws, that became robust against uncertainties. The controller's performance was evaluated in terms of reference model state variables tracking for a variety of uncertainties. In-line with the requirements for cruise conditions, the RAFL controller was able to stabilize the UAS-S4 lateral and longitudinal flight dynamics, as well as the reference model state variables; the tracking error converged to zero. In addition, Sum of Absolute Tracking Errors (SATE) results proved that the RAFL controller could handle uncertainties that were due to the controller's unknown parameters, modeling errors, and external disturbances. Small values for the weights of adaptation laws resulted in lower SATE. Based on numerical studies, for higher values of uncertainties, the controller performance degraded slightly; however, the controller could maintain the UAS-S4 state variables in the asymptotically stable region. The robust control algorithms showed that if the uncertainties surpass their boundaries, the controller cannot guarantee the reference model state variables' tracking. For further studies, we recommend the RAFL controller development by utilizing a fuzzy logic reference model to improve the RAFLC efficiency in order to reduce reference state variables tracking error.

CHAPTER 5

A Novel Flight Dynamics Modeling using Robust Support Vector Regression against Adversarial Attacks

Seyed Mohammad Hashemi ^a, Ruxandra Mihaela Botez ^b

^{a,b} Department of System Engineering, École de Technologie Supérieure, Université du Québec, 1100 Notre Dame West, Montréal, QC, H3C 1K3, Canada

This article was submitted for publication in the *SAE International Journal of Aerospace*
On November 23, 2021

Résumé

Un modèle de dynamique de vol (FDM) précis du système aérien sans pilote (UAS) nous permet de concevoir un contrôleur efficace dans ses premières phases de développement et d'augmenter la sécurité tout en réduisant les coûts. Les tests en vol sont normalement effectués pour un nombre préétabli de conditions de vol, puis des méthodes mathématiques sont utilisées pour obtenir le FDM pour l'enveloppe de vol. Pour notre UAS-S4 Ehecattl, 216 FDM locaux correspondant à différentes conditions de vol ont été utilisés pour créer son modèle de dynamique de vol de planification linéaire locale (LLS-FDM). Les données initiales de l'enveloppe de vol contenant 216 FDM locaux ont été augmentées à l'aide de méthodologies d'interpolation et d'extrapolation, augmentant ainsi le nombre de FDM locaux compensés de 216 jusqu'à 3 642. En s'appuyant sur cet ensemble de données augmenté, la méthodologie Support Vector Machine (SVM) a été utilisée dans l'algorithme de régression d'analyse comparative en raison de ses excellentes performances lorsque les échantillons d'apprentissage ne pouvaient pas être séparés d'une manière linéaire. La régression de vecteur de support (SVR) entraînée a prédit le FDM pour l'ensemble du domaine de vol. Même si le SVR-FDM ait montré d'excellentes performances, il est resté vulnérable aux attaques adverses. Par conséquent, nous l'avons modifié à l'aide d'un algorithme de défense de reconversion contradictoire en le transformant en un SVR-FDM robuste. Pour les études de validation, la

qualité du FDM de l'UAS-S4 prédit a été évaluée en se basant sur le diagramme du lieu des racines (Root Locus). La proximité des valeurs propres prédites vers les valeurs propres d'origine a confirmé la très grande précision du SVR-FDM de l'UAS-S4. La précision de la prédiction du SVR a été évaluée pour 216 conditions de vol, différents nombres de voisins, et des diverses fonctions du noyau ont également été prises en compte. De plus, les performances de régression ont été analysées en se basant sur la réponse échelonnée des variables d'état dans l'architecture de contrôle en boucle fermée. Le SVR-FDM a fourni le temps de montée et le temps de stabilisation les plus courts, mais il a échoué lorsque des attaques contradictoires ont imposé le SVR. Cependant, les propriétés de réponse pas à pas du Robust-SVR-FDM ont montré qu'elle pouvaient fournir des résultats plus précis que l'approche LLS-FDM, tout en protégeant le contrôleur des attaques adverses.

Abstract

An accurate Unmanned Aerial System (UAS) Flight Dynamics Model (FDM) allows us to design its efficient controller in early development phases, and to increase safety while reducing costs. Flight tests are normally conducted for a pre-established number of flight conditions, and then mathematical methods are used to obtain the FDM for the entire flight envelope. For our UAS-S4 Ehecatl, 216 local FDMs corresponding to different flight conditions were utilized to create its Local Linear Scheduling Flight Dynamics Model (LLS-FDM). The initial flight envelope data containing 216 local FDMs was further augmented using interpolation and extrapolation methodologies, thus increasing the number of trimmed local FDMs up to 3642. Relying on this augmented dataset, the Support Vector Machine (SVM) methodology was used as a benchmarking regression algorithm due to its excellent performance when training samples could not be separated linearly. The trained Support Vector Regression (SVR) predicted the FDM for the entire flight envelope. Although the SVR-FDM showed an excellent performance, it remained vulnerable against adversarial attacks. Hence, we modified it using an adversarial retraining defense algorithm by transforming it into a Robust SVR-FDM. For validation studies, the quality of predicted UAS-S4 FDM was evaluated based on the Root Locus diagram. The predicted eigenvalues' closeness to the original eigenvalues confirmed the high accuracy of the UAS-S4 SVR-FDM. The SVR

prediction accuracy was evaluated at 216 flight conditions, for different numbers of neighbours, and a variety of kernel functions were also considered. In addition, the regression performance was analyzed based on the state variables' step response in the closed-loop control architecture. The SVR-FDM provided the shortest rise-time and settling time, but it failed when adversarial attacks imposed the SVR. However, the Robust-SVR-FDM step response properties showed that it could provide more accurate results than the LLS-FDM approach, while protecting the controller from adversarial attacks.

5.1 Introduction

Unmanned Aerial Systems (UASs) have been successfully utilized mainly for surveillance and reconnaissance (Ebeid, Skriver, Terkildsen, Jensen, & Schultz, 2018; Saggiani et al., 2007). The fast-growing demand for UASs highlights the need of special attention to the safety and efficiency of such critical systems (Alzahrani, Oubbati, Barnawi, Atiquzzaman, & Alghazzawi, 2020). According to the aviation transportation industry, the advancement of flight dynamics modelling and control is one of the most important factors needed to improve the safety and efficiency of UASs (Stengel, 2015; Vega et al., 2020). Hence, our goal is to design such an intelligent algorithm for accurate flight dynamics modelling that improves the corresponding controller performance.

The UAS-S4 Ehecatl (designed and manufactured by Hydra Technologies company) was utilized to evaluate and experimentally validate the developed FDM algorithm and controller (R. Botez, 2018). Figure 5.1 shows the Hydra Technologies' UAS-S4 Ehecatl.



Figure 5.1 Hydra Technologies UAS-S4 Ehecattl

This UAS is equipped with elevators, ailerons, and rudders that control moments around the pitch, roll, and yaw axes. Table 5.1 lists its main specifications.

Table 5.1 The UAS-S4 geometrical and flight data Specification

Specifications	Values
Wingspan	4.2 m
Wing area	2.3 m ²
Total length	2.5 m
Mean aerodynamic chord	0.57 m
Empty weight	50 kg
Maximum take-off weight	80 kg
Loitering airspeed	35 knots
Maximum speed	135 knots
Service ceiling	15000 ft
Operational range	120 km

In this context, the "model" refers to the mathematical representation of the UAS-S4 Flight Dynamics Model (FDM). UAS modeling is crucial, as its FDM is used for the design and development of a flight dynamics controller. Having access to an UAS-S4 FDM enhances our ability to evaluate its controller performance in the early stage of its development; which allowed us to increase its flight safety while reducing flight costs (Chabir et al., 2016; Ghazi, Botez, & Achigui, 2015; Q. Wang & Stengel, 2004; Zhou et al., 2021).

The UAS-S4 FDM was obtained based on 216 flight cases (M. Kuitche, Segui, Botez, & Ghazi, 2017; M. A. J. Kuitche & Botez, 2019; M. A. J. Kuitche, Botez, Guillemain, & Communier, 2020a, 2020b) using the equipment including an UAS-S4, a wind-tunnel (R. Botez, 2018), and a level D research aircraft flight simulator (R. M. Botez et al., 2015). The flight envelope consisted of 216 local FDMs, where each FDM designed for a specific range of altitude, speed, and mass. The trimmed local FDMs were represented based on the state-space system, linearized around their corresponding equilibrium points. Given that local FDMs' accuracy decreases with the increasing distance with respect to their equilibrium points, a proper regression can thus provide a more precise FDM for the entire flight envelope (Mosbah, Botez, & Dao, 2016).

For the regression problems, data-driven algorithms have shown great efficiency in a variety of applications (Andrianantara, Ghazi, & Botez, 2021; Hashemi et al., 2020b; Segui & Botez, 2021). Despite their excellent performance in terms of regression accuracy, they have disadvantages related to their high dependency on a large quantity of training data. For data-driven regression using small-sized datasets, training data augmentation is required (Santi, Ceruti, Liverani, & Osti, 2021; Van Dyk & Meng, 2001). With this aim, “interpolation” and “extrapolation” are the most well-known practical methods for data augmentation, and both of them operate based on the k-nearest neighbors principal (Giridhara, Mishra, Venkataramana, Bukhari, & Dengel, 2019). Relying on an augmented dataset, data-driven algorithms can safely be employed for FDM regression. In this study, we utilized the Support Vector Regression (SVR) methodology as the benchmarking algorithm. Because it has excellent ability when training samples can not be separated linearly (Awad & Khanna, 2015). Figure 5.2 shows this procedure for developing the UAS-S4 Flight Dynamics Model (FDM).

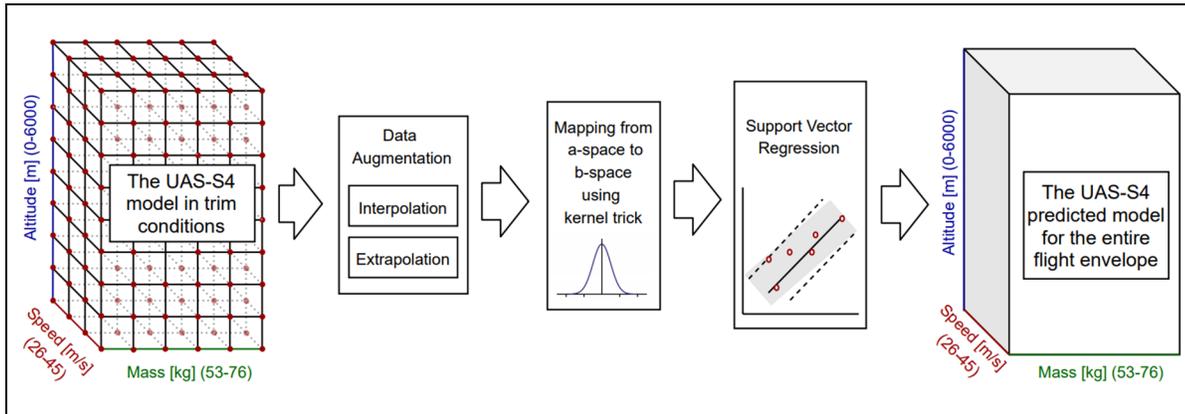


Figure 5.2 The procedure to obtain the UAS-S4 FDM for its entire flight envelope

As shown in Figure 5.2, a flight envelope containing 216 local state-space FDMs corresponding to various trim flight conditions (red nodes) for 9 altitudes (0 – 6000 m), 6 masses (53 – 76 kg), and 4 speeds (26 – 45 m/s) was constructed, and then, the number of local FDMs was increased from 216 to 3642 using interpolation and extrapolation methodologies in the data augmentation block. Next, kernel functions were used to map the data from a low dimensional space into a high dimensional space in which the data were linearly separated using support vectors. Finally, the trained SVR predicted the UAS-S4 flight dynamics model for the entire flight envelope.

Although the need of a large-scale dataset (a minor challenge) can be solved through data augmentation, security attacks (a major challenge) are potential threats for data-driven regression algorithms (Q. Liu et al., 2018). In fact, it is possible to mislead a neural network-based FDM using adversarial data, generated by carefully manipulating original data. This deception operation is known as an “Adversarial Attack” in the Artificial Intelligence (AI) community (S. Huang, Papernot, Goodfellow, Duan, & Abbeel, 2017). In our work, in case of adversarial attacks, the SVR provided a wrong local FDM for the controller in the supposed flight conditions. The controller then generates incorrect commands (for the control surfaces) which results in the UAS instability and collapse (Tuzcu et al., 2007). Development of an effective defense algorithm against adversarial attacks on SVR-based FDMs becomes therefore a priority for reliable UASs.

Four original contributions of our research paper are:

1. The design of a data augmentation procedure to prepare a large-scale dataset for flight dynamics modelling using any data-driven regression algorithm;
2. The design and testing of a data-driven regression algorithm for accurate flight dynamics modelling when training samples could not be separated linearly;
3. A realistic hacking of an UAS controller by imposing adversarial attacks on its neural network-based FDM; and
4. The design of an effective defense algorithm to render the FDM and controller robust against adversarial attacks.

This paper is composed of five sections beginning with the introduction. The UAS-S4 is modelled in Section 5.2 using scheduled local FDMs (under different flight conditions) which are then augmented through interpolation and extrapolation procedures. The SVR model was designed, fine-tuned, and trained for the UAS-S4 FDM regression in order to predict its FDM in any flight condition. Section 5.3 introduces the concept of adversarial attacks and discusses it from both white-box and black-box aspects. After imposing an adversarial attack on the UAS-S4 SVR-based FDM, we show how to render it robust by designing a defense algorithm using adversarial samples. Section 5.4 presents the results, and shows the improved UAS-S4 FDM accuracy obtained by relying on the SVR and details the SVR sensitivity to adversarial attacks. This section also evaluates the efficiency of the defense algorithm against adversarial attacks based on the performance of a controller for the UAS-S4 FDM. Finally, a comprehensive conclusion and the opportunities for future works are discussed in Section 5.5.

5.2 The UAS-S4 Flight Dynamics Model (FDM)

5.2.1 Local Linear Scheduled Flight Dynamics Model (LLS-FDM)

By considering the UAS-S4's differential equations of lateral and longitudinal motion (Caughey, 2011), its flight dynamics can be linearly modeled around its equilibrium points for 216 flight conditions. The state variables including the axial velocity u , vertical velocity w , pitch rate q , and pitch angle θ constitute the longitudinal state vector $X_{lon} = [u \ w \ q \ \theta]^T$,

where X_{lon} is controlled by the longitudinal input vector $\delta_{lon} = [\delta_e \quad \delta_T]^T$, which consists of the elevator deflection δ_e and thrust δ_T . The state variables including the lateral velocity v , roll rate p , yaw rate η , and roll angle φ constitute the lateral state vector $X_{lat} = [v \quad p \quad \eta \quad \varphi]^T$, which is controlled by the lateral input vector $\delta_{lat} = [\delta_a \quad \delta_r]^T$, composed of the aileron and ruder deflections, respectively.

For any trim condition, the UAS-S4 Flight Dynamics Model (FDM) can be represented based on the following state-space system by utilizing the state vector X , input vector δ , and output vector Y (Nelson, 1998):

$$\begin{aligned} \dot{X}(t) &= A X(t) + B \delta(t) \\ Y(t) &= C X(t) + D \delta(t) \end{aligned} \quad (5.1)$$

where the longitudinal state-space matrices are:

$$\begin{aligned} A_{lon} &= \begin{bmatrix} G_u & G_w & 0 & -g \cos \theta_0 \\ H_u & H_w & u_0 & -g \sin \theta_0 \\ M_u + M_{\dot{w}}H_u & M_w + M_{\dot{w}}H_w & M_q + u_0H_{\dot{w}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ B_{lon} &= \begin{bmatrix} G_{\delta_e} & G_{\delta_T} \\ H_{\delta_e} & H_{\delta_T} \\ M_{\delta_e} + M_{\dot{w}}H_{\delta_e} & M_{\delta_T} + M_{\dot{w}}H_{\delta_T} \\ 0 & 0 \end{bmatrix}, C_{lon} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T, D_{lon} = 0 \end{aligned} \quad (5.2)$$

and the lateral state-space matrices are:

$$\begin{aligned} A_{lat} &= \begin{bmatrix} Y_v & Y_p & -(u_0 - Y_\eta) & g \cos \theta_0 \\ L_v & L_p & L_\eta & 0 \\ N_v & N_p & N_\eta & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ B_{lat} &= \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix}, C_{lat} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T, D_{lat} = 0 \end{aligned} \quad (5.3)$$

Based on the above modeling approach, the UAS-S4 FDM was obtained using 216 local linear state-space representations that was named the Local Linear Scheduling (LLS) FDM (Lawrence & Rugh, 1995; Sadeghzadeh, Mehta, & Zhang, 2011). Each state-space FDM

expressed the linearized state variables around a specific equilibrium point corresponding to a certain combination of altitudes, speeds and masses. To obtain a large number of local linear FDMs, the use of data augmentation methodologies is required.

5.2.2 Data Augmentation

In Artificial Intelligence (AI), data augmentation refers to a procedure to generate synthetic data that are used for training data-driven algorithms to improve their performance (Dubost et al., 2019). While there is a high number of data-augmentation techniques, interpolation and extrapolation are considered as the most well-known practical methods for solving the regression problems. Figure 5.3 shows the process of local FDMs augmentation using interpolation methodology based on the k-nearest neighbors.

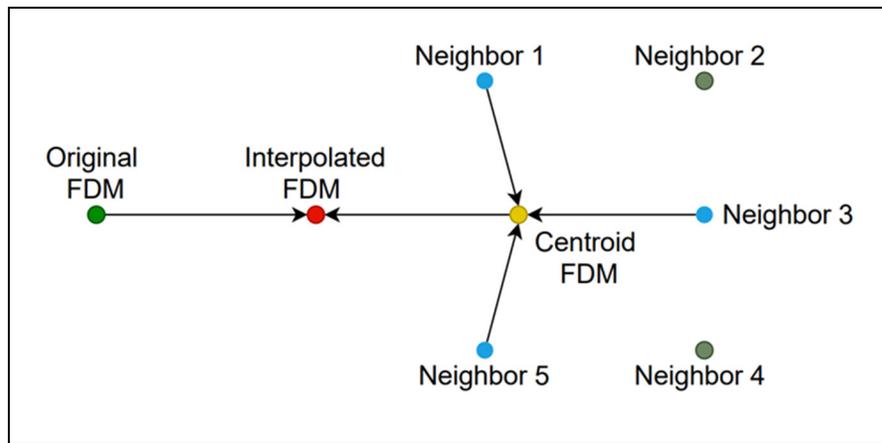


Figure 5.3 Augmenting local FDMs using an interpolation methodology based on the k-nearest neighbors (Giridhara et al., 2019)

Based on its three nearest neighbors, the centroid of the three local FDMs associated with the closest operating points embedding is computed. The new local FDM can then be generated by an interpolation between the centroid and the FDM corresponding to its original operating point. Equation (5.4) represents this generation process (Giridhara et al., 2019):

$$\bar{Z}_j = (Z_k - Z_j)\lambda_{int} + Z_j \quad (5.4)$$

where Z_k , Z_j , and \bar{Z}_j denote the computed centroid, the original, and the new local FDM embedding, respectively. λ_{int} is an adjustable factor for tuning the interpolation degree.

Following a similar methodology, data can be augmented using extrapolation. Figure 5.4 demonstrates the process for augmenting local FDMs using an extrapolation methodology based on the k-nearest neighbors.

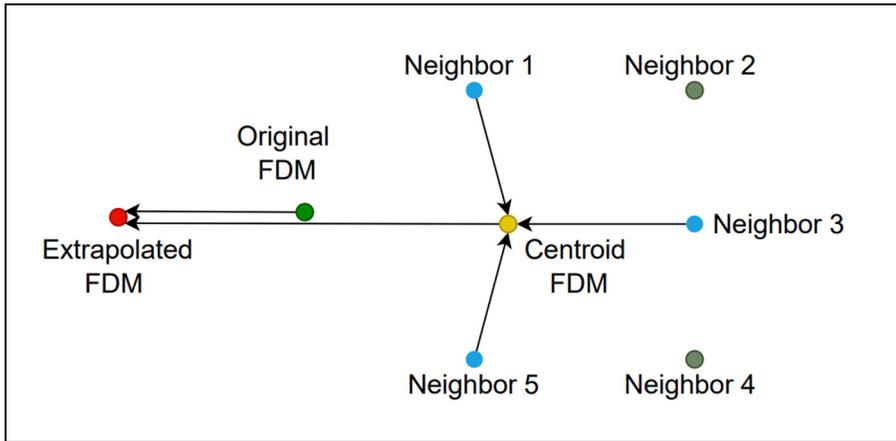


Figure 5.4 Augmenting local FDMs using an extrapolation methodology based on the k-nearest neighbors (Giridhara et al., 2019)

Using the three nearest neighbors, the centroid of the three local FDMs related to the closest operating points embedding is computed. The new local FDM can then be generated via an extrapolation between the archived centroid value and the FDM related to the original operating point, as represented in Equation (5.5) (Giridhara et al., 2019).

$$\bar{Z}_j = (Z_j - Z_k)\lambda_{ext} + Z_j \quad (5.5)$$

It should be mentioned that λ (scalar) needs to be selected carefully from its valid interpolation and extrapolation ranges, as $\lambda_{int} \in [0,1]$ and $\lambda_{ext} \in [0, \infty)$.

By use of the above-mentioned data augmentation techniques, the number of training datasets was increased from 216 to 3642. Based on the new augmented dataset, an improved regression algorithm can provide a more accurate (compared to the use of a smaller data set) FDM for any flight condition in the flight envelope.

5.2.3 Support Vector Regression (SVR) of a Flight Dynamics Model

There is no doubt regarding the excellent effectiveness of data-driven predictors in regression tasks when a large dataset is provided (Dijkstra & Veldkamp, 1988). However, small-sized datasets can be successfully enlarged using data augmentation methodologies. Relying on the augmented dataset, we utilized Support Vector Regression (SVR) as the benchmarking algorithm. It is worth noting that while the performance of an SVR depends on the dataset characteristics and its sample distributions, its computational complexity is not affected by the number of input variables or features for a dataset (input dimensionality (Cherkassky & Ma, 2004)). Besides, the SVR performs the regression task favorably when training samples cannot be separated linearly (Sebald & Bucklew, 2000). Hence, the SVR is employed for capturing data distribution.

This conventional regressor is based on the Support Vector Machines (SVM) principle, which is known as deserving “high-dimensional space” learning algorithm (Gunn, 1998). Following the “kernel trick” procedure, the SVR maps the training data into desired higher dimension space, and then learns the regression decision boundaries. A variety of kernel functions (e.g. polynomial, tanh, Gaussian) can be utilized for the mapping task (Lorenzi, Mercier, & Melgani, 2012). We carried out several experiments using these kernels and adopted the most effective one for our training data.

The SVR optimization process is represented by Equation (5.6) (D.-R. Chen et al., 2004):

$$\min \frac{1}{2} \|\Theta\|^2 \quad \text{s. t.} \begin{cases} \Theta_i x_i + b - y_i \leq \epsilon \\ y_i - \Theta_i x_i - b \leq \epsilon \end{cases} \quad (5.6)$$

where $x_i \in \mathbb{X}$ is the given input, $y_i \in \mathbb{Y}$ is the output, and Θ denotes weighting vector. Moreover, b is bias term, and ϵ adjusts the regression decision boundary, which needs to be tuned carefully. Weighting vectors are obtained during an iterative learning process, and their fine-tuning decreases the chance of overfitting (in order to avoid exactly fitting on a particular dataset and lose of generalization for other input data).

Figure 5.5 shows the trained Neural Network architecture for the UAS-S4 flight dynamics model prediction. The designed SVR learns relying on training data applied to the input layer (altitude, speed, and mass) and expected in the output containing the state matrix A and control matrix B . In details, the input neurons fully propagate the input vector into all the kernel functions that are designated for dimensional mapping. Next, the kernel functions' outputs directly update the SVR weight vector. Finally, the output layer computes A and B based on the input vector, weight vector, biased term, and decision boundaries.

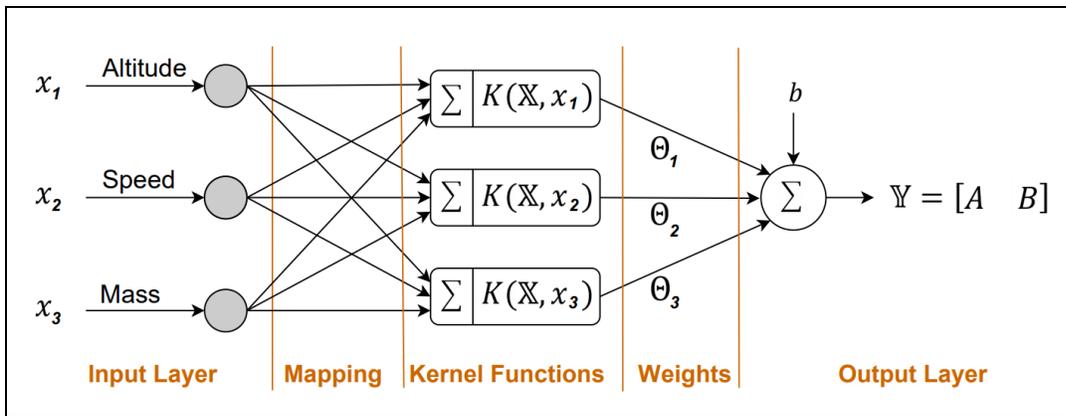


Figure 5.5 Architecture of the Trained Support Vector Regression (SVR) for the UAS-S4 Flight Dynamics Model (FDM)

Although neural network-based models can accurately perform the FDM regression task, they remain always vulnerable to adversarial attacks viewed as a potential threat. The following section describes the concept of adversarial attack, its effects on the FDM regression, and its proper countermeasure for securing the UAS-S4 data-driven FDM.

5.3 Adversarial Attacks

An “adversarial attack” is modeled as an optimization problem, solved to produce specially crafted samples that, while similar to the original samples, can deliberately mislead a prediction model (X. Huang et al., 2020). By assuming the original training samples $x_i \in \mathbb{X}$, solving the following optimization problem results in generating adversarial crafted samples:

$$\min \|x - \tilde{x}\| < \varepsilon \quad \text{s.t.} \quad \|f(x) - f(\tilde{x})\| < \gamma \quad (5.7)$$

where ε is the injected perturbation that back-propagates through the deep network and γ denotes the threshold for surpassing the eligible regression boundary.

Regarding the aim of the adversarial attack optimization problem (based on Equation (5.7)), even though the absolute value of the added perturbation is less than the random noise value, the error rate associated with the perturbation is significant, as injected perturbation exactly crafts the samples by exploiting the model's vulnerabilities (Fan et al., 2020).

Adversarial attacks can be classified based on their learning phases into two groups, known as "poisoning attacks" and "evasion attacks" (Jiang, Li, Liu, Luo, & Lu, 2020). poisoning (or causative) attacks occur during the training phase. If the dataset for training a target model is accessible, a model may be attacked by the injection of adversarial vulnerabilities. In other words, the targeted model is trained while it is sensitive to certain perturbations and so surpasses the threshold in the regression problem. Other attacks may occur during the test-phase; these are called evasion (or exploratory) attacks. Instead of manipulating a model's parameters or architecture, the evasion technique guides a model towards generating selected adversarial outputs. Since evasion-based attacks require less time and less computational effort than poisoning attacks to generate adversarial samples, they have been more common and more successful (Biggio et al., 2013). Hence, our FDM attack algorithm is designed based on the evasion procedure.

Evasion adversarial attacks can be generated using either white-box or black-box methodologies (Gil, Chai, Gorodissky, & Berant, 2019). White-box attacks use methods that require transparency of the data-driven model and its training data, while black-box attacks are used when the training set and model parameters are not available. In this way, white-box attackers generate adversarial samples by relying on their knowledge on the parameters, hyper-parameters, and architecture of the prediction model (Meng, Lin, Jung, & Wu, 2019). In this study, given our access to the training data and the SVR model details, a white-box adversarial attack methodology is used. Preparing for this type of attack is more practical for the prediction model development, as the developer can better identify model weaknesses, and can improve its robustness during the training phase. Several different white-box attack techniques have been developed for a variety of applications (Chakraborty et al., 2018; Sun, Tan, & Zhou, 2018;

Q. Wang, Liu, Xie, & Zhang, 2021). The Fast Gradient Sign Method (FGSM) as the base-line successful algorithm has been reformulated for the regression problem that can mislead FDMs.

5.3.1 Adapted Fast Gradient Sign Method (AFSGM)

The Fast Gradient Sign Method is a non-targeted attack that operates based on gradient information (Goodfellow et al., 2014). Formulation of the AFSGM for solving a regression problem can be represented as Equation (5.8) (Hashemi et al., 2020b):

$$\tilde{x} = x + \varepsilon \times \text{sign}(J(\theta, x, \sigma)) \quad (5.8)$$

where J is the cost function relying on weight vector θ , input x , and output σ , in which $\sigma \notin [a - \mu, a + \mu]$. In other words, adversarial samples are located out of the eligible band, where a denotes the bias for the eligible band while its wide is 2μ . When a float scalar perturbation ε is injected into the output layer, relying on both optimized ε and μ , the adversarial sample \tilde{x} will be generated as shown in Equation (5.8), Now we need to secure the regression model against adversarial samples through a proper defense strategy.

5.3.2 Defense against Adversarial Attacks

Generally, defense algorithms can be categorized into two aspects based on their operation mechanisms (X. Huang et al., 2020). The typical and most popular countermeasures are the adversarial training-based defenses (Mani, Moh, & Moh, 2019), In which, an expert trains the targeted model using adversarial samples that can be obtained from various optimization procedures. The other well-known defense methods rely on model structure reconfiguration (Goel, Agarwal, Vatsa, Singh, & Ratha, 2020), in which by the use of random addition of hidden layers or several parallel prediction models and their extrapolation the model mistakes are avoided. Our strategy was based on generating Adversarial samples, and then use them through the Adversarial Retraining defense methodology.

5.4 Results

The SVR performance is analyzed using different tools. As explained in Section 2, the SVR output gives the UAS-S4 FDM through the state and control matrices' prediction for any flight condition in the flight envelope. Since there are an infinite number of state-space representations for a FDM, the numerical differences between the original and predicted elements of state-space matrices cannot be used for evaluating the SVR prediction accuracy. In fact, the location of closed loop poles can represent a dynamic model behavior. Therefore, that infinite number of state space representations for a specific dynamics model can be demonstrated by a unique "Root Locus" diagram; a tool for prediction accuracy assessment. Another criterion that needs to be evaluated is the SVR regression precision. Using this criterion, the performance of the controlled UAS-S4 state variables based on the predicted FDM (provided by the SVR and Robust-SVR) can be compared to those based on the original FDM (provided by the LLS).

5.4.1 The UAS-S4 FDM Root-Locus Diagram

Assume that the UAS-S4 FDM is in the trim flight condition is expressed by the $speed = 43 \frac{m}{s}$, $altitude = 6000 m$, $mass = 53 kg$; it is important to note that this trimmed local FDM has not been used for data augmentation or as a training sample.

The original longitudinal state-space matrices under the elevator angle actuation are:

$$A_{lon} = \begin{bmatrix} -0.064 & 0.2434 & -1.087 & -9.784 \\ -0.361 & -4.261 & 43.826 & -0.251 \\ -0.136 & -1.268 & 0.4455 & -0.012 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{lonE} = \begin{bmatrix} -0.012 \\ 0.0592 \\ -0.145 \\ 0 \end{bmatrix}$$

and the predicted matrices are:

$$A_{lon} = \begin{bmatrix} -0.063 & 0.2501 & -1.095 & -9.876 \\ -0.376 & -4.362 & 41.991 & -0.269 \\ -0.137 & -1.271 & 0.4531 & -0.013 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{lonE} = \begin{bmatrix} -0.012 \\ 0.0580 \\ -0.148 \\ 0 \end{bmatrix}$$

The Root Locus diagram is used to compare the original with the predicted state-space matrices and their stability. As seen in Figure 5.6, the SVR accurately predicts the UAS-S4 trimmed FDM eigenvalues.

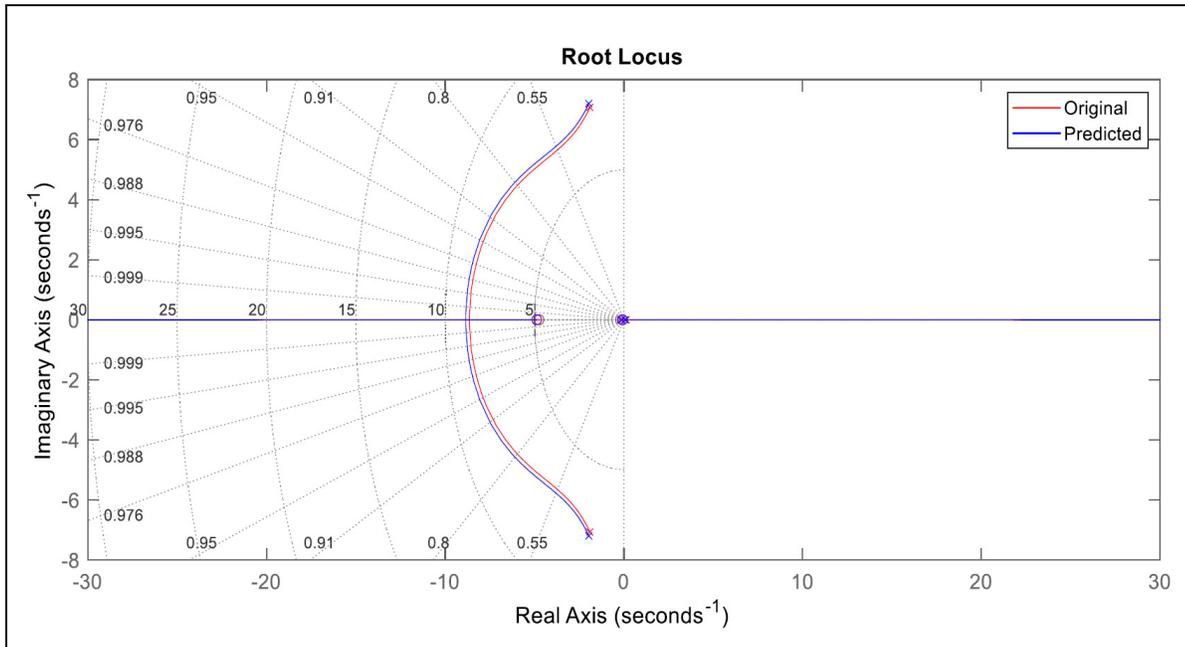


Figure 5.6 Root Locus diagram of the UAS-S4 longitudinal flight dynamics model under its elevator angle actuation

The original lateral state-space matrices under the aileron angles actuation are:

$$A_{lat} = \begin{bmatrix} -0.234 & 0.0617 & -55.24 & -9.591 \\ -0.058 & -13.99 & 0.8235 & 0 \\ 0.0943 & -0.176 & -0.642 & 0 \\ 0 & 1 & 0.0011 & 0 \end{bmatrix}, B_{lat_A} = \begin{bmatrix} 0 \\ 0.8058 \\ -0.008 \\ 0 \end{bmatrix}$$

and the predicted matrices are:

$$A_{lat} = \begin{bmatrix} -0.241 & 0.0611 & -56.05 & -9.591 \\ -0.060 & -14.23 & 0.8122 & 0 \\ 0.0955 & -0.182 & -0.649 & 0 \\ 0 & 1 & 0.0012 & 0 \end{bmatrix}, B_{lat_A} = \begin{bmatrix} 0 \\ 0.7889 \\ -0.007 \\ 0 \end{bmatrix}$$

The following Root Locus diagram shows the SVR performance in terms of the UAS-S4 lateral FDM prediction accuracy under the aileron angle actuation. As seen in Figure 5.7, the predicted eigenvalues are very close to the original ones.

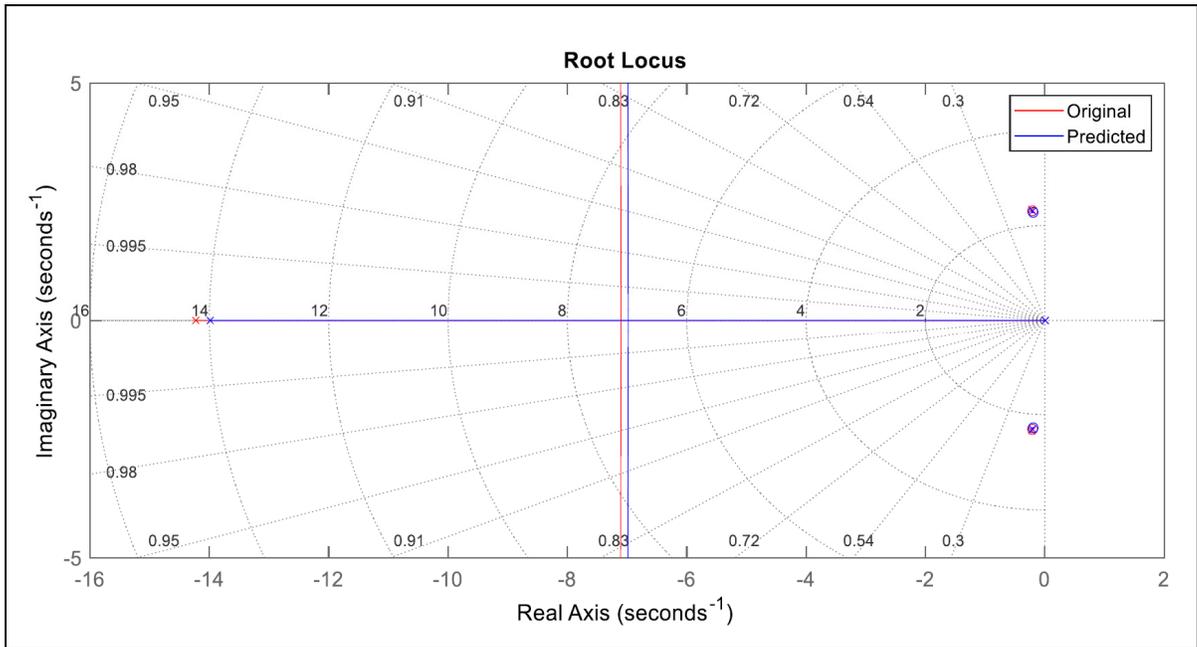


Figure 5.7 Root Locus diagram of the UAS-S4 lateral flight dynamics model under aileron angle actuation

For the lateral FDM under rudder angle actuation, the original state-space matrices are:

$$A_{lat} = \begin{bmatrix} -0.234 & 0.0617 & -55.24 & -9.591 \\ -0.058 & -13.99 & 0.8235 & 0 \\ 0.0943 & -0.176 & -0.642 & 0 \\ 0 & 1 & 0.0011 & 0 \end{bmatrix}, B_{latR} = \begin{bmatrix} -0.044 \\ 0.0091 \\ -0.199 \\ 0 \end{bmatrix}$$

and the predicted matrices are:

$$A_{lat} = \begin{bmatrix} -0.241 & 0.0611 & -56.05 & -9.591 \\ -0.060 & -14.23 & 0.8122 & 0 \\ 0.0955 & -0.182 & -0.649 & 0 \\ 0 & 1 & 0.0012 & 0 \end{bmatrix}, B_{latR} = \begin{bmatrix} -0.043 \\ 0.0082 \\ -0.196 \\ 0 \end{bmatrix}$$

Figure 5.8 shows the Root Locus diagram of the UAS-S4 lateral FDM under rudder angle actuation.

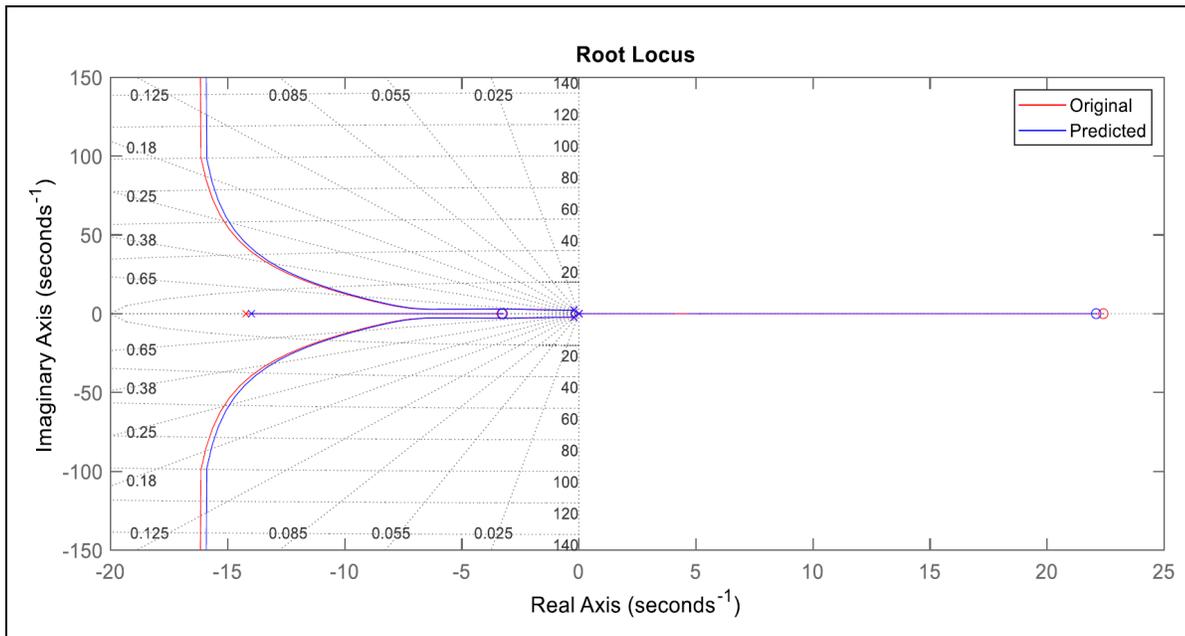


Figure 5.8 Root Locus diagram of the UAS-S4 lateral flight dynamics model under rudder angle actuation

The Root Locus diagrams in Figures 5.6, 5.7 and 5.8 graphically show the SVR's success in accurate UAS-S4 FDM eigenvalues prediction, in which the predicted eigenvalues were found to be very close to their original values. These diagrams allowed the SVR prediction accuracy to be evaluated for different flight conditions. These validation studies were performed for 5 flight conditions, as their 5 corresponding UAS-S4 FDMs have not been used for data augmentation or as training samples. As previously mentioned, the UAS-S4 FDM prediction accuracy was evaluated separately for the elevator, aileron, and rudder actuation. The Mean Absolute Error (MAE in %) was considered as a performance index calculated for the actual and the predicted eigenvalues. Table 5.2 shows the SVR prediction errors for the UAS-S4 FDM for different flight conditions and control surface actuators. As shown in Table 5.2, the SVR could accurately predict the UAS-S4 FDM for different flight conditions. Its performance was found to be slightly better for the first flight condition, where the UAV operated at the lowest altitude, speed, and mass. The Mean Absolute Error (MAE) low percentages confirm the excellent SVR prediction accuracy for a variety of flight conditions, as the MAE never exceeded 2.38%.

Table 5.2 The UAS-S4 FDM prediction accuracy using the SVR at 5 flight conditions

Flight Condition		Control Surface used for Actuation	Mean Absolute Error %
1	Altitude = 100 <i>m</i> Speed = 26 <i>m/s</i> Mass = 53 <i>kg</i>	Elevator	1.71
		Aileron	2.29
		Rudder	2.07
2	Altitude = 2250 <i>m</i> Speed = 32.3 <i>m/s</i> Mass = 57.6 <i>kg</i>	Elevator	1.74
		Aileron	2.33
		Rudder	2.11
3	Altitude = 4500 <i>m</i> Speed = 38.6 <i>m/s</i> Mass = 66.8 <i>kg</i>	Elevator	1.78
		Aileron	2.36
		Rudder	2.14
4	Altitude = 6000 <i>m</i> Speed = 43 <i>m/s</i> Mass = 76 <i>kg</i>	Elevator	1.81
		Aileron	2.38
		Rudder	2.17
5	Altitude = 6000 <i>m</i> Speed = 43 <i>m/s</i> Mass = 53 <i>kg</i>	Elevator	1.75
		Aileron	2.33
		Rudder	2.12

Since the data augmentation process used in a neighbour-based algorithm was the fundamental process for the SVR, we conducted another study to evaluate the different number of neighbours used for data augmentation. Note that the UAS-S4 FDM was considered in the trim condition, where $v=43$ *m/s*, $alt=6000$ *m* and $M=53$ *kg*. Table 5.3 shows the SVR accuracy obtained for the UAS-S4 FDM prediction when different numbers of neighbors are used in the data augmentation process.

Table 5.3 represents the UAS-S4 FDM prediction accuracy for four different numbers of neighbours under elevator, aileron, and rudder angle actuation, separately. According to the MAE values considered as the performance index, it was found that a high number of neighbours (5 neighbours compared to 4, 3, and 2 neighbours) can provide better data augmentation, and consequently more accurate FDM prediction. However, a high number of neighbours causes computational complexity in the data augmentation process. Adopting 3 nearest local FDMs in the neighbourhood of the original FDM that is supposed to be augmented is a satisfactory trade off for the interpolation and extrapolation methods.

Table 5.3 The UAS-S4 FDM prediction accuracy using SVR with different numbers of neighbors used for data augmentation

Number of neighbors for data augmentation	Control Surface used for actuation	Mean Absolute Error %
2	Elevator	1.86
	Aileron	2.45
	Rudder	2.21
3	Elevator	1.75
	Aileron	2.33
	Rudder	2.12
4	Elevator	1.71
	Aileron	2.30
	Rudder	2.07
5	Elevator	1.69
	Aileron	2.28
	Rudder	2.05

The other factor that affects the SVR performance is the type of kernel function in the hidden layer. Table 5.4 shows the UAS-S4 FDM prediction accuracy while considering a range of kernel functions in the SVR.

Table 5.4 The UAS-S4 FDM prediction accuracy while Considering a range of kernel functions for the SVR

Number of neighbors for data augmentation	Control Surface used for Actuation	Mean Absolute Error %
Polynomial $\mu = 0, \gamma = 1$	elevator	1.89
	aileron	2.47
	rudder	2.28
Gaussian $\mu = 0, \sigma^2 = 0.5$	elevator	1.75
	aileron	2.33
	rudder	2.12
RBF $\mu = 0, \sigma^2 = 0.5$	elevator	1.81
	aileron	2.39
	rudder	2.16
Sigmoidal $h = 0, C = 1$	elevator	1.83
	aileron	2.41
	rudder	2.17

Four types of kernel functions were employed: polynomial, gaussian, radial basis, and sigmoidal. Their fine-tuned parameters are described in Table 5.4. According to the calculated MAE, the SVR based on the Gaussian function gave the best performance for the UAS-S4 FDM among all the kernel functions utilized, as shown in Table 5.4. Therefore, the Gaussian-based SVR augmented data relying on 3 neighbors is considered for further studies on the regression effectiveness in the control loop.

5.4.2 Regression effectiveness under the operation of a unique LQR

According to the UAS-S4 LLS-FDM, the flight envelope contains 216 local linearized state-space representations, where each one corresponds to a specific case of altitude, speed, and mass. The accuracy of the local linear FDMs degrade when operating points move away from equilibrium points, but the FDM accuracy can be improved using nonlinear regression. Figure 5.9 graphically demonstrates this issue.

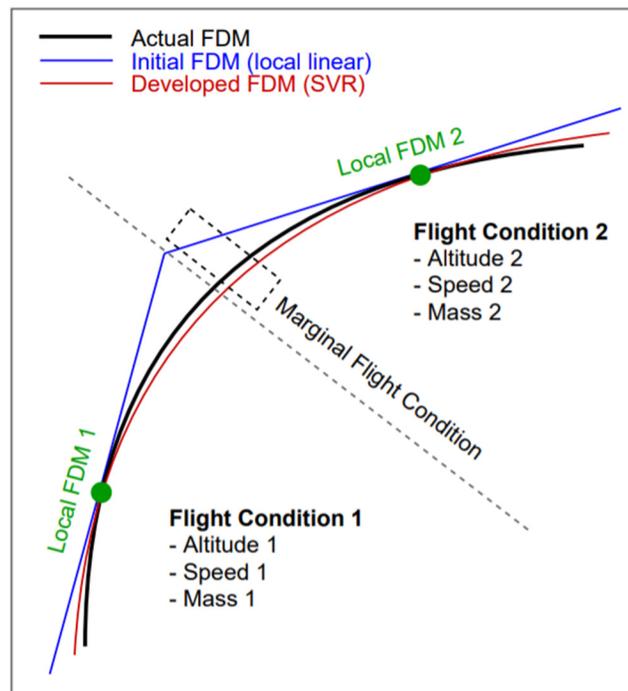


Figure 5.9 FDM design based on “Local Linear Scheduling” and “SVR” approaches

In Figure 5.9, both the “LLS” and the “SVR” approaches can provide almost the same FDM around the equilibrium points. The effectiveness of the SVR approach is the most notable in

the marginal flight condition, where it can provide a significantly more accurate FDM compared to the Local Linear approach. Following this observation, it is clear that a more accurate FDM can improve model-based controller performance. To assess the extent of these improvements, the initial (LLS) and developed (SVR) FDMs' accuracies were compared in the marginal flight condition while they were used in the same LQR control loop.

Assume that the UAS-S4 FDM is in the trim condition, where the $speed = 43 \frac{m}{s}$, $altitude = 6000m$ and $Mass = 53 kg$; its initial longitudinal LLS-FDM is represented by state-space matrices:

$$A_{LLS} = \begin{bmatrix} -0.064 & 0.2434 & -1.087 & -9.784 \\ -0.361 & -4.261 & 43.826 & -0.251 \\ -0.136 & -1.268 & 0.4455 & -0.012 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{LLS} = \begin{bmatrix} -0.012 \\ 0.0592 \\ -0.145 \\ 0 \end{bmatrix}$$

In the initial (LLS approach) flight envelope, this linearized state-space FDM was allocated to the operation range of $speed = 41.8 - 43 \frac{m}{s}$, $altitude = 5,625 - 6000 m$, $Mass = 53 - 55 kg$. We assume that the UAS-S4 is operating within the flight conditions that are at the margin of the mentioned operation range, where the $speed = 41.8 \frac{m}{s}$, $altitude = 5625 m$, $Mass = 55 kg$. When the UAS-S4 operates in this marginal flight condition, the initial state-space representation in the trim condition cannot accurately determine the flight dynamics behavior. In order to solve this problem, our designed SVR algorithm predicts the UAS-S4 FDM for any flight conditions away from the trim conditions in the flight envelope. For instance, the state-space matrices of the local UAS-S4 FDM predicted under the elevator actuation for the above-mentioned marginal flight condition are:

$$A_{SVR} = \begin{bmatrix} -0.0726 & 0.2346 & -0.9547 & -9.7830 \\ -0.3729 & -4.5992 & 43.3325 & -0.2240 \\ -0.1308 & -1.3599 & 0.4664 & -0.0118 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_{SVR} = \begin{bmatrix} -0.0133 \\ 0.0631 \\ -0.1525 \\ 0 \end{bmatrix}$$

The developed FDM (predicted by the SVR) efficiency was evaluated in control loop using the same controller as the one that was used for the UAS-S4 trimmed FDM (Yañez-Badillo et al., 2020). The Linear Quadratic Regulator (LQR) is employed as the controller in the closed-

loop architecture (Boughari, Ghazi, Botez, & Theel, 2017a, 2017b). Figure 5.10 shows the LQR controller time-domain performance during the above marginal flight conditions.

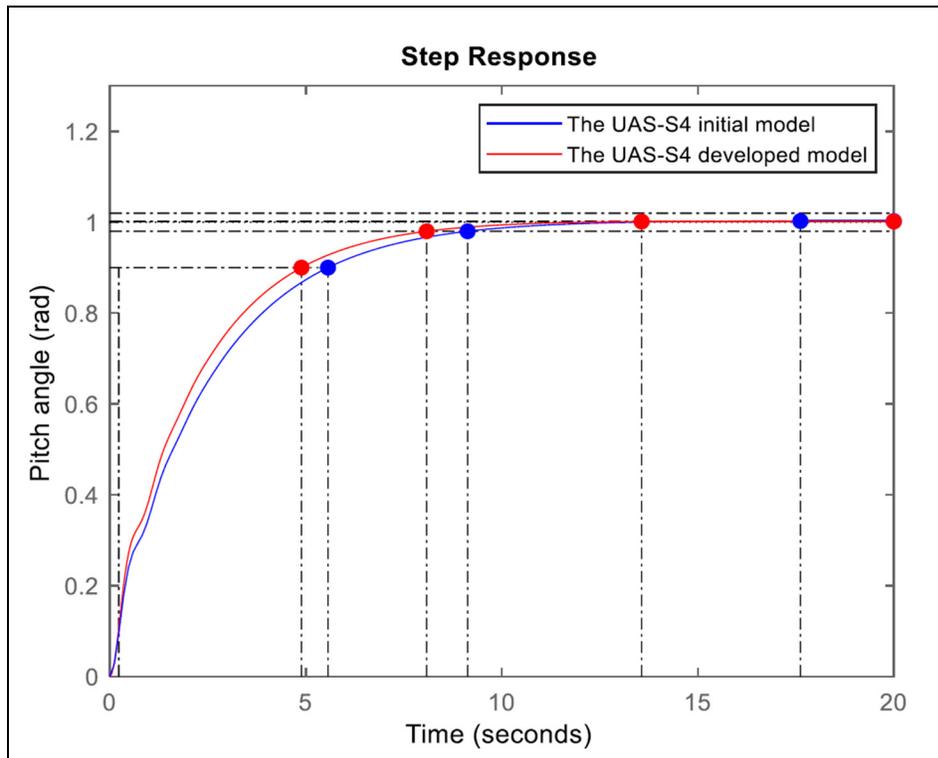


Figure 5.10 The controlled pitch angle step responses of the developed (SVR) and the initial (LLS) UAS-S4 FDMs

In this figure the controlled pitch angle step responses for the initial and the developed UAS-S4 FDMs are plotted in the marginal flight conditions. It can be observed that our designed SVR-FDM (developed model) can provide more accurate FDM, as the same LQR controller gives a better time-domain response compared to the LLS-FDM (initial model) response. Table 5.5 represents the pitch angle step response results using time-domain performance indexes for the “rise time”, the “settling time”, and “over-shoot”.

Table 5.5 The controlled step response properties of the developed and the original UAS-S4 FDM

	The developed UAS-S4 FDM	The initial UAS-S4 FDM
Rise time	4.87 sec	5.63 sec
Settling time	8.08 sec	9.13 sec
Over-shoot	0.14%	0.25%

According to Table 5.5, the SVR can provide a more accurate FDM in the marginal flight condition, as the SVR-based predicted FDM allows the same LQR controller to give a faster pitch angle step response in the marginal flight condition. In fact, for the pitch angle step response, the rise-time, settling time, and the over-shoot decreased by 0.76 sec, 1.05 sec, and 0.105% respectively, as seen in Table 5.5.

5.4.3 Robustness against Adversarial Attacks

Security issues are always a huge concern for Artificial Intelligence-based control systems. By imposing an adversarial attack on the FDM, the controller that works based on that model can be hacked. We generated an adversarial sample using the Adapted Fast Gradient Sign Method (AFGSM), and we further applied it to the UAS-S4 SVR-FDM. The adversarial sample could mislead the UAS-S4 SVR-FDM prediction. Figure 5.11 shows the effect of an imposed adversarial attack on the UAS-S4 longitudinal FDM prediction.

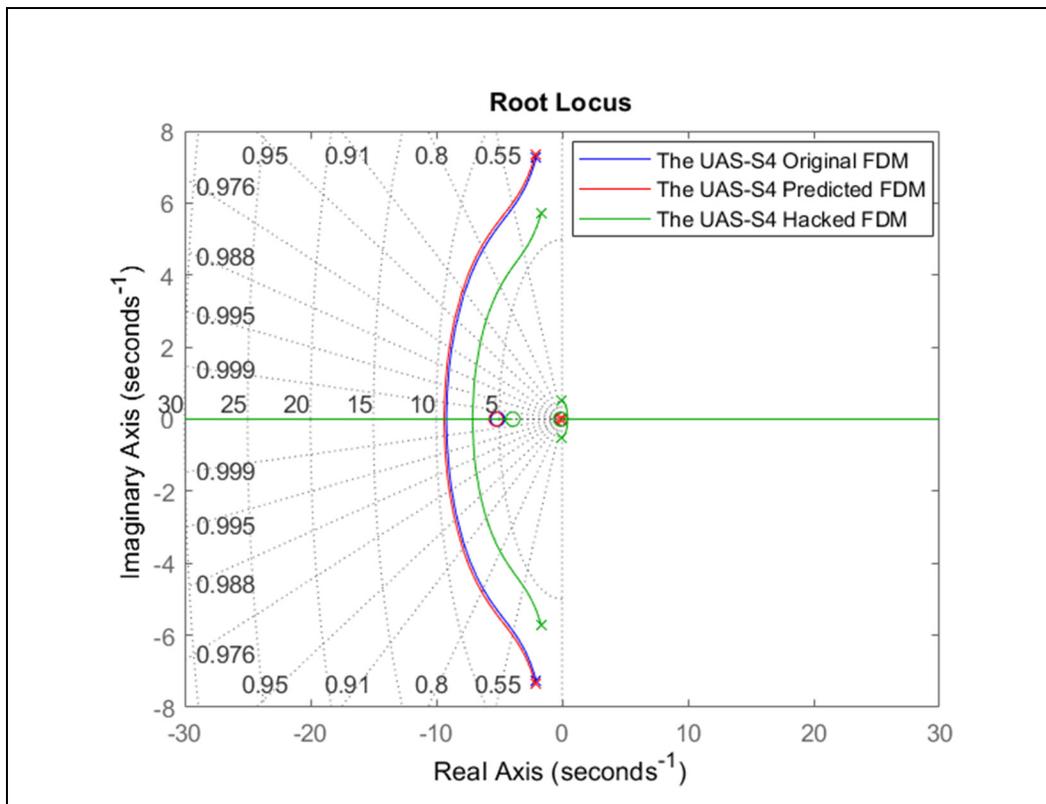


Figure 5.11 The effect of an imposed adversarial attack on the UAS-S4 longitudinal FDM prediction

According to Figure 5.11, before imposing such an adversarial attack on the SVR-FDM, the original (blue) and predicted (red) eigenvalues are very close together. However, the hacked eigenvalues (green) in the Root Locus diagram show that an adversarial sample can mislead the data-driven model, so that the SVR predicts the UAS-S4 FDM far from its original values. This incorrect prediction directly affects the FDM controller, and causes it to generate the wrong commands for the control surfaces. Assume that the UAS-S4 operates in the marginal flight condition $Speed = 42 \frac{m}{s}$, $Altitude = 5,700m$, $Mass = 55 kg$, and the LQR controls its state variables. Figure 5.12 shows the UAS-S4 pitch angle step response under LQR controller operation when different types of FDMs are used.

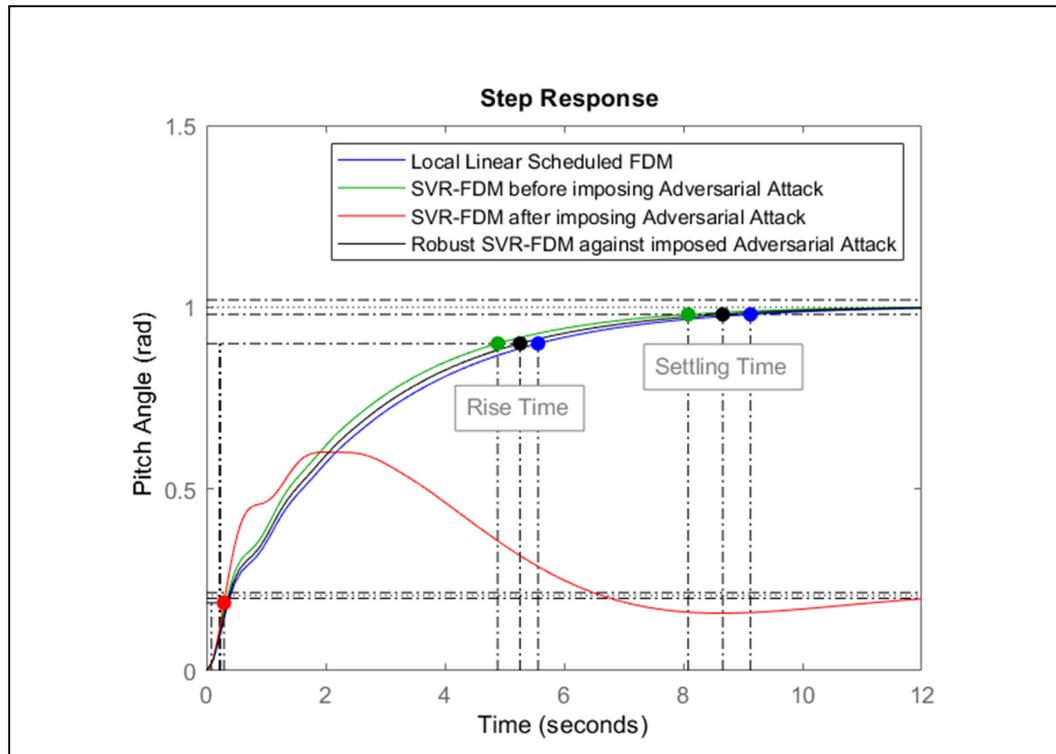


Figure 5.12 The UAS-S4 pitch angle step response using a LQR controller when different types of FDMs are used while an adversarial attack is imposed

The model-based controller shows the best performance in terms of time-domain properties when it relies on the SVR-FDM (green). However, it approached failure when an adversarial attack was imposed on the SVR-FDM (red). As shown in Figure 5.12, the Robust SVR-FDM (black) offered a better step response compared to the conventional Local Linear scheduled FDM (Blue) under an imposed adversarial attack. The time-domain properties in terms of

“rise” and “settling” times are given in Table 5.6. As seen in Figure 5.12 and Table 5.6 the SVR-FDM provided the fastest responses (4.87 sec for rise-time and 8.08 for settling time), but it failed when it approached to an adversarial sample. Meanwhile, the Robust SVR-FDM efficiency was lower than that of the SVR-FDM, and better than the LLS-FDM while remaining robust against adversarial attacks (see Table 5.6).

Table 5.6 The UAS-S4 pitch angle step response time-domain characteristics under the LQR controller when different types of FDM are used

	Rise Time (sec)	Settling Time (sec)	Adversarial Attack Examination
LLS-FDM	5.63	9.13	Pass
Robust SVR-FDM	5.29	8.64	Pass
SVR-FDM	4.87	8.08	Fail

A similar type of analysis was done for the lateral control surfaces. The UAS-S4 roll angle step responses with the LQR controller when different types of FDMs are used are presented in Figure 5.13.

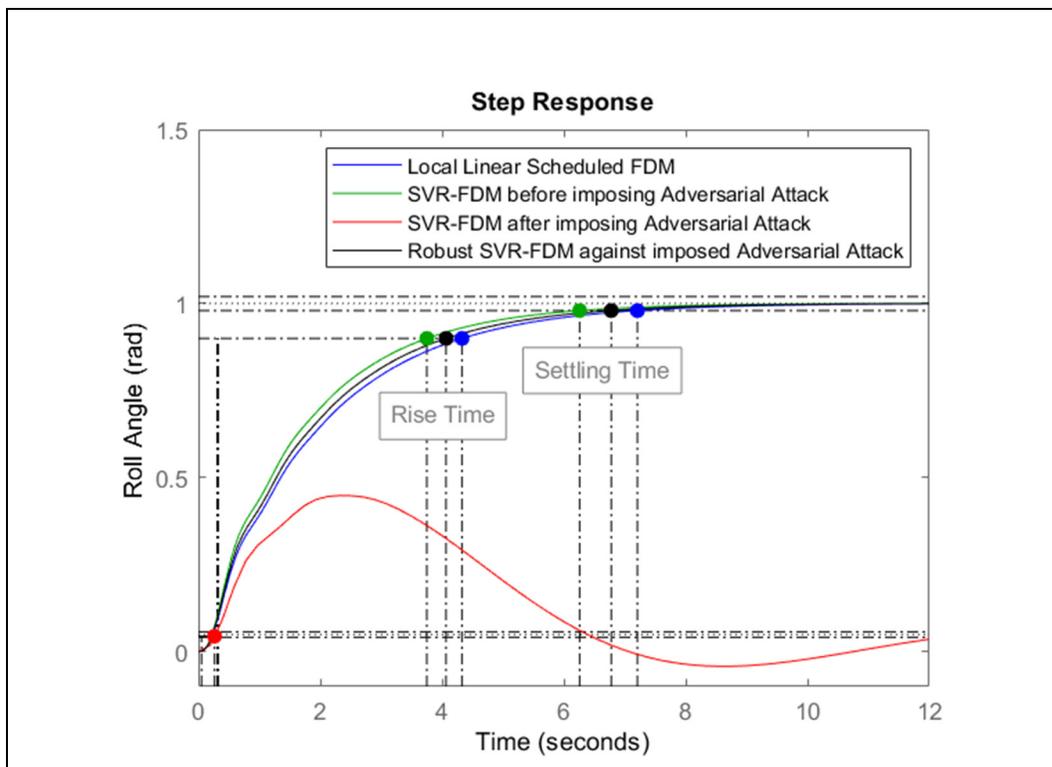


Figure 5.13 The UAS-S4 roll angle step responses using the LQR controller when different types of FDMs are used while an adversarial attack is imposed

With the SVR-FDM, the controller offers the best time-domain performance for the roll angle (green). However, it fails when is submitted to an adversarial sample (red). As shown in Figure 5.13, the Robust SVR-FDM (black) provides a faster step response compared to the conventional LLS-FDMs (blue), under an imposed adversarial attack. The time-domain properties are given in terms of rise time and settling time in Table 5.7.

Table 5.7 The UAS-S4 roll angle step response time-domain characteristics under the LQR controller when different types of FDM are used

	Rise Time (sec)	Settling Time (sec)	Adversarial Attack Examination
Local Linear FDM	4.29	7.21	Pass
Robust SVR-FDM	4.05	6.76	Pass
SVR-FDM	3.74	6.22	Fail

As indicated in Table 5.7, the SVR-FDM provided the fastest roll angle step response (3.74 sec for rise-time and 6.22 sec for settling time); however, its failure in case of adversarial attacks convinced us to improve it. The Robust SVR-FDM effectiveness was smaller than that of the SVR-FDM, but better than that of the LLS-FDM, while remaining robust against adversarial attacks.

Continuing with lateral control studies, the UAS-S4 yaw angle step response is presented in Figure 5.14, revealing that using an SVR-FDM, the LQR controller performs the best time-domain step response for the yaw angle (green). However, its failure to when perform using adversarial samples is unacceptable. In contrast, the Robust SVR-FDM (black) gave a faster step response compared to the LLS-FDMs (blue) under an imposed adversarial attack.

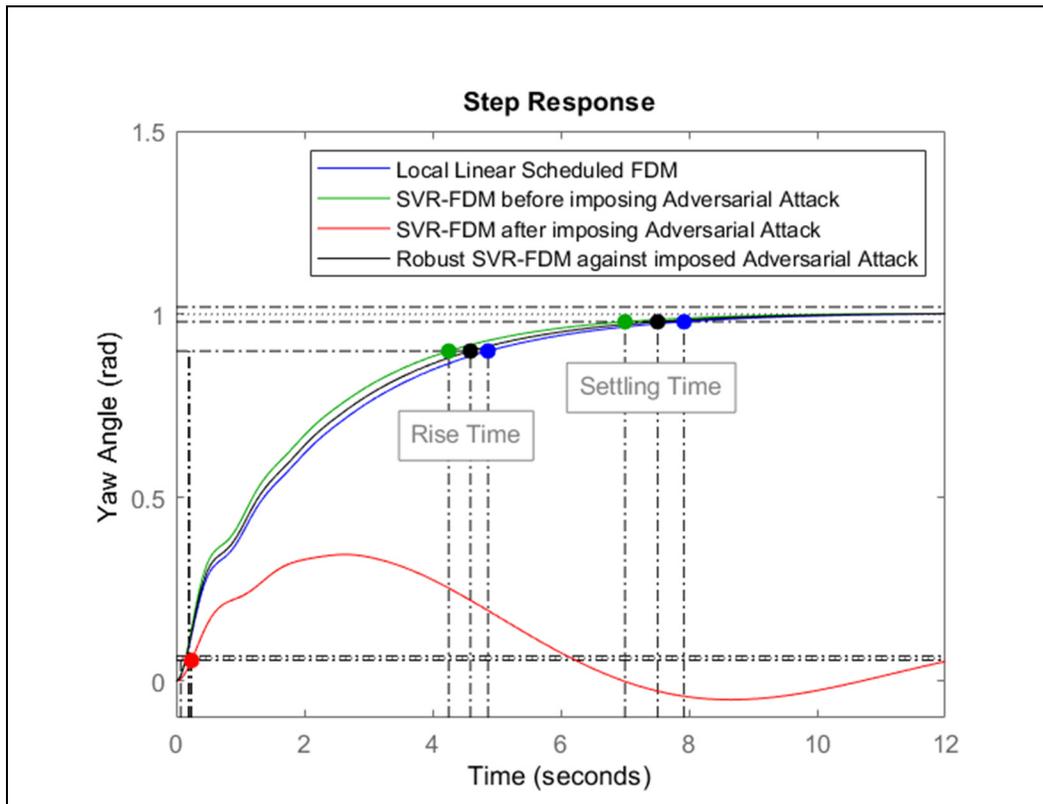


Figure 5.14 The UAS-S4 yaw angle step response with an LQR controller when different types of FDMs are used while an adversarial attack is imposed

The time-domain specifications in terms of rise time and settling time are listed in Table 5.8.

Table 5.8 The UAS-S4 yaw angle step response time-domain characteristics under the LQR controller when different types of FDM are used

	Rise Time (sec)	Settling Time (sec)	Adversarial Attack Examination
LLS-FDM	4.81	7.90	Pass
Robust SVR-FDM	4.56	7.51	Pass
SVR-FDM	4.24	6.98	Fail

As seen in Table 5.8, it is clear that using the SVR-FDM for a model-based controller resulted in the fastest yaw angle step response (4.24 sec for rise-time and 6.98 for settling time). However, its failure to perform under adversarial attacks motivated us to make more improvements. Our Robust SVR-FDM's effectiveness on yaw angle control was less than that

of the SVR-FDM and more than that of the LLS-FDM's, while it proved its robustness against adversarial attacks.

Following the lateral and longitudinal step response analyses described above, the study was performed for all marginal flight condition, and the average time-domain step response improvements of the UAS-S4 state variables are given in Table 5.9.

Table 5.9 The SVR and Robust-SVR step response improvements compared to the LLS-FDM performance for 120 marginal flight conditions.

	Average Rise Time improvement (%)			Average Settling Time improvement (%)		
	Pitch	Roll	Yaw	Pitch	Roll	Yaw
Control Surfaces Angle						
Robust SVR-FDM	6.1	5.6	5.2	5.3	6.2	4.9
SVR-FDM	13.5	12.9	11.9	11.5	13.7	11.6

Table 5.9 shows that the SVR-FDM improved time-domain step response properties much better than the Robust SVR-FDM. However, the Robust SVR-FDM could preserve the controller against adversarial attacks, as the SVR-FDM failed due to its weakness against adversarial samples.

According to the tables and figures presented above, our designed Robust-SVR outperformed the conventional flight dynamics modelling approach (Local Linear Scheduled) in terms of prediction accuracy, and also outperformed the modern data-driven flight dynamics modelling approach (Support Vector Regression) in terms of robustness against adversarial attacks.

5.5 Conclusions

In this study, 216 Local Linear Scheduling (LLS) Flight Dynamics Models (FDM) corresponding to 216 trim flight conditions were used. The flight envelope data was then augmented using interpolation and extrapolation methodologies. These methodologies gave the three closest neighbors of the original operating point supposed in the flight envelope. Next, the centroid of the embedding local FDMs was computed. Then, the new FDM was generated through interpolation and extrapolation methodologies between the centroid and the original

operating point. Following this procedure, the number of trimmed local FDMs was increased by up to 3,642. Relying on the augmented dataset, the Support Vector Regression (SVR) methodology was used as the benchmarking algorithm. The designed SVR predicted very well the UAS-S4 FDM for the entire flight envelope.

For validation studies, the initial (LLS) and the developed (SVR) UAS-S4 FDM were evaluated based on the Root Locus diagram for elevator, rudder, and aileron angle actuation. The predicted eigenvalues' closeness to the actual eigenvalues confirmed the high accuracy of the SVR-based UAS-S4 FDM. According to the Mean Absolute Errors (MAE) performance index, the excellent SVR prediction accuracy for a variety of flight conditions was confirmed, as the MAE never exceeded 2.38%. In addition, a high number of neighbours (5) compared to a lower number of neighbors (4, 3, and 2) could provide better data augmentation, and consequently more accurate FDM prediction. However, a high number of neighbours causes computational complexity for data augmentation. The SVR based on a Gaussian kernel function (with 3 neighbors) showed the best performance for the UAS-S4 FDM compared to the conditions when the kernel functions such as the RBF, Sigmoidal, and Polynomial were used.

The regression performance was then analyzed based on the pitch, roll, and yaw angle step response in closed-loop control architecture. Compared to the initial UAS-S4 FDM, our new developed UAS-S4 FDM provided more accurate local FDMs especially in marginal flight conditions. By using the new developed UAS-S4 SVR-FDM, the controller could improve the time-domain response properties for the pitch angle step response by as much as 0.76 seconds faster rise-time, 1.05 seconds faster settling time and 0.105% less over-shoot compared to the initial LLS-FDM. Moreover, the robust version of the SVR could also detect adversarial samples, and saved the controller against adversarial attacks.

CHAPTER 6

DISCUSSION OF THE RESULTS

The aim of this research study was to design and further develop the fundamental requirements for critical aerial trajectory-based operations, especially Air Traffic Management and Collision Avoidance (ATMCA) systems. Three original studies were conducted, in which novel Artificial Intelligence algorithms were developed to provide accurate Trajectory Prediction (TP), Flight Dynamics Model (FDM), and Flight Dynamics Control (FDC), respectively.

Six data-driven models were utilized to develop TP models: Logistic Regression (LR), Support Vector Regression (SVR), Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). Their performances were evaluated, and their 99.85% average regression accuracy on the air corridor revealed that our TP models gave excellent results.

Therefore, all six models are qualified for their use in trajectory-based operation systems. Meanwhile, each TP model has its own unique advantage (while some of them have their drawbacks). Among the six TP models, the LR model is the easiest in terms of implementation, but it cannot separate nonlinear samples. The SVR model implementation and its fine-tuning are complex, especially when the best kernel function must be determined, but it can separate nonlinear samples. The DNN model offers the simplest architecture among 4 deep approaches (including DNN, CNN, RNN, LSTM). The RNN model provides better performance, for datasets for which its samples are correlated. The LSTM model gives the best performance when there is a time relation between the samples. When these trade-offs between their unique abilities and architecture complexities are known, the appropriate TP model can be selected for any trajectory-based operation system.

With respect to their TP excellent performance, all the TP models were sensitive to adversarial samples. Adversarial samples caused a 100% fooling rate in all six models, while the LR and RNN models gave the lowest (0.78) and the highest (0.91) prediction confidence scores, respectively. Adversarial samples were transferable from one model to another. At least

79.16% of the total adversarial samples for the LSTM model were successfully transferable to the LR model, while 98.26% of the total adversarial samples for the RNN model were successfully transferable to the LSTM model. Finally, our designed defence algorithm based on the “Adversarial Retraining” approach using Adapted Fast Gradient Sign Method-based adversarial samples successfully secured our TP models against adversarial attacks.

By using our reliable TP models, conflicts can be detected easily, and then an efficient Flight Dynamics Control (FDC) algorithm can lead an aircraft to track a safe trajectory. Our model-based Robust Adaptive Fuzzy Logic FDC (RAFL-FDC) is designed by using the UAS-S4 Local Linear Scheduled Flight Dynamics Model (LLS-FDM) available at the LARCASE. To accurately track a trajectory, a FDC must give excellent performance in terms of flight dynamics stabilization, and in tracking state variables of the reference model. Results showed that the designed RAFL controller could stabilize the UAS-S4’s lateral and longitudinal flight dynamics very well, even in the presence of uncertainties. The RAFL controller could also track the reference model’s state variables accurately.

Following the adaptive gains of the fuzzy controller redesign, so that they would contain robust terms for handling a wide range of uncertainties, we compared the Robust-AFL (RAFL) and the AFL control approaches. In the presence of uncertainties due to external disturbances and model imperfections, the RAFL controller could track the reference model state variables with less fluctuations than the AFL controller. According to the results, the average time delays for the AFL and the RAFL controllers were 0.01 sec and 0.3 sec, respectively. These average time delays are very good in real-time operations. Therefore, it can be inferred that the RAFL controller outperformed the AFL controller, and provided a stabler flight under uncertainties.

According to the performance index based on the Sum of Absolute Tracking Errors (SATEs), the reference model state variables’ tracking accuracies were evaluated for a variety of uncertainties and for different reference models. The RAFL controller could handle larger uncertainties than the AFL controller, as long as the state variables remained in the feasible region and did not surpass the threshold defined based on the given Lyapunov stability proof. The other conclusion was that, when the reference model rigorously determined fast time-domain response, tracking the reference model state variables became more difficult for the

RAFL controller, and resulted in higher tracking error in comparison with the moderately determined reference model. In addition, selecting small values for the weights of adaptation laws resulted in lower SATEs and improved performance.

The accuracy of the FDM has a significant impact on the model-based FDC performance. Although the designed RAFL controller based on the UAS-S4 LLS-FDM showed excellent effectiveness, in the third study, a novel methodology was designed to obtain accurate FDMs. This method was based on the Support Vector Regression (SVR) application to the LLS-FDM.

The Root Locus diagram confirmed that the SVR could accurately perform the FDM regression. The performance index was based on the Mean Absolute Error (MAE), and it was obtained from the differences between the original and the predicted eigenvalues in the Root Locus diagram. The MAE percentages (%) showed the SVR excellent efficiency for a variety of flight conditions, for which they never exceeded 2.38%. Overall, the SVR provided its best performance at the lowest flight conditions expressed in terms of altitude, speed and mass.

According to the MAE, having a high number of neighbours for a local FDM, leads to a better data augmentation, and therefore to a more accurate FDM prediction. However, it significantly increases the computational complexity during the data augmentation process. Hence, a trade-off between the prediction accuracy and computational complexity can determine the appropriate number of neighbours. Relying on the MAE, the Gaussian kernel function illustrated its superiority over other kernel functions.

The SVR-FDM was utilized in a closed-loop control architecture using the Linear Quadratic Regulator (LQR) methodology. Under marginal flight conditions, the SVR-FDM allowed the LQR controller to give a step response compared to the LLS-FDM's response. For instance, for the pitch angle step response, the rise-time and the settling time decreased by 0.76 and 1.05 sec, respectively. Despite the significant improvements of time-domain properties achieved by using the SVR-FDM, the UAS-S4 SVR-FDM was sensitive to adversarial FDM samples. These adversarial samples misloaded the LQR, so that the LQR generated incorrect control commands for the actuators. The SVR-FDM was then secured against Adversarial FDM samples, using the improved model version called Robust-SVR-FDM. The results showed that

both SVR and Robust-SVR could provide more accurate FDM compared to those of the LLS approach. Although the Robust-SVR-FDM was not as accurate as the SVR-FDM, its step responses showed that it was secured against adversarial attacks.

CONCLUSION

This research was conducted to design and improve three fundamental requirements for avoiding aerial collisions. The three following studies were considered: of aircraft trajectory prediction (ATP), flight dynamics modelling (FDM), and a flight dynamics controller (FDC), that were validated on the UAS-S4 at the LARCASE.

In the first study, novel data-driven regressors were designed for trajectory prediction (TP). Although all six models could accurately predict the trajectories, deep architecture regressors (including DNN, CNN, RNN, and LSTM) showed better performance than conventional regressors (LR and SVR) in terms of regression accuracy. However, deep models were more sensitive to adversarial samples generated through the Adapted Fast Gradient Sign Method (AFGSM). Adversarial sample transferability significantly increases the importance of adversarial attacks. Hence, an adversarial retraining defence algorithm was designed to improve the six data-driven models' robustness. The results confirmed the data-driven ATP models' excellent performance in terms of prediction accuracy and adversarial attack robustness.

A model-based Fuzzy Logic FDC was designed for the UAS-S4 using its Local Linear Scheduled Flight Dynamics Model (LLS-FDM) in the second study. This Fuzzy Logic controller utilizes adaptive gains that contain robust terms. A general theorem and its stability proof were described for the designed Robust Adaptive Fuzzy Logic Controller (RAFLC). The RAFLC was evaluated in terms of state variable stabilization and reference model tracking. The results confirmed the excellent performance of the FDC at handling nonlinearities by relying on adaptive gains. The valuable impact of robust terms in the presence of uncertainties was demonstrated, as the UAS-S4 state variables could accurately track the reference model in the presence of uncertainties due to external disturbances and model imperfection.

Finally, in the third study a new methodology was developed to obtain very accurate FDMs. By considering the available LLS-FDM, the number of local FDMs was augmented from 216 to 3,642 through K-nearest neighbour interpolation and extrapolation methodologies. By relying on the enlarged FDM dataset, Support Vector Regression (SVR) was designed to

predict the FDM for the entire flight envelope. The validation studies based on Root Locus diagrams showed that the SVR could provide more accurate FDM than using LLS approach. In addition, the Mean Absolute Error (MAE) was considered as the performance index, and their values showed the superiority of the SVR-FDM in marginal flight conditions. Moreover, the SVR-FDM showed a strong resiliency against adversarial FDM samples. It could secure the FDC by utilizing Robust-SVR that had been reinforced by the adversarial retraining defense algorithm.

In this research designed highly reliable and accurate TP, FDM, and FDC algorithms were designed following fundamental requirements for any trajectory-based operations, such as air traffic control, path planning and optimization, and especially aerial collision avoidance.

RECOMMENDATIONS

Several recommendations for future work related to this research can be made. Starting with the first study, other types of Deep Neural Networks (DNNs), such as “Autoencoders”, could be investigated, as they have unique abilities when some part of the dataset is missing. Hence, when faced with communication or sensor failures, Autoencoders can still predict future trajectories very well. From the second study, reinforcement learning controllers could be utilized instead of the RAFL Controllers. These controllers have super-fast responses in the presence of uncertainties that have not yet been experienced. As for the third study, investigating other approaches for generating adversarial FDM samples would be worthwhile, such as the “deep fool”. This effort could improve the success rate of adversarial retraining defence algorithms when adversarial attacks would be approached.

BIBLIOGRAPHY

- Adesina, D., Adagunodo, O., Dong, X., & Qian, L. (2019). *Aircraft Location Prediction using Deep Learning*. Paper presented at the MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM).
- Al-Salami, N. M. (2009). Evolutionary algorithm definition. *American J. of Engineering and Applied Sciences*, 2(4), 789-795.
- Albaker, B., & Rahim, N. (2009). *A survey of collision avoidance approaches for unmanned aerial vehicles*. Paper presented at the 2009 international conference for technical postgraduates (TECHPOS).
- Albertos, P., Sala, A., & Olivares, M. (1998). *Fuzzy logic controllers. Advantages and drawbacks*. Paper presented at the VIII international congress of automatic control.
- Allison, P. D. (2012). *Logistic regression using SAS: Theory and application*: SAS institute.
- Alzahrani, B., Oubbati, O. S., Barnawi, A., Atiquzzaman, M., & Alghazzawi, D. (2020). UAV assistance paradigm: State-of-the-art in applications and challenges. *Journal of Network and Computer Applications*, 166, 102706.
- Amoroso, C., Liverani, A., Francia, D., & Ceruti, A. (2021). Dynamics augmentation for high speed flying yacht hulls through PID control of foiling appendages. *Ocean Engineering*, 221, 108115.
- Anderson, C., Vasudevan, R., & Johnson-Roberson, M. (2021). A Kinematic Model for Trajectory Prediction in General Highway Scenarios. *arXiv preprint arXiv:2103.16673*.
- Andersson, O., Wzorek, M., & Doherty, P. (2017). *Deep learning quadcopter control via risk-aware active learning*. Paper presented at the Thirty-First AAAI Conference on Artificial Intelligence.
- Andrianantara, R. P., Ghazi, G., & Botez, R. M. (2021). *Aircraft Engine Performance Model Identification using Artificial Neural Networks*. Paper presented at the AIAA Propulsion and Energy 2021 Forum.
- Anton, N., Botez, R. M., & Popescu, D. (2011). Stability derivatives for a delta-wing X-31 aircraft validated using wind tunnel test data. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(4), 403-416.
- Aqib, M., Mehmood, R., Alzahrani, A., Katib, I., Albeshri, A., & Altowaijri, S. M. (2019). Smarter traffic prediction using big data, in-memory computing, deep learning and GPUs. *Sensors*, 19(9), 2206.

- Åström, K. J., & Wittenmark, B. (2013). *Adaptive control*: Courier Corporation.
- Awad, M., & Khanna, R. (2015). Support vector regression. In *Efficient learning machines* (pp. 67-80): Springer.
- Ayhan, S., & Samet, H. (2016). *Aircraft trajectory prediction made easy with predictive analytics*. Paper presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
- Babaei, A. R., Mortazavi, M., & Moradi, M. H. (2011). Classical and fuzzy-genetic autopilot design for unmanned aerial vehicles. *Applied Soft Computing, 11*(1), 365-372.
- Bardela, P. A., & Botez, R. M. (2017). *Identification and validation of the Cessna Citation X engine component level modeling with flight tests*. Paper presented at the AIAA modeling and simulation technologies conference.
- Barratt, S. T., Kochenderfer, M. J., & Boyd, S. P. (2018). Learning probabilistic trajectory models of aircraft in terminal airspace from position data. *IEEE Transactions on intelligent transportation systems, 20*(9), 3536-3545.
- Bellingham, J., Richards, A., & How, J. P. (2002). *Receding horizon control of autonomous aerial vehicles*. Paper presented at the Proceedings of the 2002 American control conference (IEEE Cat. No. CH37301).
- Benavides, J. V., Kaneshige, J., Sharma, S., Panda, R., & Steglinski, M. (2014). *Implementation of a trajectory prediction function for trajectory based operations*. Paper presented at the AIAA Atmospheric Flight Mechanics Conference.
- Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., . . . Roli, F. (2013). *Evasion attacks against machine learning at test time*. Paper presented at the Joint European conference on machine learning and knowledge discovery in databases.
- Blažič, S., Matko, D., & Škrjanc, I. (2010). Adaptive law with a new leakage term. *IET control theory & applications, 4*(9), 1533-1542.
- Borello, F., Cestino, E., & Frulla, G. (2010). Structural uncertainty effect on classical wing flutter characteristics. *Journal of Aerospace Engineering, 23*(4), 327-338.
- Botez, R. (2018). Morphing wing, UAV and aircraft multidisciplinary studies at the Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE. *Aerospace Lab*(14), 1-11.
- Botez, R. M., Hamel, C., Ghazi, G., Boughari, Y., Theel, F., & Mendoza, A. M. (2015). *Level D research aircraft flight simulator use for novel methodologies in aircraft modeling and simulation*. Paper presented at the Third International Workshop on Numerical Modelling in Aerospace Sciences NMAS.

- Bouadi, H., Bouchoucha, M., & Tadjine, M. (2007). Sliding mode control based on backstepping approach for an UAV type-quadrotor. *World Academy of Science, Engineering and Technology*, 26(5), 22-27.
- Boughari, Y., & Botez, R. M. (2012). *Optimal flight control on the hawker 800 XP business aircraft*. Paper presented at the IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society.
- 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(2), 31.
- 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(2), 45.
- Brosilow, C., & Joseph, B. (2002). *Techniques of model-based control*: Prentice Hall Professional.
- Bucolo, M., Buscarino, A., Famoso, C., Fortuna, L., & Frasca, M. (2019). Control of imperfect dynamical systems. *Nonlinear Dynamics*, 98(4), 2989-2999.
- Camacho, E. F., & Alba, C. B. (2013). *Model predictive control*: Springer science & business media.
- Cao, C., & Hovakimyan, N. (2007). *Guaranteed transient performance with L1 adaptive controller for systems with unknown time-varying parameters and bounded disturbances: Part I*. Paper presented at the 2007 American Control Conference.
- Caughey, D. A. (2011). Introduction to aircraft stability and control course notes for M&AE 5070. *Sibley School of Mechanical & Aerospace Engineering Cornell University*.
- Ceruti, A., Bombardi, T., & Piancastelli, L. (2016). Visual Flight Rules Pilots Into Instrumental Meteorological Conditions: a Proposal for a Mobile Application to Increase In-flight Survivability. *International Review of Aerospace Engineering (IREASE)*, 9(5), 144-151.
- Ceruti, A., Curatolo, S., Bevilacqua, A., & Marzocca, P. (2015). *Image Processing Based Air Vehicles Classification for UAV Sense and Avoid Systems (0148-7191)*. Retrieved from
- Chabir, A., Boukhniifer, M., Bouteraa, Y., Chaibet, A., & Ghommam, J. (2016). Modelling and fixed order robust H_∞ control of aerial vehicle: simulation and experimental results. *Compel: International journal for computation and mathematics in electrical and electronic engineering*, 35(3), 1064-1085.

- Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., & Mukhopadhyay, D. (2018). Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.
- Chaloulos, G., Hokayem, P., & Lygeros, J. (2010). *Distributed hierarchical MPC for conflict resolution in air traffic control*. Paper presented at the Proceedings of the 2010 American Control Conference.
- Chatterjee, S., & Hadi, A. S. (2013). *Regression analysis by example*: John Wiley & Sons.
- Chen, D.-R., Wu, Q., Ying, Y., & Zhou, D.-X. (2004). Support vector machine soft margin classifiers: error analysis. *The Journal of Machine Learning Research*, 5, 1143-1175.
- Chen, H., Wang, X.-m., & Li, Y. (2009). *A survey of autonomous control for UAV*. Paper presented at the 2009 International Conference on Artificial Intelligence and Computational Intelligence.
- Cheng, C., Rees, N., Cao, S., & Feng, G. (1996). Control design of nonlinear systems using fuzzy model and genetic algorithm. *IFAC Proceedings Volumes*, 29(1), 2526-2531.
- Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural networks*, 17(1), 113-126.
- Cho, Y.-W., Seo, K.-S., & Lee, H.-J. (2007). A direct adaptive fuzzy control of nonlinear systems with application to robot manipulator tracking control. *International journal of control, Automation, and systems*, 5(6), 630-642.
- Chrif, L., & Kadda, Z. M. (2014). Aircraft control system using LQG and LQR controller with optimal estimation-Kalman filter design. *Procedia Engineering*, 80, 245-257.
- Cir, I. (2011). 328 AN/190. *Unmanned Aircraft Systems (UAS) Circular*, 10.
- Civanlar, M. R., & Trussell, H. J. (1986). Constructing membership functions using statistical data. *Fuzzy sets and systems*, 18(1), 1-13.
- Cobano, J. A., Conde, R., Alejo, D., & Ollero, A. (2011). *Path planning based on genetic algorithms and the monte-carlo method to avoid aerial vehicle collisions under uncertainties*. Paper presented at the 2011 IEEE International Conference on Robotics and Automation.
- Cockcroft, A. N., & Lameijer, J. N. F. (2003). *Guide to the collision avoidance rules*: Elsevier.
- Cook, A. (2007). *European air traffic management: principles, practice, and research*: Ashgate Publishing, Ltd.
- Cook, M. V. (2012). *Flight dynamics principles: a linear systems approach to aircraft stability and control*: Butterworth-Heinemann.

- Dai, L., Cao, Q., Xia, Y., & Gao, Y. (2017). Distributed MPC for formation of multi-agent systems with collision avoidance and obstacle avoidance. *Journal of the Franklin Institute*, 354(4), 2068-2085.
- Dancila, B. D., Beulze, B., & Botez, R. M. (2016). Geometrical vertical trajectory optimization—comparative performance evaluation of phase versus phase and altitude-dependent preferred gradient selection. *IFAC-PapersOnLine*, 49(17), 17-22.
- De Silva, C. W. (2018). *Intelligent control: fuzzy logic applications*: CRC press.
- Delahaye, D., Puechmorel, S., Tsiotras, P., & Féron, E. (2014). Mathematical models for aircraft trajectory design: A survey. In *Air Traffic Management and Systems* (pp. 205-247): Springer.
- Dijkstra, T., & Veldkamp, J. H. (1988). Data-driven selection of regressors and the bootstrap. In *On model uncertainty and its statistical implications* (pp. 17-38): Springer.
- Doitsidis, L., Valavanis, K. P., Tsourveloudis, N. C., & Kontitsis, M. (2004). *A framework for fuzzy logic based UAV navigation and control*. Paper presented at the IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004.
- Doyle, J. C., Francis, B. A., & Tannenbaum, A. R. (2013). *Feedback control theory*: Courier Corporation.
- Duan, H., Luo, Q., Shi, Y., & Ma, G. (2013). ? hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration. *IEEE Computational intelligence magazine*, 8(3), 16-27.
- Dubost, F., Bortsova, G., Adams, H., Ikram, M. A., Niessen, W., Vernooij, M., & de Bruijne, M. (2019). *Hydranet: data augmentation for regression neural networks*. Paper presented at the International Conference on Medical Image Computing and Computer-Assisted Intervention.
- Duke, E. L., Antoniewicz, R. F., & Krambeer, K. D. (1988). *Derivation and definition of a linear aircraft model* (Vol. 1207): National Aeronautics and Space Administration, Scientific and Technical
- Dullerud, G. E., & Paganini, F. (2013). *A course in robust control theory: a convex approach* (Vol. 36): Springer Science & Business Media.
- Ebeid, E., Skriver, M., Terkildsen, K. H., Jensen, K., & Schultz, U. P. (2018). A survey of open-source UAV flight controllers and flight simulators. *Microprocessors and Microsystems*, 61, 11-20.
- Ekaputri, C., & Syaichu-Rohman, A. (2013). *Model predictive control (MPC) design and implementation using algorithm-3 on board SPARTAN 6 FPGA SP605 evaluation kit*.

Paper presented at the 2013 3rd International Conference on Instrumentation Control and Automation (ICA).

- Etkin, B., & Reid, L. D. (1959). *Dynamics of flight* (Vol. 2): Wiley New York.
- Fan, Y., Wu, B., Li, T., Zhang, Y., Li, M., Li, Z., & Yang, Y. (2020). *Sparse adversarial attack via perturbation factorization*. Paper presented at the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16.
- Fine, S., Singer, Y., & Tishby, N. (1998). The hierarchical hidden Markov model: Analysis and applications. *Machine learning*, 32(1), 41-62.
- Foo, J. L., Knutzon, J., Kalivarapu, V., Oliver, J., & Winer, E. (2009). Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization. *Journal of aerospace computing, Information, and communication*, 6(4), 271-290.
- Fossen, T. I. (2011). Mathematical models for control of aircraft and satellites. *Department of Engineering Cybernetics Norwegian University of Science and Technology*, 54-58.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1): Springer series in statistics New York.
- Gaonkar, G., & PETERS, D. (1986). *Review of dynamic inflow modeling for rotorcraft flight dynamics*. Paper presented at the 27th structures, structural dynamics and materials conference.
- Gardner, R. W., Genin, D., McDowell, R., Rouff, C., Saksena, A., & Schmidt, A. (2016). *Probabilistic model checking of the next-generation airborne collision avoidance system*. Paper presented at the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC).
- Georgiou, H., Karagiorgou, S., Kontoulis, Y., Pelekis, N., Petrou, P., Scarlatti, D., & Theodoridis, Y. (2018). Moving objects analytics: survey on future location & trajectory prediction methods. *arXiv preprint arXiv:1807.04639*.
- Ghasemi Hamed, M., Gianazza, D., Serrurier, M., & Durand, N. (2013). *Statistical prediction of aircraft trajectory: regression methods vs point-mass model*.
- Ghazi, G., Botez, R., & Achigui, J. M. (2015). Cessna citation X engine model identification from flight tests. *SAE International Journal of Aerospace*, 8(2), 203-213.
- Ghazi, G., Botez, R. M., & Tudor, M. (2015). Performance database creation for cessna citation x aircraft in climb regime using an aero-propulsive model developed from flight tests. *AHS Sustainability*.

- Ghazi, G., Tudor, M., & Botez, R. (2015). *Identification of a Cessna Citation X aeropropulsive model in climb regime from flight tests*. Paper presented at the International Conference on Air Transport INAIR.
- Ghommam, J., Saad, M., Mnif, F., & Zhu, Q. M. (2020). Guaranteed performance design for formation tracking and collision avoidance of multiple USVs with disturbances and unmodeled dynamics. *IEEE Systems Journal*.
- Gil, Y., Chai, Y., Gorodissky, O., & Berant, J. (2019). White-to-black: Efficient distillation of black-box adversarial attacks. *arXiv preprint arXiv:1904.02405*.
- Giridhara, P. K. B., Mishra, C., Venkataramana, R. K. M., Bukhari, S. S., & Dengel, A. (2019). A Study of Various Text Augmentation Techniques for Relation Classification in Free Text. *ICPRAM*, 3, 5.
- Glorot, X., Bordes, A., & Bengio, Y. (2011). *Deep sparse rectifier neural networks*. Paper presented at the Proceedings of the fourteenth international conference on artificial intelligence and statistics.
- Goel, A., Agarwal, A., Vatsa, M., Singh, R., & Ratha, N. K. (2020). *DNDNet: Reconfiguring CNN for adversarial robustness*. Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.
- Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Grigorie, T. L., & Botez, R. M. (2011). New Applications of Fuzzy Logic Methodologies in Aerospace Field. In *Fuzzy Controllers, Theory and Applications*: IntechOpen.
- Grimm, L. G., & Yarnold, P. R. (1995). *Reading and understanding multivariate statistics*: American Psychological Association.
- Guan, X., Lv, R., Sun, L., & Liu, Y. (2016). *A study of 4D trajectory prediction based on machine deep learning*. Paper presented at the 2016 12th World Congress on Intelligent Control and Automation (WCICA).
- Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report*, 14(1), 5-16.
- GUO, L., YU, X., ZHANG, X., & ZHANG, Y. (2020). Safety control system technologies for UAVs: Review and prospect. *Scientia Sinica Informationis*, 50(2), 184-194.
- Hashemi, S. M., Botez, R. M., & Grigorie, L. T. (2020a). *Adaptive fuzzy control of chaotic flapping relied upon lyapunov-based tuning laws*. Paper presented at the AIAA AVIATION 2020 FORUM.

- Hashemi, S. M., Botez, R. M., & Grigorie, T. L. (2020b). New reliability studies of data-driven aircraft trajectory prediction. *Aerospace*, 7(10), 145.
- Hashemi, S. M., Menhaj, M. B., & Amani, A. M. (2006). Reconfigurable Fault-Tolerant Control by Linear Quadratic Virtual Actuator Under Control Signal Constraint. *ARPN Journal of Engineering and Applied Sciences*, 11(3), 1998-2004.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Henely, S. (2017). Traffic Alert and Collision Avoidance System II (TCASII). In *Digital Avionics Handbook* (pp. 354-362): CRC Press.
- Hernández, A. M., Magaña, E. J. C., & Berna, A. G. (2018). *Data-driven aircraft trajectory predictions using ensemble meta-estimators*. Paper presented at the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC).
- Herty, M., & Visconti, G. (2018). Kinetic methods for inverse problems. *arXiv preprint arXiv:1811.09387*.
- Hjalmarsson, H., Gevers, M., & De Bruyne, F. (1996). For model-based control design, closed-loop identification gives better performance. *Automatica*, 32(12), 1659-1673.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Huang, S., Papernot, N., Goodfellow, I., Duan, Y., & Abbeel, P. (2017). Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.
- Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., . . . Yi, X. (2020). A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37, 100270.
- Hušek, P., & Narenathreyas, K. (2016). Aircraft longitudinal motion control based on Takagi–Sugeno fuzzy model. *Applied Soft Computing*, 49, 269-278.
- Ioannou, P. A., & Sun, J. (2012). *Robust adaptive control*: Courier Corporation.
- Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. Paper presented at the International conference on machine learning.
- Iskandarani, M., Givigi, S. N., Fusina, G., & Beaulieu, A. (2014). *Unmanned aerial vehicle formation flying using linear model predictive control*. Paper presented at the 2014 IEEE International Systems Conference Proceedings.

- Jiang, W., Li, H., Liu, S., Luo, X., & Lu, R. (2020). Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles. *IEEE transactions on vehicular technology*, 69(4), 4439-4449.
- Jilkov, V. P., Ledet, J. H., & Li, X. R. (2018). Multiple model method for aircraft conflict detection and resolution in intent and weather uncertainty. *IEEE Transactions on Aerospace and Electronic Systems*, 55(2), 1004-1020.
- Jin, W., Li, Y., Xu, H., Wang, Y., & Tang, J. (2020). Adversarial attacks and defenses on graphs: A review and empirical study. *arXiv e-prints*, arXiv: 2003.00653.
- Julian, K. D., Kochenderfer, M. J., & Owen, M. P. (2019). Deep neural network compression for aircraft collision avoidance systems. *Journal of Guidance, Control, and Dynamics*, 42(3), 598-608.
- Kachitvichyanukul, V. (2012). Comparison of three evolutionary algorithms: GA, PSO, and DE. *Industrial Engineering and Management Systems*, 11(3), 215-223.
- Kada, B., & Ghazzawi, Y. (2011). *Robust PID controller design for an UAV flight control system*. Paper presented at the Proceedings of the World congress on Engineering and Computer Science.
- Kamalasadan, S., & Ghandakly, A. A. (2007). Multiple fuzzy reference model adaptive controller design for pitch-rate tracking. *IEEE Transactions on Instrumentation and Measurement*, 56(5), 1797-1808.
- Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: modeling, simulation, and control of mechatronic systems*: John Wiley & Sons.
- Karr, D. A., Vivona, R. A., Woods, S., & Wing, D. J. (2017). *Point-Mass Aircraft Trajectory Prediction Using a Hierarchical, Highly-Adaptable Software Design*. Paper presented at the AIAA Modeling and Simulation Technologies Conference.
- Kochenderfer, M. J., Holland, J. E., & Chryssanthacopoulos, J. P. (2012). *Next-generation airborne collision avoidance system*. Retrieved from
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Kuchar, J., & Drumm, A. C. (2007). The traffic alert and collision avoidance system. *Lincoln laboratory journal*, 16(2), 277.
- Kuchar, J. K., & Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on intelligent transportation systems*, 1(4), 179-189.

- Kuitche, M., Segui, M., Botez, R. M., & Ghazi, G. (2017). *New methodology for longitudinal flight dynamics modelling of the UAS-S4 Ehecatl towards its aerodynamics estimation modelling*. Paper presented at the AIAA Modeling and Simulation Technologies Conference.
- Kuitche, M. A. J., & Botez, R. M. (2019). Modeling novel methodologies for unmanned aerial systems—Applications to the UAS-S4 Ehecatl and the UAS-S45 Bálaam. *Chinese Journal of Aeronautics*, 32(1), 58-77.
- Kuitche, M. A. J., Botez, R. M., Guillemin, A., & Communier, D. (2020a). Aerodynamic modelling of unmanned aerial system through nonlinear vortex lattice method, computational fluid dynamics and experimental validation-application to the uas-s45 bálaam: Part 1. *INCAS Bulletin*, 12(1), 91-103.
- Kuitche, M. A. J., Botez, R. M., Guillemin, A., & Communier, D. (2020b). Aerodynamic modelling of unmanned aerial system through nonlinear vortex lattice method, computational fluid dynamics and experimental validation-application to the UAS-S45 bálaam: Part 2. *INCAS Bulletin*, 12(2), 99-115.
- Kuitche, M. A. J., Botez, R. M., Viso, R., Maunand, J. C., & Moyao, O. C. (2020). Blade element momentum new methodology and wind tunnel test performance evaluation for the UAS-S45 Bálaam propeller. *CEAS Aeronautical Journal*, 11(4), 937-953.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79-86.
- Landau, I. D., Lozano, R., M'Saad, M., & Karimi, A. (2011). *Adaptive control: algorithms, analysis and applications*: Springer Science & Business Media.
- Lawrence, D. A., & Rugh, W. J. (1995). Gain scheduling dynamic linear controllers for a nonlinear plant. *Automatica*, 31(3), 381-390.
- Lewis, F. L., Vrabie, D., & Syrmos, V. L. (2012). *Optimal control*: John Wiley & Sons.
- Lin, J., Zhou, J., Lu, M., Wang, H., & Yi, A. (2020). Design of robust adaptive fuzzy controller for a class of single-input single-output (siso) uncertain nonlinear systems. *Mathematical Problems in Engineering*, 2020.
- Lin, Y., Zhang, J.-w., & Liu, H. (2018). An algorithm for trajectory prediction of flight plan based on relative motion between positions. *Frontiers of Information Technology & Electronic Engineering*, 19(7), 905-916.
- Lin, Z. (2002). *Gain scheduling of aircraft pitch attitude and control of discrete, affine, linear parametrically varying systems*: Iowa State University.

- Liu, M., Egan, G. K., & Santoso, F. (2015). Modeling, autopilot design, and field tuning of a UAV with minimum control surfaces. *IEEE Transactions on Control Systems Technology*, 23(6), 2353-2360.
- Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., & Leung, V. C. (2018). A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE access*, 6, 12103-12117.
- Liu, W., & Hwang, I. (2011). Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics*, 34(6), 1779-1789.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.
- Liu, Y., & Hansen, M. (2018). Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670*.
- Lone, M., & Cooke, A. (2014). Review of pilot models used in aircraft flight dynamics. *Aerospace Science and Technology*, 34, 55-74.
- Lorenzi, L., Mercier, G., & Melgani, F. (2012). Support vector regression with kernel combination for missing data reconstruction. *IEEE Geoscience and Remote Sensing Letters*, 10(2), 367-371.
- Lungu, R., Lungu, M., & Grigorie, L. T. (2013). Automatic control of aircraft in longitudinal plane during landing. *IEEE Transactions on Aerospace and Electronic Systems*, 49(2), 1338-1350.
- Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., . . . Bailey, J. (2018). Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- Mani, N., Moh, M., & Moh, T.-S. (2019). *Towards robust ensemble defense against adversarial examples attack*. Paper presented at the 2019 IEEE Global Communications Conference (GLOBECOM).
- McEneaney, W., & Fitzpatrick, B. (2002). *Control for UAV operations under imperfect information*. Paper presented at the 1st UAV Conference.
- Mehran, K. (2008). Takagi-sugeno fuzzy modeling for process control. *Industrial Automation, Robotics and Artificial Intelligence (EEE8005)*, 262, 1-31.

- Mehrjerdi, H., Saad, M., & Ghommam, J. (2010). Hierarchical fuzzy cooperative control and path following for a team of mobile robots. *IEEE/ASME Transactions on Mechatronics*, 16(5), 907-917.
- Meng, L., Lin, C.-T., Jung, T.-P., & Wu, D. (2019). *White-box target attack for EEG-based BCI regression problems*. Paper presented at the International conference on neural information processing.
- Minchala-Avila, L. I., Garza-Castañón, L. E., Vargas-Martínez, A., & Zhang, Y. (2015). A review of optimal control techniques applied to the energy management and control of microgrids. *Procedia Computer Science*, 52, 780-787.
- Mokhtari, M. R., & Cherki, B. (2015). A new robust control for minirotorcraft unmanned aerial vehicles. *ISA transactions*, 56, 86-101.
- Mosbah, A. B., Botez, R. M., & Dao, T. M. (2016). A hybrid original approach for prediction of the aerodynamic coefficients of an ATR-42 scaled wing model. *Chinese Journal of Aeronautics*, 29(1), 41-52.
- Mousavi, M. A., Heshmati, Z., & Moshiri, B. (2013). *LTV-MPC based path planning of an autonomous vehicle via convex optimization*. Paper presented at the 2013 21st Iranian Conference on Electrical Engineering (ICEE).
- Munoz, C., Narkawicz, A., & Chamberlain, J. (2013). *A TCAS-II resolution advisory detection algorithm*. Paper presented at the AIAA Guidance, Navigation, and Control (GNC) Conference.
- Murrieta-Mendoza, A., & Botez, R. (2015). *Aircraft vertical route optimization deterministic algorithm for a flight management system (0148-7191)*. Retrieved from
- Murrieta-Mendoza, A., & Botez, R. M. (2015). Methodology for vertical-navigation flight-trajectory cost calculation using a performance database. *Journal of Aerospace Information Systems*, 12(8), 519-532.
- Murrieta-Mendoza, A., Botez, R. M., & Bunel, A. (2018). Four-dimensional aircraft en route optimization algorithm using the artificial bee colony. *Journal of Aerospace Information Systems*, 15(6), 307-334.
- Murrieta-Mendoza, A., Hamy, A., & Botez, R. M. (2017). Four-and three-dimensional aircraft reference trajectory optimization inspired by ant colony optimization. *Journal of Aerospace Information Systems*, 14(11), 597-616.
- Murrieta-Mendoza, A., Romain, C., & Botez, R. M. (2016). Commercial aircraft lateral flight reference trajectory optimization. *IFAC-PapersOnLine*, 49(17), 1-6.

- Murrieta-Mendoza, A., Ruiz, H., Kessaci, S., & Botez, R. M. (2017). *3D reference trajectory optimization using particle swarm optimization*. Paper presented at the 17th AIAA Aviation Technology, Integration, and Operations Conference.
- Murrieta-Mendoza, A., Ternisien, L., Beuze, B., & Botez, R. M. (2018). Aircraft vertical route optimization by beam search and initial search space reduction. *Journal of Aerospace Information Systems*, *15*(3), 157-171.
- Nanjundappan, D. (2016). Hybrid weighted probabilistic neural network and biogeography based optimization for dynamic economic dispatch of integrated multiple-fuel and wind power plants. *International Journal of Electrical Power & Energy Systems*, *77*, 385-394.
- Nelson, R. C. (1998). *Flight stability and automatic control* (Vol. 2): WCB/McGraw Hill New York.
- Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., & Kostaras, A. N. (2003). Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *33*(6), 898-912.
- Nolan, M. S. (2011). *Fundamentals of air traffic control*: Cengage learning.
- Nuic, A., Poles, D., & Mouillet, V. (2010). BADA: An advanced aircraft performance model for present and future ATM systems. *International journal of adaptive control and signal processing*, *24*(10), 850-866.
- Palacios, R., Murua, J., & Cook, R. (2010). Structural and aerodynamic models in nonlinear flight dynamics of very flexible aircraft. *AIAA journal*, *48*(11), 2648-2659.
- Park, S. H., Kim, B., Kang, C. M., Chung, C. C., & Choi, J. W. (2018). *Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture*. Paper presented at the 2018 IEEE Intelligent Vehicles Symposium (IV).
- Patrón, R. S. F., & Botez, R. M. (2015). Flight trajectory optimization through genetic algorithms for lateral and vertical integrated navigation. *Journal of Aerospace Information Systems*, *12*(8), 533-544.
- Pereida, K., & Schoellig, A. P. (2018). *Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions*. Paper presented at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- Pham, D.-T., Tran, N. P., Alam, S., Duong, V., & Delahaye, D. (2019). *A machine learning approach for conflict resolution in dense traffic scenarios with uncertainties*. Paper presented at the ATM Seminar 2019, 13th USA/Europe ATM R&D Seminar.

- Plancher, B., Manchester, Z., & Kuindersma, S. (2017). *Constrained unscented dynamic programming*. Paper presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade* (pp. 55-69): Springer.
- Qiu, X., Zhang, L., Ren, Y., Suganthan, P. N., & Amaratunga, G. (2014). *Ensemble deep learning for regression and time series forecasting*. Paper presented at the 2014 IEEE symposium on computational intelligence in ensemble learning (CIEL).
- Radhakrishnan, C., & Swarup, A. (2020). *Performance Comparison for Fuzzy based Aircraft Pitch using Various Control Methods*. Paper presented at the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA).
- Rocha, M., Cortez, P., & Neves, J. (2007). Evolution of neural networks for classification and regression. *Neurocomputing*, 70(16-18), 2809-2816.
- Rodriguez, L. F., & Botez, R. M. (2013). Generic new modeling technique for turbofan engine thrust. *Journal of Propulsion and Power*, 29(6), 1492-1495.
- Romeo, G., Cestino, E., Pacino, M., Borello, F., & Correa, G. (2012). Design and testing of a propeller for a two-seater aircraft powered by fuel cells. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(7), 804-816.
- Ruby, M., & Botez, R. M. (2016). Trajectory optimization for vertical navigation using the harmony search algorithm. *IFAC-PapersOnLine*, 49(17), 11-16.
- Sadeghzadeh, I., Mehta, A., & Zhang, Y. (2011). *Fault/damage tolerant control of a quadrotor helicopter UAV using model reference adaptive control and gain-scheduled PID*. Paper presented at the AIAA Guidance, Navigation, and Control Conference.
- Saggiani, G., Persiani, F., Ceruti, A., Tortora, P., Troiani, E., Giuletti, F., . . . Bentini, G. (2007). *A UAV system for observing volcanoes and natural hazards*. Paper presented at the AGU Fall Meeting Abstracts.
- Sahawneh, L. R., & Beard, R. W. (2014). *A probabilistic framework for unmanned aircraft systems collision detection and risk estimation*. Paper presented at the 53rd IEEE Conference on Decision and Control.
- Santi, G. M., Ceruti, A., Liverani, A., & Osti, F. (2021). Augmented Reality in Industry 4.0 and Future Innovation Programs. *Technologies*, 9(2), 33.
- Sebald, D. J., & Bucklew, J. A. (2000). Support vector machine techniques for nonlinear equalization. *IEEE transactions on signal processing*, 48(11), 3217-3226.

- Segui, M., & Botez, R. M. (2021). *Electric Motor Modeling using Artificial Neural Networks: Application for Drones*. Paper presented at the AIAA AVIATION 2021 FORUM.
- Shamma, J. S., & Athans, M. (1990). Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8), 898-907.
- Shearer, C. M., & Cesnik, C. E. (2007). Nonlinear flight dynamics of very flexible aircraft. *Journal of Aircraft*, 44(5), 1528-1545.
- Slotine, J.-J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199): Prentice hall Englewood Cliffs, NJ.
- Song, Y.-Y., & Ying, L. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2), 130.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
- Stengel, R. F. (2015). *Flight dynamics*: Princeton University Press.
- Sun, L., Tan, M., & Zhou, Z. (2018). A survey of practical adversarial example attacks. *Cybersecurity*, 1(1), 1-9.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). *Going deeper with convolutions*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*(1), 116-132.
- Ter Braak, C. J., & Looman, C. W. (1986). Weighted averaging, logistic regression and the Gaussian response model. *Vegetatio*, 65(1), 3-11.
- Tondji, Y., & Botez, R. (2017). Semi-empirical estimation and experimental method for determining inertial properties of the Unmanned Aerial System–UAS-S4 of Hydra Technologies. *The Aeronautical Journal*, 121(1245), 1648-1682.
- Tseng, C.-S., Chen, B.-S., & Uang, H.-J. (2001). Fuzzy tracking control design for nonlinear dynamic systems via TS fuzzy model. *IEEE Transactions on fuzzy systems*, 9(3), 381-392.
- Tuzcu, I., Marzocca, P., Cestino, E., Romeo, G., & Frulla, G. (2007). Stability and control of a high-altitude, long-endurance UAV. *Journal of Guidance, Control, and Dynamics*, 30(3), 713-721.

- U.S. Department of Transportation, F. A. A., Aviation Safety Information Analysis and Sharing (ASIAS) System. (2021). Number of Pilot-Reported Near Midair Collisions (NMAC) by Degree of Hazard. Retrieved from <https://www.bts.gov/content/number-pilot-reported-near-midair-collisions-nmac-degree-hazard>
- Van Dyk, D. A., & Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1), 1-50.
- Vega, L. F. L., Lopez-Neri, E., Arellano-Muro, C. A., González-Jiménez, L. E., Ghommam, J., & Carrasco-Navarro, R. (2020). *UAV Flight Instructional Design for Industry 4.0 based on the Framework of Educational Mechatronics*. Paper presented at the IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society.
- Wang, H., Yi, H., Peng, J., Wang, G., Liu, Y., Jiang, H., & Liu, W. (2017). Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network. *Energy conversion and management*, 153, 409-422.
- Wang, M., Luo, J., & Walter, U. (2016). A non-linear model predictive controller with obstacle avoidance for a space robot. *Advances in Space Research*, 57(8), 1737-1746.
- Wang, Q., Liu, H., Xie, G., & Zhang, Y. (2021). *Image Denoising Using an Improved Generative Adversarial Network with Wasserstein Distance*. Paper presented at the 2021 40th Chinese Control Conference (CCC).
- Wang, Q., & Stengel, R. F. (2004). Robust nonlinear flight control of a high-performance aircraft. *IEEE Transactions on Control Systems Technology*, 13(1), 15-26.
- Wang, X., Sun, T., Yang, R., Li, C., Luo, B., & Tang, J. (2019). Quality-aware dual-modal saliency detection via deep reinforcement learning. *Signal Processing: Image Communication*, 75, 158-167.
- Watts, A. C., Ambrosia, V. G., & Hinkley, E. A. (2012). Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing*, 4(6), 1671-1692.
- Weitz, L. A. (2015). *Derivation of a point-mass aircraft model used for fast-time simulation*. Retrieved from
- Whitehead, B., & Bieniawski, S. (2010). *Model reference adaptive control of a quadrotor UAV*. Paper presented at the AIAA Guidance, Navigation, and Control Conference.
- Wiens, T. S., Dale, B. C., Boyce, M. S., & Kershaw, G. P. (2008). Three way k-fold cross-validation of resource selection functions. *Ecological Modelling*, 212(3-4), 244-255.
- Wiest, J., Höffken, M., Kreßel, U., & Dietmayer, K. (2012). *Probabilistic trajectory prediction with Gaussian mixture models*. Paper presented at the 2012 IEEE Intelligent Vehicles Symposium.

- Wu, H., Chen, Z., Sun, W., Zheng, B., & Wang, W. (2017). *Modeling trajectories with recurrent neural networks*.
- Xian, B., Wang, S., & Yang, S. (2019). Nonlinear adaptive control for an unmanned aerial payload transportation system: theory and experimental validation. *Nonlinear Dynamics*, 98(3), 1745-1760.
- Xu, B., & Shi, Z. (2015). An overview on flight dynamics and control approaches for hypersonic vehicles. *Science China Information Sciences*, 58(7), 1-19.
- Yañez-Badillo, H., Kuitche, M. A. J., & Botez, R. M. (2020). *Disturbance rejection in longitudinal control for the UAS-S4 ehecatl design*. Paper presented at the AIAA AVIATION 2020 FORUM.
- Ying, H. (1998). Constructing nonlinear variable gain controllers via the Takagi-Sugeno fuzzy control. *IEEE Transactions on fuzzy systems*, 6(2), 226-234.
- Yit, K. K., Rajendran, P., & Wee, L. K. (2016). Proportional-derivative linear quadratic regulator controller design for improved longitudinal motion control of unmanned aerial vehicles. *International Journal of Micro Air Vehicles*, 8(1), 41-50.
- Yu, X., Guo, L., Zhang, Y., & Jiang, J. (2021). *Autonomous Safety Control of Flight Vehicles*: CRC Press.
- Yu, Z.-q., Liu, Z.-x., Zhang, Y.-m., Qu, Y.-h., & Su, C.-y. (2019). Decentralized fault-tolerant cooperative control of multiple UAVs with prescribed attitude synchronization tracking performance under directed communication topology. *Frontiers of Information Technology & Electronic Engineering*, 20(5), 685-700.
- Yu, Z., Zhang, Y., Jiang, B., Su, C.-Y., Fu, J., Jin, Y., & Chai, T. (2020). Decentralized fractional-order backstepping fault-tolerant control of multi-UAVs against actuator faults and wind effects. *Aerospace Science and Technology*, 104, 105939.
- Zadeh, L. A. (1988). Fuzzy logic. *Computer*, 21(4), 83-93.
- Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. arXiv 2012. *arXiv preprint arXiv:1212.5701*, 1212.
- Zeng, X.-J., Keane, J. A., & Wang, D. (2006). *Fuzzy systems approach to approximation and stabilization of conventional affine nonlinear systems*. Paper presented at the 2006 IEEE International Conference on Fuzzy Systems.
- Zhen, Z., Tao, G., Xu, Y., & Song, G. (2019). Multivariable adaptive control based consensus flight control system for UAVs formation. *Aerospace Science and Technology*, 93, 105336.

- Zhou, X., Yu, X., Guo, K., Zhou, S., Guo, L., Zhang, Y., & Peng, X. (2021). Safety Flight Control Design of a Quadrotor UAV With Capability Analysis. *IEEE Transactions on Cybernetics*.
- Zúñiga, D. C., Souza, A., & Góes, L. (2020). Flight dynamics modeling of a flexible wing unmanned aerial vehicle. *Mechanical Systems and Signal Processing*, 145, 106900.

