DECENTRALIZED ROBUST NONLINEAR MODEL PREDICTIVE CONTROLLER FOR UNMANNED AERIAL SYSTEMS

By

Gonzalo A. García Garretón

Submitted to the graduate degree program in Department of Aerospace Engineering and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairperson Dr. Shahriar Keshmiri

Dr. David Downing

Dr. Mark Ewing

Dr. Richard Hale

Dr. Craig McLaughlin

Dr. Sara Wilson

Date Defended: February 1st, 2013

The Dissertation Committee for Gonzalo A. García Garretón

certifies that this is the approved version of the following dissertation:

DECENTRALIZED ROBUST NONLINEAR MODEL PREDICTIVE CONTROLLER FOR UNMANNED AERIAL SYSTEMS

Chairperson Dr. Shahriar Keshmiri

Dr. David Downing

Dr. Mark Ewing

Dr. Richard Hale

Dr. Craig McLaughlin

Dr. Sara Wilson

Date approved: February 1st, 2013

Abstract

The nonlinear and unsteady nature of aircraft aerodynamics together with limited practical range of controls and state variables make the use of the linear control theory inadequate especially in the presence of external disturbances, such as wind. In the classical approach, aircraft are controlled by multiple inner and outer loops, designed separately and sequentially. For unmanned aerial systems in particular, control technology must evolve to a point where autonomy is extended to the entire mission flight envelope. This requires advanced controllers that have sufficient robustness, track complex trajectories, and use all the vehicles control capabilities at higher levels of accuracy. In this work, a robust nonlinear model predictive controller is designed to command and control an unmanned aerial system to track complex tight trajectories in the presence of internal and external perturbance.

The Flight System developed in this work achieves the above performance by using:

1 A nonlinear guidance algorithm that enables the vehicle to follow an arbitrary trajectory shaped by moving points;

2 A formulation that embeds the guidance logic and trajectory information in the aircraft model, avoiding cross coupling and control degradation;

3 An artificial neural network, designed to adaptively estimate and provide aerodynamic and propulsive forces in real-time; and

4 A mixed sensitivity approach that enhances the robustness for a nonlinear model predictive controller overcoming the effect of un-modeled dynamics, external disturbances such as wind, and measurement additive perturbations, such as noise and biases.

These elements have been integrated and tested in simulation and with previously stored flight test data and shown to be feasible.

Acknowledgments

I would like to express my gratitude to my Advisor Dr. Shahriar Keshmiri for his permanent support, as well as the Aerospace Department including faculty, students and staff.

My infinite gratitude is for my family, Emita, Andrés, Gonzalo and José.

Table of Contents

	Abstract	iii
	Table of Contents	v
	List of Figures	viii
	List of Tables	xi
	Glossary of Acronyms	xii
	Glossary of Symbols, Subscripts and Superscripts	xiii
0	Introduction	1
0.1	Motivation	1
0.2	Problem Statement	1
0.3	Literature Review	3
0.3.1	Aircraft Modeling	3
0.3.2	State Observers	5
0.3.3	Guidance, Navigation, and Control	6
0.3.4	Gain Scheduling of Controllers	8
0.3.5	Nonlinear Model Predictive Controller	9
0.3.6	Aircraft Formation	12
0.4	Significance and Originality	16
0.4.1	Significance	16
0.4.2	Originality	17
1	Aircraft Modeling	19
1.1	Introduction	19
1.2	Aircraft Equations of Motion	19
1.2.1	Gravity Forces $[\boldsymbol{F}_B]^g$	23

1.2.2	Propulsive Forces $[\boldsymbol{F}_B]^P$ and Moments $[\boldsymbol{M}_B]^P$	23
1.2.3	Aerodynamic Forces $[\boldsymbol{F}_B]^A$ and Moments $[\boldsymbol{M}_B]^A$	24
1.2.3.1	Linear Aerodynamic Forces $[\boldsymbol{F}_B]^A$ and Moments $[\boldsymbol{M}_B]^A$	25
1.3	Aerodynamic Coefficients Estimation	27
1.3.1	Adaptive ANN Aerodynamic Estimation	31
1.3.1.1	ANN Performance Test using Simulated Data	35
1.3.1.2	ANN Performance Test using Flight Test Data	38
1.4	Nonlinear and Time-Varying Aircraft Mathematical Description	42
2	Observer Design	44
2.1	Introduction	44
2.2	Extended Kalman Filter Design	46
2.3	Extended Kalman Filter Application	48
2.3.1	Gain Scheduling Off-line Kalman Gain Matrices	52
2.3.2	EKF Test Results	54
2.3.2.1	EKF Simulation Data Results	54
2.3.2.2	EKF Flight Test Data Results	57
2.4	Alternative Extended Kalman Filter Design	58
3	Guidance, Navigation, and Control	61
3.1	Introduction	61
3.2	Inner and Outer Loops Design	63
3.2.1	Inner Loop	63
3.2.1.1	Criterion for the Selection of Weighting Matrices	71
3.2.2	Outer Loop	72
3.3	Guidance, Navigation, and Control Testing	79

3.3.1	Gain Scheduling Testing	84
4	Nonlinear Model Predictive Control	87
4.1	Introduction	87
4.2	Integration of Guidance, Navigation and Control	89
4.3	Nonlinear Model Predictive Control Formulation	90
4.4	Aerodynamic Estimation ANN Augmentation	95
4.5	Modification of Trajectory Following Model to Follow Moving Points	96
4.6	NMPC Robustness Augmentation	98
4.6.1	Robustness Weighting Matrices Design	101
4.7	Simulations	102
4.7.1	Simulation of Trajectory Tracking	102
4.7.2	Simulation of Moving Point Tracking	105
4.7.3	Simulation of NMPC with Robustness Augmentation	106
4.7.3.1	Low Frequency Range Robustness Test	106
4.7.3.2	High Frequency Range Robustness Test	107
5	UAS Formation Centralized Command	109
5.1	Introduction	109
5.2	NMPC Decentralized Control	112
5.3	Formation Centralized Command	113
5.4	Aerodynamic Coupling in Formation Flight	118
5.5	Optimal Command Generation	121
5.6	Centralized Command Communication Delay Robustness	123
	Conclusions	125
	References	128

List of Figures

Figure 01	Standard UAS Command and Control Configuration	1
Figure 02	Adaptive, Robust NMPC Command and Control Configuration for UAS.	2
Figure 1	Coordinate Systems for Aircraft Modeling.	20
Figure 2	ANN Interconnection Structure.	29
Figure 3	Z-axis Measured and Predicted Acceleration.	36
Figure 4	Z-axis Acceleration ANN Parameters.	36
Figure 5	Z-axis Acceleration ANN Cost Function.	37
Figure 6	Z-axis Acceleration ANN Levenberg's Parameter.	37
Figure 7	X-axis Measured and Predicted Acceleration in Antarctica: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).	38
Figure 8	Y-axis Measured and Predicted Acceleration in Antarctica: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).	39
Figure 9	Z-axis Measured and Predicted Acceleration in Antarctica: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).	39
Figure 10	X-axis Measured and Predicted Acceleration in Greenland: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).	40
Figure 11	Y-axis Measured and Predicted Acceleration in Greenland: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).	40
Figure 12	Z-axis Measured and Predicted Acceleration in Greenland: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).	41
Figure 13	Kalman Gain Matrix Gain Scheduling.	53
Figure 14	Kalman Gain Matrix Subset of Entries.	54
Figure 15	Simulation Data: UAS Trajectory Following.	55

Figure 16	Simulation Data: Angle of attack (left) – Angle of attack Bias (right).	55
Figure 17	Simulation Data: Sideslip Angle (left) – Sideslip Angle Bias (right).	56
Figure 18	Simulation Data: Airspeed (left) – Airspeed Bias (right).	56
Figure 19	Simulation Data: North Wind Component (left) – Vertical Wind Component (right).	57
Figure 20	Flight Data: Angle of Attack (left) – Angle of attack Bias (right).	57
Figure 21	Flight Data: Airspeed (left) – Airspeed Bias (right).	58
Figure 22	Flight Data: Sideslip Angle (left) - Sideslip Angle Bias (right).	58
Figure 23	Augmented System.	65
Figure 24	Weighting Matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_3$ Singular Values	66
Figure 25	General LFT Interconnection.	69
Figure 26	Small Gain Theorem Interconnection.	69
Figure 27	Uncertainty Normalization.	70
Figure 28	Guidance Law Concept (left) - Switching Logic (right).	73
Figure 29	Longitudinal and Lateral Guidance Law Geometry (left) - Segment Rotations (right).	74
Figure 30	Lateral Angle Calculation (left) - Longitudinal Angle Calculation (right).	76
Figure 31	Lateral guidance Geometry (left) - Phase Portrait for Lateral guidance (right).	78
Figure 32	3D Path Following.	81
Figure 33	Engine Throttle Deflection (left) – Elevator Deflection (right).	82
Figure 34	Aileron Deflection (left) – Rudder Deflection (right).	82
Figure 35	Airspeed (left) – Ground Speed (right).	83
Figure 36	Pitch Angle (left) – Bank angle (right).	84
Figure 37	Sideslip Angle (left) – Roll Rate (right).	85

Figure 38	Yaw Rate (left) – Rudder Deflection (right).	85
Figure 39	Control Problem.	91
Figure 40	Predictive Horizon.	92
Figure 41	Time-Varying Waypoints $\vec{\boldsymbol{a}}[k]$ and $\vec{\boldsymbol{b}}[k]$.	96
Figure 42	Lateral Guiadance Logic for Time-Varying Waypoints $\vec{\boldsymbol{a}}[k]$ and $\vec{\boldsymbol{b}}[k]$.	97
Figure 43	Speed Segment Geometry (left) - Speed Command Guidance Logic (right).	98
Figure 44	Robust NMPC Configuration.	99
Figure 45	Throttle Deflection (left) - Elevator Deflection (right).	103
Figure 46	Trajectory Following (left) – Angle of attack and Pitch angle (right).	103
Figure 47	Throttle Deflection (left) – Elevator Deflection (right).	104
Figure 48	3D Views Moving Point Following.	105
Figure 49	Engine Throttle Deflection.	105
Figure 50	3D Views Trajectory Following.	106
Figure 51	Bank Angle (left) – Aileron Deflection (right).	107
Figure 52	Pitch Angle (left) – Elevator Deflection (right).	107
Figure 53	Moving Point Window Concept.	113
Figure 54	Formation Communication Concept.	114
Figure 55	Arbitrary UAS Formation.	115
Figure 56	UAS Formation Readjustment.	117
Figure 57	UAS Formation Reorientation.	118
Figure 58	Estimated Induced Lift and Rolling Moment (from Ref. [72]).	119
Figure 59	Regions of Upwash and Downwash.	120
Figure 60	Trajectory Adjustment for Obstacle Avoidance.	122

List of Tables

Table I	AAA Stability and Control Derivatives for 40% Yak 54 at Knots	52
Table II	AAA Stability and Control Derivatives for 33% Yak 54 at Knots	79
Table III	Stability and Control Derivatives Uncertainty Range	80
Table IV	Closed Loop Infinity Norm Associated with each Uncertainty	81

Glossary of Acronyms

Acronyms

2D	: Two Dimensional
3D	: Three Dimensional
AAA	: Advanced Aircraft analysis
(A)FCS	: (Autonomous) Flight Control System
ANN	: Artificial Neural Network
CG	: Center of Gravity
DOF	: Degrees of Freedom
EKF	: Extended Kalman Filter
GNC	: Guidance, Navigation and Control
GPS/INS	: Global Positioning System / Inertial Navigation System
IMU	: Inertial Measurement Unit
LFT	: Linear Fractional Transformation
LPV	: Linear Parameter-Varying
LTI	: Linear Time-Invariant
LTV	: Linear Time-Varying
NED	: North-East-Down
(N)MPC	: (Nonlinear) Model Predictive Controller
MIMO	: Multi Input Multi Output
RHC	: Receding Horizon Control
SDRE	: State Dependent Riccati Equation
SQP	: Sequential Quadratic Programming
UAS	: Unmanned Aerial System/Agent

Glossary of Symbols, Subscripts and Superscripts

Common

Symbols or Variables

$\boldsymbol{I}$	: Inertial Coordinate System.
$\bar{\boldsymbol{I}}$	: Aircraft Inertial Coordinate System.
$\bar{\boldsymbol{B}}$	: Body Coordinate System.
$\bar{\boldsymbol{S}}$	: Stability Coordinate System.
$\bar{\boldsymbol{W}}$	: Wind Coordinate System.
m	: Aircraft Mass.
$\boldsymbol{J}_B$	: Inertia Matrix Referred to Coordinate System $\bar{\boldsymbol{B}}$.
U,V,W	: Body Velocity Components.
P,Q,R	: Body Angular Rates.
ϕ,θ,ψ	: Euler Angles.
α,β	: Airflow Angles Referred to Coordinate System $\bar{\boldsymbol{S}}$.
V_T	: Airspeed.
t	: Continuous Time Variable.
k	: Discrete Time Variable.
$\boldsymbol{x}$	: Aircraft State Vector.
$\boldsymbol{u}$	: Aircraft Control Input Vector.
$\boldsymbol{y}$	: Aircraft Output Vector.
δ	: Servo Deflection.
$\boldsymbol{f}$	: Aircraft Model Dynamic Equation.
$\boldsymbol{h}$	: Aircraft Model Observation Equation.
ΔT	: Sampling Time.

T_X^i	: X-axis Propulsive Parameter for $i = 0,1,2$.
L, D, F	: Lift, Drag, and Side Forces Referred to Coordinate System $\overline{\boldsymbol{S}}$.
X, Y, Z	: Aerodynamic Forces Referred to Coordinate System $\overline{\boldsymbol{B}}$.
L, M, N	: Aerodynamic Moments Referred to Coordinate System $\overline{\boldsymbol{B}}$.
C	: Aerodynamic Coefficients or
	: Stability and Control Derivatives.
S	: Wing Reference Area.
b	: Wing Span.
$\overline{c}$	: Mean Geometric Chord.
$p_N, p_E, p_D,$	: Inertial Position Components Referred to Coordinate System $\boldsymbol{I}$.
$w_N, w_E, w_H,$	: Wind Components Referred to Coordinate System $\boldsymbol{I}$.
$\boldsymbol{v}$	: Aircraft Model Dynamic Equation Noise.
$\boldsymbol{w}$	: Aircraft Model Observation Equation Noise.
Δ	: Perturbed Variable, Difference Between a Variable and its Trim Value.
$\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3$	: Robustness Augmentation Weighting Matrices.
e	: State Variable Error, Difference Between a Commanded Variable and its Value.
η	: Guidance Angular Error.
N	: NMPC Horizon Length.
g	: Gravity Acceleration.
$\boldsymbol{o}[k+i]$	: UAS Moving Point Information at Time $[k+i]$.

Subscripts

$N, E, H \ / \ D$	: North, East and Height/Down Components Referred to Coordinate System $\boldsymbol{I}$.
X, Y, Z	: Components Referred to Coordinate System $\overline{\boldsymbol{B}}$.
L, D, y	: Partial Derivative w/r to Aerodynamic Forces L, D, Y , respectively.

l, m, n	: Partial Derivative w/r to Aerodynamic Moments L, M, N, respectively.
A	: Aerodynamic Force.
$trim$	: Flight Trim Point.
Lat, Lon	: Lateral, Longitudinal Guidance Angular Error.

Superscripts

cmd	: Control Command.
T	: Engine Throttle.
E	: Elevator.
A	: Aileron.
R	: Rudder.

Chapter 1

Symbols or Variables

$\boldsymbol{F}_I$	: Force Referred to Coordinate System $\boldsymbol{I}$.
$\boldsymbol{M}_I$	: Moment Referred to Coordinate System $\boldsymbol{I}$.
$\boldsymbol{P}_I$	: Translational Momentum Referred to Coordinate System $\boldsymbol{I}$.
$\boldsymbol{H}_I$	: Angular Momentum Referred to Coordinate System $\boldsymbol{I}$.
$\boldsymbol{F}_B$	: Force Referred to Coordinate System $\bar{\boldsymbol{B}}$.
$\boldsymbol{M}_B$	: Moment Referred to Coordinate System $\bar{\boldsymbol{B}}$.
$\boldsymbol{P}_B$	: Translational Momentum Referred to Coordinate System $\bar{\boldsymbol{B}}$.
$\boldsymbol{H}_B$	: Angular Momentum Referred to Coordinate System $\bar{\boldsymbol{B}}$.
$\tilde{\boldsymbol{W}}_I$	: Cross Product Equivalent to $\boldsymbol{W}_I \times$.
$\tilde{\boldsymbol{W}}_B$	: Cross Product Equivalent $\boldsymbol{W}_B \times$.
$\boldsymbol{V}_I$	: Translational Inertial Velocity Vector.
$\boldsymbol{W}_I$	: Angular Inertial Velocity Vector.

$\boldsymbol{H}_B^I$: Orthonormal Rotation Matrix from $\overline{\boldsymbol{B}}$ to $\overline{\boldsymbol{I}}$.

$\boldsymbol{J}_I$: Inertia Matrix Referred to Coordinate System $\overline{\boldsymbol{I}}$.

$C_L^{\bar{S}} C_D^{\bar{S}}, C_Y^{\bar{S}}$: Lift, Drag and Side Force Coefficients, Referred to Coordinate System $\overline{\boldsymbol{S}}$.

$C_l^{\bar{S}} C_m^{\bar{S}}, C_n^{\bar{S}}$: Roll, Pitch and Yaw Moment Coefficients, Referred to Coordinate System $\overline{\boldsymbol{S}}$.

$\boldsymbol{z} = [z_1, z_2, \cdots, z_q]^T$: ANN Input.

M : ANN Hidden Layer Number of Cells.

$\boldsymbol{\theta} : a_{ji}, b_i, c_j, \boldsymbol{w}_i$: ANN Parameters (weights and biases).

p : ANN Number of Output Cells.

$F_j(\boldsymbol{z})$: ANN Output Function j .

φ : ANN Activation Function.

$\boldsymbol{\varepsilon}$: ANN Estimation Error Vector.

N_W : ANN Moving Window Length.

λ : ANN Levenberg’s Parameter.

J : ANN Estimation Error Vector $\boldsymbol{\varepsilon}$ Jacobian.

W : ANN Cost Function.

∇W : ANN Cost Function Gradient.

$\nabla^2 W$: ANN Cost Function Hessian.

Superscripts

g : Gravity acceleration.

P : Propulsive Force.

Chapter 2

Symbols or Variables

$\hat{\boldsymbol{f}}$: EKF Aircraft Model Dynamic Equation.

$\hat{\boldsymbol{h}}$: EKF Aircraft Model Observation Equation.

$\hat{\boldsymbol{x}}$	: EKF State Vector.
$\hat{\boldsymbol{u}}$	: EKF Control Input Vector.
$\hat{\boldsymbol{y}}$	: EKF Output Vector.
$\boldsymbol{K}$	: Kalman Gain Matrix.
$\boldsymbol{P}$	: EKF State Error Covariance Matrix.
$\boldsymbol{F}$	: EKF Dynamic Equation Jacobian Matrix.
$\boldsymbol{H}$	: EKF Observation Equation Jacobian Matrix.
$\boldsymbol{Q}$	: EKF Dynamic Equation Noise Covariance Matrix.
$\boldsymbol{R}$	: EKF Observation Equation Noise Covariance Matrix.
$H_{X,}H_Y,H_Z$	: Magnetic Flux Referred to Coordinate System $\overline{\boldsymbol{B}}$.
C_I^B	: Rotation Matrix from $\overline{\boldsymbol{I}}$ to $\overline{\boldsymbol{B}}$.
$[k/k-1]$	: EKF One Step Prediction From Time $[k-1]$ to $[k]$.
$[k/k]$	: EKF Correction at Time $[k]$.
$v_N,v_E,v_D,$	: Inertial Velocity Components Referred to Coordinate Systems $\boldsymbol{I}$.
$\overline{a}_X,\overline{a}_Y,\overline{a}_Z$	: Translational Acceleration Measurement Referred to Coordinate Systems $\overline{\boldsymbol{B}}$.

Subscripts

b	: State Bias.
0	: Inertial Magnetic Flux Referred to Flying Area.

Superscripts

^	: EKF Signals and Functions.

Chapter 3

Symbols or Variables

$\boldsymbol{G}$	: Linear Plant.

$\boldsymbol{P}$	: Augmented Linear Plant.
$\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{C}_1, \boldsymbol{C}_2,$ $\boldsymbol{D}_{11}, \boldsymbol{D}_{12}, \boldsymbol{D}_{21}, \boldsymbol{D}_{22}$	: Augmented Linear Plant Matrices.
$\boldsymbol{u}_1, \boldsymbol{u}_2$	: Augmented Linear Plant Inputs.
$\boldsymbol{y}_1, \boldsymbol{y}_2$	: Augmented Linear Plant Outputs.
H_∞	: H-infinity Control Design.
$\lVert \ \rVert_\infty$	: Infinity Norm.
$\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{T}$	: Sensitivity, Output Sensitivity, and Complementary Sensitivity Functions.
λ	: Gamma Value.
s	: Laplace Variable.
$\boldsymbol{X}, \boldsymbol{Y}$	: H_∞ Algorithm Riccati Equation Solution Matrices.
$\overline{\boldsymbol{K}}$	: H_∞ Stabilizing Suboptimal Controller.
$\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}_1, \overline{\boldsymbol{B}}_2, \overline{\boldsymbol{C}}_1, \overline{\boldsymbol{C}}_2,$ $\overline{\boldsymbol{D}}_{11}, \overline{\boldsymbol{D}}_{12}, \overline{\boldsymbol{D}}_{21}, \overline{\boldsymbol{D}}_{22}$	: H_∞ Controller Matrices.
$\boldsymbol{F}$	: H_∞ Feedback Matrix.
$\boldsymbol{H}$	: H_∞ Observer Matrix.
Δ	: Model Uncertainty Block.
η	: Model Uncertainty Block Output
υ	: Model Uncertainty Block Input
A_S, A_T, w	: Weighting Matrices Specifications
$\vec{\boldsymbol{a}}, \vec{\boldsymbol{b}}, \vec{\boldsymbol{c}}, \ldots$	: List of Trajectory Waypoint.
$\boldsymbol{ab}$	: Trajectory Segment Between Waypoints $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$.
$\vec{\boldsymbol{p}}$	: Aircraft Location in Guidance Law Geometry.
$\vec{\boldsymbol{d}}$	: Nearest Point form Aircraft to Trajectory Segment.
$\vec{\boldsymbol{r}}$	: Trajectory Target Point.
d_{pd}	: Distance Between Waypoints $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$.

d_{pb}	: Distance Between Point $\vec{\boldsymbol{p}}$ and point $\vec{\boldsymbol{b}}$.
$\bar{d}_{dr}$	: Fixed Distance Between Point $\vec{\boldsymbol{d}}$ and Point $\vec{\boldsymbol{r}}$.
$\bar{\boldsymbol{T}}$	: Trajectory Segment Coordinate System.
$\vec{\boldsymbol{L}}$	: Position Vector Subtended Between Points $\vec{\boldsymbol{p}}$ and $\vec{\boldsymbol{r}}$.
q_d	: Parametric Description for Point $\vec{\boldsymbol{d}}$.
m_r	: Parametric Description for Point $\vec{\boldsymbol{r}}$.
$\boldsymbol{C}_I^T$	: Direction Cosine Matrix Operation.
k_p	: Guidance Law Proportional Gain.
k_i	: Guidance Law Integral Gain.
k_a	: Guidance Law Adaptive Gain.

Subscripts

e	: Equilibrium Point.
$\boldsymbol{G}$	: Linear Plant.
$\boldsymbol{P}$	: Augmented Linear Plant.
un	: Uncertain Parameter.
$track$	: Trajectory Segment Rotation from Coordinate System $\boldsymbol{I}$ To $\bar{\boldsymbol{T}}$.
$cent$	: Centripetal Acceleration.

Superscripts

i	: Controller Index.
$-$	: H_∞ Controller.
u,l	: Upper, Lower LFT.
^	: Deterministic Augmented Linear Plant $\boldsymbol{P}$, (after uncertainty extraction).
C	: Uncertain Parameter Center Value.
H	: Uncertain Parameter Half Range Value.

Chapter 4

Symbols or Variables

$\boldsymbol{u} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N\}$: NMPC Control Numerical Solution.

V : NMPC Cost Function.

$\boldsymbol{f}_d$: NMPC Discretized Aircraft Model Dynamic Equation.

$\boldsymbol{h}_e$: NMPC Aircraft Model Output Equation.

$\boldsymbol{Q}_k, \boldsymbol{R}_k$: NMPC Time-Varying Weighting Matrices.

$\boldsymbol{U}, \boldsymbol{X}$: NMPC Input, State Constraint Sets.

$\boldsymbol{n}$: NMPC Measurement Noise.

$\boldsymbol{d}$: NMPC Aircraft Disturbance.

$\boldsymbol{c}$: NMPC Constraint Vector.

λ : NMPC Constraint Multipliers.

L : NMPC Augmented Cost Function (including constraints).

$\boldsymbol{A}$: NMPC Constraint Vector $\overline{\boldsymbol{c}}$ Jacobian Matrix.

∇V : NMPC Cost Function V Gradient.

$\nabla^2 V$: NMPC Cost Function V Hessian.

$\boldsymbol{v}$: NMPC Cost Function Quadratic Decomposition Vector.

$\boldsymbol{J}$: NMPC Cost Function Quadratic Decomposition Jacobian Matrix.

V_{ab} : Segment $\boldsymbol{ab}$ Inertial Velocity.

$\boldsymbol{z}^T = [(\boldsymbol{z}^1)^T, (\boldsymbol{z}^3)^T]$: NMPC Robustness Augmentation Input.

$\boldsymbol{x}^R$: NMPC Robustness Augmentation Aircraft Model State.

$\boldsymbol{f}_d^R$: NMPC Robustness Augmentation Aircraft Model Dynamic Equation.

Subscripts

p : Number of Valid Constraints.

MIN, MAX	: NMPC Minimum, Maximum Control Value.

Superscripts

i	: NMPC Iteration Index.
$-$	: NMPC Current Valid Constraints.
R	: NMPC Robustness Augmentation.
$\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3$	: NMPC Robustness Augmentation Weighting Matrices.

Chapter 5

Symbols or Variables

$\overline{\boldsymbol{R}}$	: Formation Coordinate System.
$\boldsymbol{U}^j[k+i]$	: Information Associated to UAS j at Time $[k+i]$.
$\boldsymbol{o}^j[k+i]$	: Moving Point Information Associated to UAS j at Time $[k+i]$.
$\Im^{RP}$	: Formation Predefined Trajectory Referred to RP.
V^{RP}	: Formation Cost Function Referred to RP.
$w_X^{\boldsymbol{U}^j}, w_Y^{\boldsymbol{U}^j}, w_Z^{\boldsymbol{U}^j}$	:Wind Field Components, Associated to UAS j Referred to $\overline{\boldsymbol{R}}$.
p_X, p_Y, p_Z	:Position Components, Referred to Coordinate System $\overline{\boldsymbol{R}}$.
N_f	: Failure Communication Horizon.

Subscripts

f	: Communication Failure Horizon Length.
j	: Agent Index within the Group.

Superscripts

j	: Agent Index within the Group.
RP	: Formation Reference Point.

0. Introduction

0.1 Motivation

The majorities of Earth and Science missions are dangerous and require fixed-wing unmanned aerial systems (UAS) with longer range and high payload and power capacity flying in an unstructured and hostile environment. The related cost and complexity of large fixed-wing UASs are an order of magnitude higher than small, but mobile, UAS; however, the geometry, power, and payload limitations of small UAS reduce their functionality and practicality dramatically. The motivation of this work is to develop an advanced flight control framework to distribute sensor payloads between several smaller and less complex agents with local interactions.

UAS formation flight requires the command and control of each agent, individually, considering their mutual interaction. But, aircraft have an unsteady, uncertain, and nonlinear dynamic, which is amplified by physical interactions between agents, and formation constraints. The classical aircraft command and control approaches are inadequate for this level of accuracy, so development and integration of advanced aircraft dynamic modeling and control techniques is a must. A numerically feasible nonlinear and time-varying aircraft model integrated to an adaptive, robust and nonlinear control is required for each agent.

0.2 Problem Statement

The standard unmanned aerial system command and control is depicted in Figure 01. This system is characterized by the following subsystems:

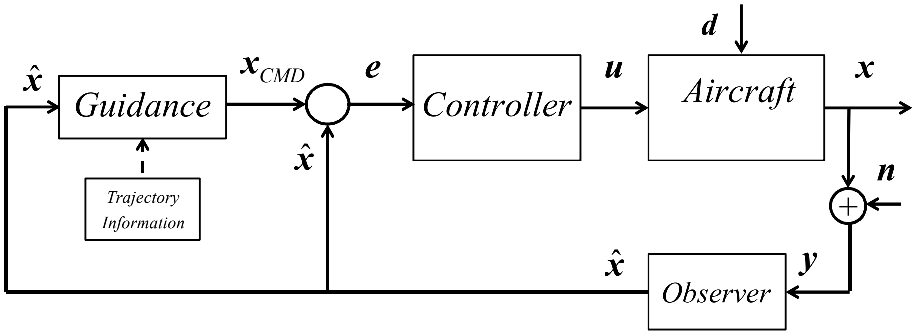

Figure 01. **Standard UAS Command and Control Configuration.**

- Controllers and observers that are linear and time-invariant, and limited to proximity of trim points.
- Controllers and observers that are based on linear dynamic models that do not account for changes in the aircraft physics-based model.
- A guidance subsystem that is independent of the control task, and designed separately.
- A trajectory subsystem that is limited to predefined curve shapes, or explicit model following.

This work is motivated to solve limitations outlined above by a feasible robust nonlinear model predictive control solution, as presented in Figure 02, coupled with an adaptive real-time system identification module to follow arbitrary trajectories.

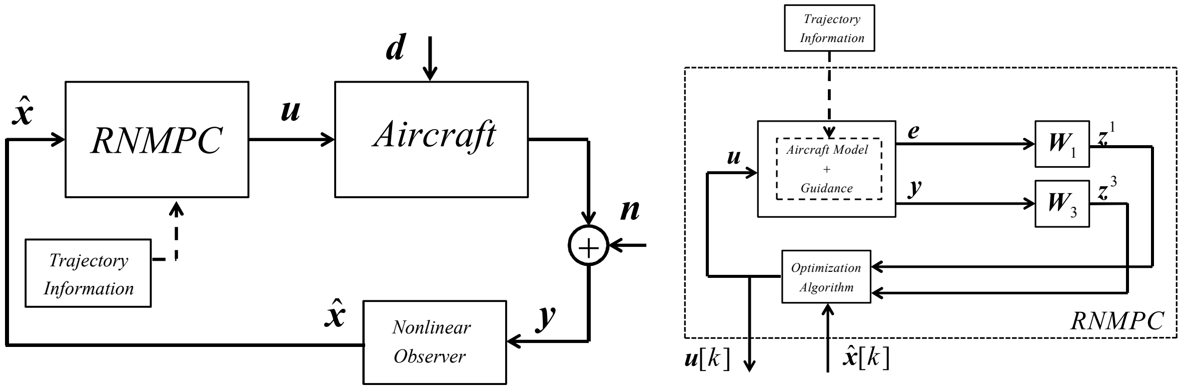

Figure 02. Adaptive, Robust NMPC Command and Control Configuration for UAS.

Specific designs and developments are tested in simulation and with previous stored flight test data. The main research goals are:

- Development of a real-time adaptive nonlinear physics-based model of an aircraft
- Development of a nonlinear attitude and speed guidance logic for flexible trajectory following based on arbitrary curves
- An integrated guidance and control task to avoid any cross coupling
- A feasible, adaptive, and robust numerical nonlinear and time-varying controller.

0.3 Literature Review

The most relevant references of the dissertation are detailed in the present section.

0.3.1 Aircraft Modeling

System modeling for control usually deals with a trade-off between complexity and tractability, i.e. how close the model is to the system against how easy is to handle control design and control implementation (Ref. [1]). Control design and control implementation, although related, need to be treated separately. Control implementation is simplified by the existence of increasingly fast and versatile hardware and software, so there is no important limitation here, but control design is still a difficult task which inherits the complexity of the system's model. The modeling should be as simple as possible while meeting control specifications.

When the control task is bounded by limited operation range, in the vicinity of one or more equilibrium points, the chosen framework for control design usually is the linear one. From a theoretical and phenomenological point of view, the aircraft dynamics and aerodynamics are highly nonlinear, time varying, and not without a significant level of uncertainty. For this reason, when the boundness premise is valid, a judicious approximation could be achieved leaving a single linear model or a set of linear models, either time invariant or time-varying, for linear control design. Ref. [2], [3] and [4] develop a very thorough description of the transition between the physical complexity and a manageable simpler description of an aircraft.

Other sources of modeling are closer to a black box type of description, where measured data is used to identify the system. Several methods and models have been reported. They range from linear to nonlinear and from adaptive to fixed. From a control perspective, these identified models should be simple and manageable enough to allow practical control design. During the process of model estimation, meaningful sub products could be obtained, as is the case of aerodynamic derivative coefficients. These elements greatly help to understand the physical behavior of the aircraft for the particular trim conditions under analysis. Enlightening works are found in Ref. [5] and [6] for parameter estimation in the time and frequency domain, respectively, where stability and control derivatives are estimated from data. A more recent work in parameter estimation is found in Ref. [7], where system identification is performed for adaptive and reconfigurable control by the incorporation of prior information, and in Ref. [8] by augmenting the model to include nonlinear terms. Unsteadiness is also addressed in Ref. [9] by estimating time-varying parameters.

Unsteady system modeling has also been addressed by fuzzy logic and Artificial Neural Networks (ANN), as these two techniques offer a powerful mapping capability (see Ref. [10] and [11] respectively). Ref. [12] and [13] use fuzzy logic for capturing an aircraft's aerodynamic unsteadiness through time varying estimation of aerodynamic coefficient derivatives, under the presence of adverse weather conditions. ANN has been extensively used to describe aerodynamics as well. Most of the applications implement the network as a fixed device, needed to be trained off-line with available data. Ref. [14] utilizes an ANN to estimate the normal force coefficient as a function of angle of attack covering a wider range. It performs the training stage off-line. Ref. [15] proposes the incorporation of first partial derivatives of the estimation errors with respect to ANN's inputs in the minimization, reporting important improvements. These results suggest the necessity of incorporating the time rate information of change of key states in the ANN training, as angle of attack or sideslip angle rate of change, to better model the aircraft unsteady dynamics. Ref. [16] trains an ANN to predict the aerodynamics unsteadiness over a wing as a function of angle of attack, and its time rate of change, by measuring the pressure at 45 points over the wing surface. Its results indicate an accurate prediction of the unsteady separation flow field as a function of the angle of attack and its rate of change. On the other hand, Ref. [17], and [18] estimate unstable aircraft stability and control derivatives under a closed loop configuration. They use a technique called the Delta method to extract these derivatives and the controller gains, which are used to stabilize the originally unstable aircraft and allow system identification, otherwise unfeasible. As a variant, Ref.

[19] proposes the use of ANN to predict ahead in time, as a way of modeling the aircraft dynamic representation, avoiding dealing explicitly with the equations of motion; and Ref. [20], proposes an approach to model the aerodynamic force and moment coefficients for lateral-directional motion from recorded flight data, based on neural partial differentiation, as an alternative to the Delta method, showing comparable results.

0.3.2 State Observers

Various studies and their applications in linear and nonlinear observing have been developed. The extended Kalman filter appears to be the most widely used, especially in navigation applications, where inertial accelerometers and gyros aided with GPS, and magnetometers are merged to get improved inertial position and velocity. This is done together with body attitude (Ref. [21]), improving measured signals and estimating inaccessible unmeasured states. More sophisticated filters have been proposed including particle filtering Ref. [22] and sigma-point Ref. [23], among others, to overcome the inherent limitation of the extended Kalman filter when the system under observation is highly nonlinear, and the Gaussian distribution assumption for system and observation noises no longer applies. Although these approaches are more accurate, their main drawback is that they are computationally expensive, which makes them not very suitable for sensitive time applications, such as an Unmanned Aerial System (UAS). A useful comparison between these filters is done in Ref. [24]. Another approach was presented in Ref. [25], that uses state dependent Riccati equation to design the observer, having the aircraft represented as a linear time varying model, without incurring in any forced approximation. With the system described in a time-varying linear format, which is not always possible, the Kalman filter is then applied.

Observers for airflow or wind components are not very common in the literature. Ref. [26] performs observation of the two horizontal wind components used to correct the velocity transformations between the involved coordinate systems. The model used is a zero time rate of change between measurements. In

Ref. [27], an unscented Kalman Filter is used to estimate the horizontal wind components, assuming small angle approximations. No three-wind components have been found in the literature.

As a way of correcting the misalignment of sensor installation, with respect to a common reference system, and inherent offset errors not detected by calibration processes, a set of sensor biases can be added to the observer as states. In order to model these new states the biases are generally assumed to be slow time-varying quantities (i.e. constant between measurements), but variable otherwise. The biases are characterized with a zero time rate of change (see Ref. [23]).

Another important feature of an observer is the management of redundant measurements from independent sensors as a way to improve the quality of the observation or estimation of states. Ref. [28] performs a comparison of two of the most used methods to fuse the data, by merging the multi-sensor data through the observation equation of the filter, or by combining the data based on a minimum-mean-square-error criterion, before entering the observer.

0.3.3 Guidance, Navigation, and Control

Any mission-based vehicle, either manned or unmanned, will require the interdependent tasks of guidance, navigation, and control, to accomplish any particular mission. In general, navigation systems indicate the position and velocity of a vehicle, and if these systems are placed in a closed loop with the vehicle’s actuators in order to control the position and velocity, guidance and control systems emerge (see Ref. [29]).

For UAS in particular, technology has evolved to a point where autonomy is extended to the entire mission, and entire flight envelope. Autonomous vehicles, developing aggressive maneuvers when following a trajectory or a moving target, require complex controllers with enough robustness and accuracy, being able to use all potential control capabilities of the aircraft. In Ref. [30], the authors presented a design of an integrated guidance and control algorithm for autonomous vehicles through a simultaneous process, achieving zero steady state error about a trim point while guaranteeing stability by avoiding possible cross coupling between loops. Their approach was based on gain scheduling theory; the trajectory tracking was achieved by switching among previously computed robust controllers. In Ref. [31], a lateral guidance and control law for UAS that handles a larger range of wind was presented. This approach is able to work with initially large cross-track errors from straight trajectory segments. A different approach is presented in Ref. [32], where an integrated trajectory tracking problem is formulated as a constrained nonlinear optimization problem. Unlike the receding horizon method which predicts the future, or an artificial neural network controller that uses the time history, the minimization must be performed with current data. Ref. [33] introduced nonlinear guidance logic for curved trajectory tracking, becoming a proportional-derivative control when applied to straight lines. This solution provided an anticipatory control element that improves the tracking capability when following a curved path. In Ref. [34] and [35], waypoint based guidance was improved by including the horizontal wind estimation, allowing the aircraft to smoothly converge to a new course after switching to the next trajectory segment in the presence of wind. This is an important development, especially for small UAS, where wind has a significant nonlinear effect on spatial orientation and consequently on the guidance algorithm. A different approach for path following is presented in Ref. [36], where 2D vector fields are used to represent the desired ground track and to generate course inputs for the inner loop attitude control, directing the aircraft to the desired track. An asymptotic stability is obtained for constant wind disturbances. More recently in Ref. [37], a Lyapunov-based adaptive back stepping approach is used to design a flight-path controller for a nonlinear F-16 model. A prescribed smooth inertial trajectory is asymptotically tracked in the presence of actuator failures and uncertainties in the stability and control derivatives. In a different approach, Ref. [38] proposes a spatial-temporal trajectory tracking capability for a fixed-wing aircraft on a time critical mission.

0.3.4 Gain Scheduling of Controllers

Gain scheduling is an approach widely used to control nonlinear systems over a broad range of operation, especially when linearity assumptions are no longer valid, by interpolating among a set of linear, and generally time invariant, controllers. When meeting predefined requirements, this linear-type technique is able to interpolate controllers matching the plant when transiting between several operating points achieving stability and performance (see Ref. [39], [40] and [41]). A number of researches and developments in this area have been done for UAS. Ref. [42] designs an H-infinity gain scheduled controller for the pitch-axis by interpolating poles, zeros, and gains from a set of four trim point controllers, removing to some extent the existence of hidden coupling terms. On the other hand, Ref. [43] proposes an interpolation of the Riccati equations from frozen longitudinal H-infinity controllers, including the flexible modes of the aircraft within the model. Another approach presented in Ref. [44] applies the Linear Parameter-Varying (LPV) control technique to missile pitch-axis dynamics expressed as a time-varying linear function of states. This solution fails to apply when nonlinear models are not expressible as linear functions of the states. Reference [45] presents two approaches to longitudinally control a nonlinear model of a UAS expressed as an LPV model, over the entire range of cruise speeds. The first one formulates the variability of the LPV model over the range of speeds as unstructured uncertainty, and designs a unique robust controller able to stabilize the system over the entire range. The second method linearly interpolates between two robust controllers calculated at the two extreme velocities, performing the gain scheduling as a function of the instantaneous total velocity. The work done in Ref. [46] applies the gain scheduling technique by representing the longitudinal dynamics of an aircraft as a second-order model. This approach allows the scheduled gains to be adjusted against rapidly-varying system states.

0.3.5 Nonlinear Model Predictive Controller

Changes in the dynamic or aerodynamic conditions due to nonlinear and time-varying nature of the UAS, together with external disturbances, can be addressed by an online scheduling version of pre-computed linear controllers (see, for example, Ref. [42] for robust designs). These pre-computed control laws, which are intended for scheduled dynamic changes, lack the ability to adjust to unforeseen events, like unexpected changes in UAS dynamics due to failures. Adaptive control assisted, for instance, by system identification (Ref. [7]), could overcome unanticipated alterations in either actuators, sensors or the UAS itself, by redesigning online new controllers that better reflect the new identified dynamics.

As for any controlled system, control and state variables and their rates of change are restricted within bounded ranges. These restrictions, eventually worsened by failures, augment the nonlinearity of the system and reduce the stability reserve. When control design does not explicitly include this constraint information, as in the case of linear control design, these constraints impose a severe difficulty in control tasks, either for the scheduling strategy or for the linear time-varying approach.

The Nonlinear Model Predictive Control (NMPC) approach, as a numerical control algorithm, is able to solve previous drawbacks in a direct way, by searching iteratively for an optimal control solution under changing dynamics and their constraints (Ref. [47]). Given the availability of computational resources for online numeric search, the NMPC algorithm is able to incorporate a complete system description and all variable restrictions without coarse approximations or simplifications. It is able to use the full nonlinear dynamic model of a plant, and all existing constraints, at each numeric cycle, even though they change from one cycle to the next. This ability is due to its numerical nature, as each cycle in the algorithm starts the search with the current system's model, independently of the one used in the previous cycle. By definition, NMPC is a nonlinear adaptive controller that is able to calculate as close to an optimal control law as possible, given the computational power, by using the best available representation of the system, possibly constrained and time-varying.

NMPC has been applied successfully in several areas, beginning with the chemical industry. Its application to unmanned vehicles is more recent and there are still promising advances to make. In Ref. [48], NMPC was applied to control a submarine in the longitudinal plane, where the vehicle was commanded to follow the sea bed at a constant distance, with no prior knowledge of its profile. Following this approach, Ref. [49] formulated a NMPC algorithm for a rotorcraft-based UAV, showing superior tracking performance over conventional multi-loop proportional-derivative controllers. Ref. [50] applied linear MPC to a small helicopter, easily incorporating control and state constraints, and reducing the typical computational burden of non-convex nonlinear programming. Another NMPC application is given in Ref. [51], where a receding horizon control algorithm is used for lateral road following, using passive sensors. NMPC has also been applied to compute evasive maneuvers in pursuit and evasion games (Ref. [52]), using a gradient-descent method.

Most past developments utilize the NMPC formulation in the outer loop for navigation and guidance purposes, leaving the inner loop control task to some commercial autopilot. Although this approach reduces the computational power requirement by reducing the size of the nonlinear model to be employed by the NMPC, it generates an undesired potential coupling between both loops, affecting the stability. Ref. [53], which employs this configuration, proved that the closed-loop system remains stable if a given initial condition can be driven into a predefined set of final states by the solution of a finite-horizon control problem. In their work, the inner loop is handled by a commercial low-level controller, and the NMPC minimizes in the outer loop the lateral cross track error to straight waypoint segments, converting the control law problem into a regulation problem.

Another interesting online application is found in Ref. [54]. The NMPC is used to control an inverted pendulum. The control law is able to work in the entire state space, including input and state constraints. This control problem is interesting to the KU Aerospace Engineering Flight System team, because it shares salient features with UAS's. It displays fast dynamics with highly nonlinear behavior, unstable about the operating point and showing a non-minimum phase nature. The online search uses a Sequential Quadratic Programming (SQP) approach, which works by breaking down the problem into a series of simpler sub-problems, each one approximated as a quadratic objective function subject to linearized constraints. This application included one control input and four states, two of them directly measured and two estimated.

A control system is said to be robust when stability is kept and an acceptable performance is reached for a given range of uncertainties and disturbances. Toward this end, a robust controller is expected to incorporate explicit information of known uncertainty ranges. An important branch of robust control is the calculation of a stabilizing control solution fulfilling a prescribed performance under the worst possible effect of uncertainty and disturbance. In the linear framework a common robust control technique that deals with the worst external disturbances and uncertainty effects is H-infinity, through an analytical LTI controller. This approach of worst case optimization represents a game theory paradigm, where the controller computes control values to minimize a given cost function while perturbations, chosen by "nature", try to maximize it (see Ref. [55]). This approach has been extended to the class of nonlinear plants described linearly by matrices with state-dependent entries (Ref. [56]), also called frozen Riccati equations. In this approach, state-dependent Riccati equations are solved to obtain a linear time-varying controller. In a more general framework, nonlinear systems showing input affine relation are addressed by seeking an approximation of the solution of the associated Hamilton-Jacobi equation (see Ref. [57]-[58]). An application to general six degree-of-freedom motion is presented in Ref. [59] and for a fixed wing UAS in Ref. [60].

A direct numeric application of the previous approach, where the control variable is obtained by minimizing the cost function while the disturbance worst case is selected to maximize it, is the robust min-max NMPC. Implementation of min-max NMPC suffers from the colossal computational burden that currently limits its applicability. Ref. [61] and [62] propose an approximation of the min-max optimal solution by reducing the problem to a minimization of the cost function in terms of future control inputs, subject to an upper bound of worst possible disturbance and uncertainty. Both cases propose an approach to compute the worst case for the consequent minimization, showing a slightly conservative control law and reducing the computational time. Another strategy to provide robustness is based on a separate design of two control loops. This is an inner linear MPC guaranteeing the tracking of the controlled variables in the absence of constraints and an outer nonlinear static law called a command governor modifying the reference signals to be delivered to the inner loop, by taking into account its limitations (see Ref. [63] and [64]). Although an interesting approach, it still keeps two separate control loops.

There is another approach to deliver a feasible robust NMPC with much less computational demand. The idea is inspired by the use of frequency-dependent weighting matrices augmenting the system's model by defining new controlled outputs to minimize, as in the linear cases of H-infinity or H-2 (see Ref. [65]) and nonlinear H-infinity (Ref. [66]). In this way, robustness is provided together with performance; two concepts originally competitors. A standard NMPC algorithm is then applied to the system augmented by the inclusion of the frequency-dependent matrices. The NMPC algorithm will numerically select the control input sequence that minimizes the new controlled output, neutralizing the effect of external disturbances, uncertainties, and noise, whose frequency content are assumed known.

0.3.6 Aircraft Formation

The study is inspired by the physics involved in birds' formations where each individual has a limited knowledge of the whole group (constrained by their physiological sensors), where there is not necessarily a leader, centralized commanding or communication and control, but where there is a high level of coordination. This is done by following a very simple set of rules achieving remarkable objectives as a whole body, called emergent behavior. Ref. [67] studies the interaction logic of animal collective behavior, whereby reconstructing the three-dimensional position of a few individuals in flocks of birds, it is shown that the interaction does not depend on the metric distance, but rather on the topological distance. Each bird modifies its position relative to the six or seven animals directly surrounding it, no matter how close or how far away those animals could be.

One of the most important joint behaviors is the optimization of the aerodynamics involved in their maneuvers, decreasing important amounts of total drag. One of these formations is the V-formation that improves the efficiency and range of flying birds, particularly over long migratory routes. Birds flying in the upwash formed by the birds ahead are able to improve their lift and reduce their induced drag. A formation of UAS could take advantage of these aerodynamic effects, decreasing their energy consumption. Ref. [68] shows that in a V-formation of 25 members, each bird can achieve a reduction of induced drag by up to 65% and as a result increase their range by 71%. On the other hand, Ref. [69] explores an approach based on simulation as an alternative to scripting the paths of each bird individually. The simulated flock is an elaboration of a particle system, each bird being a particle. The aggregated motion of the simulated flock is created by a distributed behavioral model much like that at work in a natural flock; the birds choose their own course. The paper utilizes simple mathematical models of swarms given by the following three rules: move in the same direction as its neighbors, remain close to its neighbors, and avoid collisions with its neighbors.

For control purposes, the aerodynamics of the formation needs to be well understood to be described by a simple but representative mathematical model (expected to be nonlinear and possibly time varying). This model can then be used to design a particular control solution. Ref. [70] performs a comparison of predicted and measured formation flight interference effects. It compares the data obtained from a wind-tunnel test of two delta-wing aircraft in close proximity, with predictions from a vortex lattice method. Large changes in lift, pitching moment, and rolling moments are found on the trail aircraft as it moves laterally relative to the lead aircraft. In Ref. [71] a tight formation flight problem is addressed, consisting of a leading aircraft and a wing aircraft. The lead’s trailing vortices aerodynamically couple both aircraft. In close formation, the wing aircraft experiences an upwash field generated by the preceding airplane inducing a reduction in required overall energy. The manuscript defines tight formation as one where the lateral separation between aircraft is less than the wingspan. It delivers an analytical equation for the optimal separation calculation.

Ref. [72] studies the interaction of two aircraft flying in close formation. By positioning one craft flying in the correct position within the vortex wake of the other, substantial reductions in drag could be attained, with the amount of the reduction dependent on the relative positions of the two aircraft. The precise position of the second craft relative to the first one that maximizes the drag reduction is to be determined iteratively online leading to a peak-seeking control problem. A method of peak-seeking control using a Kalman filter to estimate the characteristics of the drag reduction is derived.

A study of a nonlinear control of multiple UAS in close-coupled formation flight is done in Ref. [73]. This article studies the control of UAS flying in a close-coupled formation for the purposes of drag reduction. Inner and outer loops are employed in a two-UAS formation where the outer loop controller is based on sliding mode control and the inner uses adaptive dynamic inversion to track the commands. The controller configuration enables the wing UAS to maneuver in the lead UAS's wake, and to hold a predefined desired position within it. The induced lift on the wing UAS due to formation flight is shown to cause reduced angle-of-attack and thrust for this wing vehicle.

Decentralized control, as one of the main mimics of bird formations to be accomplished, allows more efficient control calculation by using the distributed available computational power. Each airplane is able to synthesize its own control by knowledge of its status and possibly the status of its neighbors. Even with the existence of a centralized commanding device, the control calculation is more efficient when done in a distributive way. Ref. [74] proposes a decentralized nonlinear robust control system based on outer and inner control loops for close formation flight of multiple UAS. This control system keeps a predefined (based in Ref. [73]) separation distance of the wing airplanes calculated to achieve the minimum drag. The results confirm that when the wing aircraft is positioned properly in the vortex of the leading aircraft, it experiences reduction in its required flight power. In closed-loop, prescribed separation trajectories are asymptotically tracked by each wing aircraft, while the lead aircraft is free to perform trajectory following maneuvers.

Viewed as a system, the V-formation is a highly nonlinear, time-varying, and constrained system. To guarantee a broader operation range, a set of adaptive and nonlinear control laws are required. The NMPC approach appears to be a natural candidate, as it comprises a significant level of nonlinearity and adaptation management, while allowing explicit accommodation of constraints, granting a remarkable level of optimality. A decentralized algorithm for robust constrained linear MPC is proposed in Ref. [75]. This decentralized formulation for a MPC of systems incorporate coupled constraints. A single large planning optimization is divided into small sub problems, each planning only for the states of a particular subsystem. Relevant plan data is exchanged between subsystems to ensure that all decisions are consistent with satisfaction of the coupled constraints. A typical application would be autonomous guidance of a fleet of UAS in which the systems are coupled by the need to avoid collisions, but each vehicle plans only its own path.

Ref. [76] describes the application of a methodology for high-level control and coordination of autonomous vehicle teams. The scheme employs decentralized receding horizon controllers that reside on each vehicle to achieve coordination among team members. On each vehicle, information about neighbors is used to predict their behavior and plan conflict-free trajectories that maintain coordination and achieve team objectives.

As a variant based on the nonlinear model of multi-UAS close formation, Ref. [77] proposes a control strategy using a hybrid receding horizon control and differential evolution algorithm. Three simulation experiments are performed and simulation results show the feasibility and validity of the proposed control algorithm; the aerodynamic separation is based on Ref. [73]. A distributed control scheme to generate stable flocking motion for a group of agents is proposed in Ref. [78]. In this control scheme a molecular potential field model is applied as the potential. The approach of distributed receding horizon control is adopted to drive each agent to find its optimal control input to lower its potential at every step.

0.4 Significance and Originality

0.4.1 Significance

A decentralized flight control system is developed for an unmanned aerial system to perform arbitrary trajectory following in the presence of modeling uncertainties and external disturbances. The control design is a robust, adaptive, and nonlinear solution. This work has the following four significances:

- A stable guidance logic for arbitrary trajectory following is developed. This original development is composed of nonlinear attitude command and airspeed command laws. An existent nonlinear horizontal guidance law for circular trajectories is expanded and improved. Jointly, a vertical guidance law is developed with similar capabilities. The new attitude guidance allows the following of arbitrary trajectories, defined by moving points.
- Another original contribution of this research is the unification of guidance and control loops into a single loop. Integration of a unified loop allows computation of a single solution for both guidance and robust nonlinear model predictive control. As a result, a guidance and control solution is achieved for an extended flight envelope over a broader range of operation, independent of a trim condition or linear gain scheduling interpolation between several trim conditions.
- In order to maximize the performance of the nonlinear model predictive controller and state observer, a new comprehensive and adaptive parameter identification is developed to update aircraft physics-based models in real-time. An online set of adaptive artificial neural networks are utilized to capture and characterize an aircraft's nonlinear and unsteady aerodynamic and propulsive forces, complementing aircraft modeling. The quality of observed states and performance of the controller are a function of the accuracy of the physics-based model. Consequently, adaptive, nonlinear, and unsteady

modeling of aerodynamic and propulsive forces maximizes the accuracy and performance of robust nonlinear model predictive control and the quality of extended Kalman filtering estimations.

➢ Nonlinear model predictive control robustness deficiency is compensated by applying the concept of mixed sensitivity through the addition of a frequency-dependent robustness capability. This innovative configuration is a successful numerical enhancement of low frequency tracking capability and disturbance attenuation and high frequency measurement noise and model uncertainty rejection.

For the first time in the open literature, the feasibility of a robust nonlinear model predictive control coupled with real-time and adaptive artificial neural networks, which integrates the nonlinear guidance and control laws, is achieved for a UAS. The performance of an advanced developed and integrated decentralized control solution is verified by performing challenging tasks required for highly precise flight.

0.4.2 Originality

Based on the Literature Review, this work has the following originalities:

- Development of stable guidance logic for arbitrary trajectory following composed of nonlinear attitude command and airspeed command laws.
- Guidance logic and trajectory information embedding into the aircraft modeling, providing a single guidance and control model, to be used in a nonlinear model predictive control formulation.
- Online adaptive artificial neural networks application for aerodynamic force estimation, to be used in the nonlinear model predictive control formulation.
- Feasible robustness augmentation of a nonlinear model predictive control approach, based on full fixed-wing rigid aircraft modeling, augmented by a set of adaptive artificial neural networks for nonlinear aerodynamic estimation, and by a nonlinear guidance law for arbitrary trajectory following.

0.5 Dissertation Content Description

The document comprises six chapters. The present chapter is the introductory chapter. First, second, third, and fourth chapters are directly derived from the dissertation research. Chapter five includes a conceptual description of centralized aircraft formations, and it is not part of the development. Its purpose is to solely offer a motivation for future research.

1. Aircraft Modeling

1.1 Introduction

Modeling a system has several applications. One application is for control purposes. Despite the chosen control design technique, the available knowledge of the system, composed of system variables and their relationships, is needed in the design of a control law and its application to the system. The relationships among the known variables of the system have to be reflected in a mathematical model. Its complexity will be determined by the complexity of the system, but also by the level of accuracy demanded from the control system. This model is based on available system information, and its design should be guided by usefulness rather than "truth" or exactness (Ref. [1]).

In the modeling task, two different directions are clearly distinguishable; although, generally, they are jointly used: a phenomenological approach, where the system is decomposed into simpler subsystems whose laws of physics are known, and an experimental approach, where data from experiments are used to infer a model without any prior knowledge of the system, also called system identification (Ref. [1]). Even though these two routes are self-contained and can, theoretically, provide useful models separately, the typical practice is to merge both into one solution. That is, to establish the physics of the system from the available knowledge to a certain level of complexity and use data from experiments to estimate unknown sections of the system and/or unknown physical parameters within it. This solution retains the best features of both approaches; it has a broader range of validity and it becomes a good system predictor as it is fitted to the available data.

1.2 Aircraft Equations of Motion

The building block of this research is the rigid body fixed-wing aircraft, to be modeled in a non-rotational flat earth. A complete modeling of the aircraft requires several coordinate systems. Figure 1 shows all related coordinate systems centered in the aircraft center of gravity. They consist of the inertial coordinate system $\bar{\boldsymbol{I}}$, aligned with the inertial local flat earth coordinate system $\boldsymbol{I}$, both oriented as

North-East-Down (NED), the body coordinate system $\bar{\boldsymbol{B}}$, fixed to the aircraft, the wind coordinate system $\bar{\boldsymbol{W}}$, oriented toward the relative wind, and the stability coordinate system $\bar{\boldsymbol{S}}$, oriented as shown in Figure 1. Body $\bar{\boldsymbol{B}}$, stability $\bar{\boldsymbol{S}}$, and wind $\bar{\boldsymbol{W}}$ coordinate systems are interrelated through the airflow angles, i.e. the angle of attack α and sideslip angle β, to de defined later.

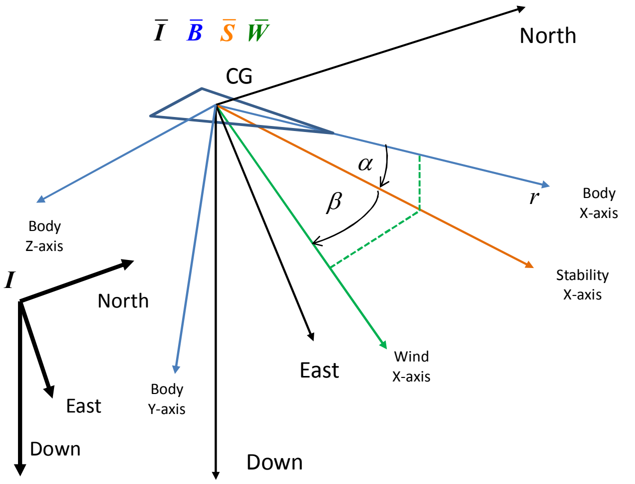

Figure 1. Coordinate Systems for Aircraft Modeling.

Based on Newton-Euler equations (Ref. [2] and [3]), the translational and rotational dynamics of the aircraft center of gravity (CG) referred to the inertial coordinate system $\bar{\boldsymbol{I}}$, (which relate the forces $\boldsymbol{F}_I$ and moments $\boldsymbol{M}_I$ acting on the system to the time rate of change of the translational momentum $\boldsymbol{P}_I$ and angular momentum $\boldsymbol{H}_I$, respectively) are defined as follows

$$Translational: \quad \frac{d\boldsymbol{P}_I(t)}{dt} = \frac{d(m(t)\boldsymbol{V}_I(t))}{dt} = m(t)\frac{d\boldsymbol{V}_I(t)}{dt} + \frac{dm(t)}{dt}\boldsymbol{V}_I(t) = \boldsymbol{F}_I(t) \tag{1}$$

$$Rotational: \quad \frac{d\boldsymbol{H}_I(t)}{dt} = \frac{d(\boldsymbol{J}_I(t)\boldsymbol{W}_I(t))}{dt} = \boldsymbol{J}_I(t)\frac{d\boldsymbol{W}_I(t)}{dt} + \frac{d\boldsymbol{J}_I(t)}{dt}\boldsymbol{W}_I(t) = \boldsymbol{M}_I(t) \tag{2}$$

where $\boldsymbol{V}_I = [V_N, V_E, V_D]^T$ is the translational inertial velocity vector, $\boldsymbol{W}_I = [W_N, W_E, W_D]^T$ is the angular velocity vector, m is the aircraft mass, and $\boldsymbol{J}_I$ is the inertia matrix defined in the inertial frame $\bar{\boldsymbol{I}}$ as

$$\boldsymbol{J}_I(t) = \begin{bmatrix} +J_{\bar{I}xx}(t) & -J_{\bar{I}xy}(t) & -J_{\bar{I}xz}(t) \\ -J_{\bar{I}yx}(t) & +J_{\bar{I}yy}(t) & -J_{\bar{I}yz}(t) \\ -J_{\bar{I}zx}(t) & -J_{\bar{I}zy}(t) & +J_{\bar{I}zz}(t) \end{bmatrix} \tag{3}$$

In order to express Equations (1) and (2) in terms of quantities referred to the body coordinate system $\bar{\boldsymbol{B}}$, and because the body reference frame $\bar{\boldsymbol{B}}$ rotates with respect to the inertial reference frame $\bar{\boldsymbol{I}}$ at an angular rate $\boldsymbol{W}_I$, the time rate of change of translational and angular momentum are expressible as

$$\textit{Translational}:\quad \frac{d\boldsymbol{P}_I(t)}{dt} = \left[\frac{d\boldsymbol{P}_B(t)}{dt}\right]_{\bar{\boldsymbol{I}}} + \boldsymbol{W}_I(t)\times\boldsymbol{P}_I(t) = \boldsymbol{H}_B^I\frac{d\boldsymbol{P}_B(t)}{dt} + \tilde{\boldsymbol{W}}_I(t)\boldsymbol{P}_I(t) = \boldsymbol{F}_I(t) \tag{4}$$

$$\textit{Rotational}:\quad \frac{d\boldsymbol{H}_I(t)}{dt} = \left[\frac{d\boldsymbol{H}_B(t)}{dt}\right]_{\bar{\boldsymbol{I}}} + \boldsymbol{W}_I(t)\times\boldsymbol{H}_I(t) = \boldsymbol{H}_B^I\frac{d\boldsymbol{H}_B(t)}{dt} + \tilde{\boldsymbol{W}}_I(t)\boldsymbol{H}_I(t) = \boldsymbol{M}_I(t) \tag{5}$$

where $\tilde{\boldsymbol{W}}_I$ is the cross-product equivalent $\boldsymbol{W}_I$, and $\boldsymbol{H}_B^I$ is an orthonormal matrix for rotation of vectors from body-axes frame to inertial-axes frame (see Ref. [2]). Equations (4) and (5) are restated as

$$\textit{Translational}:\quad \boldsymbol{H}_B^I\frac{d\boldsymbol{P}_B(t)}{dt} + \tilde{\boldsymbol{W}}_I(t)\overbrace{\boldsymbol{H}_B^I\boldsymbol{P}_B(t)}^{\boldsymbol{P}_I(t)} = \boldsymbol{H}_B^I\frac{d\boldsymbol{P}_B(t)}{dt} + \boldsymbol{H}_B^I\overbrace{(\boldsymbol{H}_B^I)^{-1}\tilde{\boldsymbol{W}}_I(t)\boldsymbol{H}_B^I}^{\tilde{\boldsymbol{W}}_B(t)}\boldsymbol{P}_B(t) = \boldsymbol{H}_B^I\boldsymbol{F}_B(t) \tag{6}$$

$$\textit{Rotational}:\quad \boldsymbol{H}_B^I\frac{d\boldsymbol{M}_B(t)}{dt} + \tilde{\boldsymbol{W}}_I(t)\overbrace{\boldsymbol{H}_B^I\boldsymbol{H}_B(t)}^{\boldsymbol{H}_I(t)} = \boldsymbol{H}_B^I\frac{d\boldsymbol{H}_B(t)}{dt} + \boldsymbol{H}_B^I\overbrace{(\boldsymbol{H}_B^I)^{-1}\tilde{\boldsymbol{W}}_I(t)\boldsymbol{H}_B^I}^{\tilde{\boldsymbol{W}}_B(t)}\boldsymbol{H}_B(t) = \boldsymbol{H}_B^I\boldsymbol{M}_B(t) \tag{7}$$

which gives

$$\textit{Translational}:\quad \frac{d\boldsymbol{P}_B(t)}{dt} + \tilde{\boldsymbol{W}}_B(t)\boldsymbol{P}_B(t) = m(t)\frac{d\boldsymbol{V}_B(t)}{dt} + \frac{dm(t)}{dt}\boldsymbol{V}_B(t) + \tilde{\boldsymbol{W}}_B(t)m(t)\boldsymbol{V}_B(t) = \boldsymbol{F}_B(t) \tag{8}$$

$$\textit{Rotational}:\quad \frac{d\boldsymbol{H}_B(t)}{dt} + \tilde{\boldsymbol{W}}_B(t)\boldsymbol{H}_B(t) = \boldsymbol{J}_B(t)\frac{d\boldsymbol{W}_B(t)}{dt} + \frac{d\boldsymbol{J}_B(t)}{dt}\boldsymbol{W}_B(t) + \tilde{\boldsymbol{W}}_B(t)\boldsymbol{J}_B(t)\boldsymbol{W}_B(t) = \boldsymbol{M}_B(t) \tag{9}$$

For the present work, both mass and inertia are assumed constant, i.e. $dm(t)/dt = 0$ and $d\boldsymbol{J}_B(t)/dt = \boldsymbol{0}$. Defining the body-axes velocity $\boldsymbol{V}_B = [U,V,W]^T$, body-axes angular rate $\boldsymbol{W}_B = [P,Q,R]^T$ and

$$\tilde{\boldsymbol{W}}_B(t) = \begin{bmatrix} 0 & -R(t) & Q(t) \\ R(t) & 0 & -P(t) \\ -Q(t) & P(t) & 0 \end{bmatrix} \tag{10}$$

the translation and rotation equations in the body-axes frame are

$$\textit{Translational}:\quad \frac{d}{dt}\begin{bmatrix} U(t) \\ V(t) \\ W(t) \end{bmatrix} + \begin{bmatrix} 0 & -R(t) & Q(t) \\ R(t) & 0 & -P(t) \\ -Q(t) & P(t) & 0 \end{bmatrix}\begin{bmatrix} U(t) \\ V(t) \\ W(t) \end{bmatrix} = m^{-1}\boldsymbol{F}_B(t) \tag{11}$$

$$Rotational: \quad \frac{d}{dt}\begin{bmatrix} P(t) \\ Q(t) \\ R(t) \end{bmatrix} + \boldsymbol{J}_B{}^{-1}\begin{bmatrix} 0 & -R(t) & Q(t) \\ R(t) & 0 & -P(t) \\ -Q(t) & P(t) & 0 \end{bmatrix}\boldsymbol{J}_B\begin{bmatrix} P(t) \\ Q(t) \\ R(t) \end{bmatrix} = \boldsymbol{J}_B{}^{-1}\boldsymbol{M}_B(t) \tag{12}$$

where $\boldsymbol{J}_B$ is the inertia matrix defined in the body-axes frame as follows, and where the subscript $\bar{\boldsymbol{B}}$ is subsequently omitted

$$\boldsymbol{J}_B = \begin{bmatrix} +J_{\bar{\boldsymbol{B}}xx} & -J_{\bar{\boldsymbol{B}}xy} & -J_{\bar{\boldsymbol{B}}xz} \\ -J_{\bar{\boldsymbol{B}}yx} & +J_{\bar{\boldsymbol{B}}yy} & -J_{\bar{\boldsymbol{B}}yz} \\ -J_{\bar{\boldsymbol{B}}zx} & -J_{\bar{\boldsymbol{B}}zy} & +J_{\bar{\boldsymbol{B}}zz} \end{bmatrix} \tag{13}$$

The two previous Equations (11) and (12), together with the following translational kinematics Equation (14), which relates the angular rate vector observed in the body-axes frame $\boldsymbol{W}_B$ to the rate of change of the Euler angle $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$ (Ref. [2]), conform a set of nine equations in the nine variable vector $[U, V, W, P, Q, R, \phi, \theta, \psi]^T$, that fully describes the rigid body aircraft over a flat, non-rotating earth

$$\begin{bmatrix} P(t) \\ Q(t) \\ R(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta(t) \\ 0 & \cos\phi(t) & \sin\phi(t)\cos\theta(t) \\ 0 & -\sin\phi(t) & \cos\phi(t)\cos\theta(t) \end{bmatrix}\begin{bmatrix} \dot{\phi}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} \tag{14}$$

Forces $\boldsymbol{F}_B$ and moments $\boldsymbol{M}_B$ are driving the system, as is apparent from Equations (11) and (12). Forces are formed by the addition of gravitational $[\cdot]^g$, propulsive $[\cdot]^P$, and aerodynamic $[\cdot]^A$ effects, i.e. $\boldsymbol{F}_B = [\boldsymbol{F}_B]^g + [\boldsymbol{F}_B]^P + [\boldsymbol{F}_B]^A$, while moments only comprise propulsive and aerodynamic, i.e., $\boldsymbol{M}_B = [\boldsymbol{M}_B]^P + [\boldsymbol{M}_B]^A$ effects, as gravity is assumed uniform. Although gravity g can be considered as an uncontrollable input to the system, it is treated as a constant. Controllable inputs to the aircraft are the engine throttle and elevator, aileron, and rudder servo deflection commands, i.e. $\boldsymbol{u}(t) = [\delta_T^{cmd}(t), \delta_E^{cmd}(t), \delta_A^{cmd}(t), \delta_R^{cmd}(t)]^T$.

Based on Ref. [2], [3], and [4], all forces and moments can be defined, up to some degree of approximation, as functions of the states of the aircraft $\boldsymbol{x}(t) = [U(t), V(t), W(t), \phi(t), \theta(t), \psi(t), P(t), Q(t), R(t)]^T$ and control inputs $\delta_T(t), \delta_E(t), \delta_A(t), \delta_R(t)$. Servo dynamics are considered an integral part of the aircraft dynamics, so servo deflections, a function of servo commands $\boldsymbol{u}(t)$ as will be seen later, will be added to the aircraft state vector $\boldsymbol{x}(t)$, i.e. $\boldsymbol{x}(t)=[U(t),V(t),W(t),\phi(t),\theta(t),\psi(t),P(t),Q(t),R(t),\delta_T(t),\delta_E(t),\delta_A(t),\delta_R(t)]^T$. Forces and moments are then generically represented by $\boldsymbol{F}_B=f_{F_B}(\boldsymbol{x},t)$ and $\boldsymbol{M}_B=f_{M_B}(\boldsymbol{x},t)$.

From a sensor perspective, airflow angles α and β, and airspeed V_T are much more accessible to measure than body velocity components U,V,W . The bijective equations, clear from Figure 1,

$$\begin{aligned} U &= V_T\cos\alpha\cos\beta \\ V &= V_T\sin\beta \\ W &= V_T\sin\alpha\cos\beta \end{aligned} \tag{15}$$

are used to interchange them. A more practical structure of the state vector is then defined as $\boldsymbol{x}(t)=[V_T(t),\alpha(t),\beta(t),\phi(t),\theta(t),\psi(t),P(t),Q(t),R(t),\delta_E(t),\delta_A(t),\delta_R(t)]^T$.

1.2.1 Gravity Forces $[\boldsymbol{F}_B]^g$

The gravity force effect in the body coordinate system is the most straightforward in terms of its definition, given its known intensity and direction. Equation (16) shows its components after rotating from the inertial coordinate system $\bar{\boldsymbol{I}}$ to body $\bar{\boldsymbol{B}}$

$$[\boldsymbol{F}_B(\boldsymbol{x},t)]^g=\begin{bmatrix} X^g(\boldsymbol{x},t) \\ Y^g(\boldsymbol{x},t) \\ Z^g(\boldsymbol{x},t) \end{bmatrix}=\begin{bmatrix} -mg\sin\theta(t) \\ mg\cos\theta(t)\sin\phi(t) \\ mg\cos\theta(t)\cos\phi(t) \end{bmatrix} \tag{16}$$

1.2.2 Propulsive Forces $[\boldsymbol{F}_B]^P$ and Moments $[\boldsymbol{M}_B]^P$

Propulsive force, or thrust, is highly dependent on the type and configuration of the engine (see Ref. [86]). For the present research, only the x-axis force is considered by assuming that the engine's effect is perfectly aligned with this axis. Equation (17) shows the propulsive force expression

$$[\boldsymbol{F}_B(\boldsymbol{x},t)]^P=\begin{bmatrix} X^P(\boldsymbol{x},t) \\ Y^P(\boldsymbol{x},t) \\ Z^P(\boldsymbol{x},t) \end{bmatrix}=\begin{bmatrix} f_{X^P}(\boldsymbol{x},t) \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

where the function $f_{X^P}(\boldsymbol{x},t)$ that represents the thrust exerted by the engine to the aircraft in the x-axis is undetermined and needs to be approximated, defining its structure and estimating its parameters, e.g. by using system identification. For the present research, the chosen structure is a polynomial expression of degree two as a function of the engine throttle position δ_T, i.e. $X^P(\boldsymbol{x},t) = T_X^2 \cdot \delta_T{}^2(t) + T_X^1 \cdot \delta_T(t) + T_X^0$, with parameters T_X^2, T_X^1, and T_X^0.

No propulsive moments are assumed in this work.

1.2.3 Aerodynamic Forces $[\boldsymbol{F}_B]^A$ and Moments $[\boldsymbol{M}_B]^A$

Determination of aerodynamic forces and moments is an intricate task. There is a complex physical dynamic and time-varying relation between the aerodynamic forces X^A, Y^A, Z^A and moments L^A, M^A, N^A and the status of the aircraft $\boldsymbol{x} = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, \delta_T, \delta_E, \delta_A, \delta_R]^T$. Forces and moments can be defined in terms of the aircraft states and, at the same time, the state of the aircraft is affected by the aerodynamic forces and moments. This interrelation makes it very difficult to analytically obtain the underlying cause-effect relation.

Following Ref. [2]-[4], it is customary to express the aerodynamic forces and moments in a normalized way, as a function of the aerodynamic force C_X, C_Y, C_Z and moment C_l, C_m, C_n coefficients.

$$[\boldsymbol{F}_B(\boldsymbol{x},t)]^A = \begin{bmatrix} X^A(\boldsymbol{x},t) \\ Y^A(\boldsymbol{x},t) \\ Z^A(\boldsymbol{x},t) \end{bmatrix} = \begin{bmatrix} f_{X^A}(\boldsymbol{x},t) \\ f_{Y^A}(\boldsymbol{x},t) \\ f_{Z^A}(\boldsymbol{x},t) \end{bmatrix} = \begin{bmatrix} \overline{q}(t) S C_X(\boldsymbol{x},t) \\ \overline{q}(t) S C_Y(\boldsymbol{x},t) \\ \overline{q}(t) S C_Z(\boldsymbol{x},t) \end{bmatrix} \tag{18}$$

$$[\boldsymbol{M}_B(\boldsymbol{x},t)]^A = \begin{bmatrix} L^A(\boldsymbol{x},t) \\ M^A(\boldsymbol{x},t) \\ N^A(\boldsymbol{x},t) \end{bmatrix} = \begin{bmatrix} f_{L^A}(\boldsymbol{x},t) \\ f_{M^A}(\boldsymbol{x},t) \\ f_{N^A}(\boldsymbol{x},t) \end{bmatrix} = \begin{bmatrix} \overline{q}(t) S b C_l(\boldsymbol{x},t) \\ \overline{q}(t) S \overline{c} C_m(\boldsymbol{x},t) \\ \overline{q}(t) S b C_n(\boldsymbol{x},t) \end{bmatrix} \tag{19}$$

where $\overline{q}(t) = 0.5\rho(t)V_T(t)$ is the dynamic pressure, defined as a function of the airspeed $V_T(t)$, and air density $\rho(t)$. In this work, air density is considered constant. The fixed parameters S, b and $\overline{c}$ are respectively called wing reference area, wing span and mean geometric cord, and are defined by the wing-fuselage properties.

There are several ways of determining the functions $f_{X^A}(\boldsymbol{x},t)$, $f_{Y^A}(\boldsymbol{x},t)$ and $f_{Z^A}(\boldsymbol{x},t)$; and $f_{L^A}(\boldsymbol{x},t)$, $f_{M^A}(\boldsymbol{x},t)$ and $f_{N^A}(\boldsymbol{x},t)$. The most invoked method is a linear time-invariant approximation technique. The present work will use a nonlinear and time-varying system identification approach to obtain these functions without resorting to any linear assumption, to provide a larger operation range.

The linear method is described below for completeness and as a reference.

1.2.3.1 Linear Aerodynamic Forces $[\boldsymbol{F}_B]^A$ *and Moments* $[\boldsymbol{M}_B]^A$

The general approach, called the component build-up method (see Ref. [4] for a detailed explanation), describes the aerodynamic forces and moments as a direct aggregation of several independent effects, each one having a physical support and being linearly described as a function of the aircraft perturbed states $\Delta\boldsymbol{x}(t)=\boldsymbol{x}(t)-\boldsymbol{x}_{trim}$ as well as the steady state, trim condition $\boldsymbol{x}_{trim}$. This approach is valid in the neighborhood of the trim point $\boldsymbol{x}_{trim},\boldsymbol{u}_{trim}$, where the assumption of linearity and independence among all physical effects is feasible.

For the case of linear and time invariant aerodynamic description, force coefficients C_X,C_Y,C_Z and moment coefficients C_l,C_m,C_n defined in the body coordinate system $\bar{\boldsymbol{B}}$ are obtained from the aerodynamic force coefficients $C_D^{\bar{\boldsymbol{S}}},C_Y^{\bar{\boldsymbol{S}}},C_L^{\bar{\boldsymbol{S}}}$ and aerodynamic moment coefficients $C_l^{\bar{\boldsymbol{S}}},C_m^{\bar{\boldsymbol{S}}},C_n^{\bar{\boldsymbol{S}}}$ defined in the stability axes $\bar{\boldsymbol{S}}$. The following conversion as a function of the trimmed angle of attack α_{trim} (Ref. [4]) is used

$$\begin{bmatrix} C_X(\boldsymbol{x}) \\ C_Y(\boldsymbol{x}) \\ C_Z(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} -C_D^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\cos\alpha_{trim}+C_L^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\sin\alpha_{trim} \\ C_Y^{\bar{\boldsymbol{S}}}(\boldsymbol{x}) \\ -C_D^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\sin\alpha_{trim}+C_L^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\cos\alpha_{trim} \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} C_l(\boldsymbol{x}) \\ C_m(\boldsymbol{x}) \\ C_n(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} +C_l^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\cos\alpha_{trim}-C_n^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\sin\alpha_{trim} \\ C_m^{\bar{\boldsymbol{S}}}(\boldsymbol{x}) \\ +C_l^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\sin\alpha_{trim}+C_n^{\bar{\boldsymbol{S}}}(\boldsymbol{x})\cos\alpha_{trim} \end{bmatrix} \tag{21}$$

Drag force coefficient $C_D^{\bar{S}}$, side force coefficient $C_Y^{\bar{S}}$, and lift force coefficient $C_L^{\bar{S}}$, along with rolling moment coefficient $C_l^{\bar{S}}$, pitching moment coefficient $C_m^{\bar{S}}$, and yawing moment coefficient $C_n^{\bar{S}}$ are assembled from the component build-up philosophy, as shown below

$$\begin{aligned} C_D^{\bar{S}}(\boldsymbol{x}) = C_{D0} + C_{D\alpha}(\overbrace{\alpha_{trim} + \Delta\alpha}^{\alpha}) + C_{D\delta_E}(\overbrace{\delta_{Etrim} + \Delta\delta_E}^{\delta_E}) + \\ C_{D\dot{\alpha}}\frac{\Delta\dot{\alpha}\bar{c}}{2U_{trim}} + C_{Dq}\frac{\Delta Q\bar{c}}{2U_{trim}} + C_{Du}\frac{\Delta U}{U_{trim}} + 2C_{L\,trim}\frac{\Delta U}{U_{trim}} - C_{L\,trim}\Delta\alpha \end{aligned} \tag{22}$$

$$\begin{aligned} C_Y^{\bar{S}}(\boldsymbol{x}) = C_{y\beta}\ (\overbrace{\beta_{trim} + \Delta\beta}^{\beta}) + C_{y\delta_A}\ (\overbrace{\delta_{Atrim} + \Delta\delta_A}^{\delta_A}) + C_{y\delta_R}\ (\overbrace{\delta_{Rtrim} + \Delta\delta_R}^{\delta_R}) + \\ C_{y\dot{\beta}}\frac{\Delta\dot{\beta}b}{2U_{trim}} + C_{yp}\frac{\Delta Pb}{2U_{trim}} + C_{yr}\frac{\Delta Rb}{2U_{trim}} \end{aligned} \tag{23}$$

$$\begin{aligned} C_L^{\bar{S}}(\boldsymbol{x}) = C_{L0} + C_{L\alpha}(\overbrace{\alpha_{trim} + \Delta\alpha}^{\alpha}) + C_{L\delta_E}(\overbrace{\delta_{Etrim} + \Delta\delta_E}^{\delta_E}) + \\ C_{L\dot{\alpha}}\frac{\Delta\dot{\alpha}\bar{c}}{2U_{trim}} + C_{Lq}\frac{\Delta Q\bar{c}}{2U_{trim}} + C_{Lu}\frac{\Delta U}{U_{trim}} + 2C_{L\,trim}\frac{\Delta U}{U_{trim}} + C_{D\,trim}\Delta\alpha \end{aligned} \tag{24}$$

$$\begin{aligned} C_l^{\bar{S}}(\boldsymbol{x}) = C_{l\beta}\ (\overbrace{\beta_{trim} + \Delta\beta}^{\beta}) + C_{l\delta_A}\ (\overbrace{\delta_{Atrim} + \Delta\delta_A}^{\delta_A}) + C_{l\delta_R}\ (\overbrace{\delta_{Rtrim} + \Delta\delta_R}^{\delta_R}) + \\ C_{l\dot{\beta}}\frac{\Delta\dot{\beta}b}{2U_{trim}} + C_{lp}\frac{\Delta Pb}{2U_{trim}} + C_{lr}\frac{\Delta Rb}{2U_{trim}} \end{aligned} \tag{25}$$

$$\begin{aligned} C_m^{\bar{S}}(\boldsymbol{x}) = C_{m0} + C_{m\alpha}(\overbrace{\alpha_{trim} + \Delta\alpha}^{\alpha}) + C_{m\delta_E}(\overbrace{\delta_{Etrim} + \Delta\delta_E}^{\delta_E}) + \\ C_{m\dot{\alpha}}\frac{\Delta\dot{\alpha}\bar{c}}{2U_{trim}} + C_{mq}\frac{\Delta Q\bar{c}}{2U_{trim}} + C_{mu}\frac{\Delta U}{U_{trim}} + 2C_{m\,trim}\frac{\Delta U}{U_{trim}} \end{aligned} \tag{26}$$

$$\begin{aligned} C_n^{\bar{S}}(\boldsymbol{x}) = C_{n\beta}\ (\overbrace{\beta_{trim} + \Delta\beta}^{\beta}) + C_{n\delta_A}\ (\overbrace{\delta_{Atrim} + \Delta\delta_A}^{\delta_A}) + C_{n\delta_R}\ (\overbrace{\delta_{Rtrim} + \Delta\delta_R}^{\delta_R}) + \\ C_{n\dot{\beta}}\frac{\Delta\dot{\beta}b}{2U_{trim}} + C_{np}\frac{\Delta Pb}{2U_{trim}} + C_{nr}\frac{\Delta Rb}{2U_{trim}} \end{aligned} \tag{27}$$

where Δ amounts for the deviation or perturbation from the trim condition $\boldsymbol{x}_{trim},\boldsymbol{u}_{trim}$. Also the quantities $C_{Dtrim}, C_{Ltrim}, C_{mtrim}$ are required. All remaining coefficients in Equations (22) to (27), which are called stability and control derivatives, are generated for the chosen trim point. A widely used method for computing these parameters is the United States Air Force Stability and Control Digital DATCOM (Ref. [87]), which in turn is based on, and implements the methods contained in, the United States Air Force

Stability and Control DATCOM. This is a collection and recording of the most comprehensive knowledge in the area of aerodynamic stability and control prediction methods, and it calculates these derivatives for fixed-wing aircraft, according to the specified flight conditions. The outputs are the corresponding dimensionless stability and control derivatives. Another accessible tool is the engineering level software called Advanced Aircraft Analysis, (AAA), (based on Ref. [4]). It delivers these parameters by interpolating among information of existing aircraft.

1.3 Aerodynamic Coefficients Estimation

A common approach in obtaining the underlying relation of the aerodynamic forces and moments, denoted $f_{X^A}(\boldsymbol{x},t)$, $f_{Y^A}(\boldsymbol{x},t)$, and $f_{Z^A}(\boldsymbol{x},t)$, and $f_{L^A}(\boldsymbol{x},t)$, $f_{M^A}(\boldsymbol{x},t)$, and $f_{N^A}(\boldsymbol{x},t)$ from Equations (18) and (19), respectively, is their identification based on measured data. This solution approach is applicable if these functions act as a cause-effect relationship to be approximated between inputs and outputs, and if signals are available (see Ref. [5] to [9]). Several techniques are applied, e.g. linear or nonlinear curve fitting. Fuzzy logic and artificial neural networks (ANN) seem to be appropriate tools, as they inherently materialize the universal approximation theorem (see Ref. [10] and [11], respectively). This capability makes ANN a highly useful tool for mapping signals, i.e. extracting the underlying cause-effect relationship between inputs and outputs.

As stated by the following theorem, a feedforward ANN with a single hidden layer, containing a finite number of neuron cells with sigmoid-type activation functions and linear activation functions at the output layer, inherently materializes the universal approximation theorem. It approximates any multivariable continuous function on a compact domain to any degree of accuracy.

Theorem 1: Universal Approximation Theorem For any given continuous function $f_j(\boldsymbol{z}) \in [0,1] \in \boldsymbol{R}$, with $\boldsymbol{z} = [z_1, z_2, \cdots, z_q]^T \in [0,1]^q \in \boldsymbol{R}^q$, and for $\varepsilon > 0$, there exists an integer M and a set of parameters (weights and biases) $b_i, c_j \in \boldsymbol{R}$, $\boldsymbol{w}_i \in \boldsymbol{R}^q$, and $a_{ji} \in \boldsymbol{R}$ with $i = 1 \cdots M$ and $j = 1 \cdots p$ such that

$$F_j(\boldsymbol{z}) = \sum_{i=1}^{M} a_{ji}\ \varphi(\boldsymbol{w}_i^T \boldsymbol{z} + b_i) + c_j \qquad (28)$$

is an approximation of the function $f_j(\boldsymbol{z})$, with $|F_j(\boldsymbol{z}) - f_j(\boldsymbol{z})| < \varepsilon$, and where $\varphi(\cdot)$ is a bounded and monotonically increasing activation function.

Demonstration Omitted, see Ref. [88].

This capability makes ANN a highly useful tool for providing mapping of signals by curve fitting. Figure 2 shows the ANN configuration that materializes Theorem 1.

With an adequate mechanism for adjusting its parameters, i.e. weights and biases, an ANN is capable of iteratively approximating any static relation to any degree of accuracy. The adjustment logic must be based on the level of precision demanded for the estimation. A standard method, the one chosen in this research, is to work with the cumulative quadratic error between the measured output of the system under identification and the estimated output of the ANN over a period of time. Reducing this error implies the ANN is able to better replicate the cause-effect relation of the system under identification.

Regularly, an ANN is trained off-line, adjusting its parameters with previously stored data in cycles called epochs. After a validation process with independent data, i.e. uncorrelated with the one used during training, the ANN is assumed operational. Training is performed iteratively through a number of epochs, where each epoch uses the whole set of training data. These data, that includes all pertinent system's inputs and outputs, are expected to capture the dynamics of the aircraft, i.e. signals should be representative of the dynamics to be identified.

As long as the dynamics of the identified system remain unchanged, there is no need to retrain the ANN and its parameters are kept fixed during operation. Two important conditions must be met for acceptable performance of an ANN as a universal approximator: a static and deterministic nature of the system under identification (the mathematical relation between inputs and outputs of the systems remains the same), and the availability of all signals involved in the cause-effect relation (i.e. inputs and outputs). Failure to meet one or both of these conditions degrades its performance.

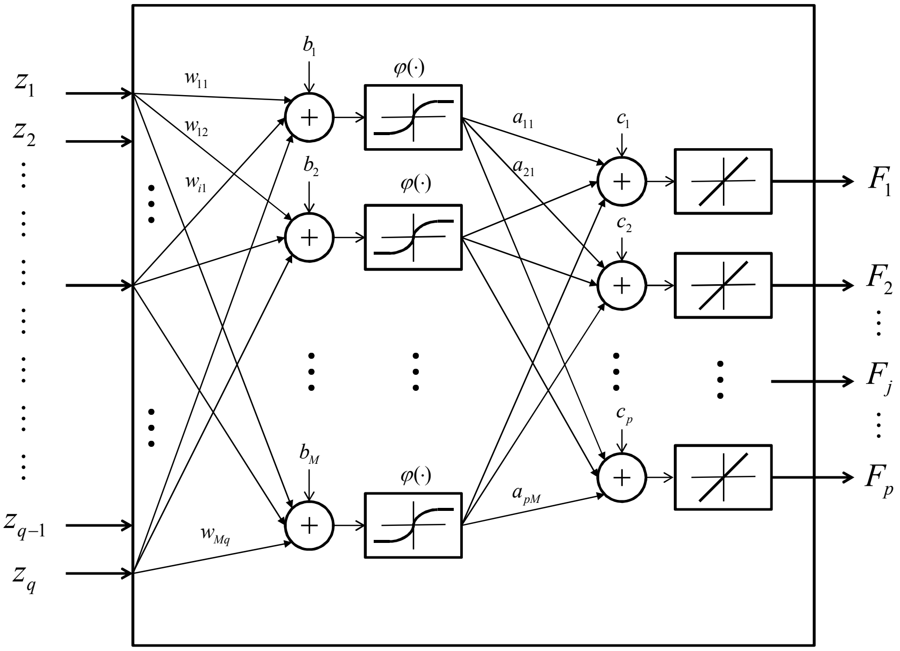

Figure 2. **ANN Interconnection Structure.**

Given the aircraft's inherently nonlinear and unsteady aerodynamics, worsened during off-nominal conditions (see Ref. [13]), ANN parameters must be readjusted in real-time, using fresh data to capture the changing dynamics of the system. The unsteadiness in the aerodynamics suggests that important dynamic behavior is unmodeled and random, and a static relation is no longer valid. This defect could be improved to some extent by expanding the system modeling. For example, it is well known that a correction needs to be made due to a convective time delay in changes in the wing-induced downwash at the tail (see Ref. [89]). This time delay could be modeled as a simple time lag, which makes the correction equivalent to an acceleration-type derivative.

An interesting solution using ANN is presented in Ref. [17] and [18] where unstable aircraft stability and control derivatives are estimated under a closed loop configuration. They use a technique called Delta method to extract these derivatives and the controller gains. The controller is used to stabilize the originally unstable aircraft and allows system identification, which is otherwise unfeasible. Ref. [20] proposes an approach to model the aerodynamic force and moment coefficients and their derivative parameters for lateral-directional motion from recorded flight data based on neural partial differentiation. It is presented as an alternative to the Delta method, showing comparable results.

In this work, an online adaptive learning algorithm is proposed to update parameters of the ANN in a continuous way. A time sliding window, where new sampled data is attached after shifting one step and the oldest portion of data is discarded, keeps the ANN in training mode permanently (sliding mode learning). The sliding mode learning concept allows error minimization between the ANN and nonlinear system by adjusting the ANN's parameters, reproducing the current dynamic behavior of the aircraft. The adjustment and convergence must be reached faster than the occurrence time of significant changes in the aircraft dynamics; i.e. ANN's time constant should be faster than the aircraft associated time constant. If this requirement is not fulfilled the outcome can be deleterious resulting in substantial performance degradation, if not failure to converge.

After failures or significant changes in the dynamics or aerodynamics of the aircraft, previously well trained ANN could become outdated, forcing new training. Weights and biases should be updated to reflect the new dynamic status. Rather than performing this process in a disconnected way, the idea is to keep the ANN under continuous training, readjusting its parameters in a permanent way, and acquiring the new dynamics embedded into the new data, uninterruptedly. By using a recursive training process, the ANN parameters are under constant update.

In the present work ANNs are fed by an aircraft subset of normalized states and servo deflections, collected as $\boldsymbol{z} = [z_1, z_2, \cdots, z_q]^T \in [-1,1]^q$. Input weights $\boldsymbol{w}_i$, output weights a_{ji}, input biases b_i, and output biases c_j are constantly adjusted in real-time in order to iteratively decrease the error between predicted and measured quantities, in a quadratic sense. At each sample time the data in the moving window is used to perform the numerical search as in the epoch concept. In classical ANN applications, the learning process is finalized after a predefined level of accuracy is achieved and the ANN is declared trained. In this research, the learning process is unstopped and weights and biases are always under adaptation, derived from the current window data. Parameters under variation are then continuously fed into the NMPC, which keeps the replicated structures of the aerodynamic ANNs, allowing them to predict body accelerations, in real-time, based on these updated aerodynamic models.

1.3.1 Adaptive ANN Aerodynamic Estimation

The adaptive online training is done by minimizing the accumulated quadratic error between a measured quantity $s[k]$ and its estimation $\hat{s}[k]$ over a period of N_W past samples, i.e. minimizing the cost function $W(\boldsymbol{\theta}) = \boldsymbol{\varepsilon}^T(\boldsymbol{\theta}) \cdot \boldsymbol{\varepsilon}(\boldsymbol{\theta}) \in \boldsymbol{R}$, with

$$\boldsymbol{\varepsilon}(\boldsymbol{\theta}) = \begin{bmatrix} \varepsilon[k - N_W + 1, \boldsymbol{\theta}] \\ \vdots \\ \varepsilon[k, \boldsymbol{\theta}] \end{bmatrix} = \begin{bmatrix} s[k - N_W + 1] - \hat{s}[k - N_W + 1, \boldsymbol{\theta}] \\ \vdots \\ s[k] - \hat{s}[k, \boldsymbol{\theta}] \end{bmatrix} \tag{29}$$

where $\boldsymbol{\theta} \in R^r$, the parameter vector including all weights and biases, is defined by

$$\boldsymbol{\theta}^T = [\theta_1, \cdots, \theta_r] = [a_{11}, \cdots a_{p1}, \cdots, a_{1M}, \cdots, a_{pM}, b_1, \cdots, b_M, w_{11}, \cdots a_{1q}, \cdots, w_{M1}, \cdots, w_{Mq}, c_1, \cdots, c_p] \tag{30}$$

with $r = p \times q + M \times q + M + p$. Using the following Newton-type search method, an iterative estimation is achieved

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] - [\nabla^2 W(\boldsymbol{\theta}[k])]^{-1} \cdot \nabla^T W(\boldsymbol{\theta}[k]) \tag{31}$$

where $\nabla W(\boldsymbol{\theta}[k])$ and $\nabla^2 W(\boldsymbol{\theta}[k])$ are the gradient and the Hessian of the cost function W with respect to the parameter vector, respectively. They are defined as follows

$$\nabla W(\boldsymbol{\theta}) = \begin{bmatrix} \dfrac{\partial W}{\partial \theta_1} & \cdots & \dfrac{\partial W}{\partial \theta_r} \end{bmatrix} \tag{32}$$

$$\nabla^2 W(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 W}{\partial \theta_1^2} & \cdots & \frac{\partial^2 W}{\partial \theta_1 \partial \theta_r} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 W}{\partial \theta_r \partial \theta_1} & \cdots & \frac{\partial^2 W}{\partial \theta_r^2} \end{bmatrix} \tag{33}$$

At each iteration, a numerical approximation of these two arrays is possible to obtain by expressing the error vector $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$ by its first order Taylor expansion

$$\boldsymbol{\varepsilon}(\boldsymbol{\theta} + \boldsymbol{\Delta\theta}) \approx \boldsymbol{\varepsilon}(\boldsymbol{\theta}) + \boldsymbol{J}(\boldsymbol{\theta}) \cdot \boldsymbol{\Delta\theta} \tag{34}$$

where the rows of $\boldsymbol{J}(\boldsymbol{\theta})$, the Jacobian of $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$, include the gradients of each element of $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$ with respect to each parameter in the vector $\boldsymbol{\theta}$, as shown below

$$\boldsymbol{J}(\boldsymbol{\theta}) = \begin{bmatrix} \nabla\varepsilon[k - N_W + 1, \boldsymbol{\theta}] \\ \vdots \\ \nabla\varepsilon[k, \boldsymbol{\theta}] \end{bmatrix} = \begin{bmatrix} \frac{\partial\varepsilon[k - N_W + 1, \boldsymbol{\theta}]}{\partial\theta_1} & \cdots & \frac{\partial\varepsilon[k - N_W + 1, \boldsymbol{\theta}]}{\partial\theta_r} \\ \vdots & \ddots & \vdots \\ \frac{\partial\varepsilon[k, \boldsymbol{\theta}]}{\partial\theta_1} & \cdots & \frac{\partial\varepsilon[k, \boldsymbol{\theta}]}{\partial\theta_r} \end{bmatrix} \tag{35}$$

Now, by using Equation (34), the cost function $W(\boldsymbol{\theta})$, evaluated at $\boldsymbol{\theta} + \boldsymbol{\Delta\theta}$, is then expressible as

$$W(\boldsymbol{\theta} + \boldsymbol{\Delta\theta}) \approx \overbrace{\boldsymbol{\varepsilon}^T(\boldsymbol{\theta}) \cdot \boldsymbol{\varepsilon}(\boldsymbol{\theta})}^{W(\boldsymbol{\theta})} + \overbrace{2\boldsymbol{\varepsilon}^T(\boldsymbol{\theta}) \cdot J(\boldsymbol{\theta})}^{\nabla W(\boldsymbol{\theta})} \cdot \boldsymbol{\Delta\theta} + \frac{1}{2}\boldsymbol{\Delta\theta}^T \cdot \underbrace{2J^T(\boldsymbol{\theta}) \cdot J(\boldsymbol{\theta})}_{\nabla^2 W(\boldsymbol{\theta})} \cdot \boldsymbol{\Delta\theta} \tag{36}$$

from where the cost function's gradient $\nabla W(\boldsymbol{\theta}[k])$ and Hessian $\nabla^2 W(\boldsymbol{\theta}[k])$ can be directly extracted as a function of the error vector $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$ and its Jacobian $\boldsymbol{J}(\boldsymbol{\theta})$, which are more feasible quantities to calculate. Then Equation (31) is rewritten in its final form as

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] - [2\boldsymbol{J}^T(\boldsymbol{\theta}[k]) \cdot \boldsymbol{J}(\boldsymbol{\theta}[k])]^{-1} \cdot [2\boldsymbol{\varepsilon}^T(\boldsymbol{\theta}[k]) \cdot \boldsymbol{J}(\boldsymbol{\theta}[k])]^T \tag{37}$$

To provide a degree of flexibility to the parameter search task, Levenberg's approach is chosen (see Ref. [90] and [91]), where the Levenberg adaptive parameter $\lambda[k]$ is employed. It allows the transition from a steepest descent type of search to a quadratic one when approaching to the minimum, adapting the speed of convergence and having a direct impact in the algorithm stability. Based on Levenberg's method, Equation (37) is modified as

$$\boldsymbol{\theta}[k+1]=\boldsymbol{\theta}[k]-[2\boldsymbol{J}^T(\boldsymbol{\theta}[k])\cdot\boldsymbol{J}(\boldsymbol{\theta}[k])+\lambda[k]\boldsymbol{I}]^{-1}\cdot[2\boldsymbol{\varepsilon}^T(\boldsymbol{\theta}[k])\cdot\boldsymbol{J}(\boldsymbol{\theta}[k])]^T \tag{38}$$

with $\boldsymbol{I}$ the identity matrix of appropriate dimensions.

Levenberg's logic states that this adaptive parameter is to be adjusted during each iteration based on the evolution of the cost function. As in principle, Levenberg's approach is intended for off-line training, it is slightly modified in this research to provide a mechanism for resetting the Levenberg parameter λ when reaching predefined extreme values, allowing an extended operation. This is intended to avoid unbounded growing of the parameter λ when the cost function gets close to a minimum and consecutive iterations get closer, being trapped by it. This is a modification required to keep the ANN under permanent training. After starting with $\lambda[k]=\lambda_0$, for a minimum-maximum range $0<\lambda_{min}<\lambda[k]<\lambda_{max}$ with decrement and increment multiplicative factors $\lambda_{dec}>0$ and $\lambda_{inc}>0$, the logic is defined as follows

$$\begin{array}{ll} IF \quad W([k+1])<W([k])\rightarrow\lambda[k+1]=\lambda[k]/\lambda_{dec}, & IF\ \lambda[k+1]<\lambda_{\min}\rightarrow\lambda[k+1]=\lambda_0 \\ IF \quad W([k+1])\geq W([k])\rightarrow\lambda[k+1]=\lambda[k]*\lambda_{inc}, & IF\ \lambda[k+1]>\lambda_{\max}\rightarrow\lambda[k+1]=\lambda_0 \end{array} \tag{39}$$

The choice of these four values has a direct impact on the search efficiency and search stability, whose definition is highly problem-dependent.

In this research, aerodynamic accelerations are estimated, i.e. $F_1(\boldsymbol{z})=\hat{a}_X$, $F_2(\boldsymbol{z})=\hat{a}_Y$, and $F_3(\boldsymbol{z})=\hat{a}_Z$, under the consideration that, due to unsteadiness, these quantities are more susceptible to sudden changes than any other aircraft dynamics. A bank of three ANNs is assembled for this purpose, each one for separate acceleration, comprising each ANN a single hidden layer with $M=10$ cells, and one output cell within the output layer, i.e. $p=1$. Input $\boldsymbol{z}$ and target f signals are normalized within the range $[-1,1]$ to keep consistency with the active range of the input activation function $\varphi(\cdot)$, a hyperbolic tangent sigmoid transfer function, a common practice as is the case in Ref. [20].

The ANN's inputs are selected from available state measurements, and servo deflection commands. No measurement of servo deflection is available. This selection is mainly based on theoretical and practical aerodynamic experience, supported by extensive trial and error experimentation. The ANN's inputs for forward, lateral, and vertical accelerations are defined as $\boldsymbol{z}_X^T=[V_T,\alpha,\beta,Q,\dot{\alpha},\delta_T,\delta_E]$,

$\boldsymbol{z}_Y^T = [V_T, \beta, P, R, \dot{\beta}, \delta_A, \delta_R]$, and $\boldsymbol{z}_Z^T = [V_T, \alpha, \beta, Q, \dot{\alpha}, \delta_T, \delta_E]$, respectively, all with $q = 7$. The ANN's target signals are measured acceleration components in the body coordinate system $\boldsymbol{B}$, as defined in Equation (18), i.e. $f_1 = a_X = X^A / m = f_{X^A}(\boldsymbol{x}, t) / m$, $f_2 = a_Y = Y^A / m = f_{Y^A}(\boldsymbol{x}, t) / m$ and $f_1 = a_Z = Z^A / m = f_{Z^A}(\boldsymbol{x}, t) / m$.

Each ANN performs at least one learning cycle (equivalent to an epoch) at each sample interval, working with the moving data window (in sliding mode learning). The window length N_W, which is mainly shaped by computational power availability and quality of measurements, is set the same for all three ANNs. Currently, all data within the window share the same weight, although a forgetting factor can be applied to give less significance to older data. The following equations show the structures of the moving windows for input signal $\boldsymbol{z}$ and target signal f for each ANN. The forward acceleration ANN is fed by airspeed, angle of attack, sideslip angle, pitch rate, angle of attack time rate of change, engine throttle deflection, and elevator deflection. The ANN for lateral acceleration uses airspeed, sideslip angle, roll rate, yaw rate, sideslip angle time rate of change, aileron deflection, and rudder deflection. The ANN for vertical acceleration shares the same inputs as the forward acceleration ANN.

$$\begin{aligned}
&\boldsymbol{z}_X : \begin{bmatrix}
V_T[k-N_W+1] & V_T[k-N_W+2] & \cdots & V_T[k-1] & V_T[k] \\
\alpha[k-N_W+1] & \alpha[k-N_W+2] & \cdots & \alpha[k-1] & \alpha[k] \\
\beta[k-N_W+1] & \beta[k-N_W+2] & \cdots & \beta[k-1] & \beta[k] \\
Q[k-N_W+1] & Q[k-N_W+2] & \cdots & Q[k-1] & Q[k] \\
\dot{\alpha}[k-N_W+1] & \dot{\alpha}[k-N_W+2] & \cdots & \dot{\alpha}[k-1] & \dot{\alpha}[k] \\
\delta_T[k-N_W+1] & \delta_T[k-N_W+2] & \cdots & \delta_T[k-1] & \delta_T[k] \\
\delta_E[k-N_W+1] & \delta_E[k-N_W+2] & \cdots & \delta_E[k-1] & \delta_E[k]
\end{bmatrix} \in \boldsymbol{R}^{q \times N_W} \\
&f_1 : \left[a_X[k-N_W+1] \quad a_X[k-N_W+2] \quad \cdots \quad a_X[k-1] \quad a_X[k] \right]^T \in \boldsymbol{R}^{N_W \times 1}
\end{aligned} \tag{40}$$

$$\begin{gathered}
z_Y : \begin{bmatrix} V_T[k-N_W+1] & V_T[k-N_W+2] & \cdots & V_T[k-1] & V_T[k] \\ \beta[k-N_W+1] & \beta[k-N_W+2] & \cdots & \beta[k-1] & \beta[k] \\ P[k-N_W+1] & P[k-N_W+2] & \cdots & P[k-1] & P[k] \\ R[k-N_W+1] & R[k-N_W+2] & \cdots & R[k-1] & R[k] \\ \dot{\beta}[k-N_W+1] & \dot{\beta}[k-N_W+2] & \cdots & \dot{\beta}[k-1] & \dot{\beta}[k] \\ \delta_A[k-N_W+1] & \delta_A[k-N_W+2] & \cdots & \delta_A[k-1] & \delta_A[k] \\ \delta_R[k-N_W+1] & \delta_R[k-N_W+2] & \cdots & \delta_R[k-1] & \delta_R[k] \end{bmatrix} \in \boldsymbol{R}^{q\times N_W} \\
f_2 : \left[a_Y[k-N_W+1] \quad a_Y[k-N_W+2] \quad \cdots \quad a_Y[k-1] \quad a_Y[k] \right]^T \in \boldsymbol{R}^{N_W\times 1}
\end{gathered} \tag{41}$$

$$\begin{gathered}
z_Z : \begin{bmatrix} V_T[k-N_W+1] & V_T[k-N_W+2] & \cdots & V_T[k-1] & V_T[k] \\ \alpha[k-N_W+1] & \alpha[k-N_W+2] & \cdots & \alpha[k-1] & \alpha[k] \\ \beta[k-N_W+1] & \beta[k-N_W+2] & \cdots & \beta[k-1] & \beta[k] \\ Q[k-N_W+1] & Q[k-N_W+2] & \cdots & Q[k-1] & Q[k] \\ \dot{\alpha}[k-N_W+1] & \dot{\alpha}[k-N_W+2] & \cdots & \dot{\alpha}[k-1] & \dot{\alpha}[k] \\ \delta_T[k-N_W+1] & \delta_T[k-N_W+2] & \cdots & \delta_T[k-1] & \delta_T[k] \\ \delta_E[k-N_W+1] & \delta_E[k-N_W+2] & \cdots & \delta_E[k-1] & \delta_E[k] \end{bmatrix} \in \boldsymbol{R}^{q\times N_W} \\
f_3 : \left[a_Z[k-N_W+1] \quad a_Z[k-N_W+2] \quad \cdots \quad a_Z[k-1] \quad a_Z[k] \right]^T \in \boldsymbol{R}^{N_W\times 1}
\end{gathered} \tag{42}$$

1.3.1.1 ANN Performance Test using Simulated Data

To test the ANN performance in simulation, vertical acceleration $Z^A(t)/m$ is chosen. The vertical ANN is constructed with one hidden layer comprising $M=5$ cells and one output layer with $p=1$ cell, where $F_1(z)=\hat{a}_Z$. The inputs to the ANN are airspeed, angle of attack, sideslip angle, pitch rate, angle of attack time rate of change, and elevator deflection angles i.e. $z^T=[V_T,\alpha,\beta,Q,\dot{\alpha},\delta_E]$, with $q=6$. The target is the vertical acceleration in the body coordinate system $\overline{\boldsymbol{B}}$, $f_1=a_Z=f_{Z^A}(\boldsymbol{x},t)/m=\overline{q}(t)SC_Z(t)/m$, linearly simulated by using Equations (18), (20), (22) and (24).

To test the ANN adaptability, a dynamic change is induced in the lift coefficient $C_L^{\overline{S}}$, defined in Equation (24) for a Yak 54 1/3 scale aircraft model. After 52 seconds of simulated flight, a reduction of 25% in the $C_{L\alpha}$ derivative is injected during a trajectory following.

As shown previously, this alteration in lift coefficient has a direct impact in the Z-axis body acceleration a_Z .

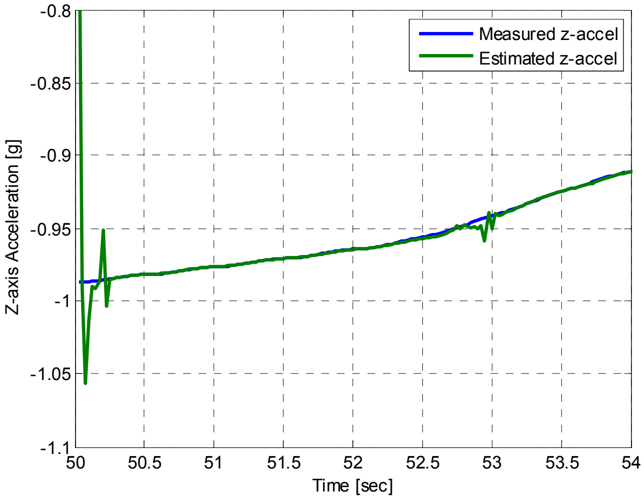

Figure 3. Z-axis Measured and Predicted Acceleration.

Figure 3 shows the effect of the dynamic change on the estimated acceleration. After the ANN adapts, it converges back to the measured acceleration. The graph also shows how the estimated acceleration converges from an arbitrary initial condition to the measured acceleration after 50 seconds of flight, the moment when the ANN was activated.

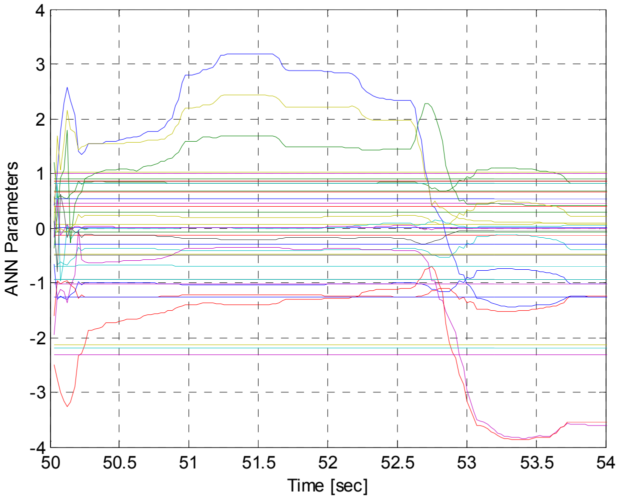

Figure 4. Z-axis Acceleration ANN Parameters.

Figure 4 shows the evolution of the parameters of the ANN after the failure, and how they adjust to decrease the estimation error. It also shows the initial adaptation after energizing the ANN.

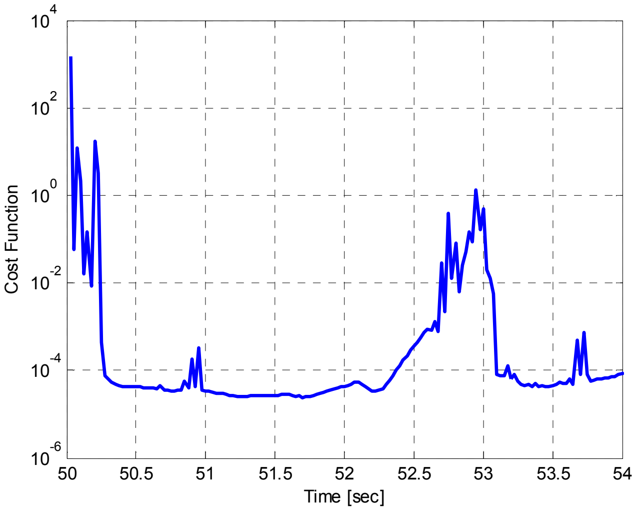

Figure 5. Z-axis Acceleration ANN Cost Function.

Figure 5 shows the evolution of the cost function, where it follows the same pattern as before, that is, increasing its value when the estimation error is larger and reducing it when the ANN starts to converge, returning to a minimum value after the adaptation of the ANN's parameters.

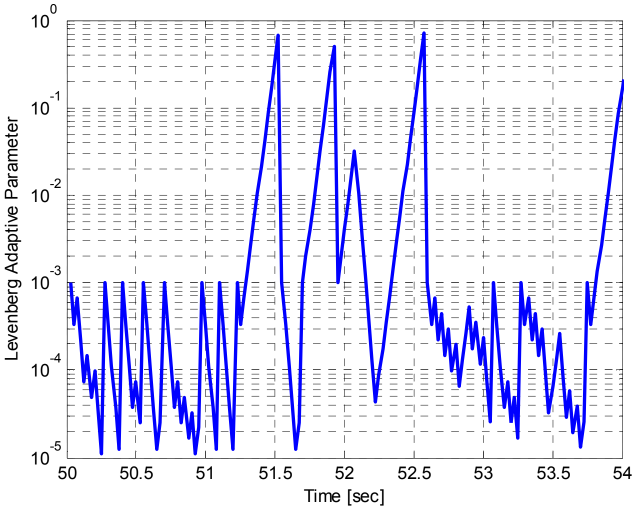

Figure 6. Z-axis Acceleration ANN Levenberg's Parameter.

As shown for this particular scenario, this minimum moves below 10^{-4}. It is important to keep in mind that this quantity reflects the summation of quadratic estimation error over the entire window. Based on extensive simulations, it has been verified that this minimum is affected mainly by the length of the moving window and the existence of noise measurements (not added in the present case). It is expected to be further increased when using measured flight data.

Finally, Figure 6 shows the evolution of the Levenberg parameter, and how it evolves between $\lambda_{\min} = 10^{-5}$ to $\lambda_{\max} = 10^{0}$, clearly following the logic defined for its confinement.

1.3.1.2 ANN Performance Test using Flight Test Data

To verify the performance of the ANN's adaptive capability with real data, two sets of stored Meridian flight test data, representing different dynamic conditions, are used; a flight from Pegasus Airfield in Antarctica during 2009 and a flight from Neem in Greenland during 2011. Both sets of data were collected with a sampling frequency of 10 Hertz. For comparison and validation purposes, the same ANN is trained and tested without adaptiveness. This off-line ANN is constructed with $M = 20$ cells in the hidden layer and fed by the same inputs as defined for the adaptive ANN. As there is no pure aerodynamic acceleration measurement, data from accelerometers is used instead. All three accelerations are estimated.

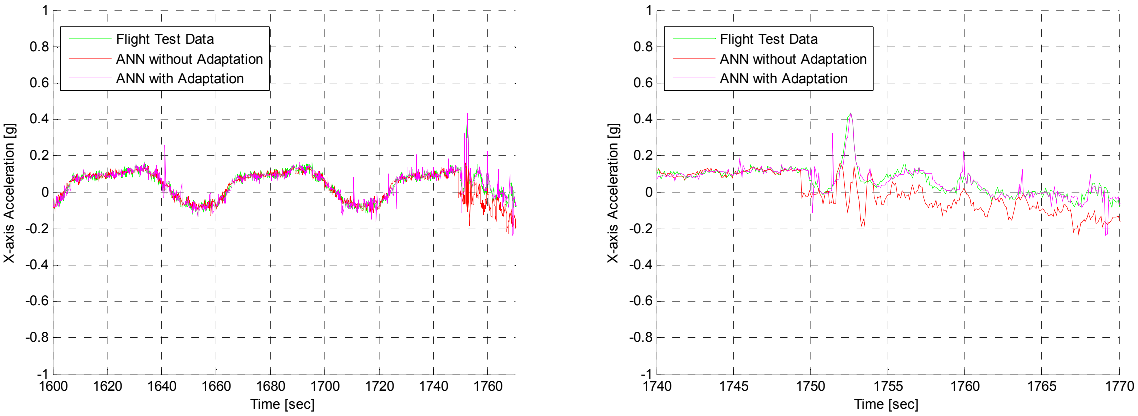

Figure 7. X-axis Measured and Predicted Acceleration in Antarctica: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).

In general, both flights share the same pattern. From a prolonged cyclic autonomous period of flight, the aircraft is switched to remote control for landing.

During autonomous flight, the non-adaptive ANN is trained. It is then tested in both the remaining autonomous flight segment, and during the remote controlled period. The adaptive ANN is tested in the same interval as the non-adaptive ANN.

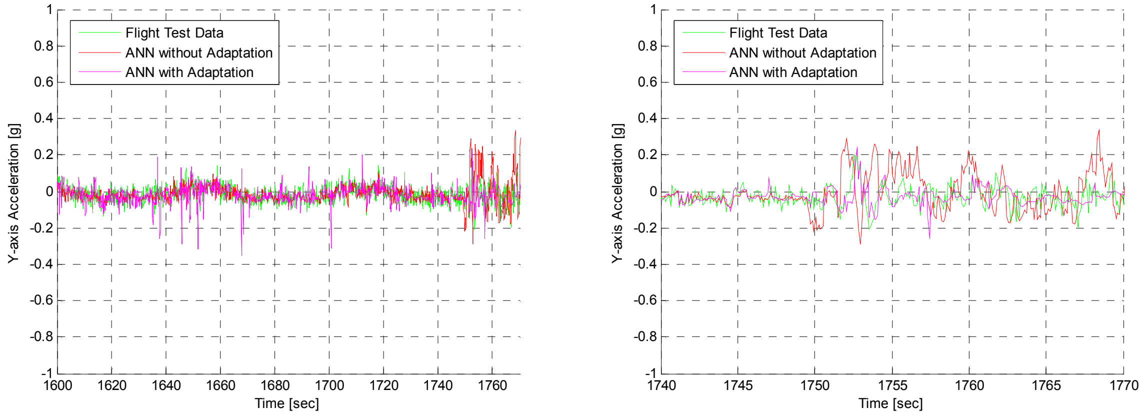

Figure 8. Y-axis Measured and Predicted Acceleration in Antarctica: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).

The non-adaptive ANN version is trained for a period of 1,000 samples (100 seconds of flight time). The adaptive ANN has a moving window of $N_W = 10$ samples and Levenberg parameters of $\lambda_0 = 0.001$, $\lambda_{min} = 0.00001$, $\lambda_{max} = 1$, $\lambda_{inc} > 2$ and $\lambda_{dec} > 3$; all parameters are determined heuristically.

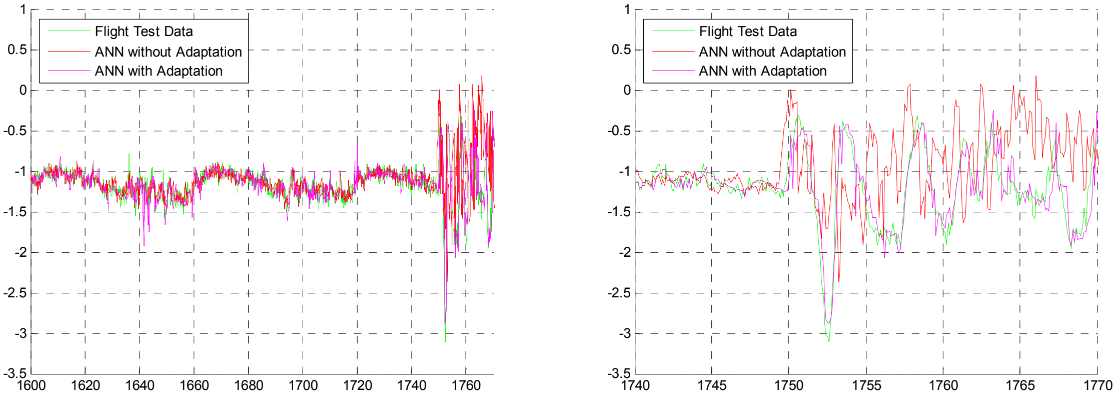

Figure 9. Z-axis Measured and Predicted Acceleration in Antarctica: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).

Figure 7, Figure 8, and Figure 9 show the Antarctica acceleration estimations, where at approximately $1,750$ seconds, the aircraft is switched from autonomous to remote control mode. Flight data is plotted in green, the non-adaptive ANN in red, and the ANN with adaptation in magenta.

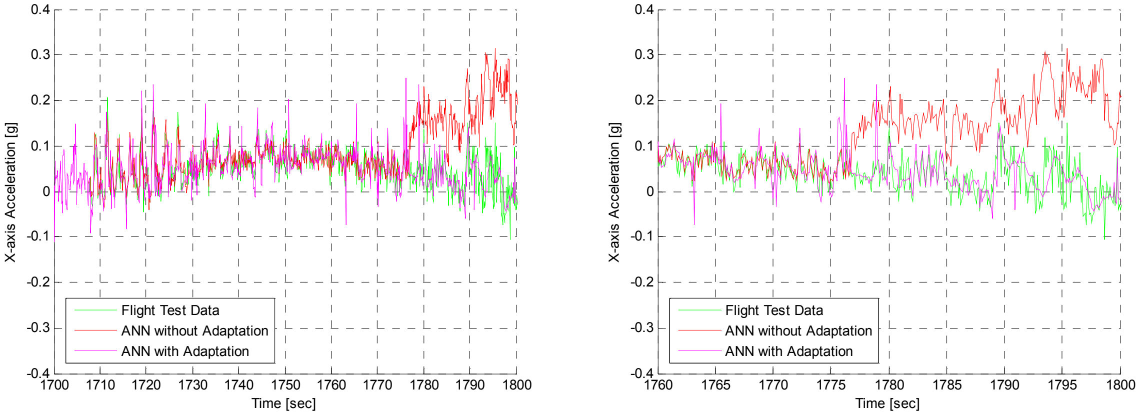

Figure 10. **X-axis Measured and Predicted Acceleration in Greenland: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).**

Figure 10, Figure 11m and Figure 12 show the Greenland accelerations, where at approximately $1,775$ seconds, the aircraft is switched from autonomous to remote control mode. Similarly, flight data is plotted in green, the ANN without adaptation in red, and the adaptive ANN in magenta.

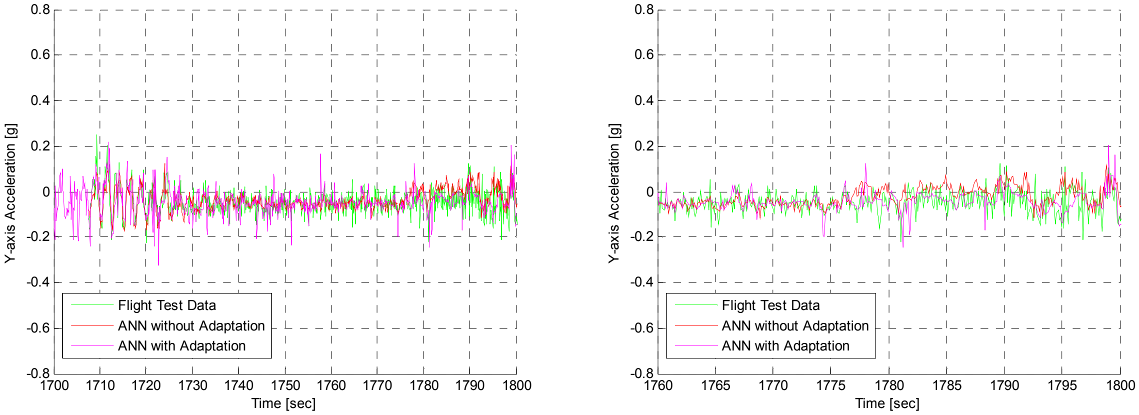

Figure 11. **Y-axis Measured and Predicted Acceleration in Greenland: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).**

As is consistent in all graphs, before mode switching, both estimations match the measured acceleration despite the high level of noise. After switching, when the aircraft is commanded to descend and approach to land, important differences are noticeable in the non-adaptive ANN.

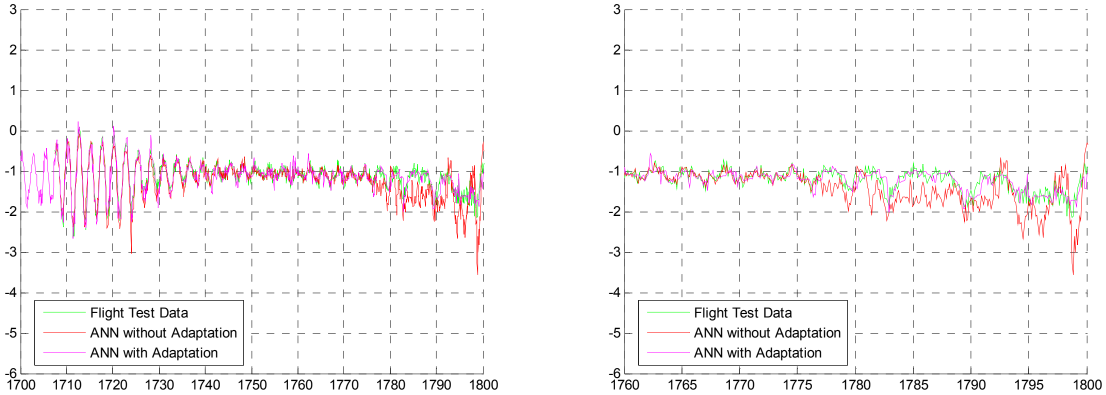

Figure 12. Z-axis Measured and Predicted Acceleration in Greenland: Mostly Autonomous Flight Portion (left) - Switch to Remote Control, Zoomed (right).

This change in flight regime is reflected in a very different dynamical characterization, and because the ANN's parameters were trained during the AUTO mode portion of the flight, the network becomes useless when experimenting a different dynamic condition.

This mismatch may also suggest that the selected inputs are not fully representative of the underlying functions. Although rate of change of some of the states were included to better reflect the physics of the aircraft interaction with its environment, extra derivatives maybe be added (see Ref. [15] and Ref. [16]) as inputs, in an effort to better characterize the dynamics in this specific flight condition (see also Ref. [12] for unsteady estimation using Fuzzy logic, during adverse weather effects).

After failures or significant changes in the dynamics or aerodynamics of a system, a previously well trained ANN could become outdated, forcing new training. Weights and biases should be updated to reflect the new dynamic status. Rather than doing this process in a separate way, the idea would be to keep the ANN under permanent training, readjusting its parameters in a continuous way, acquiring the new dynamics embedded into the data in an uninterrupted way. By using a recursive training process, the ANN parameters are continuously updated.

1.4 Nonlinear and Time-Varying Aircraft Mathematical Description

This subsection presents the assembled aircraft model, the building block of the following developments. From Equations (11), (12), (13), (14), (15) and (19) the aircraft dynamics are defined as

$$
\begin{aligned}
&\dot{V}_T = (U\dot{U} + V\dot{V} + W\dot{W})\big/V_T \\
&\dot{\alpha} = (U\dot{W} - W\dot{U})\big/(UU + WW) \\
&\dot{\beta} = (V_T\dot{V} - V\dot{V}_T)\big/(V_T^2\cos\beta) \\
&\dot{\phi} = P + \tan\theta\,(Q\sin\phi + R\cos\phi) \\
&\dot{\theta} = Q\cos\phi - R\sin\phi \\
&\dot{\psi} = (Q\sin\phi + R\cos\phi)\big/\cos\theta \\
&\dot{P} = (J_{xz}[J_{xx} - J_{yy} + J_{zz}]PQ - [J_{zz}\left(J_{zz} - J_{yy}\right) + J_{xz}^2]QR + J_{zz}L^A + J_{xz}N^A)\,/\,(J_{xx}J_{zz} - J_{xz}^2) \\
&\dot{Q} = ([J_{zz} - J_{xx}]PR - J_{xz}[P^2 - R^2] + M^A)\,/\,(J_{xx}J_{zz} - J_{xz}^2) \\
&\dot{R} = ([(J_{xx} - J_{yy})J_{xx} + J_{xz}^2]PQ - J_{xz}[J_{xx} - J_{yy} + J_{zz}]QR + J_{xz}L^A + J_{xx}N^A)\,/\,(J_{xx}J_{zz} - J_{xz}^2) \\
&\dot{\delta}_T = -\delta_T\,/\,\tau_T + \delta_T^{cmd}\,/\,\tau_T \\
&\dot{\delta}_E = -\delta_E\,/\,\tau_E + \delta_E^{cmd}\,/\,\tau_E \\
&\dot{\delta}_A = -\delta_A\,/\,\tau_A + \delta_A^{cmd}\,/\,\tau_A \\
&\dot{\delta}_R = -\delta_R\,/\,\tau_R + \delta_R^{cmd}\,/\,\tau_R \\
&\dot{p}_N = U\cos\theta\cos\psi + V(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi) + W(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) - w_N \\
&\dot{p}_E = U\cos\theta\sin\psi + V(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) + W(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) - w_E \\
&\dot{p}_H = U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta - w_H
\end{aligned}
\tag{43}
$$

where from a symmetric fixed wing angle assumption, $J_{xy} = J_{yx} = J_{yz} = J_{zy} = 0$, $J_{xz} = J_{zx}$. After replacing body velocities U,V,W by airflow angles α,β and airspeed V_T, which are much more accessible for measurement, they should be treated in (43) as intermediate variables (only kept for brevity of notation), as well as $\dot{U},\dot{V},\dot{W}$ from (11), restated here after considering Equations (16), (17) and (18)

$$
\begin{aligned}
&\dot{U} = RV - QW - g\sin\theta + (X^P + X^A)\big/m \\
&\dot{V} = -RU + PW + g\sin\phi\cos\theta + Y^A\big/m \\
&\dot{W} = QU - PV + g\cos\phi\cos\theta + Z^A\big/m
\end{aligned}
\tag{44}
$$

Throttle, elevator, aileron, and rudder servos, considered an integral part of the aircraft, are modeled in this work as first order systems with time constants $\tau_T, \tau_E, \tau_A, \tau_R$ and states $\delta_T, \delta_E, \delta_A, \delta_R$, respectively, and defined with respect to the body coordinate system $\boldsymbol{\bar{B}}$. Servos are driven by servo commands $\boldsymbol{u}^T(t) = [\delta_T^{cmd}, \delta_E^{cmd}, \delta_A^{cmd}, \delta_R^{cmd}]$, signals coming from the controller. Inertial position components p_N, p_E, p_H, referred to the inertial coordinate system $\boldsymbol{I}$, corresponding to North, East and Height of the aircraft, respectively, are also included, as their values will be used by the trajectory guidance logic as well as by the extended Kalman filter. The aircraft state vector is now extended to be $\boldsymbol{x}^T(t) = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, \delta_T, \delta_E, \delta_A, \delta_R, p_N, p_E, p_H]$.

Wind components w_N, w_E, w_H have been included as external inputs to the model in order to keep consistency between aircraft inertial and earth inertial coordinate systems. These quantities are assumed known for now. As will be seen later, particularly in the case of the extended Kalman filter, these quantities will be added as new states to the aircraft model along with new other states, i.e. measurement biases, to be dynamically estimated by the filter.

From here on, the aircraft dynamics model defined by Equation (43) and its associated equations, without the wind components w_N, w_E, w_H, will be represented by $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, with $\boldsymbol{x}(t) \in R^n$ and $\boldsymbol{u}(t) \in R^m$ as previously defined. The output model or observation equation $\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, with $\boldsymbol{y}(t) \in R^r$ will be defined opportunely, when required. This equation is highly dependent on the available sensors and their performance. System noise $\boldsymbol{v}(t)$ represents the model mistmatch with respect to the aircraft and acts through the dynamic equation $\boldsymbol{f}$. Measurement noise $\boldsymbol{w}(t)$ represents the mistmatch between the actual value and the observed value of a particular variable. It acts through the observation equation $\boldsymbol{h}$. Both noises will be defined in the sequel, especially in the filter design and also in the predictive controller design.

2. Observer Design

2.1 Introduction

Observation and estimation of unmeasured states from noisy measurements has been an important topic in navigation and flight control theory. Many works have been devoted to linear and nonlinear observing or filtering. The Kalman filter in particular seems to be the most widely used observer (see Ref. [21], [92], [93] and [94]). In aircraft navigation, different types of sensors are optimally or sub-optimally integrated through Kalman filters to improve measurements and to estimate inaccessible or recover unmeasured states.

With a varying complexity, the application of different versions of the Kalman filter mostly depends on the available computational power and the associated degree of approximation. The application is also based on the statistics of measurement noises and biases. More sophisticated filters are designed to achieve better accuracy. Particle Filtering (Ref. [22]) and Sigma-Point (Ref. [23]) are employed to overcome the inherent constraint of the extended Kalman filter dealing with highly nonlinear systems with non-Gaussian noise, or when Gaussian noise measurement undergoes nonlinear transformations and the assumption of jointly normal distributions is no longer valid. These refined filters demand an even higher computational power. Carefully regarding these restrictions, the extended Kalman filter becomes a good tradeoff solution. Ref. [24] presents a comparison between these three filters and highlights their strengths and weakness as applied to GPS/INS sensor fusion. Another approach to overcome inherent linearization approximation errors within the EKF formulation is presented in Ref. [25], where the use of the State Dependent Riccati Equation (SDRE) filter to design the observer it is proposed. An inherent requirement for the application of this solution is the ability to represent the aircraft model in a linear time varying (LTV) framework.

The development and further operation of a demanding autonomous flight control system (FCS) requires the estimation of normally inaccessible or difficult to measure states of a UAS and the improvement of those measured states contaminated by noise and subjected to unknown and possibly time-varying biases. Subsets of these states are normally needed for guidance, navigation, and control of a UAS, as well as subsequent nonlinear estimation; in this case, aerodynamic forces and moments. This work presents an EKF to perform the observation/estimation of all states of a UAS, including the three components of wind and a set of unknown biases assumed to have a slow time variation with respect to the dynamics of the aircraft. Ref. [26] and Ref. [27] present horizontal wind components estimation for UAS application.

As a way to correct installation misalignment of sensors with respect to the reference system and inherent index or offset errors not corrected by calibration processes, a set of sensor bias states are added. In order to model these new states, a slow time-variance is assumed (i.e. constant between measurements but variable otherwise). The biases are characterized with a zero time rate of change.

Another important added feature in a robust observer design is the inclusion of redundant measurements from independent sensors as a way to improve the observation/estimation of states. By the appropriate design of the EKF equations, the filter should perform a close to optimal merging of these analogous but independent measurements. Special care has to be taken in the characterization of the measurement noise statistics. Ref. [28] presents a comparison of two different techniques to fuse multisensory data. The first one merges the information within the filter and the second one combines the data prior to entering the filter. The present work fuses them within the filter formulation.

Another important design requirement for a useful observer is the accuracy of the aircraft modeling, including its dynamics, sensors, and actuators characterization. In general terms, observers perform a prediction-correction type of calculation where predicted quantities are corrected with fresh measured data. The quality of the predicted values is directly based on the quality of the model used within the observer algorithm. Adequate effort needs to be spent in improving the aircraft dynamic modeling as the

EKF performance depends on it. Bad modeling of sensors and actuators is also deleterious. Sensor noise statistics are needed within the observer formulation.

This chapter is mainly committed to the development of an EKF.

2.2 Extended Kalman Filter Design

The design of an EKF, as is the case for any observer (called filter as well, given its collateral noise filtering effect), requires a model for the prediction phase. This model is primarily based on the aircraft dynamic model recently defined, i.e. $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, and the available sensor information represented by the observation or output equation $\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t))$; to be defined. The filter functions $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{h}}$ represent the system to be observed and, as such, incorporate all available information of the aircraft characterized by functions $\boldsymbol{f}$ and $\boldsymbol{h}$, but also consider other structures to be estimated as biases and wind, i.e. $\hat{\boldsymbol{f}} \supseteq \boldsymbol{f}$ and $\hat{\boldsymbol{h}} \supseteq \boldsymbol{h}$. Available knowledge of model mismatch $\boldsymbol{v}(t)$ and sensor noise $\boldsymbol{w}(t)$ is included through the Kalman gain $\boldsymbol{K}(t)$ calculation as shown below. From this knowledge, an observer equation in continuous time (based on Ref. [95]) is assembled as

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t) &= \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)) + \boldsymbol{K}(t)(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)) \\ \hat{\boldsymbol{y}}(t) &= \hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t)) \end{aligned} \tag{45}$$

where $\hat{\boldsymbol{x}}(t) \in R^{\hat{n}}$ is the filter state vector and $\hat{\boldsymbol{u}}(t) \in R^{\hat{m}}$ is the filter input vector. In general, the filter inputs correspond to the same system inputs, i.e. $\hat{\boldsymbol{u}} = \boldsymbol{u}$ and $\hat{m} = m$. The filter output vector $\hat{\boldsymbol{y}}(t) \in R^{\hat{r}}$ is determined from the aircraft sensor information, i.e. defined to replicate the available aircraft measurements, $\boldsymbol{y}(t) \in R^{r}$. As the filter output is designed to reproduce the available measurement structure and work with the output error $\boldsymbol{y} - \hat{\boldsymbol{y}}$, normally $\hat{r} = r$. As will be seen, estimation of the aircraft states $\boldsymbol{x}(t) \in R^{n}$, the ultimate objective of the observation, will be obtained from the filter states, i.e. $\boldsymbol{x} \subseteq \hat{\boldsymbol{x}}$.

In order to facilitate the application of the extended Kalman filter algorithm, aircraft dynamics and observation equations $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{v}(t)$ and $\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t)$ are assumed to be additively corrupted by signals $\boldsymbol{v}$ and $\boldsymbol{w}$, which are assumed to be Gaussian, uncorrelated, and of appropriate dimensions, i.e. $\boldsymbol{v}(t) \in R^n$ and $\boldsymbol{w}(t) \in R^r$. From Equation (45), the gain $\boldsymbol{K}(t)$, to be computed by the extended Kalman filter algorithm, is in general a time-varying matrix that accounts for non-stationary noise statistics, and adjusts to time-varying linear representations of the nonlinear system under different dynamic conditions. Under the assumption of stationary noise, a constant Kalman gain $\boldsymbol{K}$ may be calculated off-line for a chosen trim condition, and may be used in Equation (45). This type of simplification is a viable solution when there is not enough computational power available to calculate the Kalman gain in real-time. However, this restricts the operation to the vicinity of the assumed trim condition. Despite these assumptions, the dynamic or propagation function $\boldsymbol{f}$ and the observation equation $\boldsymbol{h}$ still include all known information of the system, increasing the chance of an acceptable estimation.

Using the Euler discretization method, a same order $\hat{n}$ discrete version of the continuous model is obtained, defined at each sampling time interval ΔT, to be used in the discrete Kalman filter:

$$\begin{aligned} \hat{\boldsymbol{x}}[k+1] &= \hat{\boldsymbol{x}}[k] + \Delta T[\hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{u}}[k]) \\ \hat{\boldsymbol{y}}[k] &= \hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}[k], \hat{\boldsymbol{u}}[k]) \end{aligned} \tag{46}$$

where the index $[k]$ corresponds to time (t) sampled at $t = k\Delta T$. The discrete extended Kalman filter algorithm (Ref. [95]) is defined as follows:

$$\begin{aligned} &prediction\begin{cases} \hat{\boldsymbol{x}}[k/k-1] = \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}[k-1/k-1], \hat{\boldsymbol{u}}[k-1]) \\ \boldsymbol{P}[k/k-1] = \boldsymbol{F}[k-1]\boldsymbol{P}[k-1/k-1]\boldsymbol{F}^T[k-1] + \boldsymbol{Q}[k-1] \end{cases} \\ &correction\begin{cases} \boldsymbol{K}[k] = \boldsymbol{P}[k/k-1]\boldsymbol{H}^T[k](\boldsymbol{H}[k]\boldsymbol{P}[k/k-1]\boldsymbol{H}^T[k] + \boldsymbol{R}[k])^{-1} \\ \hat{\boldsymbol{x}}[k/k] = \hat{\boldsymbol{x}}[k/k-1] + \boldsymbol{K}[k](\boldsymbol{y}[k] - \hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}[k/k-1], \hat{\boldsymbol{u}}[k])) \\ \boldsymbol{P}[k/k] = [I - \boldsymbol{K}[k]\boldsymbol{H}[k]]\boldsymbol{P}[k/k-1] \end{cases} \end{aligned} \tag{47}$$

Two phases are defined in this discrete observer, i.e. the prediction and correction steps. In the prediction stage, no new information is considered and the one-step ahead prediction state vector at time $[k]$, i.e. $\hat{\boldsymbol{x}}[k/k-1]$, is obtained by using the propagation equation $\hat{\boldsymbol{f}}$ based on previous values of state and input vectors at time $[k-1]$. In the correction phase, the predicted state vector $\hat{\boldsymbol{x}}[k/k-1]$ is corrected, delivering $\hat{\boldsymbol{x}}[k/k]$ based on the difference between the output measurement at time $[k]$ and its predicted value, i.e. the output prediction error $\boldsymbol{y}[k]-\hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}[k/k-1],\hat{\boldsymbol{u}}[k])$, scaled by the updated Kalman gain $\boldsymbol{K}[k]$.

In Equation (47), $\boldsymbol{P}$ acts as the state error $(\boldsymbol{x}-\hat{\boldsymbol{x}})$ covariance and matrices $\boldsymbol{F}$ and $\boldsymbol{H}$ constitute the Jacobian of the propagation equation $\hat{\boldsymbol{f}}$ and observation equation $\hat{\boldsymbol{h}}$, respectively

$$\boldsymbol{F}[k-1]=\left.\frac{\partial\hat{\boldsymbol{f}}}{\partial\boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}[k-1/k-1],\hat{\boldsymbol{u}}[k-1]} \qquad \boldsymbol{H}[k]=\left.\frac{\partial\hat{\boldsymbol{h}}}{\partial\boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}[k/k-1]} \tag{48}$$

For nonlinear systems operated close to a trim point, these Jacobian matrices become nearly constants and can be computed off-line. The extended Kalman filter reduces to the Kalman filter and the algorithm in Equation (47) greatly simplifies. Matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the covariance matrices of the propagation equation noise $\boldsymbol{v}$ and observation equation noise $\boldsymbol{w}$, respectively. Under the assumption of stationary noise statistics, these covariance matrices become constant. Under the concurrence of the previous two assumptions, i.e. system linear behavior and stationary noises, matrices $\boldsymbol{P}$ and $\boldsymbol{K}$ become independent of the Kalman algorithm execution and their asymptotic values can be computed off-line. This is a further simplification of the algorithm.

2.3 Extended Kalman Filter Application

An extended Kalman filter is designed for a 40% scale Yak 54 UAS, defined by $S=21.11\,[ft^2]$, b=$10.16[ft]$, $\overline{c}=2.19\,[ft]$, $m=1.71\,[lbm]$, $\rho=0.002295\,[slug/ft^3]$, $J_{xx}=4.2277\,[slug\cdot ft^2]$, $J_{yy}=9.9008\,[slug\cdot ft^2]$, $J_{zz}=11.4716\,[slug\cdot ft^2]$ and $J_{xz}=0\,[slug\cdot ft^2]$. The filter state vector $\hat{\boldsymbol{x}}$ and input vector $\hat{\boldsymbol{u}}$=$\boldsymbol{u}$, with $\hat{n}=31$ and $\hat{m}=m=4$, are defined respectively as:

$$\hat{\boldsymbol{x}}=\begin{bmatrix} \hat{V}_T & \hat{\alpha} & \hat{\beta} & \hat{\phi} & \hat{\theta} & \hat{\psi} & \hat{P} & \hat{Q} & \hat{R} & \hat{\delta}_T & \hat{\delta}_E & \hat{\delta}_A & \hat{\delta}_R & \hat{p}_N & \hat{p}_E & \hat{p}_H & \hat{w}_N & \hat{w}_E & \hat{w}_H\cdots \\ & & & & & & \cdots & \hat{V}_{Tb} & \hat{\alpha}_b & \hat{\beta}_b & \hat{P}_b & \hat{Q}_b & \hat{R}_b & \hat{a}_{Xb} & \hat{a}_{Yb} & \hat{a}_{Zb} & \hat{H}_{Xb} & \hat{H}_{Yb} & \hat{H}_{Zb} \end{bmatrix}^T \tag{49}$$

$$\boldsymbol{u}=\begin{bmatrix}\delta_{Tcmd} & \delta_{Ecmd} & \delta_{Acmd} & \delta_{Rcmd}\end{bmatrix}^T \tag{50}$$

The output equation vector $\hat{\boldsymbol{y}} = \hat{\boldsymbol{h}}(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t))$, with $\hat{r} = 24$, is defined as:

$$\hat{\boldsymbol{h}}: \begin{cases} \hat{y}_1 = \hat{p}_N \\ \hat{y}_2 = \hat{p}_E \\ \hat{y}_3 = \hat{p}_H \\ \hat{y}_4 = \hat{\phi} \\ \hat{y}_5 = \hat{\theta} \\ \hat{y}_6 = \hat{\psi} \\ \hat{y}_7 = \hat{P} \\ \hat{y}_8 = \hat{Q} \\ \hat{y}_9 = \hat{R} \\ \hat{y}_{10} = \hat{V}_T + \hat{V}_{Tb} \\ \hat{y}_{11} = \hat{\alpha} + \hat{\alpha}_b \\ \hat{y}_{12} = \hat{\beta} + \hat{\beta}_b \\ \hat{y}_{13} = \hat{H}_X + \hat{H}_{Xb} \\ \hat{y}_{14} = \hat{H}_Y + \hat{H}_{Yb} \\ \hat{y}_{15} = \hat{H}_Z + \hat{H}_{Zb} \\ \hat{y}_{16} = \hat{P} + \hat{P}_b \\ \hat{y}_{17} = \hat{Q} + \hat{Q}_b \\ \hat{y}_{18} = \hat{R} + \hat{R}_b \\ \hat{y}_{19} = (\hat{X}^P + \hat{X}^A)/m + \hat{a}_{Xb} \\ \hat{y}_{20} = \hat{Y}^A/m + \hat{a}_{Yb} \\ \hat{y}_{21} = \hat{Z}^A/m + \hat{a}_{Zb} \\ \hat{y}_{22} = \hat{\delta}_E \\ \hat{y}_{23} = \hat{\delta}_A \\ \hat{y}_{24} = \hat{\delta}_R \end{cases} \tag{51}$$

Measurements include inertial position (GPS), Euler angles (IMU), body angular rates (IMU), airflow angles vane and airspeed probe, 3D magnetic flux, body angular rate (3D gyros), translational accelerations (3D accelerometers), and elevator, aileron and rudder deflections.

The state equation vector $\dot{\hat{\boldsymbol{x}}}(t) = \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))$ is defined as:

$$\hat{\boldsymbol{f}}:\begin{cases}
\dot{\hat{V}}_T = (\hat{U}\dot{U} + \hat{V}\dot{\hat{V}} + \hat{W}\dot{\hat{W}})\Big/\hat{V}_T \\
\dot{\hat{\alpha}} = (\hat{U}\dot{\hat{W}} - \hat{W}\dot{U})\Big/(\hat{U}\hat{U} + \hat{W}\hat{W}) \\
\dot{\hat{\beta}} = (\hat{V}_T\dot{\hat{V}} - \hat{V}\dot{\hat{V}}_T)\Big/(\hat{V}_T^2 \cos\hat{\beta}) \\
\dot{\hat{\phi}} = \hat{P} + \tan\hat{\theta}(\hat{Q}\sin\hat{\phi} + \hat{R}\cos\hat{\phi}) \\
\dot{\hat{\theta}} = (\hat{R}\cos\hat{\phi} + \hat{Q}\sin\hat{\phi}) / \cos\hat{\theta} \\
\dot{\hat{\psi}} = \hat{Q}\cos\hat{\phi} - \hat{R}\sin\hat{\phi} \\
\dot{\hat{P}} = (J_{xz}[J_{xx} - J_{yy} + J_{zz}]\hat{P}\hat{Q} - [J_{zz}\left(J_{zz} - J_{yy}\right) + J_{xz}^2]\hat{Q}\hat{R} + J_{zz}\hat{L}^A + J_{xz}\hat{N}^A) / (J_{xx}J_{zz} - J_{xz}^2) \\
\dot{\hat{Q}} = ([J_{zz} - J_{xx}]\hat{P}\hat{R} - J_{xz}[\hat{P}^2 - \hat{R}^2] + \hat{M}^A) / (J_{xx}J_{zz} - J_{xz}^2) \\
\dot{\hat{R}} = ([(J_{xx} - J_{yy})J_{xx} + J_{xz}^2]\hat{P}\hat{Q} - J_{xz}[J_{xx} - J_{yy} + J_{zz}]\hat{Q}\hat{R} + J_{xz}\hat{L}^A + J_{xx}\hat{N}^A) / (J_{xx}J_{zz} - J_{xz}^2) \\
\dot{\hat{\delta}}_T = -\hat{\delta}_T / \tau_T + \delta_{Tcmd} / \tau_T \\
\dot{\hat{\delta}}_E = -\hat{\delta}_E / \tau_E + \delta_{Ecmd} / \tau_E \\
\dot{\hat{\delta}}_A = -\hat{\delta}_A / \tau_A + \delta_{Acmd} / \tau_A \\
\dot{\hat{\delta}}_R = -\hat{\delta}_R / \tau_R + \delta_{Rcmd} / \tau_R \\
\dot{\hat{p}}_N = \hat{U}\cos\hat{\theta}\cos\hat{\psi} + \hat{V}(-\cos\hat{\phi}\sin\hat{\psi} + \sin\hat{\phi}\sin\hat{\theta}\cos\hat{\psi}) + \hat{W}(\sin\hat{\phi}\sin\hat{\psi} + \cos\hat{\phi}\sin\hat{\theta}\cos\hat{\psi}) - \hat{w}_N \\
\dot{\hat{p}}_E = \hat{U}\cos\hat{\theta}\sin\hat{\psi} + \hat{V}(\cos\hat{\phi}\cos\hat{\psi} + \sin\hat{\phi}\sin\hat{\theta}\sin\hat{\psi}) + \hat{W}(-\sin\hat{\phi}\cos\hat{\psi} + \cos\hat{\phi}\sin\hat{\theta}\sin\hat{\psi}) - \hat{w}_E \\
\dot{\hat{p}}_H = \hat{U}\sin\hat{\theta} - \hat{V}\sin\hat{\phi}\cos\hat{\theta} - \hat{W}\cos\hat{\phi}\cos\hat{\theta} - \hat{w}_H \\
\dot{\hat{w}}_N = 0 \\
\dot{\hat{w}}_E = 0 \\
\dot{\hat{w}}_H = 0 \\
\dot{\hat{V}}_{Tb} = 0 \\
\dot{\hat{\alpha}}_b = 0 \\
\dot{\hat{\beta}}_b = 0 \\
\dot{\hat{P}}_b = 0 \\
\dot{\hat{Q}}_b = 0 \\
\dot{\hat{R}}_b = 0 \\
\dot{\hat{a}}_{Xb} = 0 \\
\dot{\hat{a}}_{Yb} = 0 \\
\dot{\hat{a}}_{Zb} = 0 \\
\dot{\hat{H}}_{Xb} = 0 \\
\dot{\hat{H}}_{Yb} = 0 \\
\dot{\hat{H}}_{Zb} = 0
\end{cases} \qquad (52)$$

where, besides the aircraft model from Equation (43), wind components $\hat{w}_N, \hat{w}_E, \hat{w}_H$ are added as states, as well as a set of measurement biases, i.e. airspeed and airflow angle biases $\hat{V}_{Tb}, \hat{\alpha}_b, \hat{\beta}_b$, body angular rates biases (3D gyros) $\hat{P}_b, \hat{Q}_b, \hat{R}_b$, body translational acceleration biases $\hat{a}_{Xb}, \hat{a}_{Yb}, \hat{a}_{Zb}$, and magnetic flux biases $\hat{H}_{Xb}, \hat{H}_{Yb}, \hat{H}_{Zb}$. Subscripts X, Y, Z refer to body reference frame $\overline{\boldsymbol{B}}$ and subscript b indicates bias of the associated state. Body velocities $\hat{U}, \hat{V}, \hat{W}$ are defined in terms of $\hat{V}_T, \hat{\alpha}, \hat{\beta}$, as in Equation (15), and body accelerations $\dot{\hat{U}}, \dot{\hat{V}}, \dot{\hat{W}}$, as in Equation (44), are rewritten here

$$\begin{aligned} \dot{\hat{U}} &= \hat{R}\hat{V} - \hat{Q}\hat{W} - g\sin\hat{\theta} + (\hat{X}^P + \hat{X}^A)/m \\ \dot{\hat{V}} &= -\hat{R}\hat{U} + \hat{P}\hat{W} + g\sin\hat{\phi}\cos\hat{\theta} + \hat{Y}^A/m \\ \dot{\hat{W}} &= \hat{Q}\hat{U} - \hat{P}\hat{V} + g\cos\hat{\phi}\cos\hat{\theta} + \hat{Z}^A/m \end{aligned} \tag{53}$$

Aerodynamic forces $\hat{X}^A, \hat{Y}^A, \hat{Z}^A$ and moments $\hat{L}^A, \hat{M}^A, \hat{N}^A$, defined by Equations (18) and (19), respectively, follow the linear component build-up approach detailed in 1.2.3.1, obtained for an airspeed trim condition of $V_{Ttrim} = 70$ [knots]. Based on Chapter 1, and to complete the UAS modeling, stability and control derivatives from Ref. [99] are employed. These parameters were calculated, using the software Advanced Aircraft Analysis (AAA) developed by Design, Analysis, and Research Corporation (DARcorporation, Ref. [100]), for a straight and level flight at a constant altitude and at a constant speed of 70 [knots]. Table I summarizes the stability and control derivatives of the 1/3 scale Yak54.

Following the approach in 1.2.2, thrust T is modeled as a function of throttle deflection δ_T, i.e. $T = X^P(\boldsymbol{x}, t) = T_X^2 \cdot \delta_T^{\,2}(t) + T_X^1 \cdot \delta_T(t) + T_X^0$, where parameters $T_X^2 = 0$, $T_X^1 = 0.3815$ and $T_T^0 = 0$ were selected to approximate a linear function matching flight test data at the selected trim point.

To complete the observation equation $\hat{\boldsymbol{h}}$ in (51), the estimated body-axes magnetic flux $\hat{H}_X, \hat{H}_Y, \hat{H}_Z$ is associated to existent inertial magnetic flux H_{N0}, H_{E0}, H_{D0} through Euler angles, as follows:

$$\begin{bmatrix} \hat{H}_X \\ \hat{H}_Y \\ \hat{H}_Z \end{bmatrix} = \boldsymbol{C}_I^B\left(\hat{\phi}, \hat{\theta}, \hat{\psi}\right) \begin{bmatrix} H_{N0} \\ H_{E0} \\ H_{D0} \end{bmatrix} + \begin{bmatrix} \hat{H}_{Xb} \\ \hat{H}_{Yb} \\ \hat{H}_{Zb} \end{bmatrix} \tag{54}$$

where $\boldsymbol{C}_I^B$ corresponds to the rotation matrix from inertial coordinate system $\overline{\boldsymbol{I}}$ to body coordinate system $\overline{\boldsymbol{B}}$. Using, for example, NOAA Earth's Magnetic Field Calculator, the inertial magnetic flux

H_{N0}, H_{E0}, H_{D0} associated with the flight test area and flight date (assumed fixed), is obtained (based in Ref. [26]). Observation equation $\boldsymbol{h}$ is then completed, inserting Equations (54) into Equation (51).

Table I. AAA Stability and Control Derivatives for 40% Yak 54 at 70 **Knots**

Parameter	Value	Units	Parameter	Value	Units
C_{D0}	*0.023*	*[-]*	$C_{y\beta}$	*-0.4118*	*[1/rad]*
$C_{D\alpha}$	*-0.0027*	*[1/rad]*	$C_{y\dot{\beta}}$	*0*	*[1/rad]*
$C_{D\delta e}$	*-0.0004*	*[1/rad]*	C_{yp}	*0.0116*	*[1/rad]*
$C_{D\dot{\alpha}}$	*0*	*[1/rad]*	C_{yr}	*0.3557*	*[1/rad]*
C_{Dq}	*0*	*[1/rad]*	$C_{y\delta a}$	*0*	*[1/rad]*
C_{Du}	*0.0011*	*[1/rad]*	$C_{y\delta r}$	*0.2931*	*[1/rad]*
C_{L0}	*0*	*[-]*	$C_{l\beta}$	*-0.0336*	*[1/rad]*
$C_{L\alpha}$	*5.0060*	*[1/rad]*	$C_{l\dot{\beta}}$	*0*	*[1/rad]*
$C_{L\delta e}$	*0.4160*	*[1/rad]*	C_{lp}	*-0.3740*	*[1/rad]*
$C_{L\dot{\alpha}}$	*2.3669*	*[1/rad]*	C_{lr}	*0.0602*	*[1/rad]*
C_{Lq}	*7.3911*	*[1/rad]*	$C_{l\delta a}$	*0.2593*	*[1/rad]*
C_{Lu}	*0.0017*	*[1/rad]*	$C_{l\delta r}$	*0.0242*	*[1/rad]*
C_{m0}	*0.0103*	*[-]*	$C_{n\beta}$	*0.1417*	*[1/rad]*
$C_{m\alpha}$	*-1.0286*	*[1/rad]*	$C_{n\dot{\beta}}$	*0*	*[1/rad]*
$C_{m\delta e}$	*-1.0369*	*[1/rad]*	C_{np}	*-0.0189*	*[1/rad]*
$C_{m\dot{\alpha}}$	*-5.9000*	*[1/rad]*	C_{nr}	*-0.1971*	*[1/rad]*
C_{mq}	*-10.664*	*[1/rad]*	$C_{n\delta a}$	*-0.0071*	*[1/rad]*
C_{mu}	*0.0002*	*[1/rad]*	$C_{n\delta r}$	*-0.1674*	*[1/rad]*

2.3.1 Gain Scheduling Off-line Kalman Gain Matrices

When flight operation is expected to strictly occur in the vicinity of a trim condition, and noise features are assumed to be invariant, online calculation of the Kalman gain matrix $\boldsymbol{K}$, associated state error covariance $\boldsymbol{P}$, Jacobians $\boldsymbol{F}$ and $\boldsymbol{H}$, and covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be replaced by an off-line computation.

Both observation and propagation equations, (51) and (52), respectively, are directly or indirectly functions of yaw angle $\hat{\psi}$. Linearization of these equations over the same trim point, in terms of airspeed and flight condition, e.g. straight and level flight, but at different yaw values, will vary from one azimuth to another. This means that each linearization and its corresponding constant Kalman gain $\boldsymbol{K}$ will be valid only at one azimuth. This validity quickly degrades when moving away from the trimmed yaw angle, even if the aircraft maneuvers slowly, keeping small perturbed states.

To neutralize the dependence of the linearization of $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{h}}$ on $\hat{\psi}$, a gain scheduling technique is proposed among a set of homogeneous precomputed Kalman gains, obtained for different yaw angles. Under the assumption of small departure from a common trim condition, i.e. a particular airspeed, the Kalman gain can be interpolated among consecutive Kalman gains as a function of the current yaw angle.

For example, for the 40% scale Yak 54 UAS flying at 70 knots in a steady state horizontal flight, a set of constant Kalman gain matrices $\boldsymbol{K}_i$ are obtained for $i = 1,\cdots,18$ different equally spaced yaw values, i.e. $\psi_i = [0^\circ, 20^\circ, 40^\circ, 60^\circ, ..., 300^\circ, 320^\circ, 340^\circ, 0^\circ]$.

For the given conditions, all Kalman gain matrices $\boldsymbol{K}_i$ share the same structure and nature, allowing an interpolation technique to be used among them. The interpolation is done per each entry. This gain scheduling among each entry of consecutive matrices, as a function of the yaw angle, provides a continuous time-varying Kalman gain as a function of yaw angle, i.e. $\boldsymbol{K}(\psi)$. The idea is depicted in Figure 13.

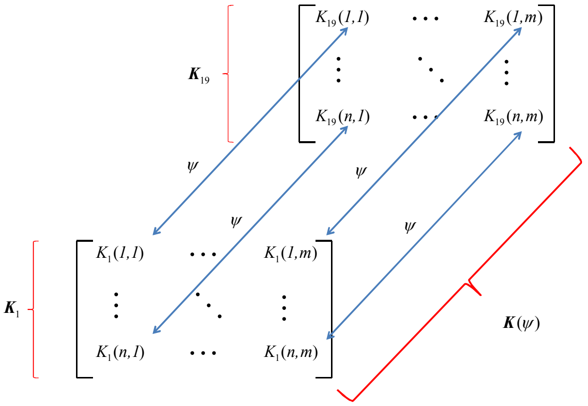

Figure 13. Kalman Gain Matrix Gain Scheduling.

The implementation is materialized by using a lookup table as a function of yaw angle where the output is the interpolated Kalman gain matrix $\boldsymbol{K}(\psi)$. Under previous assumptions of linearized behavior and stationary noises, and based on the simplified Kalman gain approach detailed in 2.2, the following filter is defined:

$$\begin{aligned} \hat{\boldsymbol{x}}[k/k-1] &= \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}[k-1/k-1], \hat{\boldsymbol{u}}[k-1]) \\ \hat{\boldsymbol{x}}[k/k] &= \hat{\boldsymbol{x}}[k/k-1] + \boldsymbol{K}(\psi)(\boldsymbol{y}[k] - \hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}[k/k-1], \hat{\boldsymbol{u}}[k])) \end{aligned} \tag{55}$$

To illustrate the severity of this yaw angle dependency, Figure 14 shows the first ten entries from the first row, $K(1,1)\cdots K(1,10)$, as a function of yaw angle ψ. A sinusoidal pattern, with important changes in absolute magnitude and sign, emerges. This graph confirms the fact that linearization of propagation and observation equations used in the current filter, at similar trim conditions but at different yaw angles, are different, and this fact needs to be considered in the application of the off-line Kalman gain filtering.

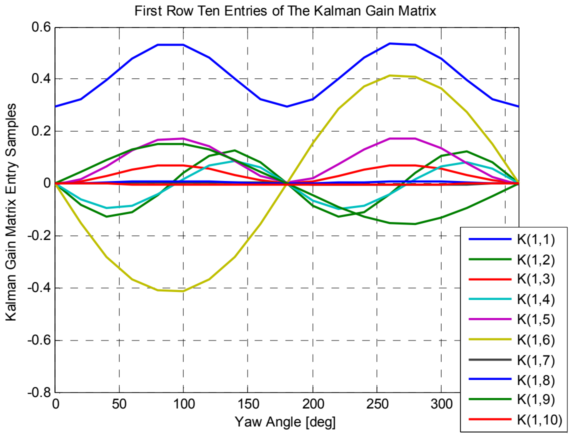

Figure 14. Kalman Gain Matrix Subset of Entries.

2.3.2 EKF Test Results

The developed EKF is applied to a 40% scale Yak 54 UAS. Previously stored flight test data, as well as simulated data, are used to test the filter.

2.3.2.1 EKF Simulation Data Results

The EKF is tested in simulation. The aircraft is engaged to follow a predefined closed trajectory, and airflow angles and airspeed measurements are subject to biases during flight. Also, wind is injected into the simulation. The following figures show some of the estimation results. Figure 15 shows the initial part of the trajectory, when the aircraft is engaged to follow it with an altitude offset.

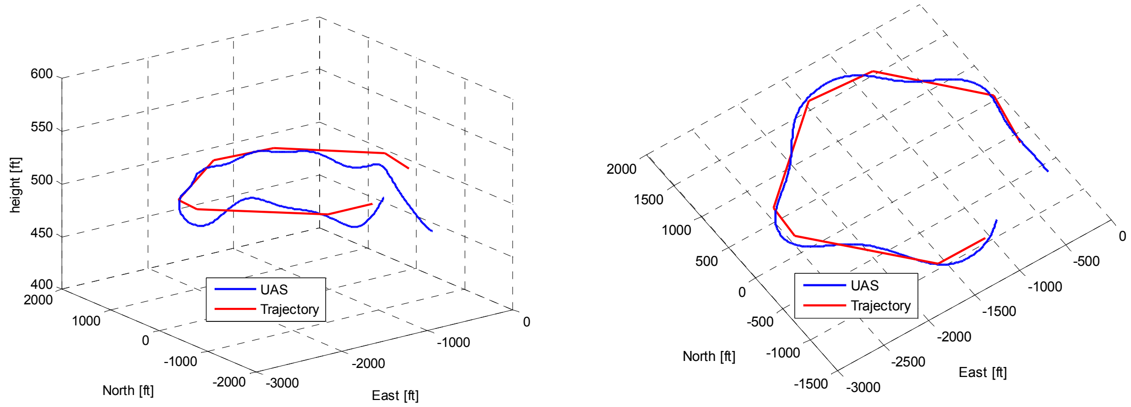

Figure 15. Simulation Data: UAS Trajectory Following.

Figure 16 (left) shows the angle of attack and the estimated one. Figure 16 (right) shows the angle of attack bias and its estimation.

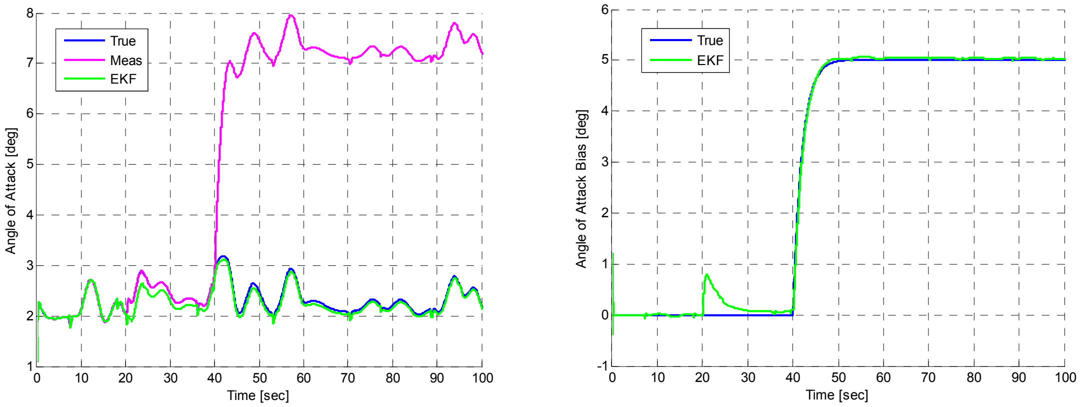

Figure 16. Simulation Data: Angle of attack (left) – Angle of attack Bias (right).

Figure 17 (left) shows the sideslip angle and the estimated one. Figure 17 (right) shows the sideslip angle bias and its estimation.

Figure 18 (left) shows the airspeed and its estimation. Figure 18 (right) shows the airspeed bias and its estimation.

All biases were estimated correctly, showing a small level of interference among each other, and some induced modulation from the trajectory itself. Airspeed bias takes the highest time to be estimated compared with both airflow angles.

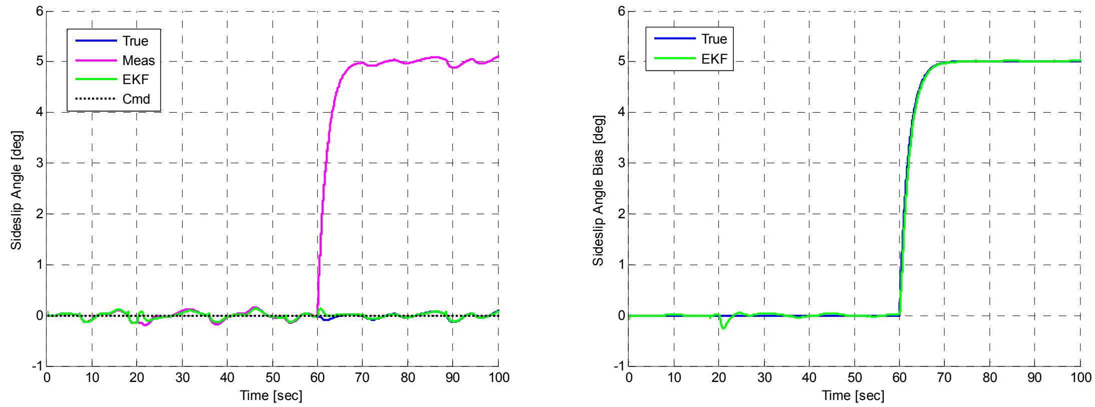

Figure 17. Simulation Data: Sideslip Angle (left) – Sideslip Angle Bias (right).

To test the wind estimation, three wind components have been introduced into the simulation at different times and different magnitudes. Figure 19 shows North and Down component estimations.

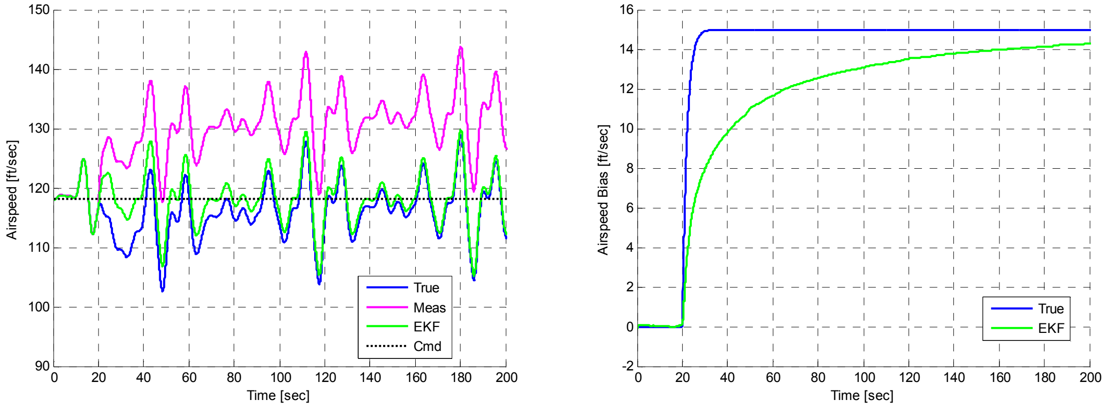

Figure 18. Simulation Data: Airspeed (left) – Airspeed Bias (right).

Wind estimation has oscillatory behavior compared with the biases, although they converge to the expected mean values. This behavior limits the type of wind acceptable for this particular observer. Gusty winds, by nature quicker and more erratic, cannot be handled optimally by this configuration. This should be tackled by the robustness of the inner loop controller as external disturbances. Only slow time-varying winds (near-constant winds) are well suited for this observer.

This behavior in biases and wind components estimation is due to the model selected to describe their nature. There is no prior knowledge about these entities so biases and wind components are modeled as constant values between measurements.

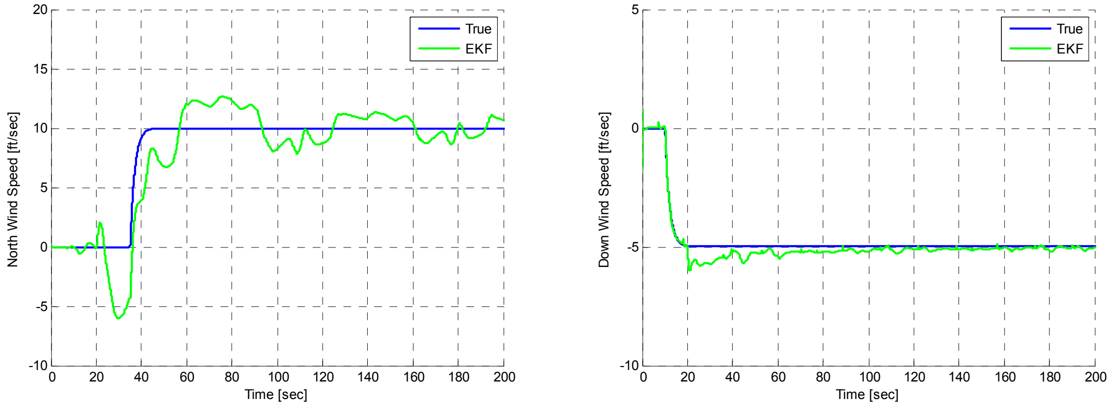

Figure 19. **Simulation Data: North Wind Component (left) – Vertical Wind Component (right).**

2.3.2.2 *EKF Flight Test Data Results*

EKF is tested with flight test data from a flight at Fort Riley, on May 08th 2012. The following figures show some of the main results.

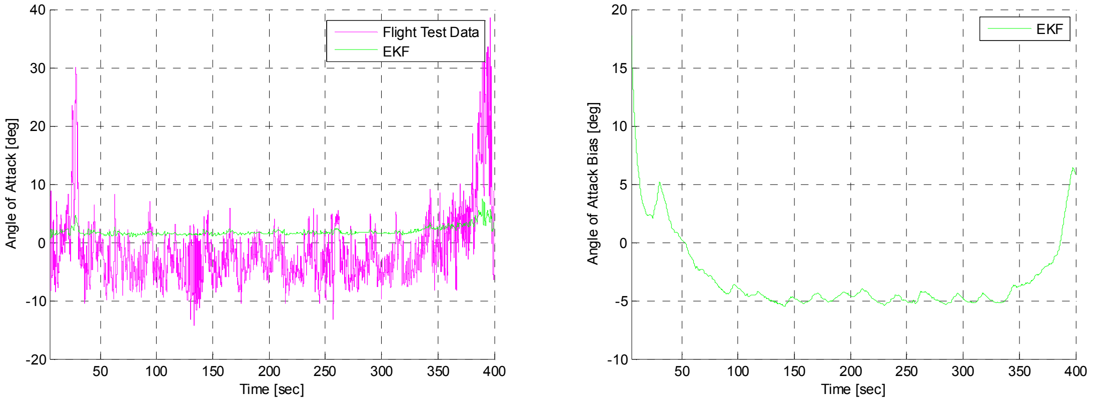

Figure 20. **Flight Data: Angle of Attack (left) – Angle of attack Bias (right).**

Figure 20 shows the measured and estimated angle of attack (left), and estimated bias (right). Figure 22 shows the measured and estimated sideslip angle (left), and estimated bias (right). Figure 21 shows the measured and estimated airspeed (left) and estimated bias (right).

Although there is no exact reference to compare with, a reasonable constant bias seems to be a good indicator of an acceptable estimation, despite the different maneuvers and flight paths. This is the case for angle of attack and sideslip angle.

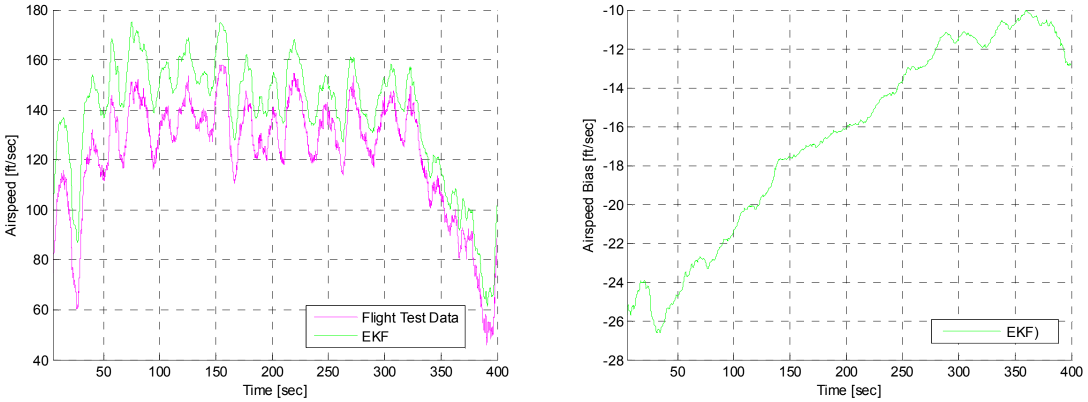

Figure 21. Flight Data: Airspeed (left) – Airspeed Bias (right).

Airspeed does not show the same pattern, although this could be consistent with a slow convergence time detected at least in simulation (see below). Presumably, in a longer flight the estimated bias should settle to a more constant value. This has yet to be tested.

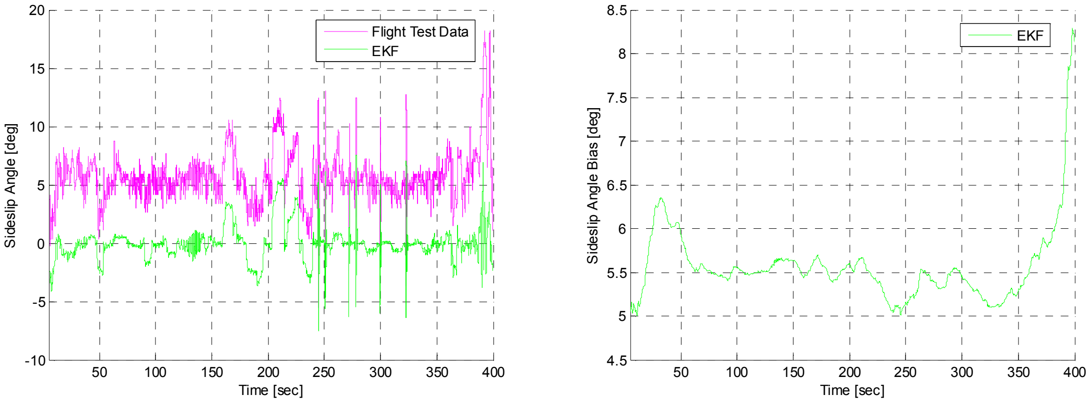

Figure 22. Flight Data: Sideslip Angle (left) - Sideslip Angle Bias (right).

2.4 Alternative Extended Kalman Filter Design

By adding extra information such as accelerometer readings to the previous EKF design, a higher degree of accuracy and broader operational ranges are possible. Previous development resorts to the linear component build-up approach and polynomial modeling, described in Chapter 1, to compute aerodynamic and propulsive forces for Equation (53), respectively. Also, due to lack of knowledge of their dynamics, it assigns the simplest possible dynamic model to wind components in the state equation $\hat{\boldsymbol{f}}$ in (52), i.e. $\dot{\hat{w}}_N = 0, \dot{\hat{w}}_E = 0, \dot{\hat{w}}_H = 0$, which corresponds to constants between measurements. Both disadvantages can potentially be mitigated by using reliable accelerometer readings as inputs to the filter.

The following modification to the filter state equation $\hat{\boldsymbol{f}}$, adds dynamic equations for the inertial velocity $\hat{v}_N, \hat{v}_E, \hat{v}_H$, defined as a function of acceleration measurements delivered by 3D accelerometers, labeled $\bar{a}_X, \bar{a}_Y, \bar{a}_Z$, and restructures the dynamic equations of inertial position $\hat{p}_N, \hat{p}_E, \hat{p}_H$, accordingly.

$$\begin{aligned}
\dot{\hat{p}}_N &= \hat{v}_N \\
\dot{\hat{p}}_E &= \hat{v}_E \\
\dot{\hat{p}}_H &= \hat{v}_H \\
\dot{\hat{v}}_N &= \bar{a}_X \cos\hat{\theta}\cos\hat{\psi} + \bar{a}_Y(-\cos\hat{\phi}\sin\hat{\psi} + \sin\hat{\phi}\sin\hat{\theta}\cos\hat{\psi}) + \bar{a}_Z(\sin\hat{\phi}\sin\hat{\psi} + \cos\hat{\phi}\sin\hat{\theta}\cos\hat{\psi}) \\
\dot{\hat{v}}_E &= \bar{a}_X \cos\hat{\theta}\sin\hat{\psi} + \bar{a}_Y(\cos\hat{\phi}\cos\hat{\psi} + \sin\hat{\phi}\sin\hat{\theta}\sin\hat{\psi}) + \bar{a}_Z(-\sin\hat{\phi}\cos\hat{\psi} + \cos\hat{\phi}\sin\hat{\theta}\sin\hat{\psi}) \\
\dot{\hat{v}}_H &= \bar{a}_X \sin\hat{\theta} - \bar{a}_Y \sin\hat{\phi}\cos\hat{\theta} - \bar{a}_Z \cos\hat{\phi}\cos\hat{\theta} - g
\end{aligned} \tag{56}$$

This modification discards the need of any approximated wind model. The filter delivers inertial position and speed, and consequently wind speed components are indirectly obtained from

$$\begin{aligned}
\hat{w}_N &= \hat{U}\cos\hat{\theta}\cos\hat{\psi} + \hat{V}(-\cos\hat{\phi}\sin\hat{\psi} + \sin\hat{\phi}\sin\hat{\theta}\cos\hat{\psi}) + \hat{W}(\sin\hat{\phi}\sin\hat{\psi} + \cos\hat{\phi}\sin\hat{\theta}\cos\hat{\psi}) - \hat{v}_N \\
\hat{w}_E &= \hat{U}\cos\hat{\theta}\sin\hat{\psi} + \hat{V}(\cos\hat{\phi}\cos\hat{\psi} + \sin\hat{\phi}\sin\hat{\theta}\sin\hat{\psi}) + \hat{W}(-\sin\hat{\phi}\cos\hat{\psi} + \cos\hat{\phi}\sin\hat{\theta}\sin\hat{\psi}) - \hat{v}_N \\
\hat{w}_H &= \hat{U}\sin\hat{\theta} - \hat{V}\sin\hat{\phi}\cos\hat{\theta} - \hat{W}\cos\hat{\phi}\cos\hat{\theta} - \hat{v}_N
\end{aligned} \tag{57}$$

Also, by studying the body translational accelerations $\dot{U}, \dot{V}, \dot{W}$ expressions, shown below, it becomes clear the terms the direct readings from accelerometers represent, as indicated:

$$\begin{aligned}
\dot{\hat{U}} &= \overbrace{\hat{R}\hat{V} - \hat{Q}\hat{W} + (\hat{X}^P + \hat{X}^A)/m}^{\bar{a}_X} - g\sin\hat{\theta} = \bar{a}_X - g\sin\hat{\theta} \\
\dot{\hat{V}} &= \overbrace{-\hat{R}\hat{U} + \hat{P}\hat{W} + \hat{Y}^A/m}^{\bar{a}_Y} + g\sin\hat{\phi}\cos\hat{\theta} = \bar{a}_Y + g\sin\hat{\phi}\cos\hat{\theta} \\
\dot{\hat{W}} &= \overbrace{\hat{Q}\hat{U} - \hat{P}\hat{V} + \hat{Z}^A/m}^{\bar{a}_Z} + g\cos\hat{\phi}\cos\hat{\theta} = \bar{a}_Z + g\cos\hat{\phi}\cos\hat{\theta}
\end{aligned} \tag{58}$$

Accelerometer measurements then replace the selected terms, directly avoiding any aerodynamic and propulsive force approximations.

A similar approach can be applied to aerodynamic moments, although it requires an intermediate step, with an associated possible level of degradation in their calculation. By analyzing the dynamics of the body angular rates in Equation (43), rewritten here:

$$\begin{aligned}
\dot{P} &= (J_{xz}[J_{xx} - J_{yy} + J_{zz}]PQ - [J_{zz}\left(J_{zz} - J_{yy}\right) + J_{xz}^{2}]QR + J_{zz}L^{A} + J_{xz}N^{A}) / (J_{xx}J_{zz} - J_{xz}^{2}) \\
\dot{Q} &= ([J_{zz} - J_{xx}]PR - J_{xz}[P^{2} - R^{2}] + M^{A}) / (J_{xx}J_{zz} - J_{xz}^{2}) \\
\dot{R} &= ([(J_{xx} - J_{yy})J_{xx} + J_{xz}^{2}]PQ - J_{xz}[J_{xx} - J_{yy} + J_{zz}]QR + J_{xz}L^{A} + J_{xx}N^{A}) / (J_{xx}J_{zz} - J_{xz}^{2})
\end{aligned} \tag{59}$$

aerodynamic moments can be obtained by recovering them from the system of equations. Inertia terms are known as well as body angular rates. Body angular rates time derivatives $\dot{P}, \dot{Q}, \dot{R}$, required to solve the system, are not directly accessible and the accuracy of their calculation, by any independent means, will determinate the accuracy level of the aerodynamic moment inference.

With a reliable procedure in obtaining accurate aerodynamic moments, original filter input vector $\boldsymbol{u} = \left[\delta_{T_{cmd}} \quad \delta_{E_{cmd}} \quad \delta_{A_{cmd}} \quad \delta_{R_{cmd}}\right]^{T}$ can be definitely replaced by the new input vector, defined as:

$$\hat{\boldsymbol{u}} = \left[\bar{a}_{X} \quad \bar{a}_{Y} \quad \bar{a}_{Z}\right]^{T} \tag{60}$$

with $\hat{m} = 3$.

3. Guidance, Navigation, and Control

3.1 Introduction

Driving a vehicle is an extremely complex task that requires the convergence of at least three interrelated tasks, i.e. guidance, navigation, and control. In general, the navigation task delivers the position and velocity of a vehicle with respect to a defined reference frame. If this navigation task is placed in closed loop with the vehicle's actuators to control its position and velocity, guidance and control tasks emerge as a direct consequence (see Ref. [29]). The guidance task is then in charge of directing the vehicle according to a predefined trajectory. It uses the information from the navigation task to determine the relative errors with respect to the trajectory. Based on this information, it actuates the controls to drive the vehicle and decrease those errors. Guidance and control tasks are normally fed by measured or estimated representative variables of the vehicle. Any vehicle-based mission, either manned or unmanned, will require the interdependent tasks of guidance, navigation, and control (GNC), to accomplish any particular mission.

Different combinations of these three tasks have been reported, but the most widely used is the employment of two control loops, denominated outer loop and inner loop. In this configuration, guidance and navigation tasks are assigned to the outer loop, while the control is assigned to the inner one. Much less common is an integrated design of these tasks. Ref. [30] presents an integrated design of guidance and control algorithms for autonomous vehicles through a simultaneous process achieving zero steady state error about a trim point, guaranteeing stability and avoiding possible cross coupling between loops.

One of the main concerns of this standard design of two separated nested loops is that, given the way they are interconnected, both loops are strongly coupled, i.e. the inner loop lies inside the outer loop. This severe condition imposes the requirement of avoiding mutual interference and loss of stability. Separately, each design may be highly stable on its own, but their interconnection can become unstable.

From some of the approaches that have been developed to implement guidance tasks for the outer loop, Ref. [31] presents a lateral track guidance based on the geometry existent between the path and the aircraft position and velocity vectors, and a control law based on fixed gains. It is designed for a UAS in the presence of strong winds. In Ref. [32], a different technique is presented where a trajectory tracking problem is formulated as a constrained nonlinear optimization problem. Ref. [33] introduces nonlinear guidance logic for curved trajectory tracking, which reduces down to a proportional-derivative control approximation when applied to straight lines.

To improve the guidance task, Ref. [34] and [35] include horizontal wind information. This is an important addition, especially for small UAS where wind has a significant effect. A different approach for path following is presented in Ref. [36] where 2D vector fields are used to represent desired ground tracks generating course inputs for an inner loop attitude control.

More recently, in Ref. [37] a Lyapunov-based adaptive back stepping approach is used to design a flight-path controller for a nonlinear F-16 model. A smooth prescribed inertial trajectory is asymptotically tracked in the presence of actuator failures and uncertainties in the stability and control derivatives. In a different approach, Ref. [38] proposes a spatial-temporal trajectory tracking capability for a fixed-wing aircraft in a time critical mission.

For the inner loop, gain scheduling seems to be the most widely used solution to actuate the aircraft controls. Gain scheduling, able in principle to control nonlinear systems over a broader range of operation where linearity assumptions are no longer valid, works by interpolating linear, time-invariant controllers. A set of predefined controllers is kept available where each controller was optimally calculated for a particular operating point, achieving stability and performance in its vicinity. Based on the aircraft dynamic condition, switching among controllers is done by interpolating their entries. Although each controller assures stability in the neighborhood of its respective trim point, transition stability is not necessarily achieved. Ref. [39] studies the transition between controllers and defines the requirement for preserving stability, and Ref. [40] and [41] present detailed analysis and design considerations.

A number of different approaches in gain scheduling have been applied to UAS. As one of the first robust gain scheduling designs, Ref. [42] designs an H-infinity gain scheduling controller for pitch-axis by interpolating poles, zeros, and gains among four fixed trim point controllers. Alternatively, Ref. [43] proposes an interpolation of the Riccati equations associated to longitudinal H-infinity designs. Ref. [44] applies the Linear Parameter-Varying (LPV) control technique to missile pitch-axis dynamics. Reference [45] presents two approaches to control a longitudinal nonlinear model of a UAS, simplified as an LPV model, over the entire range of cruise speeds. The first approach formulates the variability of the LPV model over the range of speeds as unstructured uncertainty and designs a unique robust controller able to stabilize the system over the entire range. The second approach linearly interpolates between two robust controllers calculated at the two extreme velocities, performing gain scheduling as a function of the instantaneous total velocity. The work done in Ref. [46] applies gain scheduling by using a second-order model that represents the longitudinal dynamics of an aircraft. This approach allows the scheduled gains to be adjusted against rapidly-varying system states.

The present chapter will elaborate on inner and outer loop concepts, presenting a development of gain scheduling among robust linear controllers for the 33% scale Yak 54 UAS in the inner loop, and describing stable nonlinear guidance logic for the outer loop in the lateral and longitudinal planes. In particular, guidance laws developed here will be used in the following chapter in the design of an integrated nonlinear and predictive guidance and control algorithm.

3.2 Inner and Outer Loops Design

3.2.1 Inner Loop

The inner loop is composed of a set of robust H-infinity controllers and a mechanism to perform gain scheduling based on dynamic conditions. Designing a set of linear time invariant controllers requires linear time invariant UAS representations in each equilibrium point. For this purpose, based on dynamic equations developed in Chapter 1, the UAS's nonlinear state space model is denoted $\dot{\boldsymbol{x}}(t)=\boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{u}(t))$ and $\boldsymbol{y}(t)=\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))$ for propagation and observation equations respectively, with input $\boldsymbol{u}^T(t)=[\delta_t^{cmd},\delta_e^{cmd},\delta_a^{cmd},\delta_r^{cmd}]$, output $\boldsymbol{y}(t)=\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t))=[V_T(t),\theta(t),\phi(t),\beta(t)]^T$, and state $\boldsymbol{x}(t)$. This model is to be linearized at each equilibrium point $(\boldsymbol{x}_e^i,\boldsymbol{u}_e^i,\boldsymbol{y}_e^i)$, indexed by superscript i. Subscript e refers to a particular equilibrium point, or trim point. At each equilibrium point, a truncated Taylor series expansion of the state space equations, up to first order derivatives,

$$\begin{aligned}\dot{\boldsymbol{x}}(t)&\approx\boldsymbol{f}(\boldsymbol{x}_e^i,\boldsymbol{u}_e^i)+\left.\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}}(\boldsymbol{x}(t)-\boldsymbol{x}_e^i)+\left.\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{u}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}}(\boldsymbol{u}(t)-\boldsymbol{u}_e^i)\\ \boldsymbol{y}(t)&\approx\boldsymbol{h}(\boldsymbol{x}_e^i,\boldsymbol{u}_e^i)+\left.\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}}(\boldsymbol{x}(t)-\boldsymbol{x}_e^i)+\left.\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{u}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}}(\boldsymbol{u}(t)-\boldsymbol{u}_e^i)\end{aligned}\tag{61}$$

gives the set of linear representations of the system at equilibria. The following linear time invariant model, denominated $\boldsymbol{G}^i$, is obtained for control design purposes

$$\begin{aligned}\Delta\dot{\boldsymbol{x}}(t)&=\boldsymbol{A}_G^i\Delta\boldsymbol{x}(t)+\boldsymbol{B}_G^i\,\Delta\boldsymbol{u}(t)\\ \Delta\boldsymbol{y}(t)&=\boldsymbol{C}_G^i\,\Delta\boldsymbol{x}(t)+\boldsymbol{D}_G^i\,\Delta\boldsymbol{u}(t)\end{aligned}\tag{62}$$

This is done after canceling out constant terms, defining $\Delta\boldsymbol{x}(t)=\boldsymbol{x}(t)-\boldsymbol{x}_e^i$, and relabeling the following partial derivatives evaluated at each equilibrium:

$$\begin{aligned}\boldsymbol{A}_G^i&=\left.\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}} &\qquad \boldsymbol{B}_G^i&=\left.\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{u}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}}\\ \boldsymbol{C}_G^i&=\left.\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}} &\qquad \boldsymbol{D}_G^i&=\left.\frac{\partial\boldsymbol{h}}{\partial\boldsymbol{u}}\right|_{\substack{\boldsymbol{x}_e^i\\ \boldsymbol{u}_e^i}}\end{aligned}\tag{63}$$

As shown in Figure 23, each linear model $\boldsymbol{G}^i$ is augmented by the addition of the dynamic weighting matrices $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ and $\boldsymbol{W}_3$, becoming plant $\boldsymbol{P}^i$, with inputs $\boldsymbol{u}_1\in R^{m1},\boldsymbol{u}_2\in R^{m2}$ and outputs $\boldsymbol{y}_1\in R^{p1},\boldsymbol{y}_2\in R^{p2}$. Appropriately shaping particular closed loop sensitivity functions, by using the singular values of these frequency-dependent weighting matrices, otherwise conflicting performance objectives, namely low frequency reference tracking and disturbance rejection and high frequency noise cancellation, are harmonized.

Plant $\boldsymbol{P}^i$ is then expressed as

$$\begin{aligned}\dot{\boldsymbol{x}}_P &= \boldsymbol{A}^i\boldsymbol{x}_P + \boldsymbol{B}_1^i\boldsymbol{u}_1 + \boldsymbol{B}_2^i\boldsymbol{u}_2\\ \boldsymbol{y}_1 &= \boldsymbol{C}_1^i\boldsymbol{x}_P + \boldsymbol{D}_{11}^i\boldsymbol{u}_1 + \boldsymbol{D}_{12}^i\boldsymbol{u}_2\\ \boldsymbol{y}_2 &= \boldsymbol{C}_2^i\boldsymbol{x}_P + \boldsymbol{D}_{21}^i\boldsymbol{u}_1 + \boldsymbol{D}_{22}^i\boldsymbol{u}_2\end{aligned} \tag{64}$$

where superscript i continues to be the index among the set of augmented plants $\boldsymbol{P}^i$ corresponding to each linearized plant $\boldsymbol{G}^i$. Notice that the states $\boldsymbol{x}_P$ of the augmented plant $\boldsymbol{P}^i$ contain both the original states $\Delta\boldsymbol{x}$ of plant $\boldsymbol{G}^i$ and the weighting matrices states.

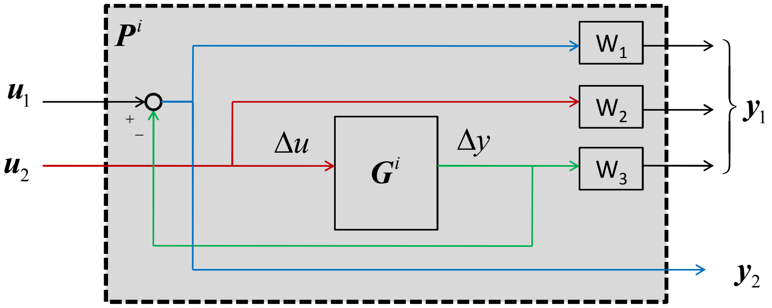

Figure 23. Augmented System.

A condition that greatly facilitates the gain scheduling procedure is to have a homogeneous augmented plant set in terms of structure, states, inputs, and outputs (or equivalently in terms of zeros and poles), and the same controller design technique. Given the aircraft dynamics do not change in nature along the range of equilibrium points, and because weighting matrices and control design technique are kept fixed for all linearization points, the condition is attained.

The chosen control design technique is the H_∞ linear robust controller. Based on Ref. [96] and [97], if the set of assumptions described below are met, then a suboptimal H_∞ control synthesis will provide a controller $\boldsymbol{K}^i$ that internally stabilizes the closed loop system such that

$$\left\|\boldsymbol{T}_{y_1u_1}(s)\right\|_\infty = \underbrace{\sup}_{jw}\bar{\sigma}[\boldsymbol{T}_{y_1u_1}(jw)] \leq \gamma \tag{65}$$

for a given $\gamma > 0$. $\boldsymbol{T}_{y_1 u_1}$ is the closed loop transfer function from exogenous inputs $\boldsymbol{u}_1$ (including references, disturbances, and noises) to controlled output $\boldsymbol{y}_1$ (including weighted error, control, and output signals). In the present work, $\boldsymbol{T}_{y_1 u_1}$ is constructed as follows:

$$\left\| \boldsymbol{T}_{y_1 u_1}(s) \right\|_\infty = \left\| \begin{matrix} \boldsymbol{W}_1(s)\boldsymbol{S}(s) \\ \boldsymbol{W}_2(s)\boldsymbol{R}(s) \\ \boldsymbol{W}_3(s)\boldsymbol{T}(s) \end{matrix} \right\|_\infty \tag{66}$$

as a combination of desired weighted closed loop objectives, $\boldsymbol{S}(s)$, $\boldsymbol{R}(s)$ and $\boldsymbol{T}(s)$, representing sensitivity, output sensitivity, and complementary sensitivity, respectively. Weighting the closed loop functions allows their joint minimization. For an effective low frequency reference tracking and disturbance rejection, weighting matrix $\boldsymbol{W}_1$ is built to have high singular values at low frequency range. For noise cancelation at higher frequencies, $\boldsymbol{W}_3$ is designed to have high singular values in those higher frequencies. The weighting matrix $\boldsymbol{W}_2$ is designed as a constant matrix to penalize the control effort from getting unrealistic values and saturating actuators. Figure 24 shows an example of the singular values of the dynamic weightings chosen for the current requirements.

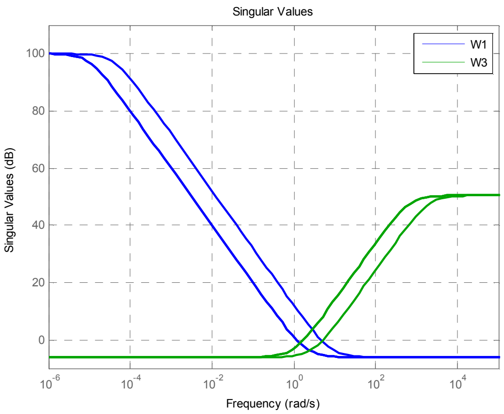

Figure 24. Weighting Matrices W_1 and W_3 Singular Values.

The previous assumptions are

a. $(\boldsymbol{A}^i, \boldsymbol{B}_2^i)$ stabilizable and $(\boldsymbol{C}_2^i, \boldsymbol{A}^i)$ detectable,

b. $rank(\boldsymbol{D}_{12}^i) = m_2$ and $rank(\boldsymbol{D}_{21}^i) = p_2$,

c. $\boldsymbol{D}_{12}^{i} = [0 \quad I_{m2}]^{T}$ and $\boldsymbol{D}_{21}^{i} = [0 \quad I_{p2}]$, and $\boldsymbol{D}_{11}^{i}$ is expressible as

$$\boldsymbol{D}_{11}^{i} = \begin{bmatrix} \underbrace{\boldsymbol{D}_{1111}^{i}}_{(p1-m2)x(m1-p2)} & \underbrace{\boldsymbol{D}_{1112}^{i}}_{(p1-m2)x(p2)} \\ \underbrace{\boldsymbol{D}_{1121}^{i}}_{(m2)x(m1-p2)} & \underbrace{\boldsymbol{D}_{1122}^{i}}_{(m2)x(p2)} \end{bmatrix} \tag{67}$$

d. $\boldsymbol{D}_{22}^{i} = 0$.

e. $rank\begin{bmatrix} \boldsymbol{A}^{i} - jwI & \boldsymbol{B}_{2}^{i} \\ \boldsymbol{C}_{1}^{i} & \boldsymbol{D}_{12}^{i} \end{bmatrix} = n + m_{2}$.

f. $rank\begin{bmatrix} \boldsymbol{A}^{i} - jwI & \boldsymbol{B}_{1}^{i} \\ \boldsymbol{C}_{2}^{i} & \boldsymbol{D}_{21}^{i} \end{bmatrix} = n + p_{2}$.

If assumption c is not achieved by the chosen weighting matrices, it still can be obtained via scaling of $\boldsymbol{u}_{2}$ and $\boldsymbol{y}_{2}$ together with a unitary transformation of $\boldsymbol{u}_{1}$ and $\boldsymbol{y}_{1}$. After previous assumptions are met, there exists some $\gamma^{i} > 0$ for which the following relations hold

$$\begin{aligned} &\max\left(\bar{\sigma}\left[\boldsymbol{D}_{1111}^{i} \quad \boldsymbol{D}_{1112}^{i}\right], \bar{\sigma}\left[\boldsymbol{D}_{1111}^{i}{}^{T} \quad \boldsymbol{D}_{1112}^{i}{}^{T}\right]\right) < \gamma^{i} \\ &\boldsymbol{X}^{i} \geq 0 \quad and \quad \boldsymbol{Y}^{i} \geq 0 \\ &\rho\left(\boldsymbol{X}^{i} \cdot \boldsymbol{Y}^{i}\right) < (\gamma^{i})^{2} \end{aligned} \tag{68}$$

This is equivalent to the existence of an internal stabilizing suboptimal controller $\bar{\boldsymbol{K}}^{i}$

$$\bar{\boldsymbol{K}}^{i} = \begin{bmatrix} \bar{\boldsymbol{A}}^{i} & \bar{\boldsymbol{B}}_{1}^{i} & \bar{\boldsymbol{B}}_{2}^{i} \\ \bar{\boldsymbol{C}}_{1}^{i} & \bar{\boldsymbol{D}}_{11}^{i} & \bar{\boldsymbol{D}}_{12}^{i} \\ \bar{\boldsymbol{C}}_{2}^{i} & \bar{\boldsymbol{D}}_{21}^{i} & 0 \end{bmatrix} \tag{69}$$

whose components are given by

$$\begin{aligned} \bar{\boldsymbol{A}}^{i} &= \boldsymbol{A}^{i} + \boldsymbol{H}^{i}\boldsymbol{C}^{i} + \bar{\boldsymbol{B}}_{2}^{i}(\bar{\boldsymbol{D}}_{12}^{i})^{-1}\bar{\boldsymbol{C}}_{1}^{i} \\ \bar{\boldsymbol{B}}_{1}^{i} &= -\boldsymbol{H}_{2}^{i} + \bar{\boldsymbol{B}}_{2}^{i}(\bar{\boldsymbol{D}}_{12}^{i})^{-1}\bar{\boldsymbol{D}}_{11}^{i} \\ \bar{\boldsymbol{B}}_{2}^{i} &= (\boldsymbol{B}_{2}^{i} + \boldsymbol{H}_{12}^{i})\bar{\boldsymbol{D}}_{12}^{i} \\ \bar{\boldsymbol{C}}_{1}^{i} &= \boldsymbol{F}_{2}^{i}\boldsymbol{Z}^{i} + \bar{\boldsymbol{D}}_{11}^{i}(\bar{\boldsymbol{D}}_{21}^{i})^{-1}\bar{\boldsymbol{C}}_{2}^{i} \\ \bar{\boldsymbol{C}}_{2}^{i} &= -\bar{\boldsymbol{D}}_{21}^{i}(\boldsymbol{C}_{2}^{i} + \boldsymbol{F}_{12}^{i})\boldsymbol{Z}^{i} \\ \bar{\boldsymbol{D}}_{11}^{i} &= -\boldsymbol{D}_{1121}^{i}\boldsymbol{D}_{1111}^{i}{}^{T}((\gamma^{i})^{2}\boldsymbol{I} - \boldsymbol{D}_{1111}^{i}\boldsymbol{D}_{1111}^{i}{}^{T})^{-1}\boldsymbol{D}_{1112}^{i} - \boldsymbol{D}_{1122}^{i} \end{aligned} \tag{70}$$

and $\bar{\boldsymbol{D}}_{12}^i$ and $\bar{\boldsymbol{D}}_{21}^i$ are any matrices satisfying the relation $\bar{\boldsymbol{D}}_{12}^i \bar{\boldsymbol{D}}_{12}^{i\,T} = \boldsymbol{I} - \bar{\boldsymbol{D}}_{1121}^i \left((\gamma^i)^2 I - \bar{\boldsymbol{D}}_{1111}^{i\,T} \bar{\boldsymbol{D}}_{1111}^i\right)^{-1} \bar{\boldsymbol{D}}_{1121}^{i\,T}$ and the relation $\bar{\boldsymbol{D}}_{21}^i \bar{\boldsymbol{D}}_{21}^{i\,T} = \boldsymbol{I} - \bar{\boldsymbol{D}}_{1112}^i \left((\gamma^i)^2 I - \bar{\boldsymbol{D}}_{1111}^{i\,T} \bar{\boldsymbol{D}}_{1111}^i\right)^{-1} \bar{\boldsymbol{D}}_{1112}^{i\,T}$, respectively. The feedback $\boldsymbol{F}^i$ and state feedback $\boldsymbol{H}^i$ matrices are defined

$$\begin{aligned}
\boldsymbol{F}^i &= \begin{bmatrix} \left.\boldsymbol{F}_{11}^i\right\}_{(m1-p2)x(n)} \\ \left.\boldsymbol{F}_{12}^i\right\}_{(p2)x(n)} \\ \left.\boldsymbol{F}_{2}^i\right\}_{(m2)x(n)} \end{bmatrix} = -(\boldsymbol{R}_a^i)^{-1}\left[\boldsymbol{D}_{1a}^{i\,T}\boldsymbol{C}_{1i}^i + \boldsymbol{B}^{i\,T}\boldsymbol{X}^i\right] \\
\boldsymbol{H}^i &= \begin{bmatrix} \underbrace{\boldsymbol{H}_{11}^i}_{(n)x(p1-m2)} & \underbrace{\boldsymbol{H}_{12}^i}_{(n)x(m2)} & \underbrace{\boldsymbol{H}_{2}^i}_{(n)x(p2)} \end{bmatrix} = \\
&\qquad -\left[\boldsymbol{B}^i\boldsymbol{D}_{1b}^{i\,T} + \boldsymbol{Y}^i\boldsymbol{C}^{i\,T}\right](\boldsymbol{R}_b^i)^{-1}
\end{aligned} \tag{71}$$

where

$$\boldsymbol{B}^i = \begin{bmatrix} \boldsymbol{B}_1^i & \boldsymbol{B}_2^i \end{bmatrix}, \quad \boldsymbol{C}^i = \begin{bmatrix} \boldsymbol{C}_1^i \\ \boldsymbol{C}_2^i \end{bmatrix} \tag{72}$$

$$\boldsymbol{Z}^i = (\boldsymbol{I} - (\gamma^i)^{-2}\boldsymbol{Y}^i \cdot \boldsymbol{X}^i)^{-1} \tag{73}$$

$$\boldsymbol{D}_{1a}^i = \begin{bmatrix} \boldsymbol{D}_{11}^i & \boldsymbol{D}_{12}^i \end{bmatrix}, \quad \boldsymbol{D}_{1b}^i = \begin{bmatrix} \boldsymbol{D}_{11}^i \\ \boldsymbol{D}_{21}^i \end{bmatrix} \tag{74}$$

$$\begin{aligned}
\boldsymbol{R}_a^i &= \boldsymbol{D}_{1a}^{i\,T}\boldsymbol{D}_{1a}^i - \begin{bmatrix} (\gamma^i)^2\boldsymbol{I}_{m1} & 0 \\ 0 & 0 \end{bmatrix} \\
\boldsymbol{R}_b^i &= \boldsymbol{D}_{1b}^i\boldsymbol{D}_{1b}^{i\,T} - \begin{bmatrix} (\gamma^i)^2\boldsymbol{I}_{p1} & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{75}$$

Matrices $\boldsymbol{X}^i$ and $\boldsymbol{Y}^i$ are the solutions of the following Riccati equations

$$\begin{aligned}
&(\boldsymbol{A}^i - \boldsymbol{B}^i(\boldsymbol{R}_a^i)^{-1}\boldsymbol{D}_{1a}^{i\,T}\boldsymbol{C}_1^i)^T\boldsymbol{X}^i + \boldsymbol{X}^i(\boldsymbol{A}^i - \boldsymbol{B}^i(\boldsymbol{R}_a^i)^{-1}\boldsymbol{D}_{1a}^{i\,T}\boldsymbol{C}_1^i) - \\
&\boldsymbol{X}^i(\boldsymbol{B}^i(\boldsymbol{R}_a^i)^{-1}\boldsymbol{B}^{i\,T})\boldsymbol{X}^i + \boldsymbol{C}_1^{i\,T}(\boldsymbol{I} - \boldsymbol{D}_{1a}^i(\boldsymbol{R}_a^i)^{-1}\boldsymbol{D}_{1a}^{i\,T})\boldsymbol{C}_1^i = 0 \\
\\
&(\boldsymbol{A}^i - \boldsymbol{B}_1^i\boldsymbol{D}_{1b}^{i\,T}(\boldsymbol{R}_b^i)^{-1}\boldsymbol{C}^i)\boldsymbol{Y}^i + \boldsymbol{Y}^i(\boldsymbol{A}^i - \boldsymbol{B}_1^i\boldsymbol{D}_{1b}^{i\,T}(\boldsymbol{R}_b^i)^{-1}\boldsymbol{C}^i)^T - \\
&\boldsymbol{Y}^i(\boldsymbol{C}^{i\,T}(\boldsymbol{R}_b^i)^{-1}\boldsymbol{C}^i)\boldsymbol{Y}^i + \boldsymbol{B}_1^i(\boldsymbol{I} - \boldsymbol{D}_{1b}^{i\,T}(\boldsymbol{R}_b^i)^{-1}\boldsymbol{D}_{1b}^i)\boldsymbol{B}_1^{i\,T} = 0
\end{aligned} \tag{76}$$

The reason for choosing a robust controller design is the inevitable existence of uncertainty in the description of the aircraft. Depending on the available model mismatch knowledge, whose complete depiction is undefined, uncertainty can have several approximate formulations. A common way to characterize model uncertainty is through parameter uncertainty. As will be shown below, because the current aircraft modeling has two different sources of stability and control derivatives, any individual derivative mismatch defines a range, i.e. a parameter uncertainty formulation.

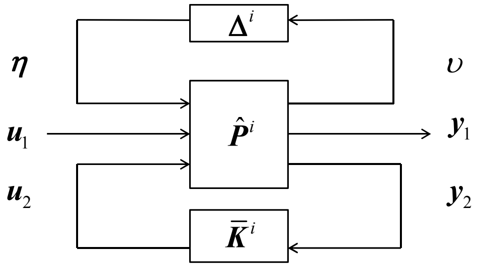

Figure 25. General LFT Interconnection.

In order to facilitate the characterization of parameter uncertainty, a linear fractional transformation (LFT) is employed. Uncertainties are extracted and packed into Δ^i following the procedure detailed in Ref. [97] shown in Figure 25, leaving a deterministic plant $\hat{\boldsymbol{P}}^i$, as opposed to the previous uncertain plant $\boldsymbol{P}^i$. The LFT interconnection is defined as follows

$$\begin{bmatrix} \upsilon \\ y_1 \\ y_2 \end{bmatrix} = \hat{\boldsymbol{P}}^i \begin{bmatrix} \eta \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{P}}^i_{11}(s) & \hat{\boldsymbol{P}}^i_{12}(s) & \hat{\boldsymbol{P}}^i_{13}(s) \\ \hat{\boldsymbol{P}}^i_{21}(s) & \hat{\boldsymbol{P}}^i_{22}(s) & \hat{\boldsymbol{P}}^i_{23}(s) \\ \hat{\boldsymbol{P}}^i_{31}(s) & \hat{\boldsymbol{P}}^i_{32}(s) & \hat{\boldsymbol{P}}^i_{33}(s) \end{bmatrix} \begin{bmatrix} \eta \\ u_1 \\ u_2 \end{bmatrix} \tag{77}$$
$$\eta = \Delta^i \upsilon \qquad u_2 = \bar{\boldsymbol{K}}^i y_2$$

or as $y_1 = \boldsymbol{F}_u(\boldsymbol{F}_l(\hat{\boldsymbol{P}}^i, \bar{\boldsymbol{K}}^i), \Delta^i) u_1 = \boldsymbol{F}_l(\boldsymbol{F}_u(\hat{\boldsymbol{P}}^i, \Delta^i), \bar{\boldsymbol{K}}^i) u_1$, where $\boldsymbol{F}_l$ corresponds to the lower LFT and $\boldsymbol{F}_u$ to the upper LFT.

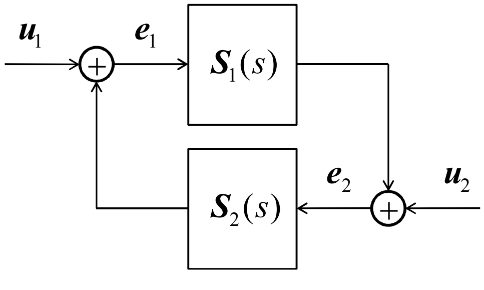

Figure 26. Small Gain Theorem Interconnection.

By the following Small Gain Theorem, assuming $\|\Delta\|_\infty \leq 1$, the interconnected system shown in Figure 26 is internally stable if and only if $\left\|\boldsymbol{F}_l(\hat{\boldsymbol{P}}^i, \bar{\boldsymbol{K}}^i)\right\|_\infty < 1$.

Theorem 2: Small Gain Theorem Under the assumption that MIMO linear systems $S_1(s)$ and $S_2(s)$ are stable, the closed loop transfer function matrix from $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ to $(\boldsymbol{e}_1, \boldsymbol{e}_2)$, as shown in Figure 26, is stable if the small gain condition

$$\|S_1(jw)\|_\infty \|S_2(jw)\|_\infty < 1 \tag{78}$$

is satisfied.

Demonstration Omitted, see Ref. [98].

For each uncertain parameter, an uncertainty range is built considering the highest and lowest values. These extreme values are used to calculate a central value and a half range value. With this information, a normalized uncertainty range is constructed and pulled out of the plant. Figure 27 shows the arrangement for a multiplicative parameter p_{un}. Rearranging p_{un} in terms of its central value p_{un}^C and half range value p_{un}^H leaves its uncertainty block with $\left\|\Delta_j^i\right\|_\infty \leq 1$. From pulling out of the system $\boldsymbol{P}^i$ each uncertainty block, an integrated block $\boldsymbol{\Delta}^i$, from $\boldsymbol{\upsilon} = [\upsilon_1, \cdots, \upsilon_j, \cdots]^T$ to $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_j, \cdots]^T$, is assembled as shown in Figure 25.

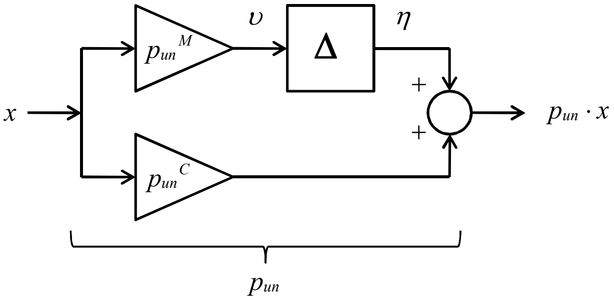

Figure 27. Uncertainty Normalization.

3.2.1.1 Criterion for the Selection of Weighting Matrices

After normalizing by γ the following conditions can be defined from Equations (65) and (66)

$$\begin{aligned} \left\| \boldsymbol{W}_1(s)\boldsymbol{S}(s) \right\|_\infty &\le 1 \\ \left\| \boldsymbol{W}_2(s)\boldsymbol{R}(s) \right\|_\infty &\le 1 \\ \left\| \boldsymbol{W}_3(s)\boldsymbol{T}(s) \right\|_\infty &\le 1 \end{aligned} \tag{79}$$

These conditions can be restated as:

$$\begin{aligned} \left\| \boldsymbol{W}_1^{-1}(s) \right\|_\infty &\ge \left\| \boldsymbol{S}(s) \right\|_\infty \\ \left\| \boldsymbol{W}_2^{-1}(s) \right\|_\infty &\ge \left\| \boldsymbol{R}(s) \right\|_\infty \\ \left\| \boldsymbol{W}_3^{-1}(s) \right\|_\infty &\ge \left\| \boldsymbol{T}(s) \right\|_\infty \end{aligned} \tag{80}$$

These relations establish a direct connection between the desired $\boldsymbol{S}(s)$, $\boldsymbol{R}(s)$ and $\boldsymbol{T}(s)$ closed loop functions and the shape of the weighting matrices' singular values. For an effective low frequency reference tracking and disturbance rejection, and high frequency additive noise and/or bias cancelation, the following frequency domain specifications can be defined:

a. Steady state error less than A_S

b. Closed-loop bandwidth close to w

c. Amplification of high frequency additive perturbation, as noise and/or biases, less than A_T

These specifications can be translated into the following weighing matrices, where the term *diag* denotes a diagonal matrix:

$$\begin{aligned} \boldsymbol{W}_1(s) &= diag\left\{ \frac{\frac{s}{M_S} + w}{s + A_S \cdot w} \right\} \\ \boldsymbol{W}_3(s) &= diag\left\{ \frac{s + \frac{w}{M_T}}{A_T \cdot s + w} \right\} \end{aligned} \tag{81}$$

For large values of $1/A_S$ and $1/A_T$, M_S and M_T are not significant and can be set to one. Closed loop bandwidth w is commonly bounded from above by the frequency content of the additive perturbations.

3.2.2 Outer Loop

The ultimate task of the inner loop controller, in this case robust, is to drive the aircraft by commanding the actuators to follow external commands generated by the outer loop. It, essentially, performs this task by minimizing the error between the external commanded states and the measured current states of the aircraft in a robust and stable way. For the actual design, this minimization is done over the errors in airspeed, pitch angle, bank angle and side slip angle, represented as:

$$\begin{aligned} e_{V_T}(t) &= {V_T}^{cmd}(t) - V_T(t) \\ e_\theta(t) &= \theta^{cmd}(t) - \theta(t) \\ e_\phi(t) &= \phi^{cmd}(t) - \phi(t) \\ e_\beta(t) &= \beta^{cmd}(t) - \beta(t) \end{aligned} \tag{82}$$

The task of the outer loop is to guide the aircraft to follow a trajectory path composed of straight segments $\{\boldsymbol{ab}, \boldsymbol{bc}, \boldsymbol{cd}, \cdots\}$ subtended between waypoints $\{\vec{\boldsymbol{a}}, \vec{\boldsymbol{b}}, \vec{\boldsymbol{c}}, \cdots\}$. Inertial speed is also assigned to each segment. To this end, it has two main subtasks: one is to generate the attitude commands ϕ^{cmd} and θ^{cmd} based on the current inertial position p_N, p_E, p_H, inertial speed v_N, v_E, v_H, and the relative location of the aircraft with respect to the segment under track. The second subtask is to generate the airspeed command V_T^{cmd} based on the segment inertial speed V_{ab} after considering the existing wind. The sideslip command β^{cmd} does not require any specific logic as it is set to zero during the entire flight.

The logic required for the generation of attitude commands ϕ^{cmd} and θ^{cmd} to keep the aircraft on track is based on a nonlinear guidance logic developed in Ref. [33] for lateral guidance, which originally generates ϕ^{cmd}.

The present research not only extends the logic to calculate θ^{cmd}, but also adds new features for both attitude angle commands, ensuring stability for any initial location and any initial attitude of the aircraft. Stability, in the sense of Lyapunov, is demonstrated for the extended logic.

The aircraft model, previously defined in Equation (43) and related equations, is augmented by embedding guidance laws and the trajectory information assumed to be known. Strictly speaking, from the predefined trajectory, which is composed of straight segments subtended between consecutive waypoints, only the current or active segment is required, allowing online trajectory adjustments. The main variables to be generated and used by the outer loop, as well as by the predictive controller formulation and operation, as will be seen in the next chapter, are two angular errors (lateral and longitudinal referred to the segment) to be defined as new variables of the model. Part of the task of the inner loop controller will be to minimize these angular errors, generating a robust and feasible control, keeping a tight trajectory following. These angular errors are not originally defined in the aircraft model so their formulation, as a result of the current trajectory segment data and the guidance logic, is incorporated as new variables. This guidance logic, which was originally proposed for lateral nonlinear guidance in Ref. [33], is extended to the longitudinal plane and modified to extend its stability.

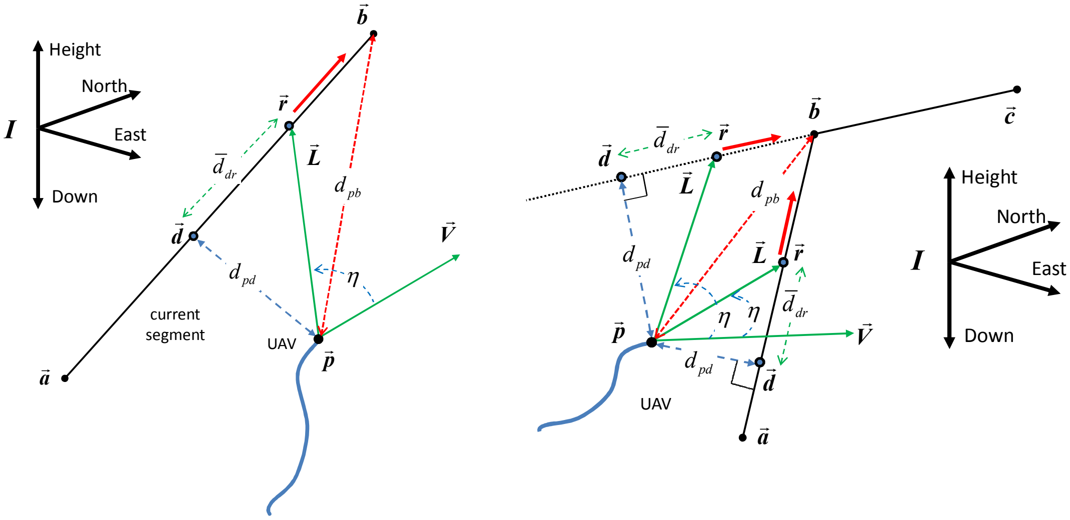

Figure 28. Guidance Law Concept (left) - Switching Logic (right).

The trajectory is composed of straight segments delimited by an ordered list of waypoints that includes their inertial location and associated inertial speed. The principle behind the trajectory following concept is to keep the UAS close to the current segment indirectly by minimizing lateral and longitudinal angular errors. A dedicated logic is incorporated to determine which segment is under tracking and when to switch to the next one in advance to avoid excessive overshoots and trajectory departures. This logic assumes the trajectory following starts from the first segment (subtended between the first two waypoints). Once the UAS has reached a predefined horizontal distance to the end of the segment (using horizontal components of d_{pb} in Figure 28 (left)), it will automatically change to the next segment (subtended between the second and third waypoints); see Figure 28 (right). This logic uses only two consecutive waypoints, allowing for the possibility of modifying future waypoints online as needed (e.g. for collision-avoidance purposes).

As shown in Figure 28 (left), the guidance logic calculates the angle η between the inertial velocity vector $\vec{\boldsymbol{V}}$ (to be distinguished from the component of body velocity V) and the vector $\vec{\boldsymbol{L}}$ subtended from the current position $\vec{\boldsymbol{p}}$ of the UAS and an imaginary point $\vec{\boldsymbol{r}}$, located in the segment between $\vec{\boldsymbol{a}}$ to $\vec{\boldsymbol{b}}$. Point $\vec{\boldsymbol{r}}$ is positioned at fixed distance $\bar{d}_{dr}$ toward the end of the segment from the closest point $\vec{\boldsymbol{d}}$ with respect to the UAS. As $\vec{\boldsymbol{d}}$ moves, $\vec{\boldsymbol{r}}$ moves in the same direction (see solid red arrow). From the geometry it is evident that a small angle η is equivalent to a small cross track error d_{pd} (minimum distance between the UAS and the segment) and the aircraft moving along the segment, or when the aircraft is asymptotically approaching the segment when situated off-track by moving toward point $\vec{\boldsymbol{r}}$. This error angle η is decomposed into two angles, lateral η_{LAT} and longitudinal η_{LON} angles, as shown in Figure 29 (left), referred to the current segment.

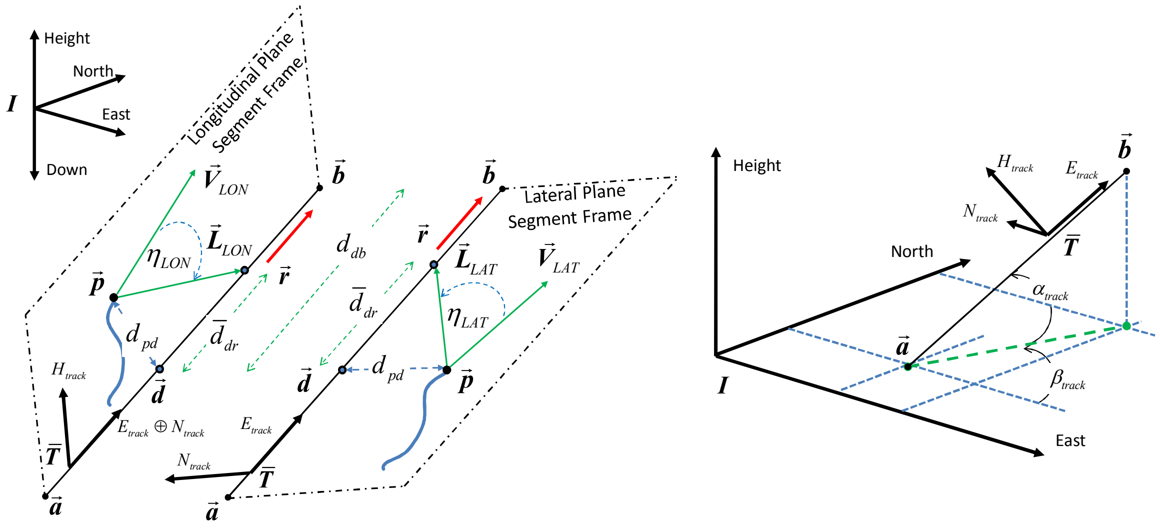

Figure 29. Longitudinal and Lateral Guidance Law Geometry (left) - Segment Rotations (right).

In order to reformulate both angles with respect to the lateral and longitudinal planes of segment $\boldsymbol{ab}$, both vectors $\vec{\boldsymbol{V}} = [\dot{p}_N, \dot{p}_E, \dot{p}_H]^T$ and $\vec{\boldsymbol{L}} = \vec{\boldsymbol{r}}(t) - \vec{\boldsymbol{p}}(t) = [r_N(t) - p_N(t), r_E(t) - p_E(t), r_H(t) - p_H(t)]^T$ must be rotated from the inertial frame $\bar{\boldsymbol{I}}$ to a coordinate frame referred to the trajectory segment, labeled $\bar{\boldsymbol{T}}$, as shown in Figure 29 (right). For this purpose, point $\vec{\boldsymbol{d}}$ is parametrically determined as

$$\vec{\boldsymbol{d}}(t) = \begin{bmatrix} d_N(t) \\ d_E(t) \\ d_H(t) \end{bmatrix} = \vec{\boldsymbol{a}}(t) + q_d(t)\left[\vec{\boldsymbol{b}} - \vec{\boldsymbol{a}}\right] \tag{83}$$

$$q_d(t) = \frac{\left[\vec{\boldsymbol{p}}(t) - \vec{\boldsymbol{a}}\right] \cdot \left[\vec{\boldsymbol{b}} - \vec{\boldsymbol{a}}\right]}{{d_{ab}}^2} \tag{84}$$

where d_{ab} represents the Euclidean distance between points $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$. The parameter $0 < q_d(t) < 1$ then determines the location of point $\vec{\boldsymbol{d}}$ between $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$, perpendicular to $\vec{\boldsymbol{p}}$.

Point $\vec{\boldsymbol{r}}$ is located at the fixed distance $\bar{d}_{dr}$ toward the end of the segment (waypoint $\vec{\boldsymbol{b}}$) and can be parametrically defined as

$$\vec{\boldsymbol{r}}(t) = \begin{bmatrix} r_N(t) \\ r_E(t) \\ r_H(t) \end{bmatrix} = \vec{\boldsymbol{d}}(t) + m_r(t)\left[\vec{\boldsymbol{b}} - \vec{\boldsymbol{d}}(t)\right] \tag{85}$$

$$m_r(t) = \frac{\bar{d}_{dr}}{d_{db}(t)} \tag{86}$$

with $d_{db}(t)$ being the distance from point $\vec{\boldsymbol{d}}$ to waypoint $\vec{\boldsymbol{b}}$. Given the characterization of the guidance law, point $\boldsymbol{r}$ is acting as an unreachable pseudo-target located in the trajectory.

Given the location of its waypoints $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$ with respect to the inertial frame $\bar{\boldsymbol{I}}$, each segment is characterized by two rotation angles, α_{track} and β_{track}, as shown in Figure 29 (right). In order to get the angular errors η_{LAT} and η_{LON}, both vectors $\vec{\boldsymbol{V}}$ and $\vec{\boldsymbol{L}}$ are converted from the inertial frame $\bar{\boldsymbol{I}}$ to the segment coordinate frame $\bar{\boldsymbol{T}}$ using the following direction cosine matrix operation, $\boldsymbol{C}_I^T$

$$\boldsymbol{C}_I^T = \begin{bmatrix} \cos\alpha_{track}\cos\beta_{track} & \sin\beta_{track} & \sin\alpha_{track}\cos\beta_{track} \\ -\cos\alpha_{track}\sin\beta_{track} & \cos\beta_{track} & -\sin\alpha_{track}\sin\beta_{track} \\ -\sin\alpha_{track} & 0 & \cos\alpha_{track} \end{bmatrix} \tag{87}$$

Inertial vectors $\vec{V}$ and $\vec{L}$, referred to the segment coordinate frame $\overline{T}$, become

$$\vec{V}_{track}(t) = C_I^T \cdot \vec{V} = C_I^T \cdot \begin{bmatrix} \dot{p}_N(t) \\ \dot{p}_E(t) \\ \dot{p}_H(t) \end{bmatrix} = \begin{bmatrix} \dot{p}_{N\ track}(t) \\ \dot{p}_{E\ track}(t) \\ \dot{p}_{H\ track}(t) \end{bmatrix} \tag{88}$$

$$\vec{L}_{track}(t) = C_I^T \cdot \vec{L}(t) = C_I^T \cdot [\vec{r}(t) - \vec{p}(t)] = C_I^T \cdot \begin{bmatrix} [r_N(t) - p_N(t)] \\ [r_E(t) - p_E(t)] \\ [r_H(t) - p_H(t)] \end{bmatrix} = \begin{bmatrix} [r_N(t) - p_N(t)]_{track} \\ [r_E(t) - p_E(t)]_{track} \\ [r_H(t) - p_H(t)]_{track} \end{bmatrix} \tag{89}$$

From Figure 30, η_{LAT} and η_{LON} are defined as follows, where the operator $x \oplus y = \sqrt{x^2 + y^2}$ is used for shortness,

$$\eta_{LAT}(t) = \arctan\left(\frac{[r_N(t) - p_N(t)]_{track}}{[r_E(t) - p_E(t)]_{track}} \right) - \arctan\left(\frac{\dot{p}_{N\ track}}{\dot{p}_{E\ track}} \right) \tag{90}$$

$$\eta_{LON}(t) = \arctan\left(\frac{\dot{p}_{H\ track}}{\dot{p}_{E\ track} \oplus \dot{p}_{N\ track}} \right) - \arctan\left(\frac{[r_H(t) - p_H(t)]_{track}}{[r_E(t) - p_E(t)]_{track} \oplus [r_N(t) - p_N(t)]_{track}} \right) \tag{91}$$

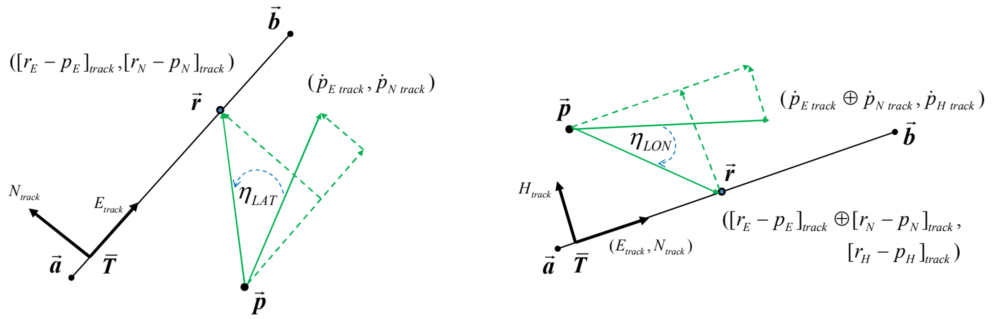
$\vec{b}$
$([r_E - p_E]_{track}, [r_N - p_N]_{track})$
$(\dot{p}_{E\ track}, \dot{p}_{N\ track})$
$\vec{r}$
η_{LAT}
N_{track}
E_{track}
$\vec{a}$
$\overline{T}$
$\vec{p}$
$(\dot{p}_{E\ track} \oplus \dot{p}_{N\ track}, \dot{p}_{H\ track})$
$\vec{p}$
η_{LON}
$\vec{b}$
$\vec{r}$
H_{track}
$([r_E - p_E]_{track} \oplus [r_N - p_N]_{track},$
$[r_H - p_H]_{track})$
(E_{track}, N_{track})
$\vec{a}$
$\overline{T}$

Figure 30. **Lateral Angle Calculation (left) - Longitudinal Angle Calculation (right).**

The angular errors are required to vary within $-\pi < \eta_{LAT,LON} < \pi$. A simple logic unwraps the lateral angle, so if it surpasses one of the limits, it is shifted pertinently in order to stay within the range. On the other hand, the longitudinal angle must be constrained to the limits to avoid inverted flight. This restriction can be included in the numerical search as a state constraint to assure that it is met. Also, no steep trajectory segments are advised.

A further conversion is applied to the angular errors η_{LAT} and η_{LON} to obtain attitude angle commands ϕ_{CMD} and θ_{CMD}, commanding variables which will be explicit elements in the augmentation of the model formulation toward an integrated representation that performs both guidance and control tasks. These angles are translated into attitude commands through the following nonlinear relations

$$\begin{aligned} \phi_{cmd} &= \tan^{-1}\{k_{a_{Lat}} \cdot \eta_{Lat} \cdot (k_{p_{Lat}} + \frac{k_{i_{Lat}}}{s})\} \\ \theta_{cmd} &= \tan^{-1}\{k_{a_{Lon}} \cdot \eta_{Lon} \cdot (k_{p_{Lon}} + \frac{k_{i_{Lon}}}{s})\} \end{aligned} \tag{92}$$

where gains k_p and k_i are convenient gains resembling PI controllers. The integral action has been added to force the steady state angle error to be zero. The arctangent function acts as a smooth limiter, avoiding prohibited growth of the commands. The adaptive gains $k_{a_{Lat}} = 2|\vec{\boldsymbol{V}}_{LAT}|^2 / |\vec{\boldsymbol{L}}_{LAT}|$ and $k_{a_{Lon}} = 2|\vec{\boldsymbol{V}}_{LON}|^2 / |\vec{\boldsymbol{L}}_{LON}|$ are used to adequately neutralize changes in the aircraft's speed due to external disturbances, such as wind gusts (see Ref. [33] for more details). Given the nature of guidance logic in Equations (92), errors will be zero only when both vectors are aligned and the UAS is flying with zero cross distance or when it is pointing toward the trajectory, with no external disturbances.

A tight trajectory following is achieved when the aircraft is able to follow these attitude commands; when the errors $e_\theta(t) = \theta_{cmd}(t) - \theta(t)$ and $e_\phi(t) = \phi_{cmd}(t) - \phi(t)$ are small. These objectives are supplemented with the minimization of the airspeed state error $e_{V_T}(t) = V_{Tcmd}(t) - V_T(t)$ and sideslip state error $e_\beta(t) = \beta_{cmd}(t) - \beta(t)$. By correcting the segment inertial speed V_{ab} based on the current wind, $V_{Tcmd}(t)$ is calculated. Also, $\beta_{cmd}(t) \equiv 0$ as defined before. These four new variables, to be understood as the new output of the model, will be the input to the inner loop controller and will be defined as one of the objectives of the NMPC minimization in the next chapter.

Stability in the Sense of Lyapunov

In order to test the stability of the guidance law, a state space representation of the lateral dynamics with $\alpha_{track} = 0$ is developed in terms of the variables $\eta_{Lat}(t)$ and $d_{pd}(t)$, the later being the shortest distance from the UAS to the trajectory. As shown in Figure 31 (left), relations are defined among guidance variables. The dynamic of the variable $d_{pd}(t)$ is described as follows

$$\dot{d}_{pd}(t) = |\vec{V}_{Lat}(t)|\sin(\eta_{Lat}(t) - \eta^{c}_{Lat}(t)) \tag{93}$$

where positive distance is as shown, and $\eta^{c}_{Lat}(t)$ is defined as $\eta^{c}_{Lat}(t) = \tan^{-1}\left(d_{pd}(t)/\bar{d}_{dr}\right)$. As a sign convention, angles measured counterclockwise are taken to be positive. For the dynamics of $\eta_{Lat}(t)$ the centripetal acceleration $a_{cent}(t)$, shown in Figure 31 (left), is defined as

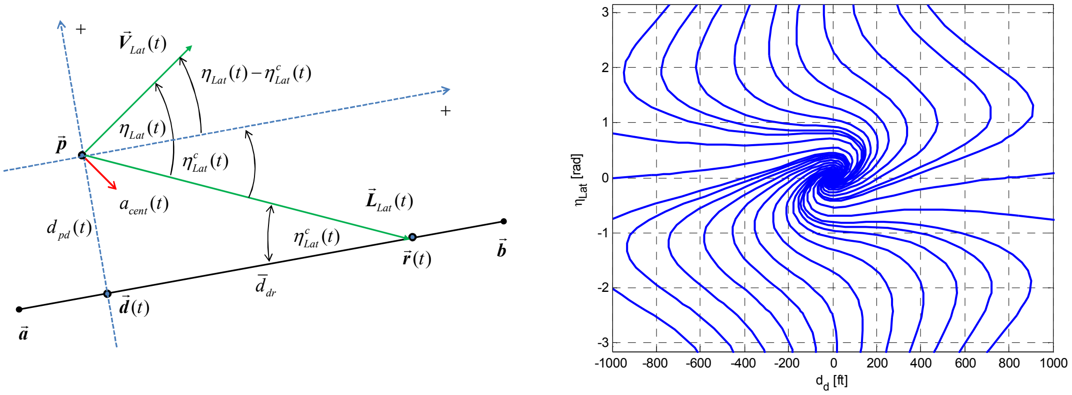

Figure 31. Lateral guidance Geometry (left) - Phase Portrait for Lateral guidance (right).

$$\frac{d}{dt}(\eta_{Lat}(t) - \eta^{c}_{Lat}(t)) = -\frac{a_{cent}(t)}{|\vec{V}_{Lat}(t)|} \tag{94}$$

By assuming that the inner controller accomplishes a close tracking of ϕ_{cmd}, and assuming $a_{cent} = g\tan\phi$, which is valid for coordinated turns, the following dynamics holds

$$\dot{\eta}_{Lat}(t) = \frac{-|\vec{V}_{Lat}(t)|\eta_{Lat}(t)}{\sqrt{d^{2}_{pd}(t) + \bar{d}^{2}_{dr}}} + \frac{|\vec{V}_{Lat}(t)|\bar{d}_{dr}}{(d^{2}_{pd}(t) + \bar{d}^{2}_{dr})}\sin(\eta_{Lat}(t) - \tan^{-1}\frac{d_{pd}(t)}{\bar{d}_{dr}} \tag{95}$$

where the speed is assumed to be constant, i.e. $|\vec{V}_{Lat}(t)| = 120\,[ft/\sec] \approx 70\,knots$, $k_{iLat} = 0$, $k_{pLat} = 1/2g$, and $\bar{d}_{dr} = 300\,[ft]$. From the dynamics defined in Equations (93) and (95), an autonomous nonlinear state space system is assembled that is useful in evaluating the guidance logic stability. Using a numerical and graphical solution, these equations are propagated in time for different initial conditions. Figure 31 (right) shows the phase portrait of the autonomous system. The lateral guidance logic is asymptotically stable in the sense of Lyapunov for any initial distance d_{pd} and $-\pi > \eta_{Lat} > \pi$, as all trajectories converge to the origin.

3.3 Guidance, Navigation, and Control Testing

In this section, the preceding guidance, navigation, and control developments are tested and compared against previous 33% scale Yak 54 UAS flight test data. The aircraft is defined by $S = 10.8997\,[ft^2]$, $b = 7.92\,[ft]$, $\bar{c} = 1.45\,[ft]$, $m = 0.89\,[lbm]$, $\rho = 0.002295\,[slug/ft^3]$, $J_{xx} = 1.0886\,[slug \cdot ft^2]$, $J_{yy} = 2.1068\,[slug \cdot ft^2]$, $J_{zz} = 3.0382\,[slug \cdot ft^2]$ and $J_{xz} = 0.05\,[slug \cdot ft^2]$.

To complete the UAS modeling from Chapter 1, stability and control derivatives from Ref. [99] are employed. These parameters were calculated using the software Advanced Aircraft Analysis (AAA) developed by Design, Analysis, and Research Corporation (DARcorporation, Ref. [100]) for a straight and level flight at a constant altitude and at a constant speed of 70 [knots]. Table II summarizes the stability and control derivatives of the 1/3 scale Yak54.

Table II. AAA Stability and Control Derivatives for 33% Yak 54 at 70 Knots

Parameter	Value	Units	Parameter	Value	Units
C_{D0}	0.0513	[-]	$C_{y\beta}$	-0.3602	[1/rad]
$C_{D\alpha}$	0.0863	[1/rad]	$C_{y\dot{\beta}}$	0	[1/rad]
$C_{D\delta e}$	0	[1/rad]	C_{yp}	0.0085	[1/rad]
$C_{D\dot{\alpha}}$	0	[1/rad]	C_{yr}	0.2507	[1/rad]
C_{Dq}	0	[1/rad]	$C_{y\delta a}$	0	[1/rad]
C_{Du}	0.0011	[1/rad]	$C_{y\delta r}$	0.1929	[1/rad]
C_{L0}	0	[-]	$C_{l\beta}$	-0.0266	[1/rad]
$C_{L\alpha}$	4.5465	[1/rad]	$C_{l\dot{\beta}}$	0	[1/rad]
$C_{L\delta e}$	0.3792	[1/rad]	C_{lp}	-0.3819	[1/rad]
$C_{L\dot{\alpha}}$	1.8918	[1/rad]	C_{lr}	0.0514	[1/rad]
C_{Lq}	5.5046	[1/rad]	$C_{l\delta a}$	0.3490	[1/rad]
C_{Lu}	0.0017	[1/rad]	$C_{l\delta r}$	0.0154	[1/rad]
C_{m0}	0.0020	[-]	$C_{n\beta}$	0.1022	[1/rad]
$C_{m\alpha}$	-0.3937	[1/rad]	$C_{n\dot{\beta}}$	0	[1/rad]
$C_{m\delta e}$	-0.8778	[1/rad]	C_{np}	-0.0173	[1/rad]
$C_{m\dot{\alpha}}$	-4.3787	[1/rad]	C_{nr}	-0.1270	[1/rad]
C_{mq}	-8.0532	[1/rad]	$C_{n\delta a}$	-0.0088	[1/rad]
C_{mu}	0.0002	[1/rad]	$C_{n\delta r}$	-0.0996	[1/rad]

The AAA is an aircraft design program that has a built-in aerodynamic database for different types of aircraft models, based on the theory developed in Ref. [4]. The AAA modeling procedure is based on the chosen trim condition and the aircraft geometric parameters.

The work also employed a vortex lattice model (VLM) as a second method for estimating some of the stability and control derivatives. For uncertainty analysis purposes, results from the AAA and AVL estimations were compared in Ref. [99] with system identification results using previous flight test data. Different values for the same parameter allow the creation of an uncertainty range. As a result of the comparison, Table III shows stability and control derivatives, level of uncertainty (maximum and minimum), and the resultant center values and half range for each uncertain parameter.

Table III. Stability and Control Derivatives Uncertainty Range

Parameter	Maximum	Minimum	Central Value	Half Range
C_{mq}	*-4.3720*	*-16.1064*	*-10.2398*	*5.8666*
$C_{m\delta e}$	*-0.7572*	*-1.2289*	*-0.9930*	*0.2359*
$C_{l\beta}$	*-0.0220*	*-0.0314*	*-0.0267*	*0.0047*
C_{nr}	*-0.1156*	*-0.2890*	*-0.2023*	*0.0867*
$C_{n\delta r}$	*-0.0996*	*-0.1404*	*-0.1200*	*0.0204*

In order to perform gain scheduling, the same set of information and analysis is required at different trim conditions. In this research, for demonstration's sake, one trim point is presented.

The 70 knots trim condition is labeled as $i = 1$ and utilizing a set of weighting matrices as described before, a robust controller $\bar{\boldsymbol{K}}^1$, with $\left\|\boldsymbol{T}_{y1\,u1}(s)\right\|_{\infty} = \gamma^1 = 0.543$, is achieved. For the present closed loop configuration, after extracting the uncertainty blocks, infinity norm $\left\|\boldsymbol{F}_l(\hat{\boldsymbol{P}}^1, \bar{\boldsymbol{K}}^1)\right\|_{\infty}$ for each uncertain derivative is calculated and shown in Table IV. The last row shows the worst case condition, which is the infinity norm for all uncertainties together. By construction $\left\|\Delta_j^1\right\|_{\infty} \leq 1$ and consequently $\left\|\Delta^1\right\|_{\infty} \leq 1$, the Small Gain Theorem ensures that the controller robustness keeps stability facing known parameter uncertainty.

The purpose of this section is to test the guidance, navigation, and control design in simulation conceptually comprisng a flight control system (FCS) and to compare its performance with Yak-54 UAS autonomous flight test equipped with a COTS autopilot system. For comparison purposes, similar flight conditions are induced in the simulation, and the same waypoints/trajectory are used for path following.

Table IV. Closed Loop Infinity Norm Associated with each Uncertainty

Parameter	$\left\|F_l(\hat{P}^1,\bar{K}^1)\right\|_\infty$
C_{mq}	*0.3846*
$C_{m\delta e}$	*0.2451*
$C_{l\beta}$	*0.0100*
C_{nr}	*0.3816*
$C_{n\delta r}$	*0.1674*
All	*0.5617*

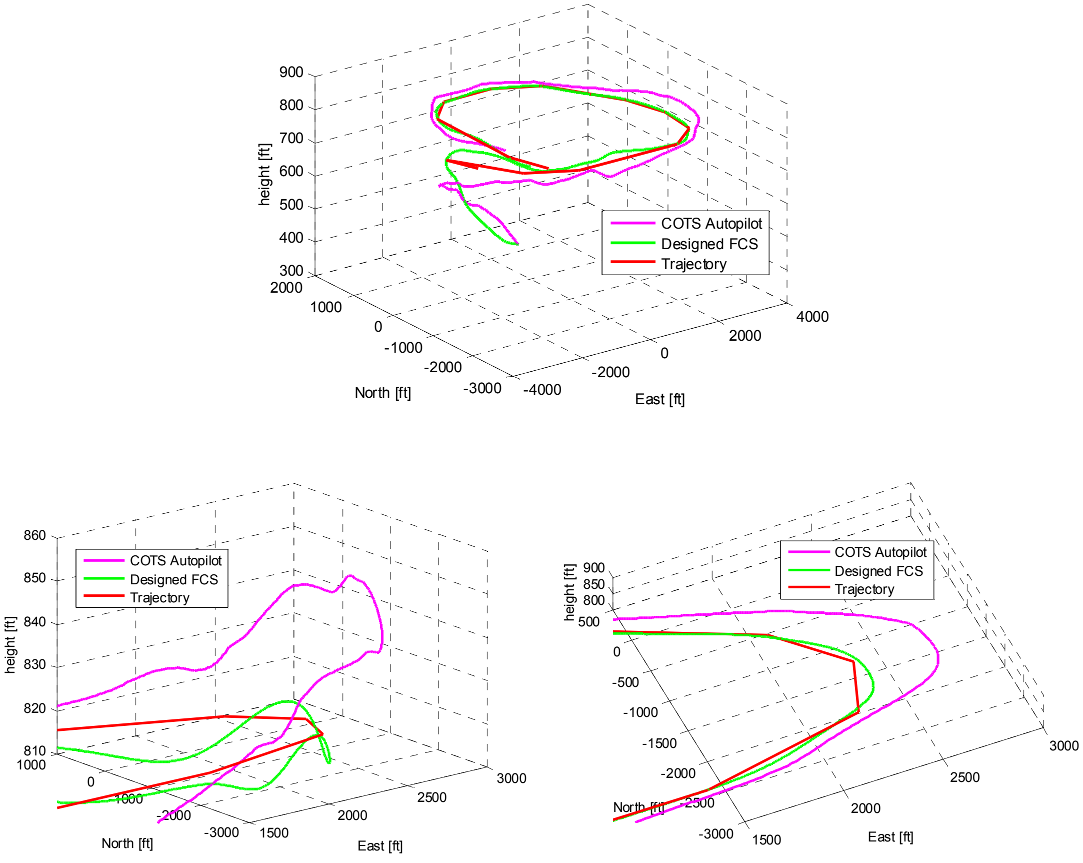

Figure 32. 3D Path Following.

As part of the comparison, inner and outer loops and extended Kalman filter, as defined before, are functioning. In this subsection, no gain scheduling and no ANN estimations are performed.

The FCS is set to command the aircraft to follow a trajectory previously flown autonomously by a Yak-54 UAS equipped with a COTS autopilot. For the similarity to flight test condition, the same initial conditions are used. Also from the flight test data, wind information is extracted to replicate the external disturbance. Figure 32 shows 3D views. The flight history of the Yak-54 UAS equipped with a COTS is shown in magenta and the simulation is shown in green.

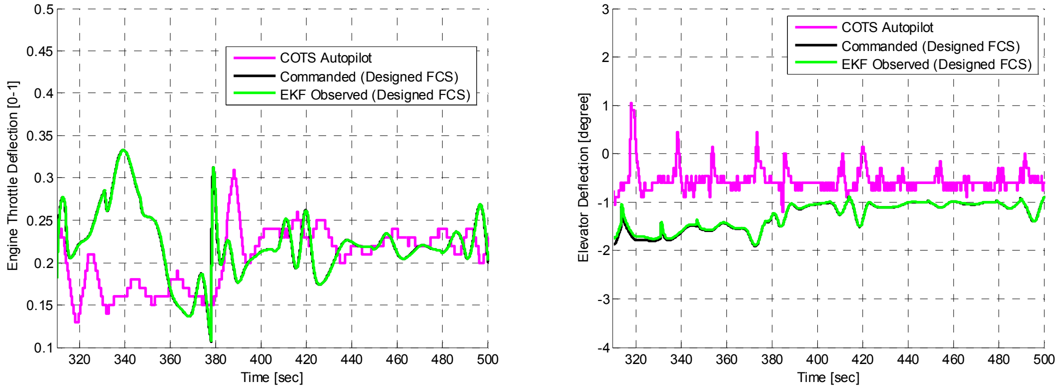

Figure 33. Engine Throttle Deflection (left) – Elevator Deflection (right).

There is an increase in speed during the rise. Although both systems follow the trajectory, the designed FCS presents a tighter tracking, with less cross track errors.

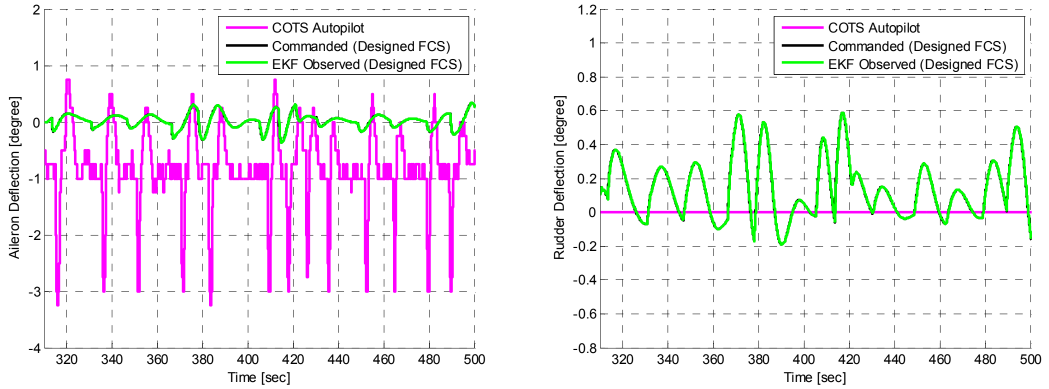

Figure 34. Aileron Deflection (left) – Rudder Deflection (right).

Figure 33 and Figure 34 show the control signals for both the designed FCS in simulation and the COTS autopilot. Some degree of correlation is apparent, with the exception of rudder deflection. By design, this COTS autopilot does not command the rudder. This certainly imposes extra demand on the aileron deflection and requires higher bank angles when initiating turns.

Figure 33 and Figure 34 also include the respective simulated servo command. In the simulation, servo commands and servo deflections are related by a first order linear system.

Despite the inherent dissimilarity between the COTS system and designed GNC, some basic but important comparisons can be made.

Figure 35 (left) shows performance comparison between the COTS and the designed FCS following airspeed command when a $14.1\,[ft/\sec]$ wind blows from 291°. The airspeed command has a step increment around 380 seconds.

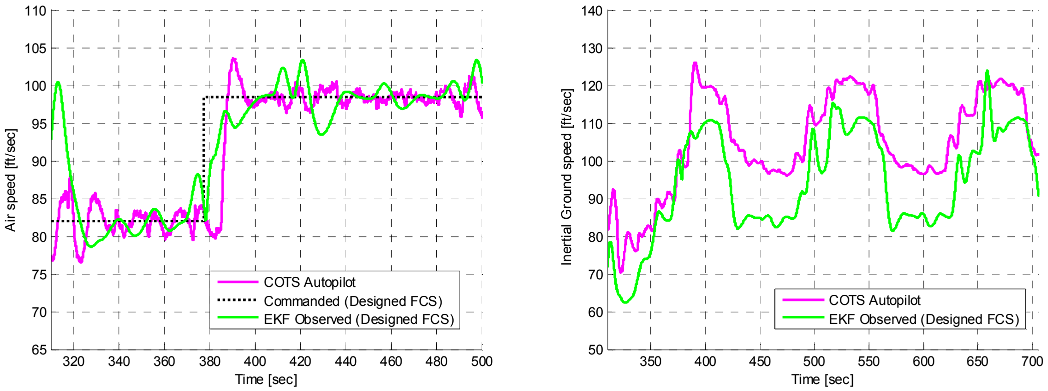

Figure 35. Airspeed (left) – Ground Speed (right).

The ground speed is shown in Figure 35 (right), for a larger portion of the flight. It seems that the error in airspeed control of the COTS autopilot was the result of bias. This presumption is confirmed by adding a constant bias of about $10-12\,[ft/\sec]$ to the measured airspeed and disabling the FCS's Kalman filter for an even comparison. After inducing the bias, ground speeds achieve much higher correlation.

Figure 36 show the attitude angles. The lack of rudder control in the COTS system is revealed in required aileron deflection and bank angle to track the trajectory accurately.

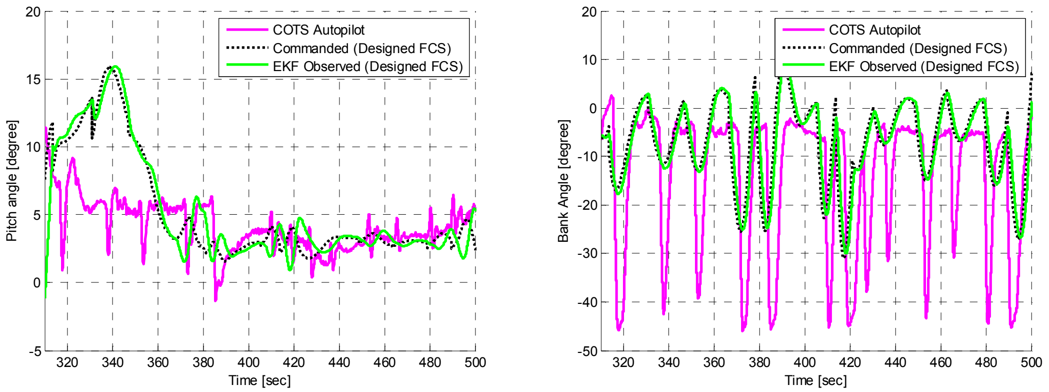

Figure 36. Pitch Angle (left) – Bank angle (right).

It is important to have in mind that the extended Kalman filter takes a few seconds to converge from the given initial conditions, affecting the first part of the estimation, but eventually showing a good command following after transients.

3.3.1 Gain Scheduling Testing

To assess the gain scheduling techniques a hypothetical failure is induced for the same aircraft flying at the trim airspeed of 70 [knots], through a change in the numerical value of a stability and control derivative.

The failure is assumed to be successfully detected and identified within a determined time, and a new controller is applied. An effective gain scheduling process adjusts the controller and keeps the aircraft stability and performance when the adverse condition happens. As a condition for successful gain scheduling, the FCS is expected to achieve a similar level of performance and uncertainty management before and after the unexpected dynamic changes.

In a simulated flight, at 120 seconds of a horizontal level flight, the numerical value of $C_{n\beta}$ (see Table II) is reduced by a factor of 4. This reflects a sudden change in the UAS dynamics, affecting the stability of UAS directionally. As a result of this change, the current controller loses its stabilizing capability and oscillation can be traced in the aircraft's lateral-directional states.

At approximately 138 seconds of flight, it is assumed that the induced failure has been successfully detected, isolated, and identified, and a new robust controller, better adapted to new dynamics, replaces the previous one. The resultant period of time taken to implement the gain scheduling (approximately 18 seconds) is due to the fact that the failure does not show up immediately, as shown in Figure 37 and Figure 38.

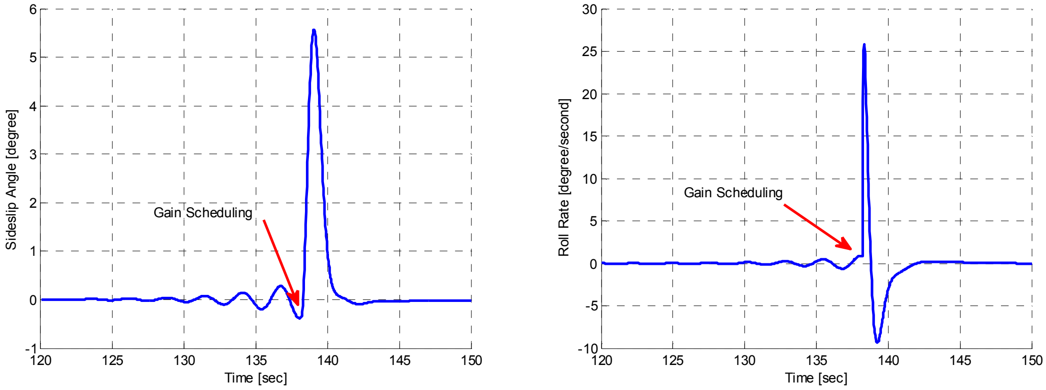

Figure 37. Sideslip Angle (left) – Roll Rate (right).

The induced failure is an alteration of the rate of change in yaw rate with respect to changes in sideslip angle, i.e. $C_{n\beta}$, so for an original constant and close to zero sideslip, no immediate effect is observed. After sideslip angle starts to change, the failure has a multiplicative effect and grows faster. This is evident after 130 seconds. So, for this particular simulation, the failure is not assumed to be detected before this moment.

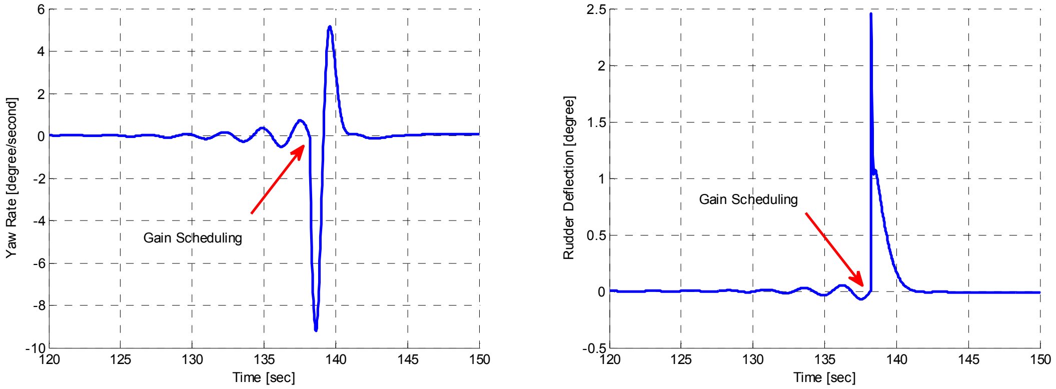

Figure 38. Yaw Rate (left) – Rudder Deflection (right).

It is interesting to note that even though the closed loop with the old controller continues to be (linearly) stable, the uncertainty robustness is lost, as the infinity norm grows over unity $\left\| \boldsymbol{F}_l(\hat{\boldsymbol{P}}^1, \bar{\boldsymbol{K}}^1) \right\|_\infty = 4.6355 > 1$.

4. Nonlinear Model Predictive Control

4.1 Introduction

Guidance, navigation, and control developed in the previous chapter was designed through two disctinct loops, i.e. the outer and inner loops, then mutual interference is not completely discarded.

Changes known in advance in the dynamic or aerodynamic conditions, due to the inherent nonlinear and time-varying nature of the UAS, can be addressed by an online scheduling version of pre-computed linear controllers (see for example Ref. [42] for robust designs). These pre-computed control laws lack the ability to adjust to unforeseen events like unexpected changes in UAS dynamics due to some failure. By redesigning an online controller, or set of them, adaptive control assisted by system identification (see Ref. [7]) offers a solution to overcome unanticipated variations in either actuators, sensors, or the UAS.

Control and state variables and their rates of change are physically restricted to some range. These restrictions can be worsened by unexpected failures, augmenting the nonlinearity of the system and reducing the stability reserve, imposing severe constraints to be satisfied by the controller.

The nonlinear model predictive controller (NMPC) approach, as a nonlinear online numerical control calculation, is potentially able to solve all previous drawbacks in an integrated and direct way, while searching for an optimal control solution (or suboptimal given limitations in computational power) for changing dynamics and conditions (Ref. [47]). Theoretically, NMPC is able to include all information without coarse approximations or simplifications. Given the required computational resources, it is able to use the full updated nonlinear description of the plant, incorporating all constraints, either invariant or time-varying, during each search cycle.

The capability of dealing with an arbitrary set of dynamic equations describing a system suggests the idea of integrating both the guidance and control tasks into one single task. Embedding the guidance laws and trajectory information into the original UAS model gives an augmented system suitable for integrated numerical work. The numerical solution search, as done by NMPC, is then performed over one system, achieving a natively integrated GNC. This, by definition, discards any mutual interference.

The NMPC has been applied successfully in several areas. Its application to unmanned vehicles is more recent and there are still promising advances to make. Ref. [48] applied NMPC to control the longitudinal plane of a submarine whose task was to follow the seabed at a constant distance with no prior knowledge of its variation. Similarly, Ref. [49] designed a NMPC solution for a rotorcraft-based UAV, showing superior tracking performance over conventional multi-loop proportional-derivative controllers. Ref. [50] applied linear MPC to a small helicopter easily incorporating control and state constraints. Another NMPC application is given in Ref. [51], where a receding horizon control algorithm is used for lateral road following using passive sensors. NMPC has also been applied to compute evasive maneuvers in pursuit/evasion games using a gradient-descent method, see Ref. [52].

Most of the past developments utilize the NMPC formulation in the outer loop, for navigation and guidance purposes, leaving the inner loop control task to some commercial autopilot. Although this approach reduces the computational power requirement by reducing the size of the nonlinear model employed by the NMPC, it does not eradicate potential cross coupling between both loops. Ref. [53], which uses NMPC as a high-level tracking controller (outer loop), proved that the complete closed-loop system remains stable if a given initial condition can be driven into a predefined set of final states. In their work, the inner loop is handled by a commercial low-level controller and the NMPC minimizes the lateral cross track error to straight waypoint segments, converting the control law into a regulation problem.

Another interesting online application is found in Ref. [54]. The NMPC was used to control an inverted pendulum. The control law is able to work for any initial condition, and including both inputs and state constraints. It displays fast dynamics with highly nonlinear behavior and it is unstable about the operating point also showing non-minimum phase nature. The online search is done by Sequential

Quadratic Programming (SQP). This application included one control input and four states, two of them directly measured and two estimated.

4.2 Integration of Guidance, Navigation and Control

The aircraft model defined in Chapter 1, represented by $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, with $\boldsymbol{x}(t) \in R^n$ and $\boldsymbol{u}(t) \in R^m$, is augmented by embedding the guidance logic defined in Chapter 3 and the trajectory information, which is assumed to be known a priori. Strictly speaking, from the UAS trajectory, which is composed of straight segments subtended between consecutive waypoints, only the current or active segment is required. The next segments can be uploaded before they are needed.

The guidance logic, as developed in the previous chapter, generates the angles η_{Lat} and η_{Lon} between UAS inertial velocity vector $\vec{\boldsymbol{V}}$ and position vector $\vec{\boldsymbol{L}}$ subtended from the current UAS position $\vec{\boldsymbol{p}}$ to an imaginary point $\vec{\boldsymbol{r}}$ running over the current segment between waypoints $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$. These angles η_{Lat} and η_{Lon} are then conveniently translated into attitude commands ϕ_{cmd} and θ_{cmd}. A tight trajectory following is achieved when the aircraft is able to follow these attitude commands closely. In effect, when $e_\theta(t) = \theta_{cmd}(t) - \theta(t)$ and $e_\phi(t) = \phi_{cmd}(t) - \phi(t)$ are small. This is supplemented with the requirement of minimization of the airspeed and sideslip states errors, i.e. $e_{V_T}(t) = V_{Tcmd}(t) - V_T(t)$ and $e_\beta(t) = \beta_{cmd}(t) - \beta(t)$, respectively. $V_{Tcmd}(t)$ is obtained based on the assigned segment inertial speed and current wind, and $\beta_{cmd}(t) \equiv 0$.

Given the discrete nature of the NMPC algorithm, the dynamics of the UAS have to be discretized, in particular their time derivatives. The choice of an appropriate discretization method is relevant, especially because using higher order discretization methods, in principle more accurate, tends to increase the number of states, and this strongly impacts the efficiency of the algorithm. The sampling rate, together with other design parameters like horizon length and precision of the numerical convergence, are also important design parameters. They have a direct impact on the NMPC performance in terms of accuracy, work load, and time consumption. Although a higher rate is theoretically better, this choice is generally bounded by the available computational power. Also, the sampling rate will have a direct influence on the amount of data the NMPC algorithm will have to manage during each sample. In the present work (as will be seen below), the sampling time $\Delta T = 1/20$ [sec], is chosen as the result of extensive simulation and current availability of computational resources.

To avoid raising the order and complexity of the model, the chosen derivative discretization technique is the Euler method. Its application gives $\boldsymbol{x}(t+\Delta T) \approx \boldsymbol{x}(t) + \Delta T \cdot \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, to be generically expressed as $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta T \cdot \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k)$, where k is the time index of the discretized variables. By defining the output of the augmented UAS model $\boldsymbol{e}(t) \in R^r$ as the controlled output vector $\boldsymbol{e}^T = [e_{V_T}, e_\theta, e_\phi, e_\beta]$, the following augmented discrete model of the aircraft is defined in terms of the original states $\boldsymbol{x}^T = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, \delta_T, \delta_E, \delta_A, \delta_R, p_N, p_E, p_H]$, and control $\boldsymbol{u} = [\delta_T^{cmd}, \delta_E^{cmd}, \delta_A^{cmd}, \delta_R^{cmd}]^T$ as

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \boldsymbol{e}_k &= \boldsymbol{h}_e(\boldsymbol{x}_k) = [e_{V_T}, e_\theta, e_\phi, e_\beta]^T \end{aligned} \tag{96}$$

It is important to note here that the augmented model output is labeled $\boldsymbol{e}$ to differentiate its origin, i.e. guidance augmentation. As will be seen later, for robustness extension purposes another output $\boldsymbol{y}$, will also be added.

4.3 Nonlinear Model Predictive Control Formulation

Following a standard NMPC formulation, as described in Ref. [47], the control problem shown in Figure 39 is stated as receding-horizon control problem P_N. It is defined as

$$\mathrm{P}_N = \min_{\boldsymbol{u}} \{ V(\boldsymbol{e}, \boldsymbol{u}) \} \tag{97}$$

for the system $\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k)$, and $\boldsymbol{e}_k = \boldsymbol{h}_e(\boldsymbol{x}_k)$ given initial condition $\boldsymbol{x}_0, \boldsymbol{u}_0$, where the controlled and control sequences are $\boldsymbol{e} = \{\boldsymbol{e}_2, \boldsymbol{e}_3, \cdots, \boldsymbol{e}_{N+1}\} \in \mathrm{R}^{r \times N}$ and $\boldsymbol{u} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N\} \in \mathrm{R}^{m \times N}$, respectively. In the present work, the cost function $V(\boldsymbol{e}, \boldsymbol{u})$ to be minimized is defined as

$$V(\boldsymbol{e},\boldsymbol{u}) = \sum_{k=1}^{N} \boldsymbol{e}_{k+1}{}^{T} \boldsymbol{Q}_{k+1}\, \boldsymbol{e}_{k+1} + \boldsymbol{u}_k{}^{T} \boldsymbol{R}_k\, \boldsymbol{u}_k \tag{98}$$

with $\boldsymbol{e}_k = \boldsymbol{h}_e(\boldsymbol{x}_k) = [e_{V_T}, e_\theta, e_\phi, e_\beta]_k{}^T \in \mathrm{R}^r$ and $\boldsymbol{u}_k = [\delta_T^{cmd}, \delta_E^{cmd}, \delta_A^{cmd}, \delta_R^{cmd}]_k{}^T \in \mathrm{R}^m$, the controlled output and control vectors at each instant k of the augmented system. Control and state sequences must satisfy the following constraints

$$\begin{gathered} \boldsymbol{u}_k \in \boldsymbol{U} \\ \boldsymbol{x}_k \in \boldsymbol{X} \end{gathered} \tag{99}$$

and as will be seen, imposed by the nature of the system under control, actual constraints can be stated as inequality constrains, accounting for limitations in variables size, i.e. maximum and minimum values.

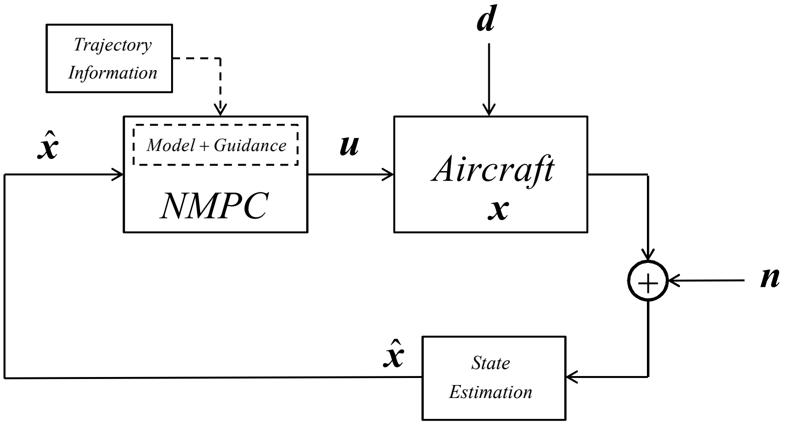

Figure 39. Control Problem.

Symmetric Matrices $\boldsymbol{Q}_k \in \mathrm{R}^{r \times r}$ and $\boldsymbol{R}_k \in \mathrm{R}^{m \times m}$ are positive semi-definite and definite matrices, respectively, allowing time-varying relative penalization among controlled and control sequences. As shown in Figure 39 the only external inputs are disturbance $\boldsymbol{d}$ acting over the aircraft dynamics and the noise $\boldsymbol{n}$ acting over the measurements, i.e. $\boldsymbol{w}^T = [\boldsymbol{n}^T, \boldsymbol{d}^T]$. As the trajectory has been embedded into the aircraft model, there is no external reference to follow, so the control has been reduced to a regulation problem. External inputs are labeled generically as $\boldsymbol{w}$, although they represent both external signals $\boldsymbol{w}$ and $\boldsymbol{v}$ to the aircraft modeling.

For a nonlinear model, analytical NMPC resolution is practically unfeasible, demanding numerical solutions in practice. As opposed to analytical solutions, numerical search does not reach the minimum in finite time, as it takes infinite number of iterations, requiring a predefined criterion to stop the search and declare an intermediate solution. At each algorithm cycle, i.e. during each time interval, the NMPC search algorithm outputs a control sequence $\boldsymbol{u}=\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_N\}$, which provides a relative degree of minimization of the cost function $V(\boldsymbol{e},\boldsymbol{u})$, for a given initial condition $\boldsymbol{x}_0$. I.e. the output control sequence has a lower cost than the initial sequence at the beginning of each numerical search. At the next sample instant, the first element of the solution input sequence $\boldsymbol{u}_1$ is applied to the UAS, the state of the aircraft $\boldsymbol{x}_1$ is measured, and then the algorithm cycle is restarted with this new initial condition, i.e. by making $\boldsymbol{x}_1 \rightarrow \boldsymbol{x}_o$. During each prediction cycle, states and outputs within the horizon are unequivocally defined by the control sequence under determination $\boldsymbol{u}=\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_N\}$, and the initial condition $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$, i.e. the measured state and the applied control input, and or equivalently by the prediction $\boldsymbol{x}_1 = \boldsymbol{f}_d(\boldsymbol{x}_0,\boldsymbol{u}_0)$.

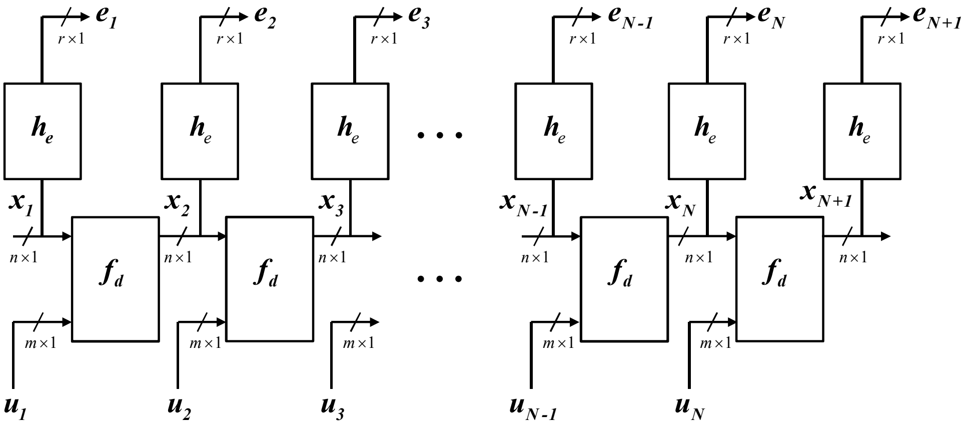

Figure 40. Predictive Horizon.

This becomes clear by recursively applying the aircraft discretized model (96) within the horizon. The state sequence is then defined as (see Figure 40)

$$\begin{aligned}
\boldsymbol{x}_2 &= \boldsymbol{f}_d(\boldsymbol{x}_1,\boldsymbol{u}_1) \\
\boldsymbol{x}_3 &= \boldsymbol{f}_d(\boldsymbol{x}_2,\boldsymbol{u}_2) = \boldsymbol{f}_d(\boldsymbol{f}_d(\boldsymbol{x}_1,\boldsymbol{u}_1),\boldsymbol{u}_2) \\
&\vdots \\
\boldsymbol{x}_N &= \boldsymbol{f}_d(\boldsymbol{x}_{N-1},\boldsymbol{u}_{N-1}) = \boldsymbol{f}_d(\boldsymbol{f}_d(\boldsymbol{x}_{N-2},\boldsymbol{u}_{N-2}),\boldsymbol{u}_{N-1}) = \cdots = \boldsymbol{f}_d(\boldsymbol{f}_d(\cdots \boldsymbol{f}_d(\boldsymbol{x}_1,\boldsymbol{u}_1),\cdots),\boldsymbol{u}_{N-1}) \\
\boldsymbol{x}_{N+1} &= \boldsymbol{f}_d(\boldsymbol{x}_N,\boldsymbol{u}_N) = \boldsymbol{f}_d(\boldsymbol{f}_d(\boldsymbol{x}_{N-1},\boldsymbol{u}_{N-1}),\boldsymbol{u}_N) = \cdots = \boldsymbol{f}_d(\boldsymbol{f}_d(\boldsymbol{f}_d(\cdots \boldsymbol{f}_d(\boldsymbol{x}_1,\boldsymbol{u}_1),\cdots),\boldsymbol{u}_{N-1}),\boldsymbol{u}_N)
\end{aligned} \tag{100}$$

and the controlled sequence as

$$\begin{aligned}
&\boldsymbol{e}_2 = \boldsymbol{h}_e(\boldsymbol{x}_2) = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{x}_1, \boldsymbol{u}_1)) \\
&\boldsymbol{e}_3 = \boldsymbol{h}_e(\boldsymbol{x}_3) = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{x}_2, \boldsymbol{u}_2)) = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{f}_d(\boldsymbol{x}_1, \boldsymbol{u}_1), \boldsymbol{u}_2)) \\
&\quad\vdots \\
&\boldsymbol{e}_N = \boldsymbol{h}_e(\boldsymbol{x}_N) = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{x}_{N-1}, \boldsymbol{u}_{N-1})) = \cdots = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{f}_d(\cdots \boldsymbol{f}_d(\boldsymbol{x}_1, \boldsymbol{u}_1), \cdots), \boldsymbol{u}_{N-1})) \\
&\boldsymbol{e}_{N+1} = \boldsymbol{h}_e(\boldsymbol{x}_{N+1}) = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{x}_N, \boldsymbol{u}_N)) = \cdots = \boldsymbol{h}_e(\boldsymbol{f}_d(\boldsymbol{f}_d(\boldsymbol{f}_d(\cdots \boldsymbol{f}_d(\boldsymbol{x}_1, \boldsymbol{u}_1), \cdots), \boldsymbol{u}_{N-1}), \boldsymbol{u}_N))
\end{aligned} \tag{101}$$

These expressions make clear that the cost function will only be dependent on the control input $\boldsymbol{u} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N\}$ under search, and the initial condition $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$, or equivalently $\boldsymbol{x}_1$, which are fixed quantities at the beginning of each cycle. The receding-horizon problem is then restated as $\mathrm{P}_N = \min_u \{V(\boldsymbol{u})\}$. The chosen approach to solve the nonlinear optimization problem is Sequential Quadratic Programming (SQP), detailed in Ref. [90] and Ref. [101]. SQP solves a sequence of optimization sub-problems, each which optimizes an approximated quadratic model of the objective subject to a linearization of the constraints.

For this application only input constraints are considered, no state constraints are accounted for. Given the restrictions of the actuators (maximum and minimum control surface deflection limits and maximum and minimum engine throttle setting), the input sequence constraints $\boldsymbol{u}_k \in \boldsymbol{U}$, in Equation (99), can be characterized by $\boldsymbol{U} = \{\boldsymbol{u} \mid \boldsymbol{c}(\boldsymbol{u}) < 0\}$. No input rate limiters are considered in the NMPC formulation as the robustness to be added later will provide a way to regulate the input sequence's bandwidth indirectly but effectively. Under the assumption of maximum and minimum limit constraints, the vector-valued function $\boldsymbol{c}(\boldsymbol{u}) = [c_1(\boldsymbol{u}) \ \cdots \ c_{2\times m\times N}(\boldsymbol{u})]^T < 0$ is able to accommodate all of them, as shown below. Minimum and maximum limits are to be updated online, during NMPC algorithm cycles, if changes in servo limits are detected.

$$\begin{bmatrix} u_{11} > u_{11_{MIN}} & \rightarrow & \boldsymbol{c}_1(\boldsymbol{u}) = -u_{11} + u_{11_{MIN}} & < 0 \\ u_{21} > u_{21_{MIN}} & \rightarrow & \boldsymbol{c}_2(\boldsymbol{u}) = -u_{21} + u_{21_{MIN}} & < 0 \\ \vdots & \vdots & \vdots & \vdots \\ u_{N1} > u_{N1_{MIN}} & \rightarrow & \boldsymbol{c}_N(\boldsymbol{u}) = -u_{N1} + u_{N1_{MIN}} & < 0 \end{bmatrix}$$
$$\begin{bmatrix} u_{11} < u_{11_{MAX}} & \rightarrow & \boldsymbol{c}_{N+1}(\boldsymbol{u}) = u_{11} - u_{11_{MAX}} & < 0 \\ u_{21} < u_{21_{MAX}} & \rightarrow & \boldsymbol{c}_{N+2}(\boldsymbol{u}) = u_{21} - u_{21_{MAX}} & < 0 \\ \vdots & \vdots & \vdots & \vdots \\ u_{N1} < u_{N1_{MAX}} & \rightarrow & \boldsymbol{c}_{2N}(\boldsymbol{u}) = u_{N1} - u_{N1_{MAX}} & < 0 \end{bmatrix}$$
$$\vdots$$
$$\begin{bmatrix} u_{1m} > u_{1m\,MIN} & \rightarrow & \boldsymbol{c}_{(2\times m-2)\times N+1}(\boldsymbol{u}) = -u_{1m} + u_{1m\,MIN} & < 0 \\ u_{2m} > u_{2m\,MIN} & \rightarrow & \boldsymbol{c}_{(2\times m-2)\times N+2}(\boldsymbol{u}) = -u_{2m} + u_{2m\,MIN} & < 0 \\ \vdots & \vdots & \vdots & \vdots \\ u_{Nm} > u_{Nm\,MIN} & \rightarrow & \boldsymbol{c}_{(2\times m-1)\times N}(\boldsymbol{u}) = -u_{Nm} + u_{Nm\,MIN} & < 0 \end{bmatrix}$$
$$\begin{bmatrix} u_{1m} < u_{1m\,MAX} & \rightarrow & \boldsymbol{c}_{(2\times m-1)\times N+1}(\boldsymbol{u}) = u_{1m} - u_{1m\,MAX} & < 0 \\ u_{2m} < u_{2m\,MAX} & \rightarrow & \boldsymbol{c}_{(2\times m-1)\times N+2}(\boldsymbol{u}) = u_{2m} - u_{2m\,MAX} & < 0 \\ \vdots & \vdots & \vdots & \vdots \\ u_{Nm} < u_{Nm\,MAX} & \rightarrow & \boldsymbol{c}_{2\times m\times N}(\boldsymbol{u}) = u_{Nm} - u_{Nm\,MAX} & < 0 \end{bmatrix} \qquad (102)$$

Violated constraints must be included as part of the cost function, with the help of time-varying Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1 \ \cdots \ \lambda_p]^T$. The new objective function is so defined as

$$L(\boldsymbol{u},\boldsymbol{\lambda}) = V(\boldsymbol{u}) - \boldsymbol{\lambda}^T \bar{\boldsymbol{c}}(\boldsymbol{u}) \qquad (103)$$

where constraints are added as equality constraints only when they are valid, this is when they are violated during the numerical search. An updated constraint vector $\bar{\boldsymbol{c}}(\boldsymbol{u}) = [\bar{c}_1(\boldsymbol{u}) \ \cdots \ \bar{c}_p(\boldsymbol{u})]^T = 0$ incorporates only the current p valid constraints; the upper bar is used to differentiate hypothetical in (102) from currently violated limits. Constraints are otherwise removed from the augmented objective function $L(\boldsymbol{u},\boldsymbol{\lambda})$ and consequently from $\bar{\boldsymbol{c}}(\boldsymbol{u})$, along with respective elements in vector $\boldsymbol{\lambda}$. This condition is easily evaluated based on the sign of the respective multipliers (Ref. [90]).

SQP solves the nonlinear problem with a series of sub-problems, each one minimizing a quadratic approximation of the objective function with a linearized approximation of the constraints. In the present formulation, as the constraints are defined as linear in the control variables, there is no need to linearize them. In the case of no valid constraints for a particular iteration, SQP reduces to a Newton-type numerical method. Applying the SQP method from Ref. [90] to the present case, the following correction is made each numerical search step

$$\begin{bmatrix} \boldsymbol{u}^{i+1} \\ \boldsymbol{\lambda}^{i+1} \end{bmatrix}_{(m\times N+p)\times 1} = \begin{bmatrix} \boldsymbol{u}^{i} \\ \boldsymbol{\lambda}^{i} \end{bmatrix}_{(m\times N+p)\times 1} + \begin{bmatrix} \Delta\boldsymbol{u}^{i} \\ \Delta\boldsymbol{\lambda}^{i} \end{bmatrix}_{(m\times N+p)\times 1} \tag{104}$$

where the subscript $i = 1, 2, \cdots$ indexes the iterations within a cycle, p is the number of inequality constraints valid at iteration i, and $[(\boldsymbol{u}_1^i)^T, (\boldsymbol{u}_N^i)^T, \cdots, \lambda_1^i, \cdots, \lambda_p^i]^T \in \mathrm{R}^{(m\times N+p)}$ is a column vector. The correction term $[(\Delta\boldsymbol{u}^i)^T, \Delta\boldsymbol{\lambda}^i]^T$ is obtained by solving the following linear system of equations each iteration

$$\begin{bmatrix} \nabla^2 V & -\boldsymbol{A}^T \\ \boldsymbol{A} & 0 \end{bmatrix}_{(m\times N+p)\times(m\times N+p)} \begin{bmatrix} \Delta\boldsymbol{u}^{i} \\ \Delta\boldsymbol{\lambda}^{i} \end{bmatrix}_{(m\times N+p)\times 1} = \begin{bmatrix} -\nabla V + \boldsymbol{A}^T\boldsymbol{\lambda}^i \\ \overline{\boldsymbol{c}} \end{bmatrix}_{(m\times N+p)\times 1} \tag{105}$$

where matrix $\boldsymbol{A}^T(\boldsymbol{u}^i) = [\nabla\overline{c}_1(\boldsymbol{u}^i) \quad \cdots \quad \nabla\overline{c}_p(\boldsymbol{u}^i)]$ is the Jacobian of the constraints vector $\overline{\boldsymbol{c}}$. Remaining is the calculation of the Hessian and the gradient $\nabla^2 V$ and ∇V, respectively. Based on Ref. [54], and given the quadratic symmetry of cost function $V(\boldsymbol{u}^i)$, it is expressed as $V(\boldsymbol{u}^i) = \boldsymbol{v}^T(\boldsymbol{u}^i)\boldsymbol{v}(\boldsymbol{u}^i)$ with $\boldsymbol{v}^T(\boldsymbol{u}^i) = [(\boldsymbol{Q}_2^{1/2}\boldsymbol{e}_2^i)^T, \cdots, (\boldsymbol{Q}_{N+1}^{1/2}\boldsymbol{e}_{N+1}^i)^T, (\boldsymbol{R}_1^{1/2}\boldsymbol{u}_1^i)^T, \cdots, (\boldsymbol{R}_N^{1/2}\boldsymbol{u}_N^i)^T]$ and $\boldsymbol{v}(\boldsymbol{u}^i) \in \mathrm{R}^{n\times N+m\times N}$. The Hessian and gradient of the cost function can be approximately obtained by expanding the error vector $\boldsymbol{v}(\boldsymbol{u}^i)$ in a Taylor series up to the first order term as $\boldsymbol{v}(\boldsymbol{u}^i + \Delta\boldsymbol{u}^i) \approx \boldsymbol{v}(\boldsymbol{u}^i) + \boldsymbol{J}(\boldsymbol{u}^i)\Delta\boldsymbol{u}^i$, where $\boldsymbol{J}(\boldsymbol{u}^i)$ is the Jacobian matrix of $\boldsymbol{v}(\boldsymbol{u}^i)$, and then forming,

$$\begin{aligned} V(\boldsymbol{u}^i + \Delta\boldsymbol{u}^i) &\approx (\boldsymbol{v}(\boldsymbol{u}^i) + \boldsymbol{J}(\boldsymbol{u}^i)\Delta\boldsymbol{u}^i)^T(\boldsymbol{v}(\boldsymbol{u}^i) + \boldsymbol{J}(\boldsymbol{u}^i)\Delta\boldsymbol{u}^i) \\ &\approx \overbrace{\boldsymbol{v}^T(\boldsymbol{u}^i)\boldsymbol{v}(\boldsymbol{u}^i)}^{V(\boldsymbol{u}^i)} + (\Delta\boldsymbol{u}^i)^T \overbrace{2\boldsymbol{J}^T(\boldsymbol{u}^i)\boldsymbol{v}(\boldsymbol{u}^i)}^{\nabla V(\boldsymbol{u}^i)} + \frac{1}{2}(\Delta\boldsymbol{u}^i)^T \overbrace{2\boldsymbol{J}^T(\boldsymbol{u}^i)\boldsymbol{J}(\boldsymbol{u}^i)}^{\nabla^2 V(\boldsymbol{u}^i)}\Delta\boldsymbol{u}^i \end{aligned} \tag{106}$$

4.4 Aerodynamic Estimation ANN Augmentation

To take full advantage of the NMPC adaptability, the online aerodynamic nonlinear estimation developed in Chapter 1, is used in the controller's aircraft model formulation.

4.5 Modification of Trajectory Following Model to Follow Moving Points

The previous UAS guidance task was designed for straight lines, called segments, subtended between consecutive waypoints. To provide more flexibility for a centralized UAS formation command, with centralized commands and decentralized control, for example, a modification is developed in the present section to allow the UAS to track a moving point, i.e. to follow arbitrary curves shaped by moving points.

The idea is to have a centralized command that delivers the moving points to the UAS flying alone or within a formation, so the moving point information is received by the UAS as an external reference. This is not mandatory and the moving point information could be already stored onboard the aircraft before taking off. Based on this moving point information, the UAS generates its own control actions by making use of its original guidance structure.

The original waypoint guidance works by tracking a segment, i.e. a straight line, subtended between two consecutive waypoints $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{b}}$, generating the angles η_{Lat} and η_{Lon} referred to the trajectory, and consecutively converting these values into attitude commands ϕ_{cmd} and θ_{cmd}. When the aircraft reaches a predefined distance to the end of the segment $\boldsymbol{ab}$ it switches to the next segment $\boldsymbol{bc}$.

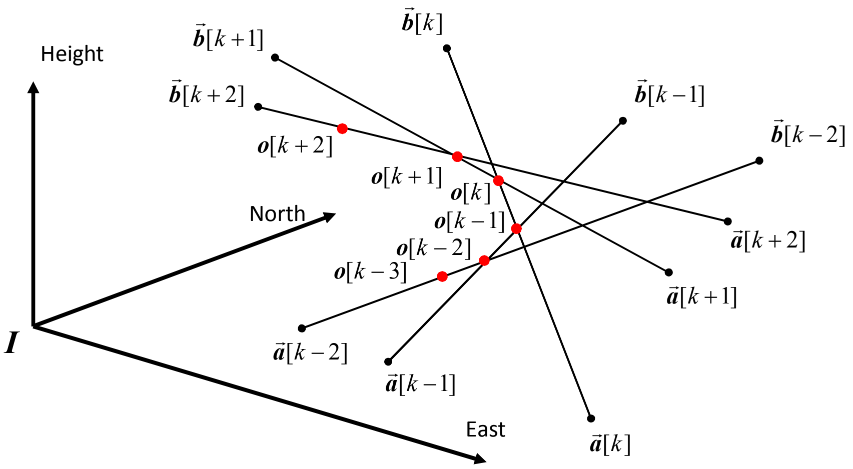
$\vec{\boldsymbol{b}}[k]$
$\vec{\boldsymbol{b}}[k+1]$
Height
$\vec{\boldsymbol{b}}[k+2]$
$\vec{\boldsymbol{b}}[k-1]$
$\vec{\boldsymbol{b}}[k-2]$
$\boldsymbol{o}[k+2]$
$\boldsymbol{o}[k+1]$
$\boldsymbol{o}[k]$
North
$\boldsymbol{o}[k-1]$
$\boldsymbol{o}[k-2]$
$\boldsymbol{o}[k-3]$
$\vec{\boldsymbol{a}}[k+2]$
$\vec{\boldsymbol{a}}[k+1]$
$\boldsymbol{I}$
$\vec{\boldsymbol{a}}[k-2]$
$\vec{\boldsymbol{a}}[k-1]$
East
$\vec{\boldsymbol{a}}[k]$

Figure 41. Time-Varying Waypoints $\vec{\boldsymbol{a}}[k]$ and $\vec{\boldsymbol{b}}[k]$.

The idea is to use the original guidance structure without modifying it. The modification is done over the waypoints of the current segment $\boldsymbol{ab}$. This is, using the moving point inertial position and speed coordinates, defined as $\boldsymbol{o}[k]:\{\vec{\boldsymbol{p}}^{\boldsymbol{O}}_{NEH}[k]=p^{O}_{N}[k],p^{O}_{E}[k],p^{O}_{H}[k]\},\{\vec{\boldsymbol{v}}^{\boldsymbol{O}}_{NEH}[k]=v^{O}_{N}[k],v^{O}_{E}[k],v^{O}_{H}[k]\}$, and its history information, i.e. $\{\boldsymbol{o}[k-1],\boldsymbol{o}[k-2],\cdots\}$, which needs to be stored by the UAS, computes online time-varying waypoints $\vec{\boldsymbol{a}}[k]$ and $\vec{\boldsymbol{b}}[k]$, and new time-varying segment inertial speed $V_{ab}[k]$, and then override preexisting ones each sample time. The logic is sketched in Figure 41 and Figure 42. As shown in Figure 41, time-varying waypoints $\vec{\boldsymbol{a}}[k]$ and $\vec{\boldsymbol{b}}[k]$ are located in the extremes of a line intersecting current and previous moving point, i.e. $\boldsymbol{o}[k]$ and $\boldsymbol{o}[k-1]$. The distance at which waypoints are located should be greater than the distance required by the segment switching logic, but otherwise is irrelevant.

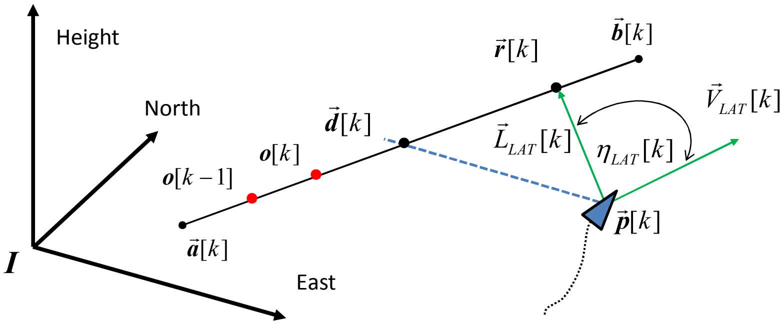
Height
North
East
I
$\boldsymbol{o}[k-1]$
$\boldsymbol{o}[k]$
$\vec{\boldsymbol{a}}[k]$
$\vec{\boldsymbol{d}}[k]$
$\vec{\boldsymbol{r}}[k]$
$\vec{\boldsymbol{b}}[k]$
$\vec{L}_{LAT}[k]$
$\eta_{LAT}[k]$
$\vec{V}_{LAT}[k]$
$\vec{\boldsymbol{p}}[k]$

Figure 42. Lateral Guidance Logic for Time-Varying Waypoints $\vec{a}[k]$ and $\vec{b}[k]$.

As shown in Figure 42 (for the lateral plane), guidance logic applies unchanged, independently of the waypoints new time-varying nature. For the time-varying segment inertial speed $V_{ab}[k]$ a specific new logic is required. Originally this quantity was fixed, as were the waypoints. Now it will be a variable quantity, so a mathematical description is required.

Previous modification will keep the aircraft close to the time-varying segment $\boldsymbol{ab}$, irrespectively of its relative location with respect to the moving point $\boldsymbol{o}[k]$, as the attitude guidance works with the line subtended between the current and previous moving points, i.e. $\boldsymbol{o}[k]$ and $\boldsymbol{o}[k-1]$, and not their relative location. In order to asymptotically get closer to the moving point $\boldsymbol{o}[k]$, the absolute distance d_{do} between point $\boldsymbol{o}[k]$ and the projected position of the aircraft to the segment $\vec{\boldsymbol{d}}[k]$, see Figure 43 (left), and its sign $sign(d_{ob}[k]-d_{db}[k])$, are shown to be adequate to control this guidance. Equation (107) shows the chosen mathematical relation

$$V_{ab}[k] = \left|\boldsymbol{v}^{\boldsymbol{O}}_{NEH}[k]\right| - k^{1}_{Vo}\tan^{-1}\left(\frac{1}{k^{2}_{Vo}}sign(d_{ob}[k]-d_{db}[k])\cdot d_{do}[k]\right) \tag{107}$$

where $\left|\vec{\boldsymbol{v}}^{\boldsymbol{O}}_{NEH}[k]\right|$ is the speed magnitude of the moving point, and k^{1}_{Vo} and k^{2}_{Vo} are two positive constants that help shape the magnitude and rate of speed correction, respectively, and d_{ob} and d_{db} are the distances from waypoint $\vec{\boldsymbol{b}}$ to point $\boldsymbol{o}[k]$ and to point $\vec{\boldsymbol{d}}$, respectively.

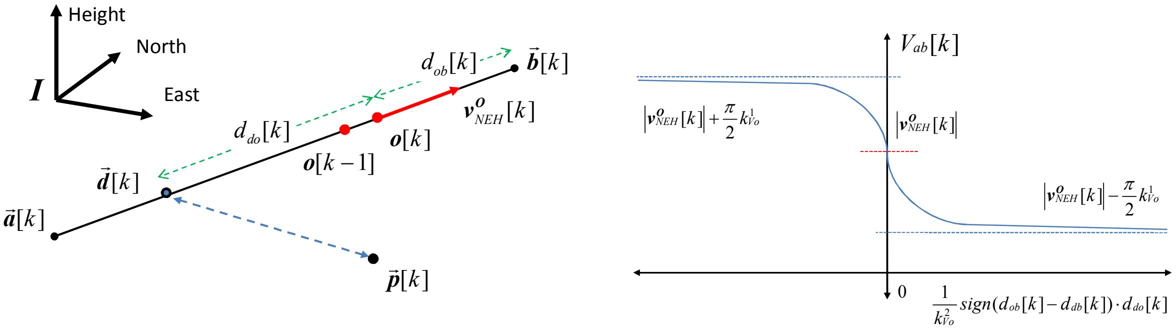
Height
North
East
$\boldsymbol{I}$
$d_{ob}[k]$
$\vec{\boldsymbol{b}}[k]$
$d_{do}[k]$
$\boldsymbol{v}^{\boldsymbol{O}}_{NEH}[k]$
$\boldsymbol{o}[k]$
$\boldsymbol{o}[k-1]$
$\vec{\boldsymbol{d}}[k]$
$\vec{\boldsymbol{a}}[k]$
$\vec{\boldsymbol{p}}[k]$
$V_{ab}[k]$
$\left|\boldsymbol{v}^{\boldsymbol{O}}_{NEH}[k]\right| + \frac{\pi}{2}k^{1}_{Vo}$
$\left|\boldsymbol{v}^{\boldsymbol{O}}_{NEH}[k]\right|$
$\left|\boldsymbol{v}^{\boldsymbol{O}}_{NEH}[k]\right| - \frac{\pi}{2}k^{1}_{Vo}$
0
$\frac{1}{k^{2}_{Vo}}sign(d_{ob}[k]-d_{db}[k])\cdot d_{do}[k]$

Figure 43. Speed Segment Geometry (left) - Speed Command Guidance Logic (right).

From Figure 43 (right) it is clear that there is a bounded range for the speed command to vary centered in the current moving point speed. This shape allows a smooth correction and avoids excesively large speeds.

4.6 NMPC Robustness Augmentation

There is more than one way to grant robustness to NMPC. Although robustness is one of the main purposes and achievements of the action of feedback, it can be further improved using the available knowledge of uncertainties and external disturbances. Robust control aims to use existing uncertainty and disturbance bounds as additional information in the controller design to provide a degree of robustness. NMPC has no explicit robustness built-in, leaving a severe restriction in its employment. In general there are two main robustness augmentation schemes. One is the Min-max design; where minimization of a cost function is perform jointly with the maximization of the external disturbance effect (see for example Ref. [61] and [62]), and the other is the judicious shaping of some characteristic closed loop functions (see Ref. [65] for H-2 and Ref. [66] for nonlinear H-infinity), indirectly providing robustness. The first one, probably more optimal, becomes quickly unfeasible when controlling medium size order systems in numerical solutions like NMPC. The second, a common approach as is the case of linear LQG, H-2, or H-infinity, is less optimal, but offers a tractable online solution. The present section integrates this concept into an NMPC framework, providing it with robustness without endangering the feasibility of its application to an aircraft control. Figure 44 sketches the proposed solution.

Note that internal signals do not show a time index as they are independent of the absolute time k ; NMPC works during the time interval between samples, based on the measured/estimated status of the system, and outputs its solution at the next sample time k , after several iterations.

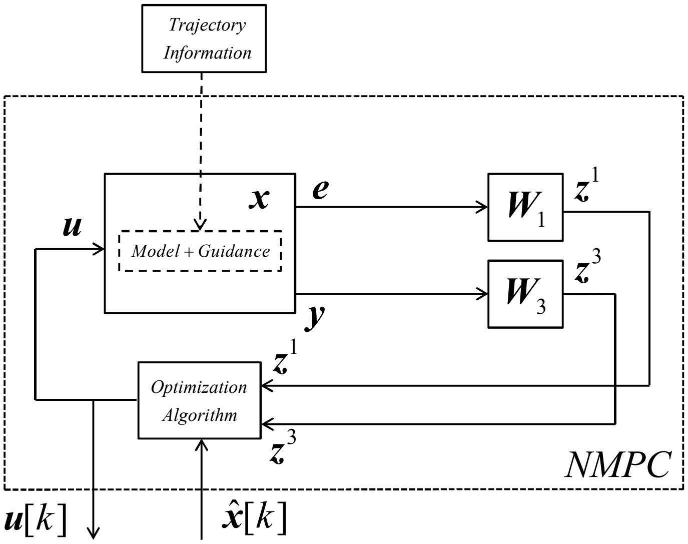

Figure 44. Robust NMPC Configuration.

The robust NMPC solution is inspired by the robust augmentation developed for the mixed sensitivity technique in linear cases such as H-infinity and H-2 controllers. It is based on the assumptions that external disturbances and internal uncertainties have specific and disjoint frequency content, which is not unpractical. Disturbances are characterized by a low frequency range, while measurement noises and model uncertainties are normally defined by a high frequency range. This assumption leads to the successful application of the mixed sensitivity concept, shaped by dynamic weighting matrices $\boldsymbol{W}_1$ and

$\boldsymbol{W}_3$. This frequency modulation allows joint minimization of original opponent objectives, namely sensitivity (associated with $\boldsymbol{W}_1$) and complementary sensitivity (associated with $\boldsymbol{W}_3$) closed loop functions. This linear tool, which has been extended to a particular nonlinear control case in Ref. [66], for example, is also applied in the present work to the nonlinear model predictive controller.

The signals to be minimized are an extension from the ones in the original NMPC developed in 4.2. As shown in Figure 44, the new minimization variable $\boldsymbol{z}^T = [(\boldsymbol{z}^1)^T, (\boldsymbol{z}^3)^T]$, is entering the optimization algorithm, instead of only the original sequence $\boldsymbol{e}$. Besides $\boldsymbol{e}$, a new intermediate output is defined $\boldsymbol{y}_k^T = \boldsymbol{h}_y(\boldsymbol{x}_k) = [V_T, \theta, \phi, \beta]_k$, i.e. $\boldsymbol{y} = \{\boldsymbol{y}_2, \boldsymbol{y}_3, \cdots, \boldsymbol{y}_{N+1}\} \in \mathrm{R}^{r \times N}$. Both signals become, through weighting matrices, the new controlled output $\boldsymbol{z}$. The control problem is then reformulated as the NMPC problem $\mathrm{P}_N = \min_{\boldsymbol{u}} \{V(\boldsymbol{z}, \boldsymbol{u})\}$ with $V(\boldsymbol{z}, \boldsymbol{u}) = \sum_{k=1}^{N} \boldsymbol{z}_{k+1}{}^T \boldsymbol{Q}_{k+1} \boldsymbol{z}_{k+1} + \boldsymbol{u}_k{}^T \boldsymbol{R}_k \boldsymbol{u}_k$ for a given initial condition $\boldsymbol{x}_0$.

The aircraft dynamic model is expanded as $\boldsymbol{x}_{k+1}^R = \boldsymbol{f}_d^R(\boldsymbol{x}_k^R, \boldsymbol{u}_k)$ to include the dynamics of the weighting matrices, described as follows

$$
\begin{aligned}
&\boldsymbol{W}_1 : \begin{cases}
x_{k+1}^{W_1^1} = A^{W_1^1} x_k^{W_1^1} + B^{W_1^1} e_k^1 & z_k^{W_1^1} = C^{W_1^1} x_k^{W_1^1} + D^{W_1^1} e_k^1 \\
x_{k+1}^{W_1^2} = A^{W_1^2} x_k^{W_1^2} + B^{W_1^2} e_k^2 & z_k^{W_1^2} = C^{W_1^2} x_k^{W_1^2} + D^{W_1^2} e_k^2 \\
\qquad \vdots & \qquad \vdots \\
x_{k+1}^{W_1^r} = A^{W_1^r} x_k^{W_1^r} + B^{W_1^r} e_k^r & z_k^{W_1^r} = C^{W_1^r} x_k^{W_1^r} + D^{W_1^r} e_k^r
\end{cases} \\
&\boldsymbol{W}_3 : \begin{cases}
x_{k+1}^{W_3^1} = A^{W_3^1} x_k^{W_3^1} + B^{W_3^1} y_k^1 & z_k^{W_3^1} = C^{W_3^1} x_k^{W_3^1} + D^{W_3^1} y_k^1 \\
x_{k+1}^{W_3^2} = A^{W_3^2} x_k^{W_3^2} + B^{W_3^2} y_k^2 & z_k^{W_3^2} = C^{W_3^2} x_k^{W_3^2} + D^{W_3^2} y_k^2 \\
\qquad \vdots & \qquad \vdots \\
x_{k+1}^{W_3^r} = A^{W_3^r} x_k^{W_3^r} + B^{W_3^r} y_k^{W_1^r} & z_k^{W_3^r} = C^{W_3^r} x_k^{W_3^r} + D^{W_3^r} y_k^3
\end{cases}
\end{aligned}
\tag{108}
$$

with $\boldsymbol{x}^R = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, \delta_T, \delta_E, \delta_A, \delta_R, p_N, p_E, p_H, x_k^{W_1^1}, x_k^{W_1^2}, \cdots, x_k^{W_1^r}, x_k^{W_3^1}, x_k^{W_3^2}, \cdots, x_k^{W_3^r}]^T$ the new vector states of dimension n^R. The set of matrices $\{A^{W_1^i}, B^{W_1^i}, C^{W_1^i}, D^{W_1^i}\}$ and $\{A^{W_3^i}, B^{W_3^i}, C^{W_3^i}, D^{W_3^i}\}$, with $i = 1 \cdots r$, are designed to achieve frequency response singular values in the shape of $\boldsymbol{W}_1$ and $\boldsymbol{W}_3$, respectively. Given the configuration, as shown in Figure 44, these frequency-dependent matrices will allow the joint minimization of the low frequency range sequence $\boldsymbol{e}$ and the high frequency range $\boldsymbol{y}$

(defined by the bandwidth of the respective matrices' singular values), originally opposed objectives. In sum, after necessary tuning, the right matrices will provide low frequency regulation and disturbance rejection, and high frequency measurement noise and model uncertainty attenuation. The choice of the sampling interval ΔT, in this case $1/20$ seconds, which is bounded basically by the computational onboard available power, has a direct impact on the frequency range for which the weighting matrix $\boldsymbol{W}_3$ is effective. In theory, noise frequencies higher than $(2\Delta T)^{-1}$ [Hz] will show up unfiltered in the lower frequency range due to aliasing. This imposes the requirement of a previous independent analogous filtering stage.

4.6.1 Robustness Weighting Matrices Design

The design of the discrete-time weighting matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_3$ follows the same approach carried out in the continuous-time robust design problem (see 3.2.1.1).

Low frequency reference tracking and disturbance rejection, and high frequency additive noise and/or bias cancelation are achieved by setting:

a. Steady state error $A_S < \overline{A}_S$

b. Closed-loop bandwidth $w < \overline{w}$

c. High frequency noise amplification $A_T < \overline{A}_T$

where $\overline{A}_S$ and $\overline{A}_T$ are the maximum expected closed loop steady state error and high frequency noise amplification. The frequency bandwidth $\overline{w}$, similarly to the continuous case, is closely defined by the measurement additive perturbation. In the present case, this factor is also determined by the discrete sampling frequency.

The previous criterion, which gives analytical design specifications based on known statistics of disturbances and noise, serves as a starting point and should be adjusted by the outcome of high-fidelity simulations. For the present case, the following empirical conditions, which are highly problem-dependent, are useful in the setting of the weighting matrices' parameters:

I. For each element in the error vector $\boldsymbol{e}^T = [e_{V_T}, e_\theta, e_\phi, e_\beta]$, adjust respective parameter $\overline{A}_S$ in each transfer function in $\boldsymbol{W}_1$, to achieve acceptable steady state error and disturbance attenuation.

II. For each element in the output vector $\boldsymbol{y}^T = [V_T, \theta, \phi, \beta]$, adjust respective parameter $\overline{A}_T$ in each transfer function in $\boldsymbol{W}_3$ to achieve acceptable measurement additive perturbation rejection.

III. Adjust the closed loop bandwidth parameter w to achieve acceptable control reaction time.

IV. Perform steps I, II and III for different perturbations, in terms of intensity and frequency content, monitoring the evolution of the states and control variables of the aircraft in terms of amplitude and time duration, satisfying their respective constraints, and keeping the closed loop stability.

4.7 Simulations

In this section, trajectory and moving point tracking are tested in simulation for a UAS controlled to follow a predefined trajectory. Tracking is simulated both using the full NMPC formulation previously designed and making use of ANNs to model the translational accelerations. In both simulations, failures are induced to test the online detection and NMPC adaptation. Also in this section, robustness augmentation is tested in a separate simulation for both frequency ranges.

4.7.1 Simulation of Trajectory Tracking

In this simulation the Meridian UAS is commanded to fly a straight trajectory at 100 knots.

Considering the Meridian modal analysis (Ref. [102]), the targeted time for real-time adaptive ANN estimation and NMPC numerical search is set to be around one second. It is very important to notice that both tasks are parallel tasks; they do not need to be accomplished sequentially, so their respective times do not add together. It is expected that using a dedicated microprocessor for both the ANN and NMPC, associated computational times could be decreased further.

For the present section, the simulation is run at frequency of 20 Hertz. The NMPC is designed with a horizon of $N = 20$ sample times, giving a horizon time of one second of predicted flight. Also, one iteration has been employed. The ANN is trained using a moving window of $N_W = 50$.

To test the performance of the ANN and NMPC performance in off-nominal conditions, two scenarios are simulated. An adverse onboard condition (e.g. damage) and external hazards and disturbances (e.g. icing) are induced. To implement these changes in the aircraft dynamics, while it is commanded to follow a trajectory, key aerodynamic derivatives are altered in Equations (24) and (22), respectively.

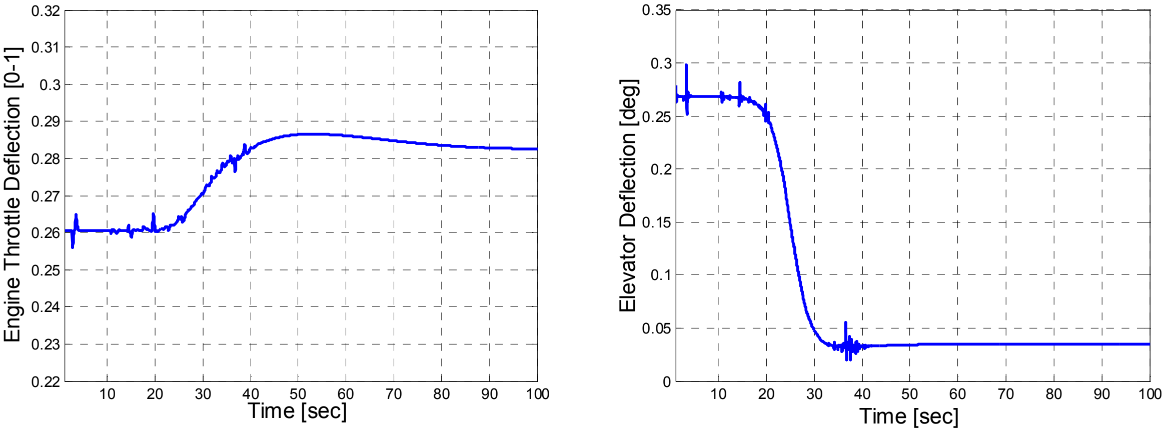

Figure 45. Throttle Deflection (left) - Elevator Deflection (right).

It is shown that the adaptive ANN algorithm successfully identify the alteration in the UAS dynamics, reacting to it by adjusting the ANN parameters and consequently updating the NMPC physics-based model. Alterations are induced approximately after 20 seconds of north flight at 100 knots.

Alteration of C_{L0}

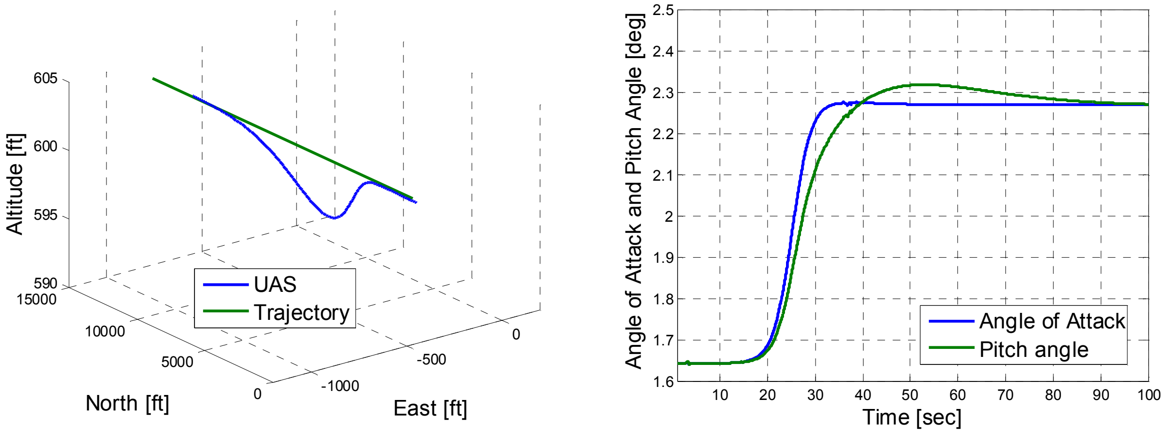

Figure 46. Trajectory Following (left) – Angle of attack and Pitch angle (right).

The first scenario is created by decreasing the coefficient C_{L0} by 20%. This decrement simulates the effect of decreasing overall lift due to a wing loss incident. Figure 45 and Figure 46 show some of the relevant variables involved. As shown in Figure 45, longitudinal controls react accordingly to keep the UAS on the trajectory after the change is induced.

Figure 46 (left) shows the direct impact on the altitude of the UAS after losing lift, and the consequent correction and approach to the trajectory. Figure 46 (right) shows the increase in angle of attack to compensate the loss in the overall lift, and the correction in pitch angle to restore the horizontal flight.

Alteration of C_{D0}

The second scenario is doubling the coefficient C_{D0} due to external hazards and change in the surface roughness due to icing. This increase in the total drag is counteracted mainly by increasing the thrust through the throttle deflection. Figure 47 shows the controls reaction. No significant departure from the trajectory was observed.

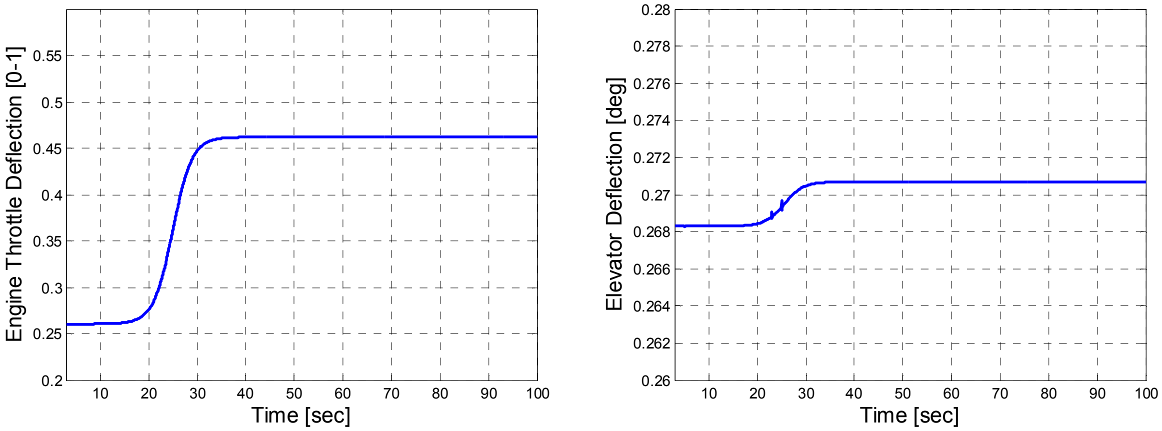

Figure 47. Throttle Deflection (left) – Elevator Deflection (right).

Particularly in the first scenario, some high frequency oscillations are observed when the dynamic change was induced. This could be reflecting an incipient interference coupling between the NMPC and the ANN. One way to counteract this possible cause is to provide robustness to the NMPC.

Previous simulations did not include robust design augmentation.

4.7.2 Simulation of Moving Point Tracking

A 33% scale Yak 54 UAS is commanded to follow a moving point flying at 70 knots, starting from an offset location.

For comparison, the same conditions are applied to the design from Section 3.2.

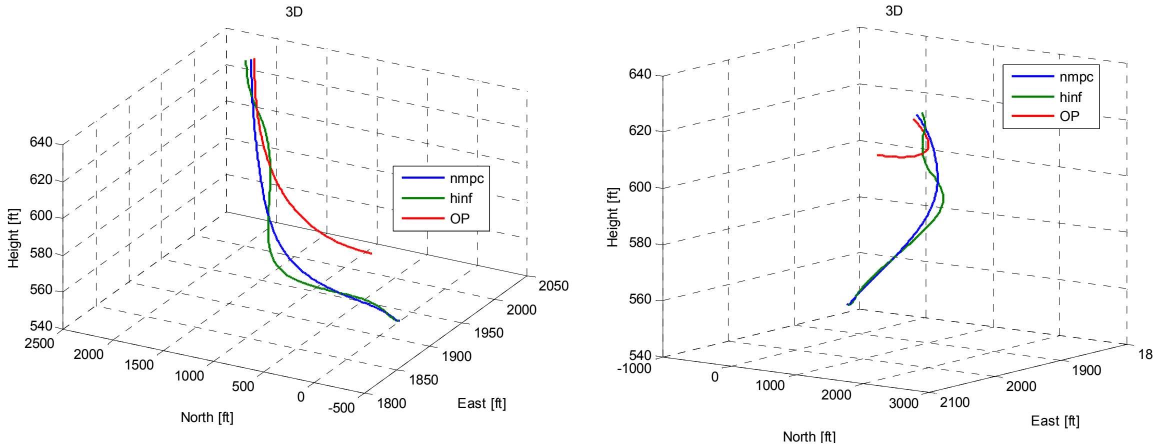

Figure 48. 3D Views Moving Point Following.

Figure 48 shows 3D views of the moving point following, which is drawn in red. The aircraft starts with a crosstrack offset of 50 feet horizontally and 50 feet vertically.

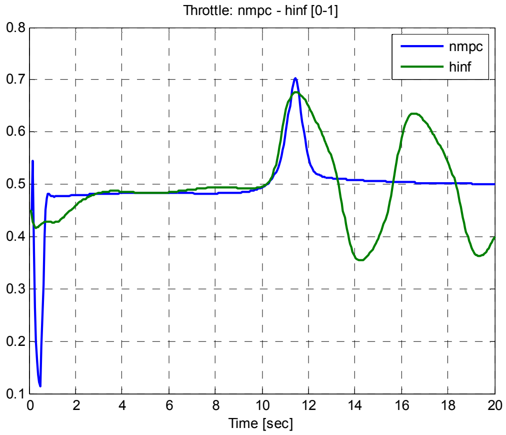

Figure 49. Engine Throttle Deflection.

Figure 49 shows the engine throttle deflection. The two-loops guidance is suffering from important oscillations during the tracking task, whereas the integrated NMPC guidance shows quick convergence to final values without any oscillation. This is a good example of the potential cross coupling between outer and inner loops. As shown before, both loops are separately stable, but not jointly.

4.7.3 Simulation of NMPC with Robustness Augmentation

Robustness tests are performed in low and high frequency ranges. The Meridian UAS is set to fly a straight trajectory.

4.7.3.1 Low Frequency Range Robustness Test

For the trajectory, while the UAS is flying a segment close to 100 knots, the NMPC solution is tested against a low frequency content external disturbance. It is subject to a constant wind of 5 knots from the East. Two similar simulations are compared: an NMPC solution without robustness and NMPC with robustness, given by the use of the frequency-dependent weighting function $W_1 = 1/(z-1)$, in the bank angle channel.

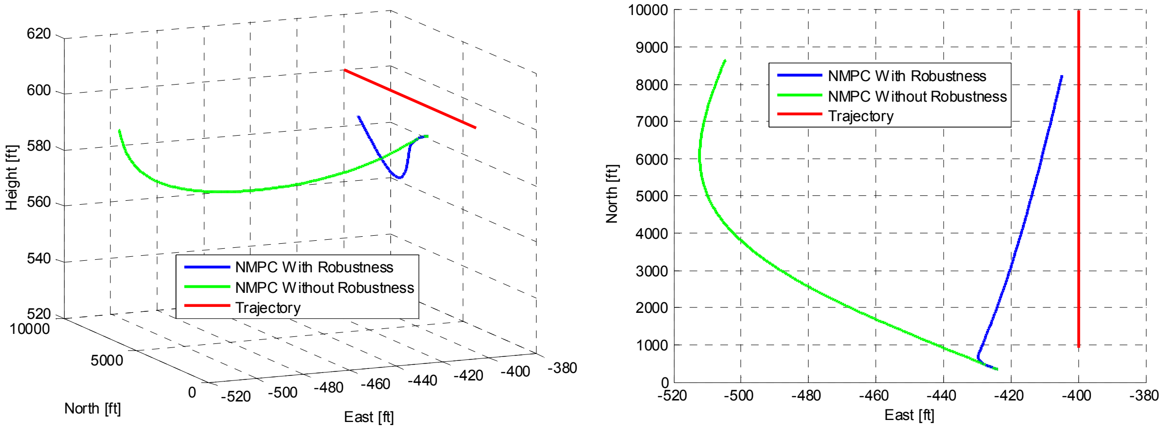

Figure 50. 3D Views Trajectory Following.

Figure 50 shows 3D views where the effects of cross wind on the NMPC without low frequency disturbance rejection and low frequency regulation are apparent.

Figure 51 shows the evolution of the bank angle and aileron as the main implicated variables. Although having greater excursion, both signals still have reasonable values in the robust case, while compensating for the disturbance.

Different weighting functions W_1 can be used, without substantial behavior changes. It will have an impact on the disturbance rejection performance and the consequent flying conditions of the aircraft.

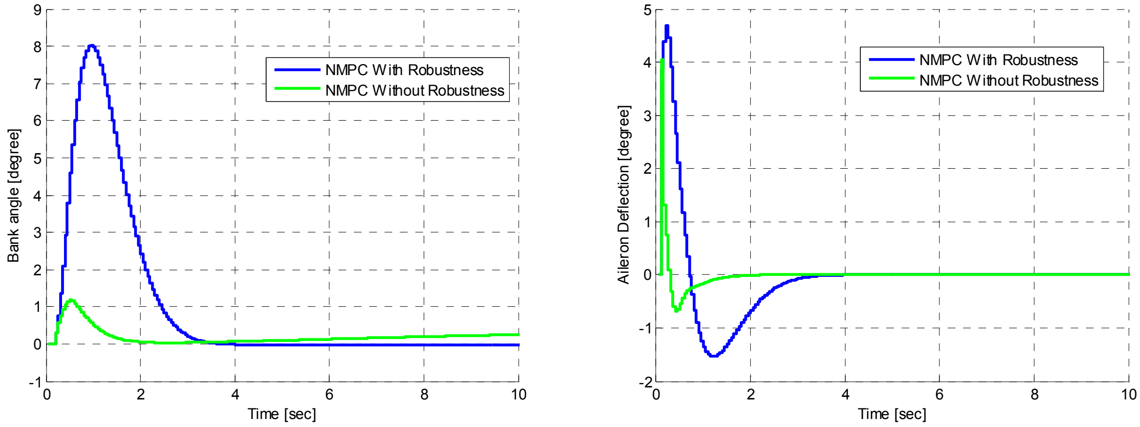

Figure 51. Bank Angle (left) – Aileron Deflection (right).

4.7.3.2 High Frequency Range Robustness Test

The NMPC solution is now tested against higher frequency content measurement noise. During the same type of tracking as before, the UAS is subject to additive measurement noise injected into the pitch angle measurement. Its spectral density is concentrated between 0.5 to 2 [Hz]. Figure 52 show the variables mainly affected.

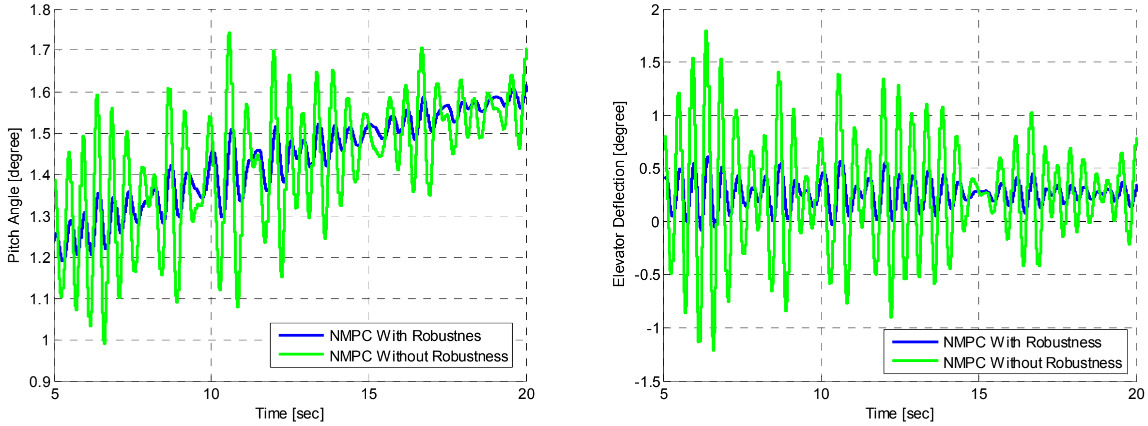

Figure 52. Pitch Angle (left) – Elevator Deflection (right).

Although in general terms, the trajectory following was not severely affected (not shown), it is evident the deleterious effect the noise has on the pitch angle and elevator deflection (as a result of the NMPC elevator command solution) in the non-robust case. By the use of a frequency-dependent function W_3 in the pitch angle channel, the NMPC solution has better performance with fewer oscillations in both signals.

5. UAS Formation Centralized Command

This chapter is understood as the research's motivational incentive. The complete work is embodied in Chapters 1 to 4, and this one is seen as a complement that inspires their development. All efforts put into their expansion, were illuminated by the ideas proposed in Chapter 5.

The content of this Chapter, an exciting and thrilling proposal, is viewed as a possible continuation of what has been studied before, i.e. a starting point.

5.1 Introduction

Studies of aircraft formations are inspired by the physics of birds in formations. Each individual has a limited knowledge of the whole group (constrained by their physiological sensors), there may not be a leader or centralized command, communication or control, but there is a high level of coordination. This coordination follows a very simple set of rules, achieving remarkable objectives as a whole body, so called emergent behavior. Ref. [67] studies the interactions ruling collective animal behavior by reconstructing the three-dimensional position of a few individuals in a flock of birds. It is shown that the interaction does not depend on the metric distance, but rather on the topological distance. Each bird modifies its position relative to the six or seven animals directly surrounding it no matter how close or how far away those animals may be.

One of the most important joint behaviors is optimizing the aerodynamics involved in the birds' maneuvers, decreasing significant amounts of total drag. One of these formations is the V-formation that improves the efficiency and range of flying birds, particularly over long migratory routes. As birds fly in the upwash formed by forward birds in the formation they achieve a better lift to induced drag ratio. A formation of UAS could take advantage of these aerodynamic effects, thereby decreasing their energy consumption. Ref. [68] showed that each bird in a V-formation of 25 members can achieve a reduction of induced drag by up to 65% and 71% increase in range. Ref. [69] explores an approach based on simulating the paths of each bird individually. The simulated flock is an elaboration of a particle system where the simulated birds are the particles. The aggregate motion of the simulated flock is created by a distributed behavioral model much like that at work in a natural flock; the birds choose their own course. The paper utilizes simple mathematical models of swarms given by the following three rules: birds must move in the same direction as their neighbors, remain close to their neighbors, and avoid collisions with their neighbors.

To develop a formation command and control, both the dynamics and the aerodynamics of the formation need to be understood and described by a simple but useful mathematical model (expected to be nonlinear and possibly time-varying). This model can then be used to design a particular control solution. Ref. [70] performs a comparison of predicted and measured formation flight interference effects. It compares the data obtained from a wind-tunnel test of two delta-wing aircraft in close proximity with predictions from a vortex lattice method. Large changes in lift, pitching moment, and rolling moments are found on the trailing aircraft as it moves laterally relative to the lead aircraft. Ref. [71] studies the tight formation flight problem consisting of a leading aircraft and a wing aircraft. The lead's trailing vortices aerodynamically couple both aircraft. In close formation, the wing aircraft experiences an upwash field generated by the preceding aircraft, leading to a reduction in required overall energy. The manuscript defines tight formation as one where the lateral separation between aircraft is less than the wingspan. It delivers an analytical equation for the optimal separation calculation.

Ref. [72] studies the interaction of two aircraft flying in close formation. By positioning one craft flying in the correct position within the vortex wake of the other, substantial reductions in drag may be attained. The reduction amount is dependent on the relative positions of the two aircraft. To maximize the drag reduction, the precise position of the second craft relative to the first, is to be determined online, leading to a peak-seeking control problem. A method of peak-seeking control, using a Kalman filter to estimate the characteristics of the drag reduction, is derived.

Ref. [73] studies the nonlinear control of multiple UAS in close-coupled formation flight for the purposes of drag reduction. Inner and outer loops are employed in a two-UAS formation, where the outer loop controller is based on sliding mode control and the inner uses adaptive dynamic inversion to track the commands. The controller also enables the wing UAS to maneuver in the lead UAS's wake, and to hold a predefined desired position within it. The induced lift on the wing UAS due to formation flight is shown to cause reduced angle-of-attack and thrust for the wing vehicle.

Decentralized control is the main feature of bird formations to be mimicked because it allows more efficient control calculation by using the distributed available computational power. Each airplane is able to synthesize its own control by knowledge of its own status, and possibly that corresponding to its neighbors. Even with the existence of a centralized commanding device, the control calculation is more efficient when done in a distributive way. Ref. [74] proposed a decentralized nonlinear robust control system, based on outer and inner control loops, for close formation flying of multiple UAS. This control system keeps a predefined (based in Ref. [73]) separation distance of the wing airplanes, calculated to achieve the minimum drag. The results confirm that when the wing aircraft is positioned properly in the vortex of the leading aircraft, it experiences reduction in its required flight power. In closed-loop, prescribed separation trajectories are asymptotically tracked by each wing aircraft while the lead aircraft is free to perform trajectory following maneuvers.

As a system, the V-formation is a highly nonlinear and time-varying system, so to guarantee a broader operation range, a set of adaptive and nonlinear control laws are required. The NMPC appears to be a natural candidate as it comprises the desired level of nonlinearity and adaptation management. It also allows explicit accommodation of constraints, granting a remarkable level of optimality. Ref. [75] proposes an algorithm for robust constrained MPC which incorporates coupled constraints. A single large planning optimization is divided into small sub problems, each planning only for the states of a particular subsystem. Relevant plan data is exchanged between subsystems to ensure that all decisions are consistently satisfying the coupled constraints. A typical application would be autonomous guidance of a fleet of UAS, in which the individual aircraft are coupled by the need to avoid collisions, but each vehicle only plans its own path.

Ref. [76] describes the application of a methodology for high-level control and coordination of autonomous vehicle teams. The scheme employs decentralized receding horizon controllers that reside on each vehicle to achieve coordination among team members. For each vehicle, information about its neighbors is used to predict their behavior and plan conflict-free trajectories that maintain coordination and achieve team objectives.

As a variant, Ref. [77] proposes a control strategy using hybrid receding horizon control (RHC) and a differential evolution algorithm based on the nonlinear model of a multi-UAS close formation. Three simulation experiments are performed and the simulation results show the feasibility and validity of the proposed control algorithm. The aerodynamic separation is based on Ref. [73], and Ref. [78] proposes a distributed control scheme to generate stable flocking motion for a group of agents. In this control scheme, a molecular potential field model is applied as the potential. The approach of distributed receding horizon control is adopted to drive each agent to find its optimal control input to lower its potential at every step.

5.2 NMPC Decentralized Control

The NMPC algorithm is highly demanding on computational power and generally limited by the available computational resources on board the aircraft. With available technology, a full centralized formation control is still unfeasible; although decentralized control is becoming a reachable solution (Ref. [79], [80] and [81]). Decentralized control relies on each UAS (also called agent, interchangeably) to compute its own control commands. To have a coherent formation, a centralized command is required.

Each UAS NMPC requires information about its future moving points, i.e. during its horizon length, to perform the predictive calculations. To solve this requirement, and at the same time allow a centralized command to define the location of each aircraft within the formation, centralized trajectory generation logic is required to keep a moving window of the future trajectory of each moving point for each UAS. As defined in the previous chapter, moving points $\boldsymbol{o}$ are the locations where UAS are expected to be, so each element of the moving window, say $\boldsymbol{o}[k+i]$, is the reference location for a UAS to be at time $[k+i]$. This information is required by the NMPC algorithm at each time interval.

The centralized trajectory generator is required to send each UAS the new entry of its moving point window every sample time, i.e. the furthest in time, and each UAS is required to keep the previous sent points. Figure 53 shows the moving window logic for a horizon of N samples, where UAS $\boldsymbol{U}^1$ is approaching its moving point $\boldsymbol{o}$.

Although a desirable capability, each agent is not required to assure collision avoidance within the formation; certainly each agent is not restricted to have the sensors and the logic to determine weakness in safety. Collision avoidance within the formation is one of the main tasks of the centralized command generator. This is one of the main concerns when designing the formation and its dynamics.

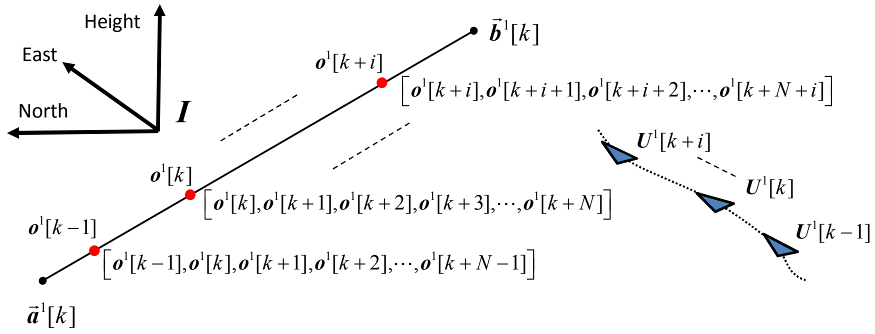
Height
East
North
$\boldsymbol{I}$
$\vec{\boldsymbol{b}}^1[k]$
$\boldsymbol{o}^1[k+i]$
$\left[\boldsymbol{o}^1[k+i],\boldsymbol{o}^1[k+i+1],\boldsymbol{o}^1[k+i+2],\cdots,\boldsymbol{o}^1[k+N+i]\right]$
$\boldsymbol{U}^1[k+i]$
$\boldsymbol{U}^1[k]$
$\boldsymbol{o}^1[k]$
$\left[\boldsymbol{o}^1[k],\boldsymbol{o}^1[k+1],\boldsymbol{o}^1[k+2],\boldsymbol{o}^1[k+3],\cdots,\boldsymbol{o}^1[k+N]\right]$
$\boldsymbol{U}^1[k-1]$
$\boldsymbol{o}^1[k-1]$
$\left[\boldsymbol{o}^1[k-1],\boldsymbol{o}^1[k],\boldsymbol{o}^1[k+1],\boldsymbol{o}^1[k+2],\cdots,\boldsymbol{o}^1[k+N-1]\right]$
$\vec{\boldsymbol{a}}^1[k]$

Figure 53. Moving Point Window Concept.

5.3 Formation Centralized Command

The philosophy of the UAS formation presented here is to have decentralized control, but a centralized command. This is a centralized command logic running and located within the formation, i.e. onboard any formation agent with the capability of synthesizing its own controls and the formation commands. Every agent will be computing its control actions based on the formation command, a very demanding task.

Three structures for multi-agent formation decentralized control and centralized command are present in the literature: leader-follower, virtual structure, and the behavioral approach. Leader-follower seems to be the most used, Ref. [81]. For general applications it provides a simple and direct solution. For model predictive control, where each UAS needs to know the leader's future status within the receding horizon length, it becomes a disadvantage. This is the case of Ref. [77] where it is assumed that the leader will not change its states within the horizon. Similarly in Ref. [82], each vehicle in a centralized formation requires the full future information of all other vehicles over the look-ahead horizon; or in a decentralized formation, each vehicle is required to assume that every other vehicle keep its last actuator commands frozen within the horizon. And in Ref. [83] it is assumed that leader's acceleration vector is known within the horizon, or at least that it is bounded and converges to a constant.

The centralized command generator broadcasts new entries for the moving point windows to each UAS, and each UAS transmits back to the centralized command generator its current position and velocity. Both happens once each time interval, and it is expected to be the only communication. Figure 54 sketches the sharing information organization between the agents and the centralized command

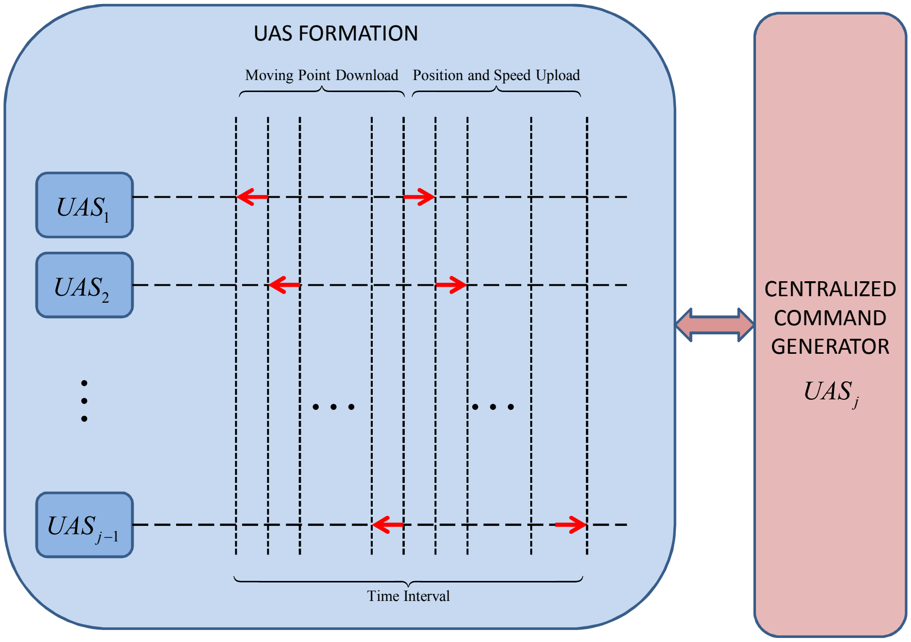

Figure 54. Formation Communication Concept.

generator within a formation of j agents. One of the main requirements for an effective information interchange is the availability of a reliable and efficient communication link between all agents.

The chosen formation structure is a version of the virtual structure approach (Ref. [81] and Ref. [84]). In this approach, the formation has a defined structure and all its agents have defined locations within the formation. When the virtual structure modifies its shape and/or alters its motion based on broadcasted information, all its agents independently manipulate their controls to follow their moving points.

As the command generator keeps sending the formation future commands (i.e. at time $[k]$, the future moving points $\boldsymbol{o}^1[k+N],\boldsymbol{o}^2[k+N],\cdots,\boldsymbol{o}^j[k+N]$ are sent to the formation of j agents), each UAS's NMPC does not need to guess or make assumptions in the case of leader-follower. This requires the command generator be able to determine the virtual structure in advance.

The formation structure has a reference point, denominated $\boldsymbol{o}^{RP}[k]$, and every moving point $\boldsymbol{o}^j[k]$, is referred to it. Figure 55 shows an arbitrary formation at two different times $[k-1]$ and $[k]$, with $j=3$ agents, their moving points located within the formation and its reference point.

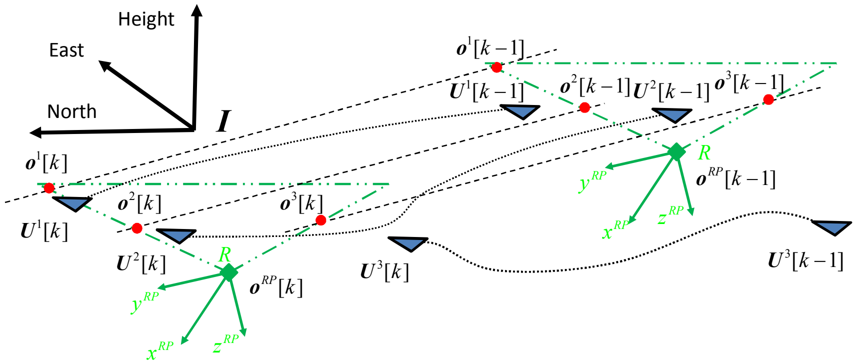

Figure 55. Arbitrary UAS Formation.

It is useful to define a formation coordinate system $\bar{\boldsymbol{R}}$ centered in the reference point $\boldsymbol{o}^{RP}[k]$, which is at the same time referred to the inertial coordinate system $\boldsymbol{I}$ by three rotation angles $\{\phi^{RP}[k],\theta^{RP}[k],\psi^{RP}[k]\}$, an inertial position $\{p_N^{RP}[k],p_E^{RP}[k],p_H^{RP}[k]\}$, and inertial speed $\{v_N^{RP}[k],v_E^{RP}[k],v_H^{RP}[k]\}$. Any formation shape readjusting, formation speed increase or decrease, or formation reorientation is performed over the reference point $\boldsymbol{o}^{RP}[k]$ and its coordinate system. Figure 55 also shows the coordinate system $\bar{\boldsymbol{R}}$ and its current orientation with respect to the inertial frame $\boldsymbol{I}$.

Each moving point $\boldsymbol{o}^{j}[k]$ is referred to the reference point $\boldsymbol{o}^{RP}[k]$ by three components in the coordinate system $\bar{\boldsymbol{R}}$, and if there are no modifications to the formation in the near future, their positions appear static. To compute the moving point's time-varying inertial location and speed, these relative quantities are transferred to the inertial frame. For moving point $\boldsymbol{o}^{j}[k]$, associated to UAS j, located at $\{p_X^{\boldsymbol{O}^j}[k], p_Y^{\boldsymbol{O}^j}[k], p_Z^{\boldsymbol{O}^j}[k]\}$ in $\bar{\boldsymbol{R}}$, its inertial position coordinates $\{p_N^{\boldsymbol{O}^j}[k], p_E^{\boldsymbol{O}^j}[k], p_H^{\boldsymbol{O}^j}[k]\}$ are given by

$$\begin{aligned}
p_N^{\boldsymbol{O}^j}[k] &= p_N^{RP}[k] + \\
&\quad p_X^{\boldsymbol{O}^j}[k]\cos\phi^{RP}[k]\cos\psi^{RP}[k] + \\
&\quad p_Y^{\boldsymbol{O}^j}[k](-\cos\phi^{RP}[k]\sin\psi^{RP}[k] + \sin\phi^{RP}[k]\sin\theta^{RP}[k]\cos\psi^{RP}[k]) + \\
&\quad p_Z^{\boldsymbol{O}^j}[k](\sin\phi^{RP}[k]\sin\psi^{RP}[k] + \cos\phi^{RP}[k]\sin\theta^{RP}[k]\cos\psi^{RP}[k]) \\
p_E^{\boldsymbol{O}^j}[k] &= p_E^{RP}[k] + \\
&\quad p_X^{\boldsymbol{O}^j}[k]\cos\theta^{RP}[k]\sin\psi^{RP}[k] + \\
&\quad p_Y^{\boldsymbol{O}^j}[k](\cos\phi^{RP}[k]\cos\psi^{RP}[k] + \sin\phi^{RP}[k]\sin\theta^{RP}[k]\sin\psi^{RP}[k]) + \\
&\quad p_Z^{\boldsymbol{O}^j}[k](-\sin\phi^{RP}[k]\cos\psi^{RP}[k] + \cos\phi^{RP}[k]\sin\theta^{RP}[k]\sin\psi^{RP}[k]) \\
p_H^{\boldsymbol{O}^j}[k] &= p_H^{RP}[k] + \\
&\quad p_X^{\boldsymbol{O}^j}[k]\sin\theta^{RP}[k] - \\
&\quad p_Y^{\boldsymbol{O}^j}[k]\sin\phi^{RP}[k]\cos\theta^{RP}[k] - \\
&\quad p_Z^{\boldsymbol{O}^j}[k]\cos\phi^{RP}[k]\cos\theta^{RP}[k]
\end{aligned} \tag{109}$$

It is clear that even for a fixed rotation from $\boldsymbol{I}$ to $\bar{\boldsymbol{R}}$, i.e. time invariant, inertial location will vary as the whole structure is moving. For a local speed $\{v_X^{\boldsymbol{O}^j}[k], v_Y^{\boldsymbol{O}^j}[k], v_Z^{\boldsymbol{O}^j}[k]\}$, referenced to $\boldsymbol{o}^{RP}[k]$ in $\bar{\boldsymbol{R}}$, when the structure is changing its shape, for example, its inertial speed $\{v_N^{\boldsymbol{O}^j}[k], v_E^{\boldsymbol{O}^j}[k], v_H^{\boldsymbol{O}^j}[k]\}$ is given by

$$
\begin{aligned}
v_N^{\boldsymbol{O}^j}[k] = {} & v_N^{RP}[k] + \\
& v_X^{\boldsymbol{O}^j}[k]\cos\phi^{RP}[k]\cos\psi^{RP}[k] + \\
& v_Y^{\boldsymbol{O}^j}[k](-\cos\phi^{RP}[k]\sin\psi^{RP}[k] + \sin\phi^{RP}[k]\sin\theta^{RP}[k]\cos\psi^{RP}[k]) + \\
& v_Z^{\boldsymbol{O}^j}[k](\sin\phi^{RP}[k]\sin\psi^{RP}[k] + \cos\phi^{RP}[k]\sin\theta^{RP}[k]\cos\psi^{RP}[k]) \\
v_E^{\boldsymbol{O}^j}[k] = {} & v_E^{RP}[k] + \\
& v_X^{\boldsymbol{O}^j}[k]\cos\theta^{RP}[k]\sin\psi^{RP}[k] + \\
& v_Y^{\boldsymbol{O}^j}[k](\cos\phi^{RP}[k]\cos\psi^{RP}[k] + \sin\phi^{RP}[k]\sin\theta^{RP}[k]\sin\psi^{RP}[k]) + \\
& v_Z^{\boldsymbol{O}^j}[k](-\sin\phi^{RP}[k]\cos\psi^{RP}[k] + \cos\phi^{RP}[k]\sin\theta^{RP}[k]\sin\psi^{RP}[k]) \\
v_H^{\boldsymbol{O}^j}[k] = {} & v_H^{RP}[k] + \\
& v_X^{\boldsymbol{O}^j}[k]\sin\theta^{RP}[k] - \\
& v_Y^{\boldsymbol{O}^j}[k]\sin\phi^{RP}[k]\cos\theta^{RP}[k] - \\
& v_Z^{\boldsymbol{O}^j}[k]\cos\phi^{RP}[k]\cos\theta^{RP}[k]
\end{aligned}
\tag{110}
$$

Even when a fixed structure changes its orientation, i.e. $\{\phi^{RP}[k], \theta^{RP}[k], \psi^{RP}[k]\}$ change, the agent's moving point inertial speed also experiences changes.

Figure 56 and Figure 57 show two basic formation maneuvers, a readjustment and a horizontal turn of its agent moving point's positions. The first maneuver consists of switching from a V-formation to a line formation and the second, a reorientation of the formation where moving points change their formation coordinates from $\{p_X^{\boldsymbol{O}^j}[k], p_Y^{\boldsymbol{O}^j}[k], p_Z^{\boldsymbol{O}^j}[k]\}$ to $\{p_X^{\boldsymbol{O}^j}[k+i], p_Y^{\boldsymbol{O}^j}[k+i], p_Z^{\boldsymbol{O}^j}[k+i]\}$, after i interval times, and their inertial coordinates from $\{p_N^{\boldsymbol{O}^j}[k], p_E^{\boldsymbol{O}^j}[k], p_H^{\boldsymbol{O}^j}[k]\}$ to $\{p_N^{\boldsymbol{O}^j}[k+i], p_E^{\boldsymbol{O}^j}[k+i], p_H^{\boldsymbol{O}^j}[k+i]\}$.

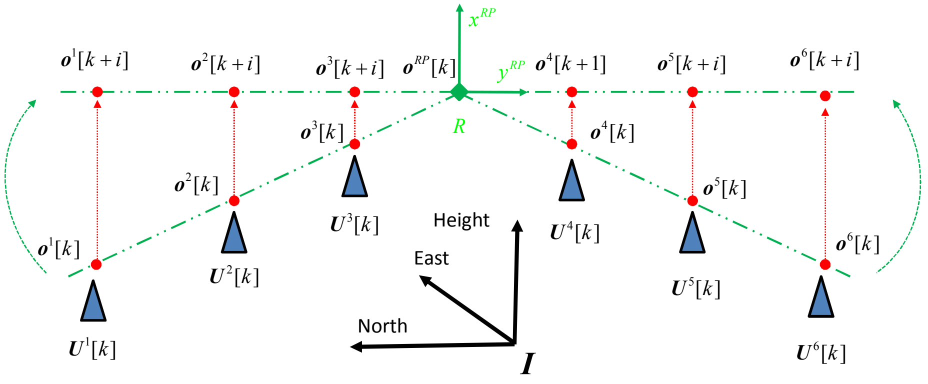

Figure 56. UAS Formation Readjustment.

One essential assumption for the present centralized commanding approach is the requirement of position information of each agent location, either referred to the inertial coordinate system $\boldsymbol{I}$ or referred to the reference point. It is assumed that each UAS is able to determine its own location by using a passive sensor, i.e. a GPS receiver, and send this information to the centralized device; or, it is assumed that there is an active sensor onboard the formation, i.e. a radar onboard one or more agents, able to determine the relative position of the agents within the formation.

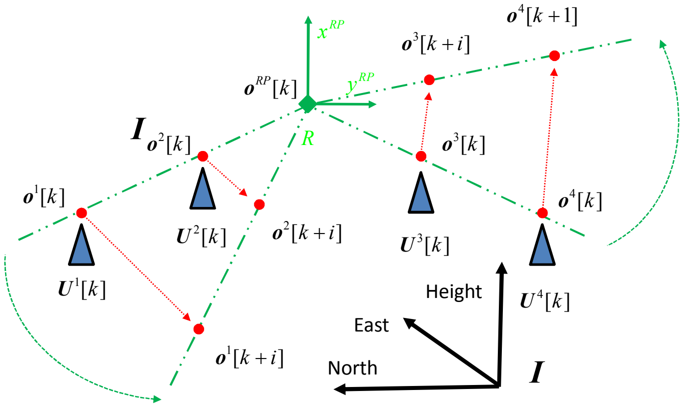

Figure 57. UAS Formation Reorientation.

This assumption is critical from the safety point of view. When an agent fails to follow its moving points due to some failure, its current location still is useful to the centralized command generator to adjust the formation to avoid any collision derived from this anomaly.

5.4 Aerodynamic Coupling in Formation Flight

A significant number of papers have been devoted to the study of the aerodynamic interaction between airplanes flying close enough to couple their surrounding wind fields. Different techniques have been used to model this interaction. In principle, the physics behind the interaction has a field interpretation, i.e. a vector field, and a rigorous description should consider a 3D continuous approach. This complexity is intractable from a practical control and command design point of view, and simplifications are required in order to arrive at a feasible controlled and commanded system.

Studies are traced back to Ref. [103] where a vortex-lattice method was applied to a formation of three wings to calculate forces and moments and the potential benefits of flying aircrafts in formation. Two formations were analyzed, the V-formation and the double row formation, showing the same overall benefits as a whole. Their ranges were potentially increased from 46% to 67%. They showed that the following aircraft in a V-formation required trimming in roll, while aircraft in the second row in a double row formation do not have out-of-balance rolling moments. Also, the second line aircraft saw double the benefits compared to the V-formation following aircraft, while the first line does not benefit at all. Lastly, it was found that both formations are insensitive to small longitudinal movements, but benefits (and trimming requirements) decrease rapidly with lateral and vertical movements from the optimum position.

Wind tunnel experiments have also been conducted, as in Ref. [70] and Ref. [104]. The latest worked with two delta-wing aircraft in close proximity and made comparisons between measured and predicted effects. Large changes in lift, pitching moment, and rolling moment are found on the trailing aircraft as it moves relative to the leading aircraft. A maximum induced drag reduction of 25% is measured as opposed to a predicted 40% in the trailing aircraft. Important flight tests have been done largely confirming the estimated benefits, see Ref. [105].

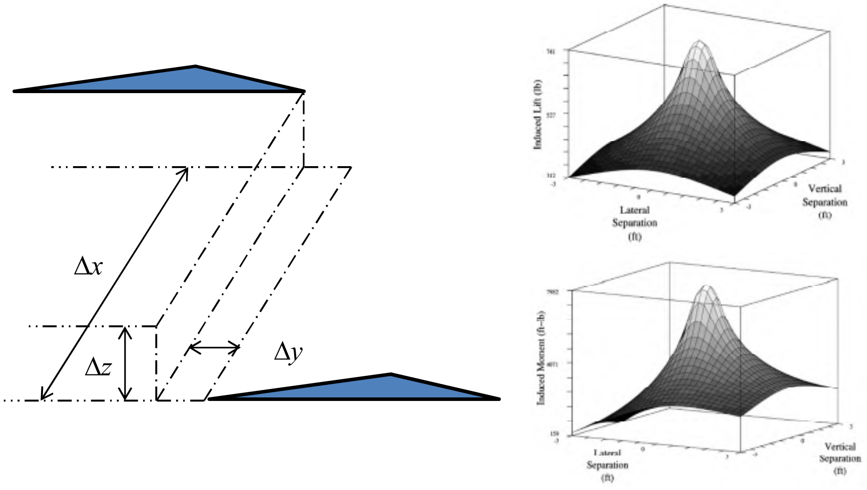

Figure 58. Estimated Induced Lift and Rolling Moment (from Ref. [72]).

Another research branch has been focused on synthetically mapping primary induced aerodynamic effects, i.e. induced drag and lift forces, and induced pitching and rolling moments experienced by a following aircraft. By using a horseshoe vortex representation and assuming inviscid flow in order to capture the dominant effects, maps with defined maximum peaks are obtained as a function of vertical and lateral separations (Ref. [72] and [108]), as shown in Figure 58 (based on Ref. [72]). Over a minimum safety distance, i.e. on the order of a few wing spans, longitudinal distance becomes irrelevant as it has little effect compared with the other two, so normally it is discarded.

Based on the convergent conclusions of existing literature, present research is attempting to consider this aerodynamic effect in the command design of the formation. The approach has more concept proofing and command generation enabling rather than confirming or contributing to the modeling of these aerodynamic couplings.

The chosen solution attempts to capture the most relevant physical features but at the same time to keep the aerodynamic coupling model as simple as possible to facilitate a feasible formation commanding design.

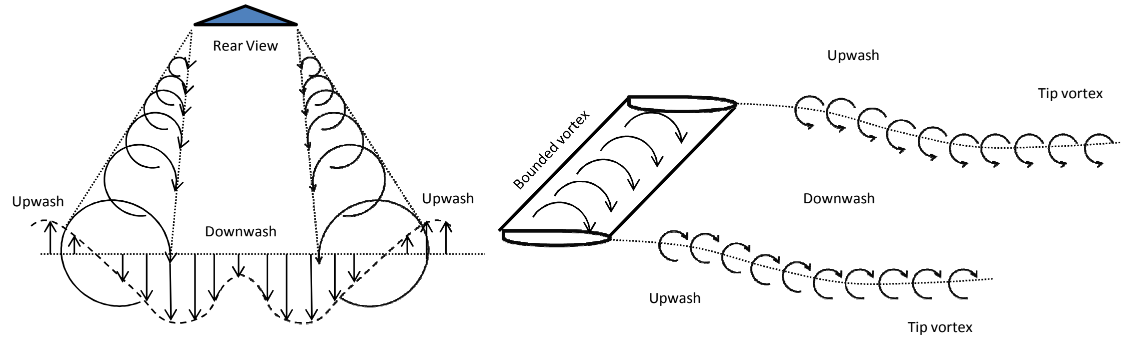

Figure 59. Regions of Upwash and Downwash.

The idea, sketched in Figure 59 (based on Ref. [106]), will be based on a discrete wind field pattern formed behind each aircraft, shaped as a function of states of the aircraft states, and time dependent. Equation (111) defines the relationship

$$\begin{aligned}\vec{\boldsymbol{w}}^{U^j}_{XYZ}[x,y,z,k] = (&w_X^{U^j}[V_T^{U^j},\alpha^{U^j},\beta^{U^j},\phi^{U^j},\theta^{U^j},\psi^{U^j},P^{U^j},Q^{U^j},R^{U^j},p_X^{U^j},p_Y^{U^j},p_Z^{U^j},k],\\ &w_Y^{U^j}[V_T^{U^j},\alpha^{U^j},\beta^{U^j},\phi^{U^j},\theta^{U^j},\psi^{U^j},P^{U^j},Q^{U^j},R^{U^j},p_X^{U^j},p_Y^{U^j},p_Z^{U^j},k],\\ &w_Z^{U^j}[V_T^{U^j},\alpha^{U^j},\beta^{U^j},\phi^{U^j},\theta^{U^j},\psi^{U^j},P^{U^j},Q^{U^j},R^{U^j},p_X^{U^j},p_Y^{U^j},p_Z^{U^j},k])\end{aligned} \tag{111}$$

where $w_X^{U^j}$, $w_Y^{U^j}$ and $w_Y^{U^j}$ are the wind vector field components associated to agent $\boldsymbol{U}^j$, referred to coordinate systems $\bar{\boldsymbol{R}}$. These components will be a function of the states of each agent $\boldsymbol{U}^j$, i.e. $\{V_T^{U^j},\alpha^{U^j},\beta^{U^j},\phi^{U^j},\theta^{U^j},\psi^{U^j},P^{U^j},Q^{U^j},R^{U^j}\}$ and relative location $\{p_X^{U^j}[k],p_Y^{U^j}[k],p_Z^{U^j}[k]\}$. The extension of the wind field body should be carefully bounded, i.e. where there are no significant changes with respect to the free stream, there is no requirement to perform any wind field calculation.

The compound wind field, adding contributions from all aircraft within the formation, is then described as

$$\sum_{n=1}^{j} \vec{\boldsymbol{w}}^{\boldsymbol{U}^j}[x,y,z,k] = (\sum_{n=1}^{j} w_X^{\boldsymbol{U}^j}[x,y,z,k], \sum_{n=1}^{j} w_Y^{\boldsymbol{U}^j}[x,y,z,k], \sum_{n=1}^{j} w_Z^{\boldsymbol{U}^j}[x,y,z,k]) \tag{112}$$

The overall wind field, i.e. the one filling the whole formation, will be the superposition of all patterns and the inertial wind existing in the area. The accuracy or level of fidelity of the 3D wind field is not the objective here; its only purpose is to provide a simple characterization of a wind field to be used by the command generator optimization task. Availability of a better 3D wind field may replace the current model, but the optimal moving point generator logic will still be the same.

5.5 Optimal Command Generation

The outputs of the command generator are the future moving points, i.e. the newest element of the respective moving windows $\boldsymbol{o}^1[k+N], \boldsymbol{o}^2[k+N], \cdots, \boldsymbol{o}^j[k+N]$ within the formation.

The formation is expected to fulfill some objective, e.g. transferring the aircraft from one location to another while minimizing their fuel consumption, achieving particular land coverage when carrying some type of remote sensor, etc. For this task, a formation cost function V^{RP} will be in place as the algorithm's core of the command generator. It will fed by different sources of information, the wind field being one of the most important. Equation (113) defines the problem to be solved

$$\mathrm{P}_N^{RP} = \min_{\boldsymbol{o}^1[k+N], \boldsymbol{o}^2[k+N], \cdots, \boldsymbol{o}^j[k+N]} \left\{ V^{RP}(\Im^{RP}, \boldsymbol{o}^1[k+N], \boldsymbol{o}^2[k+N], \cdots, \boldsymbol{o}^j[k+N]) \right\} \tag{113}$$

where the cost function V^{RP} is defined over a predefined trajectory $\Im^{RP}$, either loaded before taking off, updated or uploaded during flight. The outputs of the minimization process are the future moving points $\boldsymbol{o}^1[k+N], \boldsymbol{o}^2[k+N], \cdots, \boldsymbol{o}^j[k+N]$.

For the given assumptions, a numerical minimization approach will also allow the inclusion of obstacle avoidance, for particular agents or for the entire formation, and possible redistribution of the agents. Although obstacles could be classified as geographical and assumed static as well as non-geographical and assumed mobile, from the avoidance logic both categories are indistinguishable and treated in the same way.

The obstacle collision avoidance logic is supported by the requirement of an active sensor centered in the formation with a predefined range of detection. This range is an important parameter to be considered in the formation's cost function. This sensor is assumed to be able to detect the obstacle within the range and give its relative location.

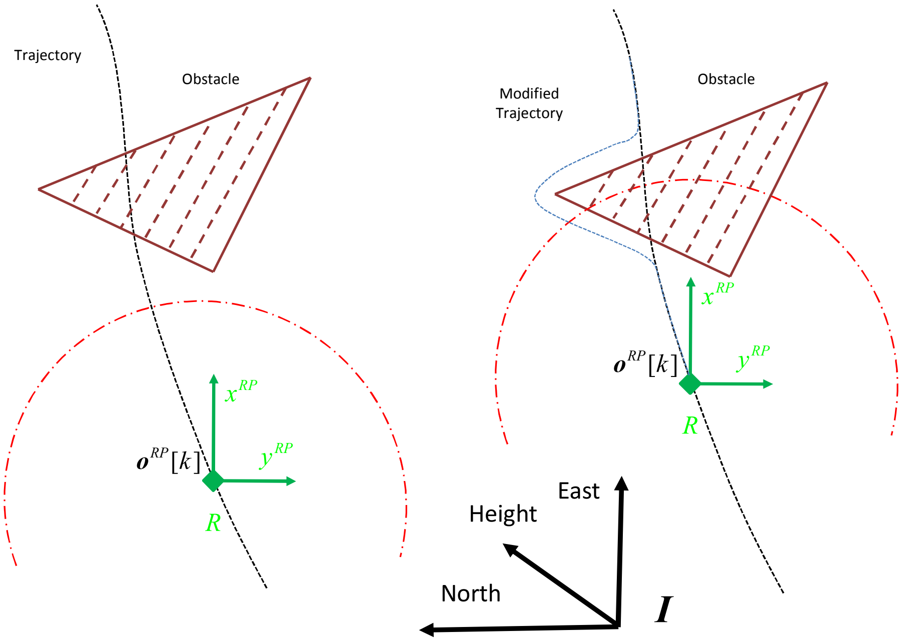

Figure 60. Trajectory Adjustment for Obstacle Avoidance.

The idea of avoidance is stated in two independent logics that could be integrated or kept separate, based on the available computational power. The first one is the formation trajectory adjustment and is independent of the current formation distribution. It only requires knowledge of the maximum formation shape dimension. Its main task is to redefine the trajectory of the formation to avoid the obstacle while minimizing the departure from the original trajectory, see Figure 60. The second logic is the redistribution of the agents within the formation, see Figure 56 and Figure 57. The idea is that both logics could work separately or integrated, if feasible.

Both logics will be based on cost function minimization and will not deal with aircraft dynamics. They will minimize kinematic relations involving positions and velocities of the agent's moving points or positions of the adjusted trajectory. This greatly simplifies the numerical complexity of the algorithm.

5.6 Centralized Command Communication Delay Robustness

The philosophy of the centralized command of sending future moving points $\boldsymbol{o}^1[k+N], \boldsymbol{o}^2[k+N], \cdots, \boldsymbol{o}^j[k+N]$ inherently provides a moderate robustness, in the face of communication delays.

Each agent keeps its full future horizon N available for NMPC computations, i.e. $\boldsymbol{o}^j[k], \boldsymbol{o}^j[k+1], \cdots, \boldsymbol{o}^j[k+N]$. In a temporary communication link failure, characterized by $f < N$ lost moving points, where UAS misses moving points $\boldsymbol{o}^j[k+N-f+1], \boldsymbol{o}^j[k+N-f+2], \cdots, \boldsymbol{o}^j[k+N]$, the natural solution is to adaptively shrink its horizon length to include only previously received moving points, i.e. $\boldsymbol{o}^j[k], \boldsymbol{o}^j[k+1], \cdots, \boldsymbol{o}^j[k+N-f]$. Once the communication is restored, the UAS can use the recently received moving points and the previous ones before the incident, to interpolate the missing elements and build an estimated full horizon.

The length of the NMPC horizon N is normally dictated by the available onboard computational power, using a horizon length as large as possible. As a way to increase communication robustness, it is possible to define a second horizon $N_f > N$, not used by the NMPC but put in place to lessen the effects of communication failure. If N_f is sufficiently large, then the chance of affecting the NMPC horizon could be reduced.

The choice of the design parameter N_f should be guided by two considerations. One is the expected length of a communication loss, based on existing data or existing statistics, and the other, the expected level of unpredicted obstacles coupled wih the extent of associated early warnings. The last consideration will be strongly shaped by the availability and type of sensors. In the present work no particular sensor is chosen, and a real-time detection region centered in the formation is assumed for the purpose of the formation cost function V^{RP} minimization.

With respect to the expected length of communication failure, this estimation can be guided by existing similar data. Ref. [85] reports an uplink communication drop out between ground station and aircraft COTS autopilot up to 2.5 seconds. Although this case is not fully representative, it at least gives a real value under comparable conditions, worthy as a starting point.

Conclusions

An integrated Trajectory/Guidance/State Estimation/Controller system was developed motivated by the desire to create an advanced Flight Control System for future unmanned aerial vehicles that have nonlinear/time-varying dynamics, track complex trajectories and are subject to external disturbances. As a result of laboratory testing using standard simulation techniques as well as flight test data, the following conclusions have been reached:

- ➢ A new nonlinear guidance law provides a stable attitude and airspeed command generation for arbitrary trajectory following. This development is expanding an existing nonlinear lateral guidance law by the addition of a new longitudinal law, providing a full nonlinear attitude guidance law. Specifically, the new guidance law can achieve a stable and steady state error free guidance through the use of integral effect, and by lifting constraints related to maximum distances and relative position with respect to the trajectory, expanding them to the entire state space.
- ➢ An original contribution has been achieved by the integration of the guidance and control loops into a single task. This provides an effective solution to minimize mutual interference and cross coupling between these two distinct loops, in contrast to decoupled classical inner and outer loop approaches. The integrated formulation allows the computation of a single solution for both guidance and control tasks. This solution is achieved without the induction of oscillations and reduction of stability seen in nested inner and outer loops.
- ➢ Using the Meridian flight test data, it is demonstrated that adaptive artificial neural networks utilizing the moving window concept and modified adaptive Levenberg's parameter have the ability to adjust and capture fast changes in aerodynamic characteristics of the unmanned aerial vehicle in real-time. In contrast with static artificial neural networks, the adaptive artificial neural networks provide an accurate online, nonlinear, and unsteady estimation of aircraft dynamics regardless of the flight control modes (e.g. manual or autonomous). This original development shows that online

adaptive neural networks are capable of updating the aircraft physics-based model accurately and in real-time with a 10 Hz update rate.

- By the application of the mixed sensitivity theory through the addition of frequency-dependent weights, the lack of robustness of the time-based nonlinear model predictive control approach is attenuated dramatically. This novel configuration has successfully enhanced the low frequency tracking capability and disturbance attenuation, and high frequency measurement noise and model uncertainty rejection. Specifically for the Meridian unmanned aerial system, the added robustness to the nonlinear model predictive controller reduces the commanded control values while keeping the attitude in the desired range. For example, in a specific case, when the Meridian was commanded to follow a trajectory in presence of 5 Knots crosswind, the aileron commanded value was reduced by 300% and the aircraft achieved a better trajectory following with maximum 15 degrees bank angle.

The feasibility of a robust nonlinear model predictive control for an unmanned aerial system, enhanced by the addition of adaptive artificial neural networks, and by the integration of nonlinear command and control tasks, has been achieved in high-fidelity simulations.

Future Work

The following topics are seen as logic next steps derived from the present research.

Near future:

Adapt the MATLAB scripts to an operation-based structure, using Simulink software.

Intermediate future:

Perform flights to test the developed decentralized solution for individual agents.

Distant future:

Develop a centralized formation command based on the flight tested decentralized solution.

References

[1] Ljung, L., *System Identification, Theory for the User*, 2nd ed. Prentice Hall, Englewood Cliffs, NJ, 1999.

[2] Stengel, R., *Flight Dynamics*, Princeton University Press, Princeton, New Jersey, 2004.

[3] Stevens, B. and Lewis, F., *Aircraft Control and Simulation*, 2nd ed., John Wiley & Sons, Hoboken, New Jersey, 2003.

[4] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls (part I)*, DARcorporation, Lawrence, KS, 2003.

[5] Maine, R., E., and Iliff, K., W., "Application of Parameter Estimation to Aircraft stability and Control, The output error Approach", NASA Reference Publication 1168, 1986.

[6] Donegan, J., J., Robinson, S., W., Jr., and Gates, O., B., Jr., "Determination of Lateral-Stability Derivatives and Transfer Function Coefficients from Frequency Response Data for Lateral Motion", NACA Report-1225, 1955.

[7] Chandler, P. R., Pachter, M., and Mears, M., "System Identification for Adaptive and Reconfigurable Control", *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, pp. 516-524, 1995.

[8] Smith, B., and Malcolm, G., "Determination of Non-Linear Dynamic Aerodynamic Coefficients for Aircraft", SBIR N96-017 Report, Naval Air Systems Command, Torrance, CA, 1997.

[9] Goman, M., and Khrabrov, A., "State-Space Representation of Aerodynamics Characteristics of an Aircraft at High Angles of Attack", *Journal of Aircraft*, Vol. 31, No. 5, pp. 1109-1115, 1994.

[10] Wang, L.-X., and Mendel, J., M., "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-Squares Learning", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 807-814, 1992.

[11] Stinchcombe, M., and Whit, H., "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, Vol. 2, pp. 359-366, 1989.

[12] Chang, R., C., Ye, C.-E., Lan, C. E., and Lee, Y., “Nonlinear and Dynamic Aerodynamic Models for Commercial Transport Aircraft with Adverse Weather effects”, Hindawi Publishing Corporation, *Mathematical Problems in Engineering*, Vol. 2010, Article, ID 924275, 17 pages.

[13] Lan, C.E.; Keshmiri, S.; and Hale, R., “Fuzzy-Logic Modeling of a Rolling Unmanned Vehicle in Antarctica Wind Shear”, *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, pp. 1538-1547, 2012.

[14] Larsson, M., De Raedt, P., and Hedlund, M., “Aerodynamic Identification using Neural Networks”, Department of Electrical Engineering, Linköping University, Rept. LiTH-154-1937, Linköping, Sweden, 1997.

[15] Linse, D., J., and Stengel, R., F., “Identification of Aerodynamic Coefficients Using Computational Neural Networks”, *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, pp. 1018-1025, 1993.

[16] Faller, W., E., and Schreck, S., J., “Real Time Prediction of Unsteady Aerodynamics: Application for Aircraft Control and Maneuverability Enhancement”, *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, pp. 1461-1468, 1995.

[17] Raisinghani, S. C., Reddy, G. V. S., and Ghosh, A. K., “Parameter Estimation of An Augmented Aircraft using Neural Networks”, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Portland, OR, 1999.

[18] Raisinghani, S. C., and Ghosh, A. K., “Parameter Estimation of An Unstable Aircraft in Turbulent Atmosphere using Neural Networks”, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, CA, 2002.

[19] Peyada, N., K., and Ghosh, A., K., “Aircraft Parameter Estimation using a New Filtering Technique based upon Neural Network and Gauss-Newton Method”, *The Aeronautical Journal*, Vol. 113, No. 1142, pp. 243-252, 2009.

[20] Das, S., Kuttieri, R., A., Sinha, M., and Jategaonkar, R., “Neural Partial Differential Method for Extracting Aerodynamic Derivatives from Flight Data”, *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, pp. 376-384, 2010.

[21] Rios, J. A., and White, E., “Fusion Filter Algorithm Enhancements for a MEMS GPS/IMU”, Proceedings of the 14th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2001), pp. 1382-1393, Salt Lake City, Utah, September 2001.

[22] Ren, Y., and Ke, X., “Particle Filter Data Fusion Enhancements for MEMS-IMU/GPS”, *Intelligent Information Management*, Vol. 2, pp. 417-421, 2010.

[23] Van der Merwe, R. and Wan, E., “Sigma-Point Kalman Filters for Integrated Navigation”, Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION), Dayton, Ohio, 2004.

[24] Gross, J., et al., “A Comparison of Extended Kalman Filter, sigma-Point Kalman Filter, and Particle Filter in GPS/INS Sensor Fusion”, AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, 2010.

[25] Nemra, A., and Aouf, N., “Robust INS/GPS Sensor Fusion for UAV Localization using SDRE Nonlinear Filtering”, *IEEE Sensors Journal*, Vol. 10, No. 4, April 2010.

[26] Eldredge, A., “Improved State Estimation for Miniature Air Vehicles” M.Sc. Thesis, Department of Mechanical Engineering, Brigham Young University, Provo, Utah, 2006.

[27] Pachter, M., Ceccarelli, N., and Chandler, P., “Estimating MAV’s Heading and the Wind Speed and Direction Using GPS, Inertial, and Air Speed Measurements”, AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, August, 2008.

[28] Gan, Q., and Harris, C., J., “Comparison of Two Measurement Fusion Methods for Kalman-Filter-Based Multisensor Data Fusion”, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No.1, pp. 273-279, 2001.

[29] Duncan R., C., and Gunnersen, A., S., Jr., “Inertial Guidance, Navigation, and Control”, *AIAA Journal of Spacecraft and Rockets*, Vol. 1, No. 6, pp. 577-587, 1964.

[30] Kaminer, I., Pascoal, A., Hallberg, E., and Silvestre, C., “Trajectory Tracking for Autonomous Vehicles: An Integrated Approach to Guidance and Control”, *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, January-February, 1998.

[31] Niculescu, M., “Lateral Track Control Law for Aerosonde UAV”, 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2001.

[32] Boyle, D., and Chamitoff, G., “Robust Nonlinear LASSO Control: A New Approach for Autonomous Trajectory Tracking”, AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas, 2003.

[33] Park, S., Deyst, J. and How, J., “A New Nonlinear Guidance Logic for Trajectory Tracking”, AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, 2004.

[34] Osborne, J., and Rysdyk, R., “Waypoint Guidance for Small UAVs in Wind”, Infotech@Aerospace, AIAA, Arlington, Virginia, 2005.

[35] Rysdyk, R., “Unmanned Aerial Vehicle Path Following for Target Observation in Wind”, *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, September-October 2006.

[36] Nelson, D., Blake Barber, D., McLain, T. W., and Beard, R., “Vector Field Path Following for Miniature Air Vehicle”, *IEEE Transactions on Robotic*, Vol. 23, No. 3, June, 2007.

[37] Sonneveldt, L, van Port, E. R., Chu, Q. P., and Mulder, J. A., “Nonlinear Adaptive Trajectory Control Applied to an F-16 Model”, *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, January-February 2009.

[38] Low., C., B., “A Trajectory Tracking Control Design for Fixed-wing Unmanned Aerial Vehicles”, IEEE International Conference on Control Applications, pp. 2118-2123, Yokohama, Japan, 2010.

[39] Stilwell, D., J., and Rugh, W., J., “Stability Preserving Interpolation Methods for the Synthesis of Gain Scheduled Controllers”, *Automatica*, Vol. 36, No. 5, pp. 665-671, 2000.

[40] Rugh, W., and Shamma, J., “Research on Gain Scheduling”, *Automatica*, Vol. 36, No. 10, pp. 1401–1425, 2000.

[41] Leith, D., J., and Leithhead, W., E., “Survey of Gain-scheduling Analysis & Design”, *Int. Journal of Control*, Vol. 73, No. 11, pp. 1001-1025, 2000.

[42] Nichols, R., Reichert, R., and Rugh, W., “Gain Scheduling for H-Infinity Controllers: A Flight Control Example”, *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 2, pp. 69–79, 1993.

[43] Aouf, N., and Boulet, B., “A Gain Scheduling Approach for a Flexible Aircraft”, in Proc. American Control Conference, Anchorage, AK, May, pp. 4439-4442, 2002.

[44] Wu, F., Packard, A., and Balas, G., “Systematic Gain-Scheduling Control Design: A Missile Autopilot Example”, *Asian Journal of Control*, Vol. 4, No. 3, pp. 341-347, 2002.

[45] Natesan, K., Gu, D.-W., Postlehwaite, I., and Chen, J., “Design of Flight Controllers based on Simplified LPV model of a UAV”, Proc. of the 45th IEEE Decision and Control, San Diego, CA, USA, Dec., 2006.

[46] Richardson, T., Lowenberg, M., Jones, C., and Dubs, A., “Dynamic Gain Scheduled Control of a Hawk Scale Model”, *The Aeronautical Journal*, Vol. 111, No. 1121, pp. 461-469, 2007.

[47] Mayne, D. Q., Rawlings, C., Rao, C., and Scokaert, P. O. M., “Constrained Model Predictive Control: Stability and Optimality”, *Automatica*, Vol. 36, No. 6, 2000.

[48] Sutton, G., and Bitmead, R., “Experiences with Model Predictive Control applied to a Nonlinear Constrained Submarine”, Proceedings of the 37th IEEE Conference on Decision & Control, Vol. 2, Tampa, Florida, pp. 1370-1375, 1998.

[49] Kim, H. J., Shim, D. H., and Sastry, S., "Nonlinear Model Predictive Tracking Control for Rotorcraft-based Unmanned Aerial Vehicles", Proceedings of the American Control Conference, Vol. 5, Anchorage, Alaska, pp. 3576-3581, 2002.

[50] Castillo, C. L., Moreno, W., and Valavanis, K. P., "Unmanned Helicopter Waypoint Trajectory Tracking using Model Predictive Control", Proceedings of the 15th Mediterranean Conference on Control and Automation, Athenas, pp. 1-8, 2007.

[51] Frew, E., "Comparison of Lateral Controllers for Following Linear Structures Using Computer Vision", American Control Conferences, Minneapolis, 2006.

[52] Eklund, J. M., Sprinkle, J., and Sastry, S., "Implementing and Testing a Nonlinear Model Predictive Tracking Controller for Aerial Pursuit/Evasion Games on a Fixed Wing Aircraft", American Control Conferences, Vol. 3, pp. 1509-1514, 2005.

[53] Kang, Y., and Hedrick, J. K., "Linear Tracking for a Fixed-Wing UAV Using Nonlinear Model Predictive Control", *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 5, 2009.

[54] Mills, A., Wills, A., and Ninness, B., "Nonlinear Model Predictive Control of an Inverted Pendulum", American Control Conferences, St. Louis, pp. 2335-2340, 2009.

[55] Kwakernaak, H., "Robust Control and H-infinity Optimization – Tutorial Paper", *Automatica*, Vol.29, No. 2, pp. 255, 273, 1993.

[56] Cloutier J. R., "State-Dependent Riccati Equation Techniques: An Overview", American Control Conference Albuquerque, New Mexico, 1997.

[57] Huang, J., and Lin, C.-F., "Numerical Approach to Computing Nonlinear H-infinity Control Laws", Journal of Guidance, Control, and Dynamics, Vol.18, No. 5, pp. 989, 994, 1995.

[58] Huang, Y., and Lu, W.-M., "Nonlinear Optimal Control: Alternatives to Hamilton-Jacobi Equations", 35th Conference on Decision and Control, Kobe, Japan, pp. 3942, 3947, 1996.

[59] Yang, C.-D., and Kung, C.-C., "Nonlinear H-infinity Flight Control of General Six Degree-of-Freedom Motions", American Control Conference, Chicago, Illinois, pp. 1852, 1856, 2000.

[60] Ferreira, H., Baptista, R., Ishihara, J., and Borges, G., "Disturbance Rejection in a Fixed Wing UAV using Nonlinear H-infinity State Feedback", 9th IEEE International Conference on Control and Automation, Santiago, Chile, pp. 386, 391, 2011.

[61] Ramirez, D. R., Alamo, T., Camacho, E. F., and Muñoz de la Peña, D., "Min-Max MPC Based on a Computationally Efficient Upper Bound of the Worst Case Cost", *Journal of Process Control*, 16 (2006), pp. 511, 519.

[62] Alamo, T., Ramirez, D. R., Muñoz de la Peña, D., and Camacho, E. F., "Min-Max MPC using a Tractable QP Problem", *Automatica*, 16 (2007), pp. 693, 700.

[63] Casavola, A., Mosca, E, and Angeli, D., "Robust Command Governors for Constrained Linear Systems", *IEEE Transactions on Automatic Control*, Vol. 45, No. 11, 2007, pp. 2077, 2077.

[64] Famularo, D., Martino, D., and Mattei, M., "Constrained Control strategies to Improve Safety and Comfort on Aircraft", *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 6, 2008, pp. 1782, 1792.

[65] Kwakernaak, H., "H2-Optimization–Theory and Applications to Robust Control Theory", *Annual Reviews in Control*, 26 (2002), pp. 69, 79.

[66] Ferreira, H., Rocha, P. H., and Sales, R. M., "Nonlinear H-infinity Control and the Hamilton-Jacobi-Isaac Equation", 17th World Congress, The international Federation of Automatic Control, Seoul, Korea, pp. 188, 193, 2009.

[67] Ballerini, M. et. al. "Interaction Ruling Animal Collective Behavior Depends on Topological Rather Than Metric Distance: Evidence from a Field Study", *PNAS*, Vol. 105, pp. 1232-1237, 2008.

[68] Lissaman P. B. S. and Schollenberg, "Formation Flight of Birds", *Science*, Vol. 168, pp. 1003-1005, 1970.

[69] Reynolds, C., “Flocks, Herds, and Schools: A Distributed Behavioral Model”, *Computer Graphics*, Vol. 21, No. 4, 1987.

[70] Blake, W., and Gingras, D., “Comparison of Predicted and Measured Formation Flight Interference Effects”, *Journal of Aircraft*, Vol. 41, No. 2, pp. 201-207, 2004.

[71] Patcher, M., D’Azzo, J. J., and Proud, A. W., “Tight Formation Flight Control“, *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, pp. 246-254, 2001.

[72] Chichka, D., Speyer, J., Fanti, C., and Park, C. G., “Peak-Seeking Control for Drag Reduction in Formation Flight”, *Journal of Guidance, Control, and Dynamics*, Vol. 29, pp. 1221-1230, 2006.

[73] Schumacher, C., and Singh, S., “Nonlinear Control of Multiple UAVs in Close-Coupled Formation Flight”, AIAA Guidance, Navigation, and Control Conference, Denver, CO, 2000.

[74] Singh, S., Zhang, R., Chandler, P., and Banda, S., “Decentralized Nonlinear Robust Control of UAVs in Close Formation”, *Intl. Journal of Robust and nonlinear Control*, Vol. 13, pp. 1057-1078, 2003.

[75] Richards, A., and How, J., “A Decentralized Algorithm for Robust Constrained Model Predictive Control”, Proc. of American Control Conference, Boston, MA, pp. 4261-4266, 2004.

[76] Keviczky, T., and et. al. “Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations”, *IEEE Trans. On Control System Technology*, Vol. 16, No. 1, pp. 19-33, 2008.

[77] Zhang, X., Duan, H., and Yu, Y., “Receding Horizon Control for Multi-UAVs Close Formation Control based on Differential Evolution”, *Science China*, Vol. 53, No. 2, pp. 223-235, 2010.

[78] Zhang, Y., Duan, H., and Zhang, X., “Stable Flocking of Multiple Agents Based on Molecular Potential Field and Distributed Receding Horizon Control”, *Chin. Phys. Lett.*, Vol. 28, No. 4, 2011.

[79] Weihua, Z., and Go, T., H., "Robust Decentralized Formation Flight Control", *International Journal of Aerospace Engineering*, Hindawi Publishing Corporation, Vol. 2011, Article ID 157590, 2011.

[80] Izadi, H., A., Gordon, B., and Zhang, Y., "Decentralized Model Predictive Control for Cooperative Multiple Vehicles Subject to Communication Loss", *International Journal of Aerospace Engineering*, Hindawi Publishing Corporation, Vol. 2011, Article ID 198308, 2011.

[81] Chao Z., Zhou, S., L., Ming, L., and Zhang, W., G., "UAV Formation Flight Based on Nonlinear Model Predictive Control", *International Journal of Aerospace Engineering*, Hindawi Publishing Corporation, Vol. 2012, Article ID 261367, 2012.

[82] Shin, J., and Kim, H., J., "Nonlinear Model Predictive Formation Flight", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol.39, No. 5, 2009.

[83] Shin, H., S., and Thak, M., J., "Nonlinear Model Predictive Control for Multiple UAVs Formation Using Passive Sensing", International of Aeronautical & Space Science, Vol. 12, No. 1, pp. 16-23, 2011.

[84] Ren, W., and Beard, R., W., "A decentralized Scheme for Spacecraft Formation Flying via the Virtual Structure Approach", Proceedings of the American Control Conference, pp.1746-1751, 2003.

[85] Leong, H., I., Keshmiri, S., and Kager, R., "Evaluation of a COTS Autopilot and Avionics System for UAVs", AIAA Infotech@Aerospace Conference, Seattle, WA, 2009.

[86] Phillips, W., F., *Mechanics of Flight*, 2nd ed., John Wiley & Sons, Hoboken, New Jersey, 2009.

[87] The USAF Stability and Control Digital Datcom, AFFDL-TR-79-3032, 1979.

[88] Cybenko, G., "Approximations by Superpositions of Sigmoidal Functions", *Mathematics of Control, Signals, and Systems*, Vol. 2 No. 4, pp. 303-314, 1989.

[89] Greenwell, D., "A Review of Unsteady Aerodynamic Modeling for Flight Dynamics of Maneuverable Aircraft", AIAA Atmospheric Flight Mechanics Conference and Exhibit, Providence, RI, 2004.

[90] Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer-Verlag, New York, 1999.

[91] Roweis, S., T., "Levenberg-Marquardt optimization", http://www.cs.toronto.edu/~roweis/notes/lm.pdf.

[92] Li, Y., Wang, J., Rizos, C., Mumford, P., and Ding, W., "Low-cost Tightly Coupled GPS/INS Integration based on a Nonlinear Kalman Filtering Design", Proceedings of the ION National Technical Meeting, pp. 958-966, Monterey, California, January, 2006.

[93] Zhang, P., Gu, J., Milios, E. E., Huynh, P., "Navigation with IMU/GPS/Digital Compass with Unscent Kalman Filter", Proceedings of the IEEE International Conference on Mechatronics and Automation, pp. 1497-1502, Niagara Falls, Canada, Lujy, 2005.

[94] Gross, J., et al., "A Comparison of Extended Kalman Filter, sigma-Point Kalman Filter, and Particle Filter in GPS/INS Sensor Fusion," AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, 2010.

[95] Simon, D., *Optimal State Estimation*, John Wiley & Sons, Hoboken, NJ, 2006.

[96] Glover, K. and Doyle, "State-Space Formulae for all Stabilizing Controllers that Satisfy an H-infinity Norm bound and Relations to Risk Sensitivity", *Systems & Control Letters*, Oct. 1988, pp. 167-172.

[97] Zhou, K., and Doyle, J., *Essentials of Robust Control*, Prentice Hall, Upper Saddle River, New Jersey, 1998.

[98] Levine, W., *The Control Handbook*, 2nd Ed., CRC Press, 2010.

[99] Leong, E., "Development of a 6DOF Nonlinear Simulation Model Enhanced with Fine Tuning Procedures", November 19, 2008, The University of Kansas.

[100] DARcorporation, *Advanced Aircraft Analysis*, Software Package, KS, 2009.

[101] Boggs, P. T., and Tolle, J. W., "Sequential Quadratic Programming," Acta Numerica, pp. 1-51, 1996.

[102] Royer, D., Keshmiri, S., Sweeten, B. "Modeling and Sensitivity Analysis of the Meridian Unmanned Aircraft," AIAA Infotech@Aerospace Conference, Atlanta, Georgia, 2010.

[103] Maskew, B., "Formation Flying Benefits Based on Vortex Lattice Calculations", NASA CR-151974, May, 1997.

[104] Myatt, J., and Blake, W., B., "Aerodynamic Database Issues for Modeling Close Formation Flight", AIAA 99-4194, 1999.

[105] Vachon, M., J., Ray, R., J., Walsh, K., R., and Ennix, K., "F/A-18 Performance Benefits Measured During the Autonomous Formation Flight Project", NASA/TM-2003-210734, 2003.

[106] Thien, H., P., Moelyadi, M., A., and Muhammad, H., "Effects of Leader's Position and Shape on Aerodynamic Performances of V Flight Formation", ICIUS, Bali, Indonesia, Oct. 2007.

[107] Iglesias, S., and Mason, W., H., "Optimum Spanloads in Formation Flight", 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2002.

[108] Chichka, D., Speyer, J., Fanti, C., and Park, C. G., "Peak-Seeking Control with Application to Formation Flight", Proceedings of the 38th Conference on Decision & Control, Phoenix, Arizona, Dec., 2009.