REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No.0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 31/12/2014	**2. REPORT TYPE** Final Technical Report	**3. DATES COVERED** *(From-To)* 30/11/2012- 31/12/2014

4. TITLE AND SUBTITLE Artificial Immune System for Flight Envelope Estimation and Protection	**5a. CONTRACT NUMBER** HR0011-13-C-0024
	5b. GRANT NUMBER 1005857R
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Mario G. Perhinschi Dr. Hever Moncayo Dr. Dia Al Azzawi	**5d. PROJECT NUMBER** 10016675
	5e. TASK NUMBER **1**
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) West Virginia University, Morgantown, West Virginia	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency, Tactical Technology Officer	**10. SPONSOR/MONITOR'S ACRONYM(S)** DARPA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Peter Erbland, ADPM, DARPA Research Services, Defense Technical Information Center, Christopher L. Glista, Frank Friedl, Michael Oppenheimer, Chris Hall Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The general goal of this research effort was to design and develop a novel biologically inspired framework for flight envelope estimation and protection, which is conceptually integrated with the abnormal detection and accommodation processes and is general and comprehensive such that it can operate at all conditions, normal and abnormal, a priori known or unknown. Two major methodologies have been proposed and investigated: one based on a dendritic cell mechanism and one on structuring of the self/non-self. Data for development and demonstration has been obtained from the WVU motion-based flight simulator for a supersonic fighter aircraft. Results show promising capabilities of the artificial immune system paradigm for providing an integrated and comprehensive solution to the problem of aircraft abnormal condition detection, identification, evaluation, and accommodation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*

Sponsored by: Defense Advanced Research Projects Agency, Tactical Technology Officer (TTO)
Program: Artificial Immune Systems for Flight Envelope – West Virginia University
Issued by DARPA/CMO under Contract No. HR0011-13-C-0024

"The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government."

Artificial Immune System for Flight Envelope Estimation and Protection

Technical Point of Contact:

Dr. Mario Perhinschi
Department of Mechanical and Aerospace Engineering,
West VirginiaUniversity
PO Box 6106, ESB 521
Morgantown, WV26505-6106
Phone Number: (304) 293-3301
Fax Number: (304) 293-6689
E-Mail: Mario.Perhinschi@mail.wvu.edu

ERAU Technical Point of Contact:

Dr. Hever Moncayo
Department of Aerospace Engineering,
Embry-Riddle Aeronautical University
600 South Clyde Morris Blvd
Daytona Beach, FL 32114
Phone Number: (386) 323-8092
E-Mail: moncayoh@erau.edu

Administrative Point of Contact:

Mr. Alan Martin
West Virginia University Research Corporation
886 Chestnut Ridge Road
P.O. Box 6845
Morgantown, WV26506
Phone Number: (304) 293-3998
Fax Number: (304) 293-7435
E-Mail: Alan.Martin@mail.wvu.edu

12/31/2014

Table of Contents

Standard Form 298 1
Title Page 2
Table of Contents 3
List of Figures 6
List of Tables 10
List of Acronyms 12

Project Summary 13
1. Task Objectives 13
2. Technical Problems 14
3. General Methodology 15
4. Technical Results 16
5. Important Findings and Conclusions 16
6. Significant Hardware Development 17
7. Special Comments 17
8. Implications for Future Research 17

Chapter 1 General Framework for Immunity-based Abnormal Condition Detection, Identification, Evaluation, and Accommodation 18
1.1. Biological Immune System 18
1.2. Artificial Immune System Paradigm 23
1.3. Definitions, Nomenclature, and Notation 26
1.4. Aircraft Abnormal Condition Management Process 31
1.5. Definition of the Self and Non-Self 34
1.5.1. General Algorithm 34
1.5.2. Raw Data Set Union Method 38
1.5.3. Cluster Set Union Method 40
1.5.4. Comparison of the Proposed Methods for Antibody Generation 42
1.6. Hierarchical Multi-Self Strategy 43
1.7. Selection of Development and Demonstration Test Cases 45
1.7.1. Aircraft Subsystems 45
1.7.2. Types of Subsystem Abnormal Conditions 45
1.7.3. Flight Envelope Analysis Parameters 45
1.7.4. Features Selection 47
1.8. WVU Simulation Environment 49
1.9. WVU Self/Non-self Analysis Tool 51

Chapter 2 Immunity-based Detection of Aircraft Subsystem Abnormal Conditions 53
2.1. Detection through Direct Self/Non-self Discrimination 53
2.1.1. General Aspects 53
2.1.2. Generation of Detectors 54
2.1.3. Analysis of Detection Performance of Individual Selves 55
2.1.4. Detection Logic for Direct Self/Non-self Discrimination Scheme 56
2.1.5. Detection Performance Using Self/Non-self Discrimination 59
2.2. Dendritic Cell Mechanism for Abnormal Condition Detection 61

2.2.1. The Artificial Dendritic Cell 61
2.2.2. Detection Performance Using the DC Mechanism 65

Chapter 3 Immunity-based Identification of Aircraft Subsystem Abnormal Conditions 67
3.1. Structured Non-self Approach for Abnormal Condition Identification 67
3.1.1. Principles of AC Identification Using the Structured Non-self Approach 67
3.1.2. Identification Algorithm Using the Structured Non-self Approach 68
3.1.3. Generation of Identifiers 71
3.1.4. The Two Phases of the SNSA Approach 73
3.1.5. AC Identification Performance Using Self/Non-self Discrimination 75
3.2. Dendritic Cell Mechanism for Abnormal Condition Identification 76
3.2.1. Dendritic Cell Mechanism for AC Identification 76
A). Feature-pattern (F-P) approach for AC identification 76
B). Projection-pattern (P-P) approach for AC identification 76
C). Matrix-pattern (M-P) approach for AC identification 77
3.2.2. Performance of AC Identification Using a Dendritic Cell Mechanism 78

Chapter 4 Immunity-based Direct Evaluation of Aircraft Subsystem Abnormal Conditions 81
4.1. Structured Non-self Approach for Abnormal Condition Direct Evaluation 81
4.1.1. Structured Non-self Approach for Qualitative Direct Evaluation 81
4.1.2. Structured Non-self Approach for Quantitative Direct Evaluation 83
4.1.3. Performance of Direct Evaluation Using the SNSA 85
4.2. Dendritic Cell Mechanism for Abnormal Condition Direct Evaluation 88
4.2.1. Dendritic Cell Approach for Qualitative Direct Evaluation 88
A). Feature-pattern (F-P) approach for AC qualitative evaluation 89
B). Projection-pattern (P-P) approach for AC qualitative evaluation 89
C). Matrix-pattern (M-P) approach for AC qualitative evaluation 90
4.2.2. Dendritic Cell Approach for Quantitative Direct Evaluation 91
4.2.3. Performance of Direct Evaluation Using the Dendritic Cell Approach 92

Chapter 5 Flight Envelope Estimation Under Abnormal Conditions Using Artificial Immune System 95
5.1. Problem Formulation 95
5.2. Estimation Algorithms for Actuator AC 98
5.2.1. Stabilator Failure 98
5.2.2. Aileron Failure 100
5.2.3. Rudder Failure 102
5.2.4. Throttle Failure 103
5.3. Estimation Algorithms for Sensor AC 108
5.3.1. Roll Rate Sensor Bias 108
5.3.2. Pitch Rate Sensor Bias 109
5.3.3. Yaw Rate Sensor Bias 110
5.4. Estimation Algorithms for Structural Damage 112
5.4.1. Wing Damage 112
5.4.2. Horizontal Tail Damage 115
5.4.3. Vertical Tail Damage 117

5.5. Estimation Algorithms for Propulsion AC 119
5.6. Performance of Flight Envelope Assessment Under AC 122
5.6.1. Evaluation Metrics for Flight Envelope Assessment 122
5.6.2. Flight Envelope Assessment Under Actuator Failure 123
5.6.3. Flight Envelope Assessment Under Sensor Failure 124
5.6.4. Flight Envelope Assessment Under Structural Damage 125
5.6.5. Flight Envelope Assessment Under Propulsion System Failure 126

Chapter 6 Investigation of Immunity-based Accommodation of Aircraft Subsystem Abnormal Conditions 128
6.1. Immunity Approaches for AC Accommodation 128
6.2. The Humoral Immune Mechanism for Abnormal Condition Accommodation 131
6.2.1. General Control Architecture 131
6.2.2. Analysis of AIS Compensation Performance 134
6.3. Structured Self/Non-self for Control 138
6.3.1. Formulation of Mathematical Model 138
6.3.2. Generating AIS Memory 140
6.3.3. Matching Algorithm 142
6.3.4. Example Results Using the Structured Self/Non-self for Control 142
A). Simulation Scenarios 142
B). System Implementation 143
C). Simulation Results for Symmetric Climb 143
D). Simulation Results for Coordinated Half-Turn 146

Chapter 7 Integrated Scheme for Aircraft Subsystem ACDIE 151
7.1. Integrated ACDIE Scheme Using the Structured Self/Non-self Approach 151
7.2. Integrated ACDIE Scheme Using the DC mechanism-based Approach 153

Chapter 8 Demonstration of the Integrated Scheme for Aircraft Subsystem ACDIE 154
8.1. Implementation of the Integrated Scheme for Aircraft Subsystem ACDIE 154
8.2. Demonstration of the Integrated Scheme for Aircraft Subsystem ACDIE Through Simulation 157
8.3. Demonstration of the AIS AC Detection Through Flight Testing Using a UAV Research Platform 160
8.3.1. Flight Test Scenario 160
8.3.2. Test Platform 160
8.3.3. Simulink Models 162
8.3.4. Flight Test Results 164

References 170

APPENDIX A - List of Publications Resulting from the DARPA Sponsored Research Effort 172
APPENDIX B - Aircraft Subsystem ACDIE Scheme – Simulink Implementation (on DVD) 175
APPENDIX C - Installation and User Manual for the Aircraft Subsystem ACDIE Scheme 176

List of Figures

Figure 1.1. Phagocytes in the Peripheral Tissue 19
Figure 1.2. A Dendritic Cell Processing an Antigen 20
Figure 1.3. Activation of T_h-, B-, and T_c-Cells by the APC 21
Figure 1.4. The Negative Selection Process 22
Figure 1.5. Design of AIS-Based Abnormal Condition Detection Scheme 24
Figure 1.6. Self/Non-Self 2-Dimensional Projection 30
Figure 1.7. AIS-Based ACM Process 31
Figure 1.8. AIS-Based ACM System Design 32
Figure 1.9. On-Line AC Detection, Identification, Evaluation, and Accommodation 32
Figure 1.10. Post-Processing of Flight Data and ACDIEA Outcomes 33
Figure 1.11. Potential of Sub-Selves to Capture the Abnormal Conditions 35
Figure 1.12. Self/Non-Self Generation Process 37
Figure 1.13. Two Methods for AIS Antibodies Generation 38
Figure 1.14. 2-D Self Clusters with AIS Antibodies Generated Using the Raw Data Set Union Method 39
Figure 1.15. 2-D Self with 22000 Clusters 41
Figure 1.16. 2-D Self with 6343 Clusters 41
Figure 1.17. Sample 2-D Self Clusters with AIS Antibodies Generated Using the Cluster Set Union Method 41
Figure 1.18. Detection Performance Comparison of Detectors Obtained with CSUM and RDSUM for a Stabilator Failure 42
Figure 1.19. The Hierarchical Multi-Self Strategy for ACDIE 44
Figure 1.20. The WVU 6-DOF Motion-based Flight Simulator 49
Figure 1.21. Simulink Supersonic Fighter Aircraft Model. 50
Figure 1.22. Testing Flight Envelope 50
Figure 1.23. The Interactive Tool for Self/Non-self Analysis 51
Figure 1.24. The Zooming Action of the Interactive Tool for Self/Non-self Analysis 52

Figure 2.1. Cluster Set Union Based Logic 54
Figure 2.2. Detection Rate of Left Aileron Locked at 2.5 deg Failure 57
Figure 2.3. Detection Rate of Left Stabilator Locked at 2.0 deg Failure 57
Figure 2.4. Detection Rate of 6% Left Wing Loss Failure 57
Figure 2.5. Detection Rate of Pitch Sensor Bias of 5 deg/sec Failure 58
Figure 2.6. Detection Rate of Self 87 for 4 Different Failures 58
Figure 2.7. On-Line Failure Detection Logic with SNSA 59
Figure 2.8. Block Diagram of the DC Algorithm for Failure Detection 64
Figure 2.9. Data Structure of an Artificial Dendritic Cell 64
Figure 2.10. Sum of Residual Cytotoxic T-cells vs. Time of a Nominal Flight Test 65
Figure 2.11. Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: Right Aileron Locked at 8°. Actual Failure Time = 40 s; Detection Time = 40.76 s 65
Figure 2.12. Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: 3° LFDB in Yaw Rate Sensor. Actual Failure Time = 40 s; Detection Time = 43.38 s 65
Figure 2.13. Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: 15% Loss of the Left Wing. Actual Failure Time = 40 s; Detection Time = 40.52 s 65

Figure 3.1. Failed Sub-System Identification Using the Structured Non-Self Approach 67
Figure 3.2. (a) Self#3 with Left Aileron Failure; b) Self#3 with Right Wing Structural Damage 70
Figure 3.3. (a) Self#30 with L Aileron Failure, (b) Self#30 with R Wing Structural Damage 70
Figure 3.4. Structured Non-Self Approach Logic 71
Figure 3.5. Flowchart of the Identifier Generation Process 72
Figure 3.6. Concentric Hypercubes and Radii Variation With Respect to Distance from the Self 72
Figure 3.7. Generation of Library of Reference Patterns for Identification Using the DC Mechanism for Identification and the Naïve Bayes Classifier 78
Figure 3.8. Variation of the Feature-Pattern Vector with Time of a Failed Right Wing Subsystem 79
Figure 3.9. Reference Features-Pattern of a Failed Right Wing 79
Figure 3.10. Reference Features-Pattern of a Failed Left Aileron 79
Figure 3.11. Reference Features-Pattern of a Failed Yaw Rate Sensor 80
Figure 3.12. Sample Current Feature-Pattern of a Failed Right Wing 80

Figure 4.1. Integration of Direct Evaluation and Identification Within SNSA 82
Figure 4.2. Failure Qualitative Evaluation Using the Structured Non-Self Approach 83
Figure 4.3. Failure Direct Quantitative Evaluation Based on Distance to Self 84
Figure 4.4. Failure Direct Quantitative Evaluation Using the Structured Non-Self Approach 85
Figure 4.5. Generation of Library of Reference Patterns for Qualitative Evaluation Using the DC Mechanism and the Naïve Bayes Classifier 88
Figure 4.6. DC Mechanism for Simultaneous AC Identification and Qualitative Evaluation 90
Figure 4.7. Training the Naïve Bayes Classifier for AC Detection and Qualitative Evaluation 91
Figure 4.8. AC detection and Qualitative Evaluation Scheme Using the DC Mechanism 92
Figure 4.9. Reference Feature Patterns for a Stabilator Under Low Severity Failure 93
Figure 4.10. Reference Features-Pattern for a Stabilator Under High Severity Failure 93
Figure 4.11. Reference Features-Pattern for a Wing Under Low Severity Failure 93
Figure 4.12. Reference Features-Pattern for a Wing Under High Severity Failure 93
Figure 4.13. Reference Features-Pattern for a Roll Rate Sensor Under Low Severity Failure 93
Figure 4.14. Reference Features-Pattern for a Roll Rate Sensor Under High Severity Failure 93

Figure 5.1. Post-Failure Flight Envelope Estimation Using the AIS Paradigm 97
Figure 5.2. Aircraft Forces in the Plane of Symmetry at Trim 112
Figure 5.3. Post-Failure Range of the Pitch Rate Under Stabilator Failure 124

Figure 6.1. Block Diagram of the Immune System Feedback Model 128
Figure 6.2. Failure Accommodation Based on Structured Non-Self 130
Figure 6.3. Humoral Immune System Representation 131
Figure 6.4. General Block Diagram of the Baseline Controller 132
Figure 6.5 General Block Diagram of the Baseline controller (PID+NLDI) Augmented with ANN 132
Figure 6.6. Block Diagram of the PID-AIS-based Adaptive Mechanism 133

Figure 6.7. Baseline Controller Augmented with AIS-based Mechanism 134
Figure 6.8. Results of Average PI for Different ACs 134
Figure 6.9. Roll Response Under High Magnitude Stabilator Failure 135
Figure 6.10. Pitch Response Under High Magnitude Stabilator Failure 135
Figure 6.11. Total Pilot Activity Under High Magnitude Stabilator Failure 135
Figure 6.12. Average Performance Index for Tracking Error 136
Figure 6.13. Task Execution Time Histogram 137
Figure 6.14. Time per Simulation vs Simulation Count 137
Figure 6.15. Failure Accommodation Definition 138
Figure 6.16. Accommodation Problem Main Components 139
Figure 6.17. AIS Memory Structure 141
Figure 6.18. Symmetric Uniform Climb 143
Figure 6.19. Coordinated Half Turn 143
Figure 6.20. Aircraft Altitude at Nominal 144
Figure 6.21. Longitudinal Channel Commands 144
Figure 6.22. Aircraft Altitude 144
Figure 6.23. Longitudinal Channel Commands 144
Figure 6.24. Lateral Channel Commands 145
Figure 6.25. Lateral Aircraft Position Coordinate 145
Figure 6.26. Aircraft Bank Angle 145
Figure 6.27. Aircraft Altitude Under Sensor Failure 145
Figure 6.28. Longitudinal Channel Commands 145
Figure 6.29. Lateral Channel Commands 145
Figure 6.30. Aircraft Bank Angle 146
Figure 6.31. Coordinated Turn Flight Trajectory Under Nominal Conditions 146
Figure 6.32. Lateral Channel Commands for Coordinated Turn at Nominal Conditions 147
Figure 6.33. Aircraft Bank Angle 147
Figure 6.34. Longitudinal Channel Commands 147
Figure 6.35: Aircraft Altitude - Nominal Conditions Coordinated Half Turn 148
Figure 6.36. Coordinated Turn Trajectory Under Stab Failure 148
Figure 6.37. Longitudinal Channel Commands Under Stab Failure 148
Figure 6.38. Lateral Channel Commands Under Stab Failure 149
Figure 6.39. Aircraft Bank Angle Under Stab Failure 149
Figure 6.40. Flight Trajectory for Coordinated Turn Under Roll Rate Sensor Failure 149
Figure 6.41. Lateral Channel Commands Under Roll Rate Sensor Failure 149
Figure 6.42. Longitudinal Channel Commands for Coordinated Turn Under Roll Rate Sensor Failure 150

Figure 7.1. ACDIE with the Structured Non-Self Approach 152
Figure 7.2. ACDIE with the DC Mechanism Approach 153

Figure 8.1. Top-level Simulink Model of the Integrated Aircraft ACDIE Using the Structured Non-Self Approach 154
Figure 8.2. Simulink Block Diagram of the Structured Non-Self Approach 154
Figure 8.3. Top-level Simulink Model of the Integrated Aircraft ACDIE Scheme Using the DC Mechanism 155

Figure 8.4. Simulink Block Diagram of the DC Mechanism 155
Figure 8.5. ACDIE Outcome Visualization Interface 156
Figure 8.6. Demonstration Results for a Nominal Flight 159
Figure 8.7. Demonstration Results for a 15% Damage of the Left Wing 159
Figure 8.8. Skywalker 1880 RC 161
Figure 8.9. Onboard APM 2.5 161
Figure 8.10. Simulink Model Top Level 163
Figure 8.11. Simulink Sensor Model 163
Figure 8.12. Kalman Filter Architecture 164
Figure 8.13. Self #1 Detector Activity for Left Aileron Locked at Bank Maneuver 167
Figure 8.14. Self #17 Detector Activity for Left Aileron Locked at Bank Maneuver 167
Figure 8.15. Self #18 Detector Activity for Left Aileron Locked at Bank Maneuver 168
Figure 8.16. Method 2 Detector Activity for Left Aileron Locked at Bank Maneuver 168
Figure 8.17. Method 1 Detector Activity for Left Aileron Locked at Bank Maneuver 169

List of Tables

Table 1. Activity Chart for the Project 14

Table 1.1. Main Biological Terms and Their AIS Paradigm Representations 24
Table 1.2. Parameters for AC Indirect Evaluation Analysis 45
Table 1.3. Features for Self/Non-self Definition 47

Table 2.1. Features of Selected Projections 55
Table 2.2. Detection Performance of Selected Individual Selves 56
Table 2.3. Aircraft Subsystems Considered for AC Detection 59
Table 2.4. Average Detection Performance of the SNSA Scheme 60
Table 2.5. Average DC-based Detection Rate for Different Subsystems 65

Table 3.1. Self/Non-self 2-D Projections for SNSA Approach 69
Table 3.2. Phase I SNSA Subsystem Failures 73
Table 3.3. Detection Sensitivity of a Sample Set of Projections (% DR) 74
Table 3.4. Total Number of Projections Activated per Failure 75
Table 3.5. Average Identification Rates Using the DC Mechanism 80

Table 4.1. Projections Used for Simultaneous Identification and Direct Evaluation 86
Table 4.2. Total Number of Projections Used for Simultaneous Identification and Direct Evaluation 86
Table 4.3. Performance of AC Identification and Direct Evaluation 87
Table 4.4. AC Detection and Qualitative Evaluation Rates of the DC Mechanism Scheme 94

Table 5.1. Performance of Flight Envelope Prediction Under Stabilator Failure 123
Table 5.2. Performance of Flight Envelope Prediction Under Aileron Failure 124
Table 5.3. Performance of Flight Envelope Prediction Under Rudder Failure 124
Table 5.4. Performance of Flight Envelope Prediction Under Roll Rate Sensor Failure 125
Table 5.5. Performance of Flight Envelope Prediction Under Pitch Rate Sensor Failure 125
Table 5.6. Performance of Flight Envelope Prediction Under Yaw Rate Sensor Failure 125
Table 5.7. Performance of Flight Envelope Prediction Under Wing Damage 126
Table 5.8. Performance of Flight Envelope Prediction Under Horizontal Tail Damage 126
Table 5.9. Performance of Flight Envelope Prediction Under Engine Failure 127

Table 6.1. Computer Characteristics for Execution Time Assessment 136
Table 6.2. Task Execution Time Assessment 136

Table 8.1. Detection Results for a Set of Demonstration Tests 157
Table 8.2. Identification Results for the Demonstration Tests 157
Table 8.3. Failure Type Evaluation Results for the Demonstration Tests 158
Table 8.4. Failure Severity Evaluation Results for the Demonstration Tests 158
Table 8.5. Failures Injected in Flight Tests 160
Table 8.6. Skywalker Dimensions and Mass Properties 161
Table 8.7. Self/Non-self Features for Flight Testing 164

Table 8.8. Skywalker Self Projections 165
Table 8.9. Detection Rate and False Alarm Rate for 4 Failures 165
Table 8.10. Detection Rates and False Alarms for 3 Configurations of Selves 166

List of Acronyms

AC	abnormal condition
ACDIEA	abnormal condition detection, identification, and evaluation
ACDIEA	abnormal condition detection, identification, evaluation, and accommodation
ADALINE	adaptive linear network
AIS	artificial immune system
ANN	artificial neural network
APC	antigen presenting cells
CSUM	cluster set union method
DC	dendritic cells
DIV	directly involved variable
DOF	degrees-of-freedom
DQEE	decentralized quadratic estimation error
DR	detection rate
EDIV	equivalent directly involved variables
ERAU	Embry-Riddle Aeronautical University
ERV	envelope relevant variables
F-P	feature-pattern
FA	false alarm
FN	false negative
FP	false positive
HMS	hierarchical multi-self
IQE	indirect quantitative evaluation
M-P	matrix-pattern
MHC II	major histocompatibility complexes II
MQEE	main quadratic estimation error
NCE	no-cost-extension
NLDI	nonlinear dynamic inversion
NS	negative selection
OQEE	output quadratic estimation error
P-P	projection-pattern
PID	proportional-integral-derivative
PR	prediction rate
PS	positive selection
RDSUM	raw data set union method
SNSA	structured non-self approach
TN	true negative
TO	technical objectives
TP	true positive
UAV	unmanned aerial vehicle
WVU	West Virginia University

Project Summary

1. Task Objectives

The general goal of this research effort was to design and develop a novel biologically inspired framework for flight envelope estimation and protection, which is conceptually integrated with the abnormal detection and accommodation processes and is general and comprehensive such that it can operate at all conditions, normal and abnormal, a priori known or unknown. The proposed methodology relies primarily on the artificial immune system (AIS) paradigm. To reach this objective, the following 3 technical objectives (TO) with a total of 14 main research tasks and one additional task over the no-cost-extension period have been planned.

TO #1. Develop a novel bio-mimetic framework for flight envelope estimation and protection under upset flight conditions.

Task #1.1. *Problem formulation including definition of abnormal conditions addressed, parameter ranges, and envelope parameters targeted.*

Task #1.2. *Development of schemes for qualitative failure evaluation.*

Task #1.3. *Development of schemes for direct quantitative failure evaluation.*

Task #1.4. *Development of schemes for indirect quantitative failure evaluation. Flight envelope reduction evaluation.*

Task #1.5. *Design and execution of flight simulator tests for AIS abnormal condition evaluation schemes development.*

TO #2. Investigate the use of the artificial immune system paradigm for aircraft adaptive guidance and control under abnormal conditions.

Task #2.1. *Formulation of the AIS-based framework for flight envelope protection.*

Task #2.2. *Immune adaptive control laws design and implementation.*

Task #2.3. *System integration.*

Task #2.4. *Design and execution of flight simulator tests for development of AIS-based abnormal condition accommodation schemes.*

TO #3. Implement and demonstrate the proposed syst. on a motion based flight simulator.

Task #3.1. *Design of flight simulator tests for proposed methodology demonstration.*

Task #3.2. *Customization of existing computational tools for aircraft simulation, adaptive control laws, and artificial immune system generation.*

Task #3.3. *Implementation and integration of a simulation environment for design, testing, and demonstration of the novel flight envelope estimation and protection methodology.*

Task #3.4. *Execution of the 6-degrees-of-freedom (DOF) flight simulator tests and data processing.*

Task #3.5. *Analysis of results and evaluation of the novel flight envelope estimation and protection methodology.*

Task #3.6 *Reduced system demonstration on an unmanned aerial vehicle (UAV).*

The following project milestones were defined and scheduled as follows:

M#1. Formulation of a generalized conceptual framework for flight envelope estimation based on the AIS paradigm (End of Month #8)

M#2. Flight envelope protection algorithms implemented and tested (End of Month #14)

M#3. Integrated system demonstration in the motion-based flight simulator (End of Month #18, re-scheduled at the end of no-cost-extension period)

The activity chart for the project is presented in Table 1. The periods of performance for all tasks over the initial duration of the project are marked in blue. The re-scheduled periods for Tasks 1.4, 2.3, 3.3-3.6 and Milestone #3 over the no-cost-extension (NCE) period are marked in light blue. The due dates for the three project milestones are marked with green lines.

Table 1. Activity Chart for the Project

		Months during 1st Year						Months during 2nd Year						NCE					
		1-4		5-8		9-12		1-2		3-4		5-6		7-9		10-11		12-13.5	
TO #1	1.1	🟦	🟦																
	1.2		🟦	🟦	🟦	🟦	🟦	🟦	🟦										
	1.3		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦									
	1.4		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	▨	▨	▨	▨				
	1.5		🟦	🟦	🟦														
TO #2	2.1		🟦	🟦	🟦	🟦													
	2.2		🟦	🟦	🟦	🟦	🟦												
	2.3				🟦	🟦	🟦	🟦	▨	▨	▨	▨	▨	▨	▨	▨			
	2.4			🟦	🟦	🟦	🟦												
TO #3	3.1					🟦	🟦	🟦	🟦										
	3.2		🟦	🟦	🟦	🟦	🟦												
	3.3		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	▨	▨	▨	▨				
	3.4						🟦	🟦	🟦	🟦	🟦	🟦	▨	▨	▨	▨	▨		
	3.5								🟦	🟦	🟦	🟦	🟦	▨	▨	▨	▨	▨	▨
	3.6													▨	▨	▨	▨	▨	▨
					M#1				M#2										M#3

2. Technical Problems

In this project, the AIS paradigm was used to accomplish critical steps towards the development of a holistic framework for an effective, comprehensive, and integrated solution to the problem of aircraft operation under abnormal conditions (AC). The complete aircraft AC management process is envisioned as consisting of four major components: AC detection, identification, evaluation, (ACDIE) and accommodation (ACDIEA). From this perspective, performing aircraft AC management becomes an extremely challenging, complex, and multi-dimensional task.

In continuation of previous research establishing conceptual and practical approaches for AIS-based AC detection and identification, this project was primarily focused on addressing the qualitative and quantitative evaluation of abnormal flight conditions consisting of failures/damages affecting aircraft actuators, sensors, propulsion system, and aerodynamic surfaces. The AC evaluation process is highly specific and requires individualized approaches due to the synergistic interaction between aircraft subsystems, aircraft states, and the nature of the ACs. Within the project, classes of AC were considered that alter subsystem aerodynamics in a manner consistent with and predictable based on aircraft operation at normal condition, referred to as “self” within the AIS paradigm. Integration of AC evaluation with prior detection and identification was given significant attention.

The multi-dimensionality of the feature (or relevant variable) space implied by the development of a comprehensive AIS creates computational and logical issues that need to be addressed carefully. The use of lower dimensional projections of the self (sub-selves) within the hierarchical multi-self strategy was proposed in this project. It should be noted that the issue of computational time for off-line and on-line processing has not been specifically targeted within the project.

The AC accommodation problem was also addressed within the project. The possibility of using immunity mechanisms for adaptive control and compensation of abnormal flight conditions was investigated along two directions. One relied on the antibody activation/suppression mechanism converted into an adaptive control approach. The other, relied on extending the classification capabilities of the AIS and using them not only to detect and diagnose the problem, but also to select or find the solution. In other words, once a certain abnormal situation is detected and identified, the AIS-based system can determine the adequate compensation.

3. General Methodology

Developing a generalized conceptual framework (comprehensive and integrated) for the detection, identification, evaluation, and accommodation of aircraft sub-system abnormal conditions requires specific tools that can handle the high complexity and multi-dimensionality of aircraft dynamic response in the context of abnormal conditions. The tools for the development of the proposed methodology rely on a new concept inspired from the biological immune system. The immune system protects the body against intruders by recognizing and destroying harmful cells or molecules through simple in principle yet powerful mechanisms, such as negative/positive selection, mutation, and cloning, which exhibit robust, adaptive, and highly distributed cognitive capabilities. It can be thought of as a *robust adaptive system* that is capable of dealing with an enormous variety of known and unknown disturbances and uncertainties. Another critical aspect of the bio-immune system is that it can "remember" how previous encounters were successfully handled. As a result, it can respond faster to similar encounters in the future. The AIS is capable of mimicking these mechanisms (such as negative/positive selection, antibody activation/suppression, state/resource assessment, and memory) to solve the problem of aircraft subsystem ACDIEA in a general and comprehensive manner.

The failure evaluation process must address several distinct aspects such as determining the type of the failure, its magnitude or severity, and evaluating the reduction of the flight envelope due to the failure, in the most general sense. Evaluation requires the previous correct detection and identification of the anomalous conditions. Detection is the process of declaring that a generic malfunction of the system has occurred in any of the subsystems, while the identification implies determining what is the failed subsystem and/or component. These two aspects are important to diagnose the effect of the failure on the aircraft operational limits and provide necessary information to the pilot and the automatic control system to avoid commands that might lead to loss of control and other dangerous/catastrophic situations.

The computational issues associated with high-dimensionality hyper-spaces is addressed by using lower dimensional projections of the self/non-self within the hierarchical multi-self (HMS) strategy, which relies on two main concepts. First, the normal/abnormal discrimination capability of complete sets of projections is the same as the entire hyper-self under certain conditions. Second, projections along specific feature axes have better capabilities in capturing the dynamic fingerprint of specific AC than others. This hierarchical ranking of sub-selves with respect to targeted AC can be used to develop less computationally intensive detection, identification, and evaluation schemes.

The AIS paradigm is based only on data acquisition and structuring and does not need sophisticated modeling of the targeted system. However, this requires extensive amount of data, mostly at normal operational conditions, and specific processing in some instances. For the purpose of this project, data for a supersonic fighter were acquired from a motion-based 6-DOF flight simulator.

This research effort has been performed as a close collaboration between researchers from West Virginia University (WVU) and Embry-Riddle Aeronautical University (ERAU).

4. Technical Results

The proposed AIS-based generalized conceptual framework for aircraft AC management has been demonstrated to be able to handle simultaneously several major aircraft subsystems, under known and unknown abnormal conditions of different types, and severities. The subsystems considered under abnormal conditions were: actuators (left/right stabilator, aileron, rudder), sensors (pitch, roll, yaw gyros), propulsion (throttle), structural components (left/right wing, horizontal tail, vertical tail).

Two different methodologies were proposed, developed, and successfully tested for AC detection, identification, and evaluation: dendritic cell (DC) mechanism and structured non-self approach. Integrated schemes were implemented and evaluated in terms of false alarms and success rates for detection, identification, and evaluation. In most cases considered, the proposed ACFDIE schemes achieve zero false alarms and success rates between 95-100%.

Specific algorithms for post-failure flight envelope evaluation have been developed, implemented and successfully tested. The post-failure flight envelope evaluation is defined in terms of relevant features and targets classes of AC that affect the aerodynamics of the system in a manner consistent with the operation at normal conditions.

The preliminary investigation of AIS-based abnormal condition accommodation included the use of antibody activation-suppression mechanism for adaptive control laws and the transfer of immunity memory capabilities to AIS for enabling for pilot compensatory commands extraction. Both approaches have been implemented and successfully tested demonstrating very promising potential.

The outcomes of this project have been widely disseminated. The record to date of peer-reviewed journal publications includes 3 published papers, 7 in review, and 1 in preparation. The record to date of peer-reviewed conference publications includes 7 papers presented and published and 2 papers accepted (see Appendix A for complete list of titles).

5. Important Findings and Conclusions

The technical outcomes of this project include the formulation of novel concepts and methodologies that are expected to improve pilot situational awareness and improve the guidance and control capabilities of aircraft systems under normal and abnormal flight conditions. The proposed framework for ACDIEA has been demonstrated to possess the capability for providing a comprehensive solution to the problem of aircraft AC management including flight envelope estimation and protection at post-failure conditions.

A set of integrated computational tools with a high degree of flexibility, portability, and modularity have been implemented and successfully tested. They can be used for the design, development, testing, simulation, and evaluation of fault tolerant control laws and schemes for ACDIEA.

Specific algorithms within the AIS paradigm have been developed and demonstrated to achieve high performance rates for various phases of the aircraft ACDIEA process.

The proposed methodology has the capabilities for facilitating the design of advanced aircraft with high survivability, improved operation safety, and optimized performance at both normal and abnormal/upset conditions.

The proposed framework was used to design, implement, and test through simulation integrated schemes for aircraft AC detection, identification, and evaluation that achieved excellent performance for a large variety of scenarios.

6. Significant Hardware Development

Development of hardware was not part of this project.

7. Special Comments

There are no special comments relative to the technical aspects of this project.

8. Implications for Future Research

The outcomes of this project create premises for research continuation in the following directions.

1). *Extension of AC spectrum for post-failure evaluation* . The algorithms developed within this project address specific classes of ACs. For a complete and comprehensive AC management process, additional classes of AC should be investigated.

2). *Further investigation and development of AIS-based abnormal condition accommodation.* Solving the abnormal condition accommodation problem within the AIS paradigm is expected to provide a compact, consistent, and relatively easy to implement framework that will address completely all major processes related to fault tolerant control (abnormal condition detection, identification, evaluation, and accommodation). This research effort should produce a novel adaptive architecture based on hybrid approaches and artificial intelligence techniques to increase robustness, autonomy, and safety of aircraft systems.

3). *Testing the proposed methodology in flight using a remotely piloted reduced size aircraft***.** The proposed AIS-based framework for aircraft AC management should be implemented and tested under more realistic conditions using existing aerial platforms. This project is expected to provide the opportunity to apply and investigate lessons learned, fine-tune and extend algorithms, and validate the proposed methodology through actual flight tests. It is expected that issues related to memory allocation and computational time during both off-line development and on-board operation will be addressed.

4). *Extension of the AIS paradigm into an alternative or complementary modeling methodology for dynamic systems.* The versatility of the AIS and its capabilities to store extensive system information can be expanded through combination with mechanisms inspired by genetics to develop methodologies for building data driven system models that are expected to remove or mitigate the burden of complex and challenging modeling tasks. The proposed strategy is to rely on experimental and analytical data and information that is typically available for a technical system for building a representation of the system that includes all the elements necessary for system build up and operation, similar to the genome of biological organisms. This will create the premises for the development of intelligent, evolving system with a wide range of applications.

Chapter 1

General Framework for Immunity-based Abnormal Condition Detection, Identification, Evaluation, and Accommodation

1.1. Biological Immune System

The biological immune system consists of two equally important components: the innate system and the adaptive system[1]. The former is an inherited system that functions as the first line of defense against invading entities, while the latter is an acquired system built through previous exposures to invading agents. The innate immune system is always active and responds immediately to any class of pathogen without distinction. On the other hand, the adaptive immune system is normally silent and is much more potent in recognizing specific antigens with slower response.

The immune system is composed of cells that are developed from stem cells in the bone marrow and differentiate into different populations, the most important of which in the immune response are macrophages (MΦs), dendritic cells (DCs), T-cells, and B-cells. The MΦ and DC populations form what is known as *phagocytes* (part of the innate system) whereas populations of T-cells and B-cells form the *lymphocytes* (part of the adaptive system).

Phagocytes reside in the peripheral tissues searching for antigens (Figure 1.1). Proteins (or any other markers) of an antigen are recognized by the surface of the phagocyte. The phagocyte engulfs the pathogen and breaks it up into its constituent molecules. Special proteins in the phagocyte, called *major histocompatibility complexes* II (MHC II), bind to subsets of these molecules (specifically, *peptides*) and transport to the outer surface of the phagocyte along with the bound antigen peptides (Figure 1.2). MHC II play a critical role in activating the cell-mediated response of the adaptive system when the phagocyte migrates from the peripheral tissues to the lymph nodes where lymphocytes reside. B-cells process antigens in a very similar way to the phagocytes, except that phagocytes process general, non-specific antigens whereas B-cells can only process very specific types of antigens. For this reason, phagocytes are called professional *antigen presenting cells* (APCs). This communication is so crucial that the adaptive response cannot take place without the innate response first.

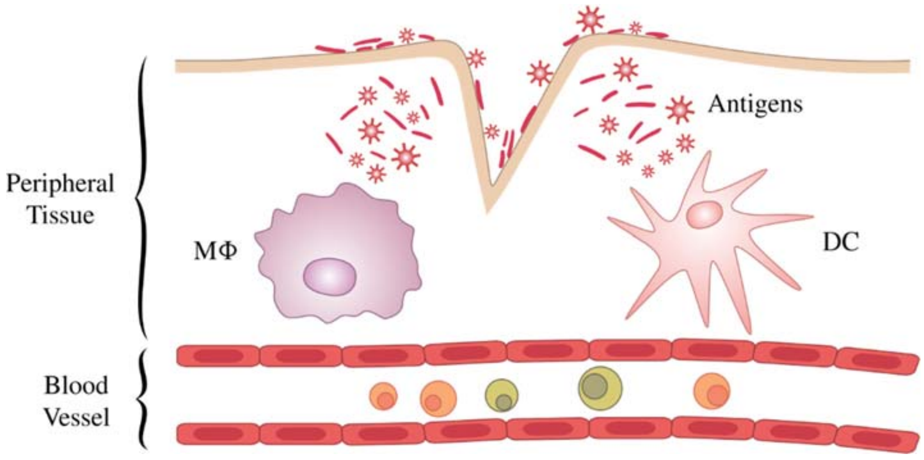


Figure 1.1. Phagocytes in the Peripheral Tissue

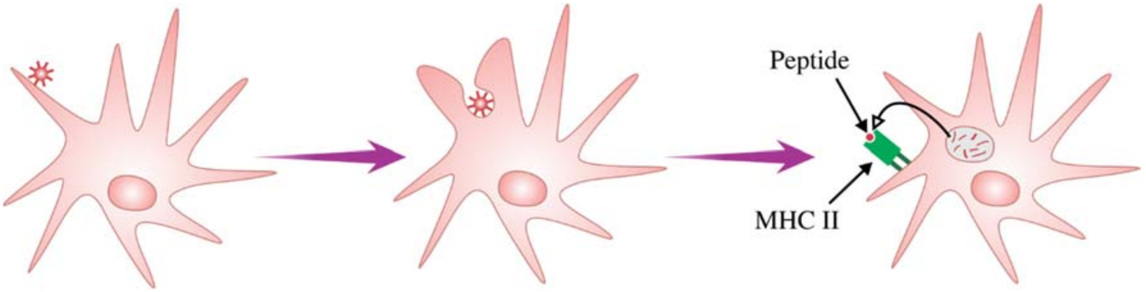


Figure 1.2. A Dendritic Cell Processing an Antigen

After they are developed from stem cells in the bone marrow, DCs move to peripheral tissues that are in contact with the environment, such as skin and inner lining of the nose, lungs, stomach, intestines, and oral cavities where they are in their initial "immature" state. They become mature and migrate to regions rich in T-cells, primarily the lymph nodes, whether they have already captured and processed an antigen or not (steady-state)[2]. In the lymph nodes, they either present peptides of the processed antigen to the T-cells to activate the adaptive immune response or to induce immune tolerance to "harmless" antigens, including those from the body's own tissues, cells, and proteins which prevents the immune response from attacking self cells, a disease known as *autoimmunity*[3,4].

T-cells, which mature in the thymus, exist in two main types in the lymph nodes: Helper T-cells (T_h) and cytotoxic T-cells (T_c). Referring to Figure 1.3, when a DC migrates to the lymph node carrying MHC II-peptide complexes, it attracts T_h-cells to bind to that MHC II-peptide complex. Note that only those T_h-cells with receptors "specific" to the MHC II-peptide complex of the migrated DC can bind to it. Once bound, a T_h-cell is said to be activated, upon which it proliferates and differentiates into "memory" T_h-cells and "effector" T_h-cells. Memory T_h-cells stay in the body for years to provide faster response when the same antigen infects the body once again; this is a feature of the adaptive immune system the innate system lacks, where the adaptive memory provides stronger and faster response which effectively stops the infection with less reliance on the innate system when it occurs the second time. Effector T_h-cells, on the other hand, raise the alarm by releasing small cell-signaling protein molecules known as cytokines responsible of activating both T_c-cells and B-cells. It is worth mentioning here that any infected cell in the body displays peptides of the infectious antigen on their surface via MHC I (MHC I molecules are displayed by any nucleated cell to enable the body to recognize infected cells whereas MHC II molecules are displayed by APCs, such as DCs, MΦs, and B-cells to recognize epitopes of exogenous antigens and discriminate self from non-self). The role of the T_c-cells is to kill the infected cells by binding its epitopes to that "specific" MHC I-peptide complex and releasing special proteins. These proteins are the *perforin*, which inserts itself into the infected cell membrane and forms a pore and the *granzyme*, which induces *apoptosis* (the healthy programmed cell death) in the infected cell. This adaptive immune response from T-cells is referred to as *cell-mediated response*.

B-cells, which mature in the bone marrow, carry "specific protein complexes on their surfaces known as *membrane-bound antibodies* formed by shuffling the DNA during the maturation of these B-cells. These antibodies bind to foreign pathogens with the same protein combination. Upon binding, B-cells process the antigen in a similar way to that done by phagocytes, as mentioned earlier, and display part of the processed antigen on their surface via the MHC II-peptide complex. However, B-cells are not activated until they receive the cytokines released by the effector T_h-cells that are stimulated by the migrated DCs. When activated, B-cells proliferate and differentiate into "memory" B-cells and "effector" B-cells. Like memory T_h-cells, memory B-cells live for years in the body to provide faster B-cell response in case the same pathogen invades the body once again, while "effector" B-cells produce clones of their membrane-bound antibodies and release them as *free antibodies*. One of the important functions of these free antibodies is *opsonization*, in which the antibodies bind to foreign antigens and mark them for attack by phagocytes. The adaptive immune response from B-cells is called *humoral response*.

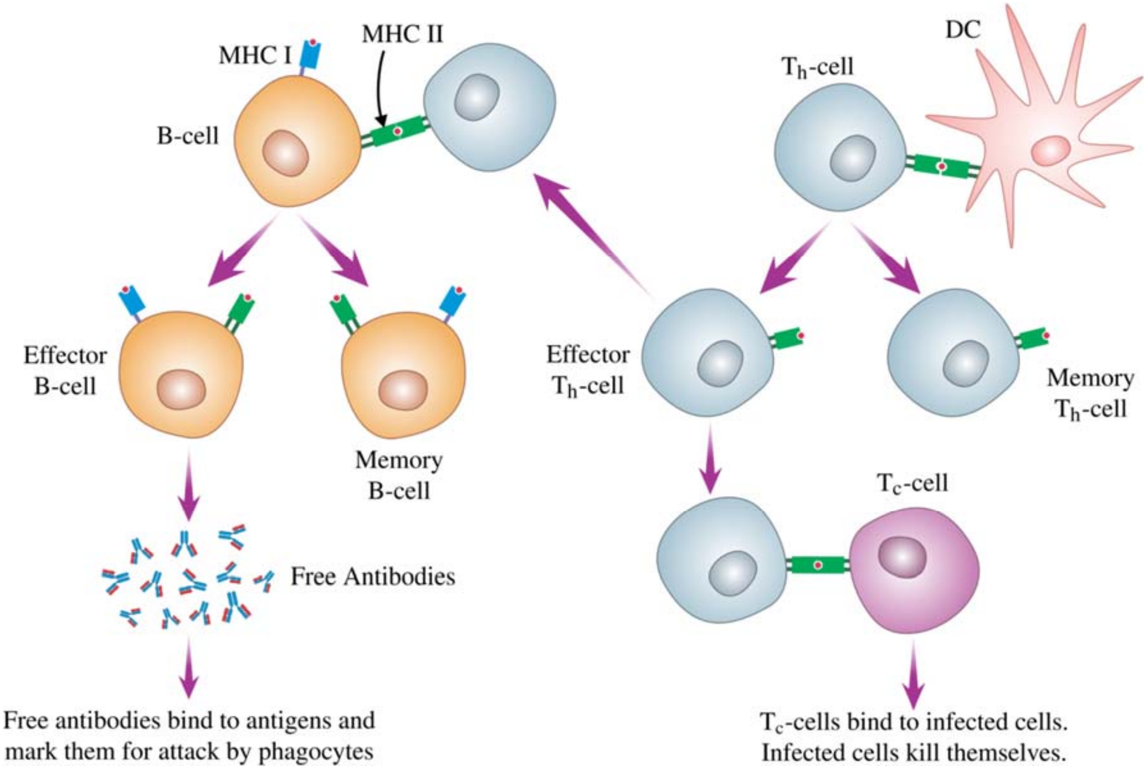

MHC I
MHC II
B-cell
DC
T_h-cell
Effector B-cell
Memory B-cell
Effector T_h-cell
Memory T_h-cell
T_c-cell
Free Antibodies
Free antibodies bind to antigens and mark them for attack by phagocytes
T_c-cells bind to infected cells. Infected cells kill themselves.

Figure 1.3. Activation of T_h-, B-, and T_c-Cells by the APC

The negative selection (NS) process, through which important constituents of the immune system are generated, produces biological agents that have the capability to detect microbial and non-microbial exogenous entities (referred to as non-self) while not reacting to the cells of their own organism (referred to as self). Briefly, the process allows the proliferation of cells that do not match the self, but match the non-self. Therefore, they are only compatible with the non-self and will be capable to detect its presence. The concept is illustrated in Figure 1.4, where the term "antibodies" is used in a most generic way.

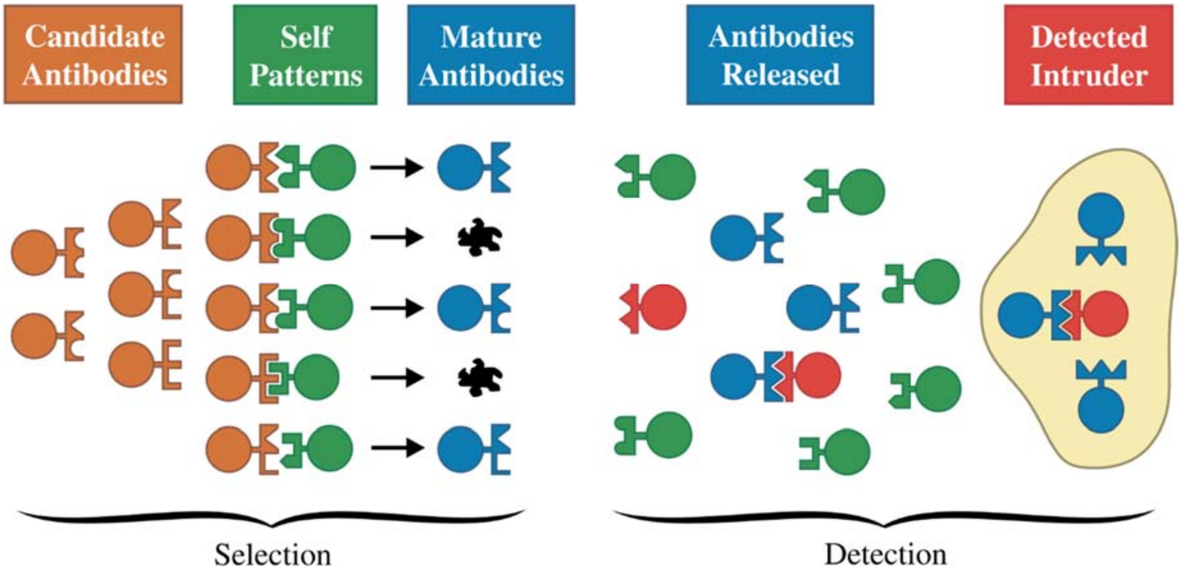


Figure 1.4. The Negative Selection Process

1.2. Artificial Immune System Paradigm

The AIS emerged in recent years as a new computational intelligence paradigm[5,6] with a variety of applications in areas such as anomaly detection, data mining, computer security, adaptive control, and pattern recognition. All initial efforts were directed towards immunity-based fault detection[7,8], which operates in a similar manner as does the biological immune system - according to the principle of self-non-self discrimination - when it detects exogenous antigens while not reacting to the self cells. Prior to this project, an integrated set of methodologies[9-12] for AIS-based detection and identification of a wide variety of aircraft sub-system failures/damages has been designed and implemented at WVU by the authors of this report. Integrated high-performance AIS-based failure detection and identification schemes have been demonstrated to be capable of handling several categories of sub-system abnormal conditions over extended areas of the flight envelope[13]. The potential of the AIS paradigm for flight envelope reduction assessment at post-failure conditions has also been investigated with promising results[14].

The basic idea supporting the AIS paradigm for sub-system fault detection is that an abnormal situation (i.e. failure of one of the aircraft sub-systems, which is considered similar to an invasion by antigens) can be declared when a current configuration of "features" does not match with any configuration from a pre-determined set known to correspond to normal situations. These "features" – similar to the biological chemical markers – represent the encoding of the self. They can include various sensor outputs, states estimates, statistical parameters, or any other information expected to be relevant to the behavior of the system and able to capture the signature of abnormal situations. Extensive experimental data are necessary to determine the self or the hyper-space of normal conditions. Adequate numerical representations of the self/non-self must be used and the data processed such that they are manageable given the computational and storage limitations of the available hardware. The artificial counterpart of the antibodies - the detectors - must then be generated and optimized. This process typically attempts to mimic the variation followed by selection of the T-cells. The mechanisms through which DC acquire, store, process, and transfer information can be used to handle the large amounts of data involved in the ACDIEA process. Handling computationally the large feature hyper-space as a whole is impractical if at all possible for a comprehensive approach. Alternatively, lower order projections of the self can be used, but then self/non-self discrimination outcomes from a large number of such projections must be handled. A DC-based mechanism was developed for this purpose that will be described in details in following sections. A summary of main components of the AIS paradigm and their biological source is presented in Table 1.1.

The NS process through which main cellular components of the biological immune system are generated is mimicked in the generation of detectors. Evolutionary algorithms can be used for optimization in this process[11]. As an alternative mechanism, positive selection (PS) has been explored for AIS design. Through the PS strategy, the detectors are generated to coincide with the self and the process is equivalent to clustering the self data. In this case, an abnormal situation is declared if the explored current configuration does not match any of the detectors. For detection purposes, NS and PS are equivalent; however, using PS within a detection scheme is typically more computationally intensive than using NS because it is necessary to test the complete set of positive antibodies before classifying a sample as abnormal. With the NS approach, the activation of a single negative antibody is enough to declare the presence of an abnormal situation.

A detection logic must be designed for real time operation with a high detection rate and low number of false alarms. The block diagram of the AIS design process for aircraft sub-system abnormal condition detection is presented in Figure 1.5.

Table 1.1. Main Biological Terms and Their AIS Paradigm Representations

Biological Term	AIS Paradigm Representation
Organic markers (proteins and other compounds)	System features or characteristic variables and their values.
Organism or self	Data clusters of feature values acquired from development tests under normal operating conditions.
Alien entities or non-self	Set of complementary clusters covering the feature hyper-space outside of self clusters.
Antigen	Abnormal condition or set of feature values at abnormal conditions
Antibody	Cluster of the non-self or detector
Dendritic cell	A computational unit that processes the outcomes of the self/non-self discrimination process.

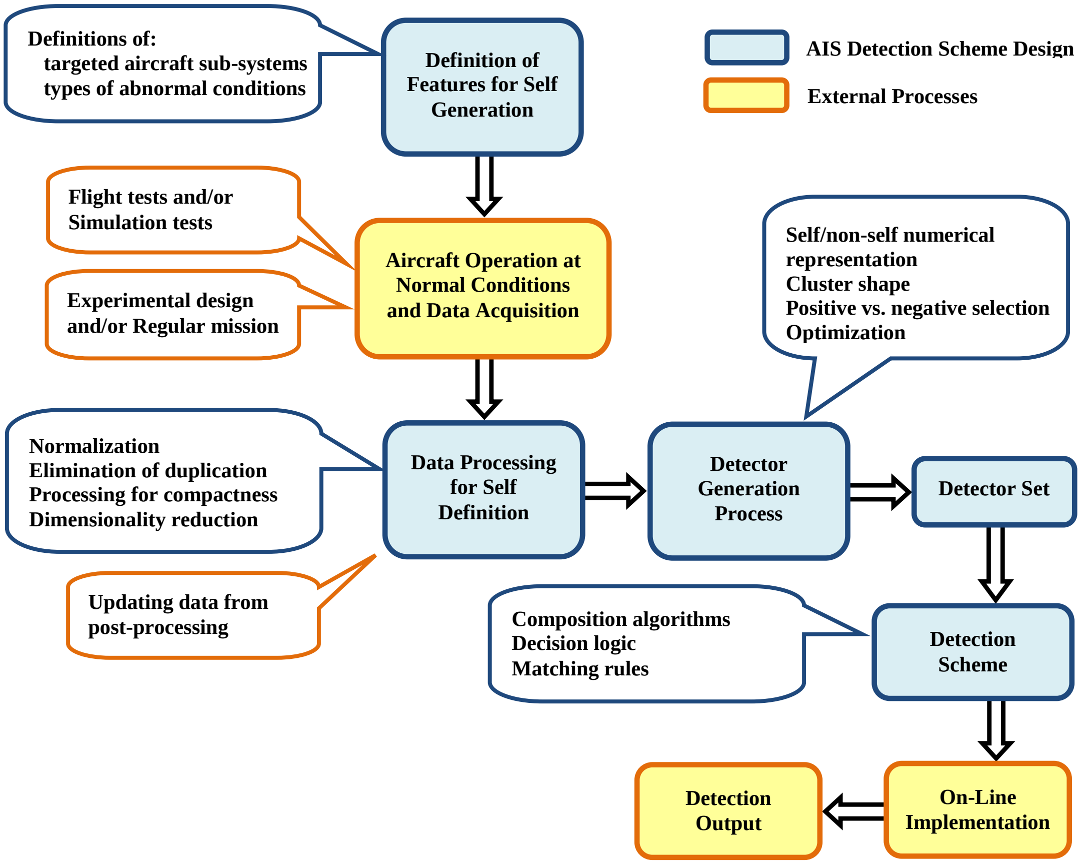

Definitions of:
targeted aircraft sub-systems
types of abnormal conditions
Definition of Features for Self Generation
AIS Detection Scheme Design
External Processes
Flight tests and/or Simulation tests
Experimental design and/or Regular mission
Aircraft Operation at Normal Conditions and Data Acquisition
Self/non-self numerical representation
Cluster shape
Positive vs. negative selection
Optimization
Normalization
Elimination of duplication
Processing for compactness
Dimensionality reduction
Data Processing for Self Definition
Detector Generation Process
Detector Set
Updating data from post-processing
Composition algorithms
Decision logic
Matching rules
Detection Scheme
Detection Output
On-Line Implementation

Figure 1.5. Design of AIS-Based Abnormal Condition Detection Scheme

For the identification and evaluation phases, additional information is needed to structure the non-self hyperspace, which can be obtained through a PS-type of mechanism using data under failure conditions. In particular, for the quantitative direct evaluation, the non-self must be separated into sub-regions corresponding to the different “identifiable” level of failures and PS must be used to assess the distribution of the failure signature within the non-self by labeling the antibodies corresponding to different magnitudes of the failure.

1.3. Definitions, Nomenclature, and Notation

For a general formulation, it will be assumed that the targeted system is the "*aircraft*", including the vehicle itself with all its subsystems (such as actuators, sensors, propulsion, etc.), the human pilot, and the environment. However, the proposed methodology is generally formulated and can be applied to any system and its components.

The generic term of "*abnormal conditions*" (AC) refers to faults and failures of hardware, human pilot-related abnormal situations, operational upset conditions, extreme environmental conditions, and any other situations that require specific attention and/or action for safety purposes. It should be noted that for the practical design and implementation of ACDIEA schemes, the designer is required to define the nature, type, and severity of the AC targeted.

The *detection* represents the process through which the existence of an AC is assessed or, in other words, a failure within at least one of the aircraft sub-systems has been acknowledged. The outcome of the detection process Det is binary:

$$Det \in \{0, 1\}, \quad \begin{array}{ll} Det = 0 & \text{for normal conditions} \\ Det = 1 & \text{for abnormal conditions} \end{array} \tag{1.1}$$

The *identification or isolation* process determines which subsystem has failed. Depending on the complexity of the targeted system, the AC identification can be performed in several phases. For example, a first identification phase could distinguish between actuator and sensor failures. If an actuator failure is identified, a second phase could determine whether it is an elevator, aileron, or rudder failure. Finally, a third identification phase could establish that the left elevator has failed or the left in-board elevator has failed, if the aircraft is equipped with additional redundancy. The outcome of the identification process, Idt, can be formulated as an Ns-dimensional vector with binary components, where Ns is the total number of sub-systems considered:

$$Idt = \begin{bmatrix} id_1 & id_2 & \cdots & id_{Ns} \end{bmatrix}, \quad \begin{array}{ll} id_j = 0 & \text{for subsystem } j \text{ at normal conditions} \\ id_j = 1 & \text{for subsystem } j \text{ at abnormal conditions} \end{array} \tag{1.2}$$

Alternatively, the outcome of the identification process can be formulated as a set of integers labeling the failed sub-system:

$$Idt = \lfloor id_1 \quad id_2 \quad \cdots \quad id_{Nsf} \rfloor \tag{1.3}$$

where Nsf is the number of failed sub-systems and id_j represents the label associated to each subsystem, typically:

$$id_j \in \{1 \quad 2 \quad \cdots \quad Ns\} \tag{1.4}$$

Once a failure has been detected and correctly identified, the AC *evaluation* process must address three aspects. One is of a qualitative nature and involves determining the type of the failure. For example, the qualitative evaluation is expected to determine if an actuator failure consists of a locked actuator, or a freely moving control surface, or a reduction of control efficiency. The other two aspects are of a quantitative nature and can be defined as direct and indirect. The direct failure evaluation consists of estimating the magnitude or severity of the failure (e.g. left aileron locked at +10deg). The indirect failure evaluation includes the re-assessment of the flight envelope and prediction of the limitations and constraints on the performance and handling qualities inflicted by the presence of the failure.

The outcome of the qualitative evaluation, $Ev1$, is an Nsf -dimensional string of integers labeling the type of failures for all Nsf affected sub-systems out of a list of Nft_k, $k = 1,2,\ldots Ns$, potential types associated to each sub-system id_j, $j = 1,2,\ldots Nsf$:

$$Ev1 \in \{1 \quad 2 \quad \ldots \quad Nft_k\} \tag{1.5}$$

Let the set F of all failures considered be expressed as:

$$F = \{f_1 \quad f_2 \quad \cdots \quad f_{N_F}\} \tag{1.6}$$

where N_F is the total number of AC considered and:

$$N_F = \sum_{k=1}^{N_S} Nft_k \tag{1.7}$$

The outcome of the direct quantitative evaluation process, $Ev2$, can take a numerical value that relates to the magnitude or severity of the AC. For example:

$$Ev2 = 40\% \tag{1.8}$$

where 40% represents, let's say, the relative area of the left aileron affected by structural damage. While possible in some cases, this level of accuracy is, obviously, difficult to achieve in general. Alternatively, the direct quantitative evaluation could provide a severity estimate mapped on a discrete set:

$$Ev2 \in \{low\ severity \quad medium\ severity \quad high\ severity\} \tag{1.9}$$

or, it could provide a severity evaluation based on fuzzy logic and expressed as:

$$Ev2 = [v_1 \quad v_2 \quad \ldots \quad v_m] \tag{1.10}$$

where v_i are fuzzy membership values with respect to the *m* linguistic values associated to the severity metric $Ev2$ (e.g. low severity, medium severity, and high severity). $Ev2$ can also be expressed as a scalar by using a defuzzification algorithm:

$$Ev2 = \Delta([v_1 \quad v_2 \quad \ldots \quad v_m]) \tag{1.11}$$

where Δ is the defuzzification operator.

The outcome of the indirect quantitative evaluation process, $Ev3$, typically represents a set of new ranges at post failure conditions of the variables y_i that define the flight envelope, in its most general meaning. Let the number of these variables be Ne. Then:

$$Ev3 = \left[\left(y_{1min}, y_{1max} \right) \ \left(y_{2min}, y_{2max} \right) \ \ldots \ \left(y_{Ne\,min}, y_{Ne\,max} \right) \right] \tag{1.12}$$

More specifically, let a *directly involved variable* (DIV) v_δ in the AC be a variable whose alteration or abnormal variation is directly and significantly the result of the AC. Typically, DIV are used to define/characterize the AC. They may be part of the feature set or not. If they are not, then a relationship between the DIV and some other variable(s) in the feature set must be established. This process will define *equivalent directly involved variables* (EDIV), v_ε, which are part of the feature set. For example, consider the case of the left stabilator locked failure. The DIV can be defined as the left stab deflection δ_{eL}. It obviously defines the failure; however, let us assume that it is not part of the feature set that has been established. A relationship can be formulated between the left stab deflection δ_{eL} and the longitudinal stick displacement d_e - which presumably is a feature – of the form $d_e = f(\delta_{eL})$. Therefore, the EDIV is in this case, d_e. For each failure considered, a set of variables v_E must be determined that are affected by the failure, that are part of the feature set, and that are relevant from the point of view of aircraft operation (they are flight envelope relevant variables). These variables may be determined through analytical means but also through the analysis of the 2-D selves.

The *accommodation* process can take two forms: *passive accommodation* through warnings and information displayed in the cockpit and *active accommodation* through direct compensation as integral part of the control laws. The passive accommodation represents actually the re-processing of the outcomes of the detection, identification, and evaluation and their delivery to the pilot in a form that is easy to perceive and understand. The direct accommodation involves triggering of pre-existing compensating modules within the control laws and/or actual computation of commands at post-failure conditions.

The *feature variables* or shortly *features* are the variables φ_i that completely define the targeted system and are expected to capture the fingerprints of all AC considered, in terms of occurrence, presence, and severity. They can be (sub-)system states, inputs, control system variables, estimated values, etc. The set of all features $\Im$:

$$\Im = \left\{ \varphi_i \quad | \quad i = 1,2,\ldots N \right\} \tag{1.13}$$

defines a *feature point P* as a set of simultaneous values of all features φ_i that can be obtained through measurements or simulation, at normal or abnormal conditions. The set $\Im$ defines an *N*-dimensional hyper-space $\mathscr{U}$, which will be referred to as the "*Universe*". An orthogonal coordinate system (CS) equally denoted *U* is associated to this hyper-space. Its origin is at point *O* with coordinates $\left[\varphi_1 = 0 \quad \varphi_2 = 0 \quad \ldots \quad \varphi_N = 0 \right]$. Therefore, the feature point *P* can be represented by the position vector with respect to O, $\vec{r}^{\,OP}$, whose coordinates with respect to CS $\mathscr{U}$ are denoted as:

$$\left\lfloor \vec{r}^{\,OP} \right\rfloor_{\mathscr{U}} = \left[\varphi_{1P} \quad \varphi_{2P} \quad \dots \quad \varphi_{NP}\right]_{\mathscr{U}}^{T} \tag{1.14}$$

The self $\mathscr{S}$ is defined as the hyper-sub-space of all possible feature points at normal conditions. All other points in U form the non-self $\hat{\mathscr{S}}$. Therefore:

$$\mathscr{S} \bigcup \hat{\mathscr{S}} = \mathscr{U} \qquad \text{and } \mathscr{S} \bigcap \hat{\mathscr{S}} = \Phi \tag{1.15}$$

For computational tractability and practical reasons, the self points are clustered. These *self clusters* can be shaped as hyper-spheres, hyper-rectangles, or hyper-ellipsoids. Clustering of the non-self generates what is referred to as *detectors*. They may be directly used in the detection process. Structuring the non-self by adding information to the detectors makes them capable to be used in the failure identification process. Such upgraded detectors will be referred to as *identifiers*. Clusters of the self or non-self that are processed to be used for evaluation or accommodation will be referred to as *evaluators* and *compensators*, respectively.

The threshold between "normal" and "abnormal" (i.e. between self and non-self) is represented by an *N*-dimensional surface:

$$\Sigma(\varphi_1, \ \ \varphi_2, \ \ \dots, \ \ \varphi_N) = 0 \tag{1.16}$$

Lower dimensional projections of the self will be referred to as *sub-selves*. Let us assume that we consider a 2-dimensional projection (or sub-self) defined by features φ_1 and φ_2 and we collect all possible feature points at normal conditions. Area 0 will result as shown in Figure 1.6. Note that typically normalized values are used for the features; therefore, a working sub-region of the universe $\mathscr{U}$ is used next, which is represented as a hyper-cube of side 1. The projection of Σ is represented as straight line segments for convenience, without loss of generality. All points outside Area 0 belong to the non-self $\hat{\mathscr{S}}$. That means that whenever a pair of values (φ_1, φ_2) is measured that lies outside Area 0, a failure can be declared. However, whenever a pair of values (φ_1, φ_2) is measured that lies inside Area 0, it cannot be necessarily concluded that the conditions are normal. AC whose dynamic fingerprints depend on one or more other features can produce projections inside Area 0.

The *order* of an AC, denoted as *Ord(AC)*, is defined as the minimum number of features necessary to completely detect that AC. A *complete detection* is said to be possible if all feature points produced at AC fall within the non-self. For example, a first order failure would require only one feature φ^* for detection. In other words, the presence of the failure necessarily produces always values of φ^* that are outside a normal range. In Figure 1.6, if Area 1 is the projection of all points under a certain failure, then that failure is first order with $\varphi^* = \varphi_1$. Similarly, Area 2 represents an AC of degree 2, and Area 3 a failure of degree 3 or higher. A complete detection of the failure represented by Area 1 can be performed using the 2-dimensional self defined by φ_1 and φ_2, but also using only the projection, or the 1-dimensional sub-self defined by φ_1. The 2-dimensional self is necessary and sufficient to completely detect the failure represented by Area 2. Using the sub-self defined by φ_1 will never succeed detecting this failure, while using the sub-self defined by φ_2 will allow detection in some instances, but not a complete detection. Using the 2-dimensional self to detect the third failure will not ensure a complete detection. However, it should be noted that, in practice, the approach could achieve very high rates of detection if the feature points that are projected inside the self are reached with very low probability.

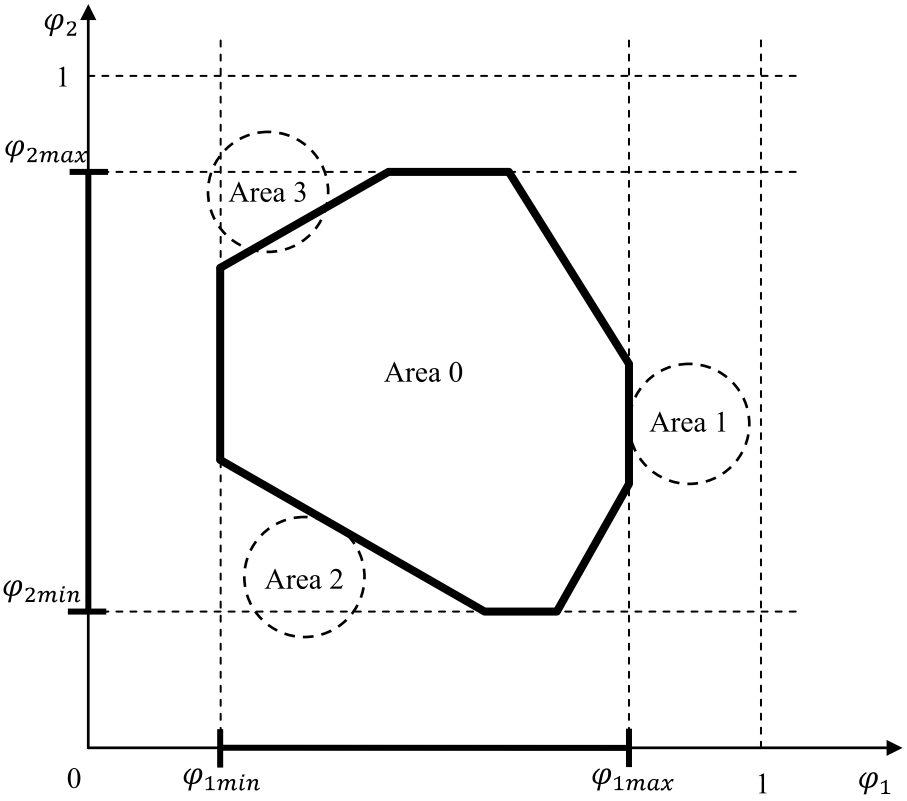


Figure 1.6. Self/Non-Self 2-Dimensional Projection

1.4. Aircraft Abnormal Condition Management Process

The aircraft AC management (ACM) based on the AIS paradigm can be considered to include three main components functionally connected in a closed loop as shown in Figure 1.7:

- off-line ACM system design and implementation
- on-line AC detection, identification, evaluation, and accommodation
- post-processing of flight data and ACDIEA outcomes

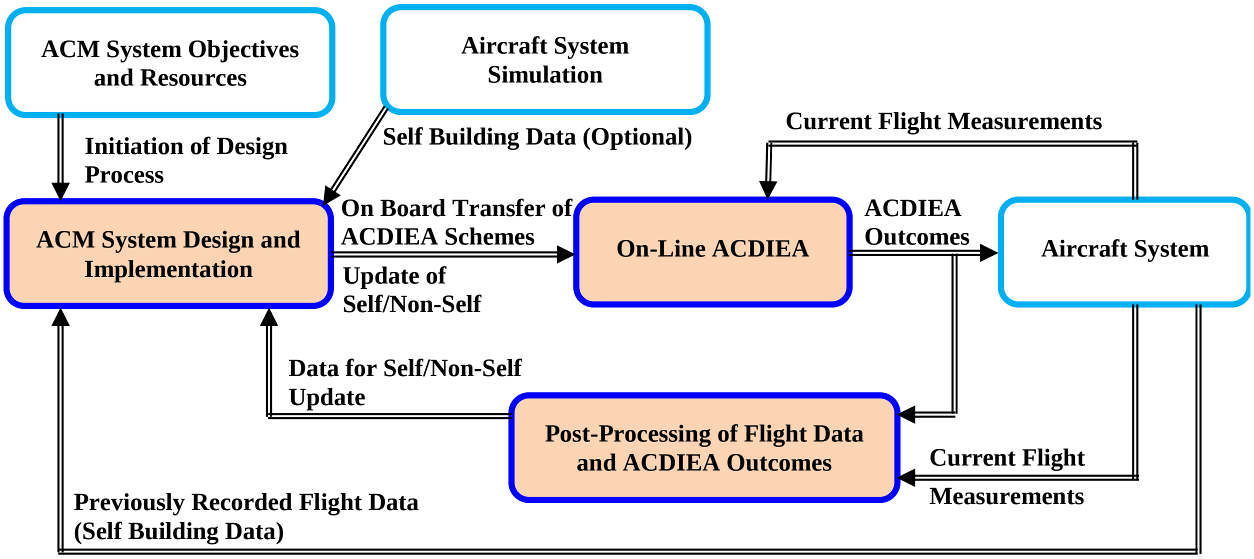

ACM System Objectives and Resources
Aircraft System Simulation
Initiation of Design Process
Self Building Data (Optional)
Current Flight Measurements
ACM System Design and Implementation
On Board Transfer of ACDIEA Schemes
Update of Self/Non-Self
On-Line ACDIEA
ACDIEA Outcomes
Aircraft System
Data for Self/Non-Self Update
Post-Processing of Flight Data and ACDIEA Outcomes
Current Flight Measurements
Previously Recorded Flight Data (Self Building Data)

Figure 1.7. AIS-Based ACM Process

The *off-line ACM system design and implementation* has as outcome the development of an integrated and comprehensive ACDIEA scheme. The design depends on a clear definition of the aircraft sub-systems that are targeted, the types of AC (including known and unknown failures), the failure severity scale, the flight envelope variables, and the nature and level of passive and active accommodation. The development of the ACDIEA scheme requires the availability of large amounts of measured data that must be pre-processed for self/non-self generation and structuring. For a comprehensive solution, acquiring and processing these data is considerably less difficult and expensive than developing extensive accurate models, as required by alternative approaches, still with questionable level of success. The block diagram of the ACM design component is presented in Figure 1.8.

The ***on-line ACDIEA*** process implies the real time operation of the ACDIEA scheme. Sets of current values of the features measured in flight at a certain sampling rate are compared against the detectors, identifiers, evaluators, and compensators and the outcomes of the ACDIEA are generated. These outcomes are transferred to the pilot, the on-board monitoring and recording system, and the automatic fault tolerant control laws. The block diagram of the on-line ACDIEA component is presented in Figure 1.9.

The ***post-processing*** of ACDIEA outcomes and the analysis of false alarms and failed detections in conjunction with current measured values of the features can potentially be used to modify/extend the sets of detectors, identifiers, evaluators, and compensators and improve the overall performance for future operation. The block diagram of the post-processing component is presented in Figure 1.10.

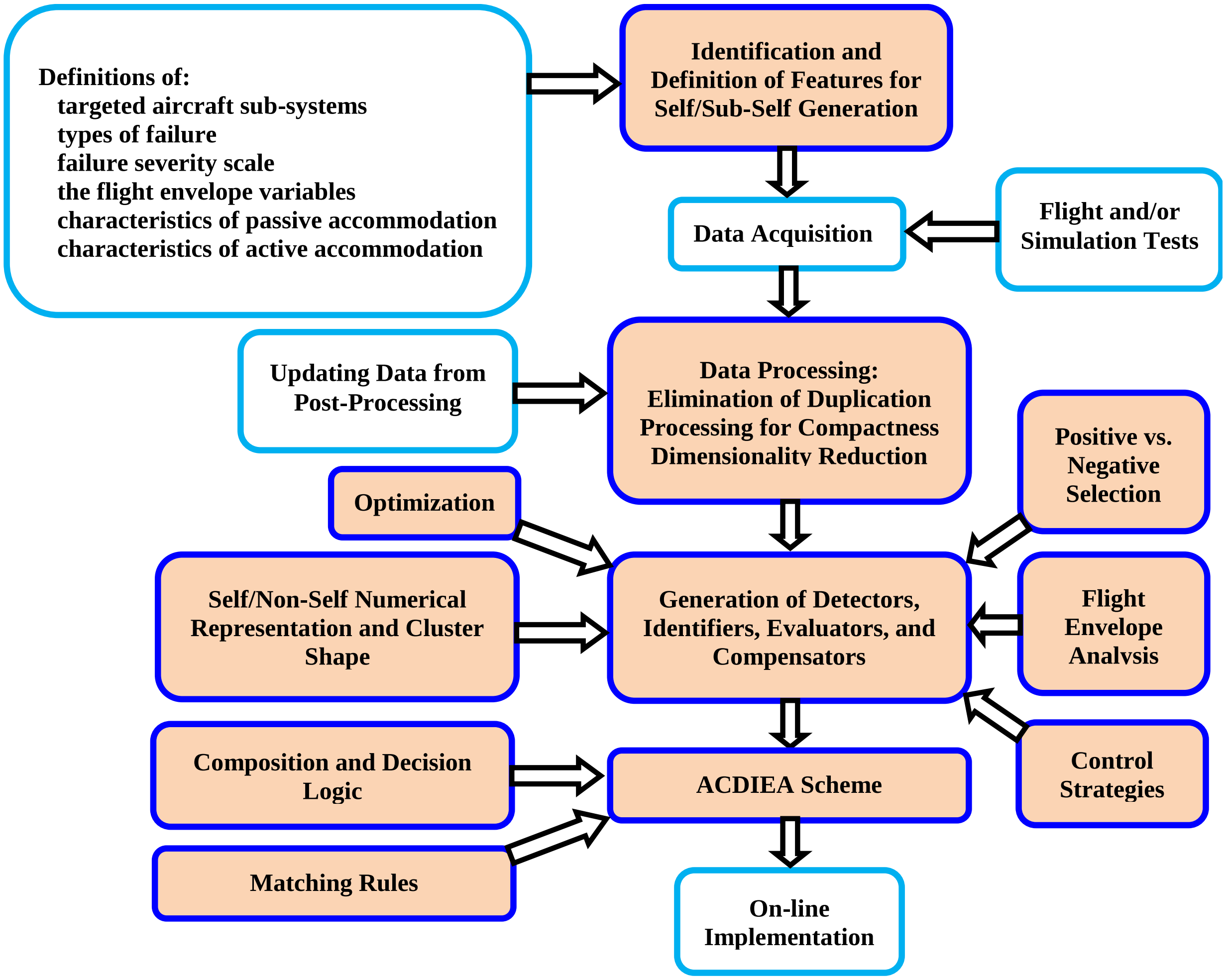

Definitions of:
targeted aircraft sub-systems
types of failure
failure severity scale
the flight envelope variables
characteristics of passive accommodation
characteristics of active accommodation
Identification and Definition of Features for Self/Sub-Self Generation
Data Acquisition
Flight and/or Simulation Tests
Updating Data from Post-Processing
Data Processing: Elimination of Duplication Processing for Compactness Dimensionality Reduction
Positive vs. Negative Selection
Optimization
Self/Non-Self Numerical Representation and Cluster Shape
Generation of Detectors, Identifiers, Evaluators, and Compensators
Flight Envelope Analysis
Composition and Decision Logic
ACDIEA Scheme
Control Strategies
Matching Rules
On-line Implementation

Figure 1.8. AIS-Based ACM System Design

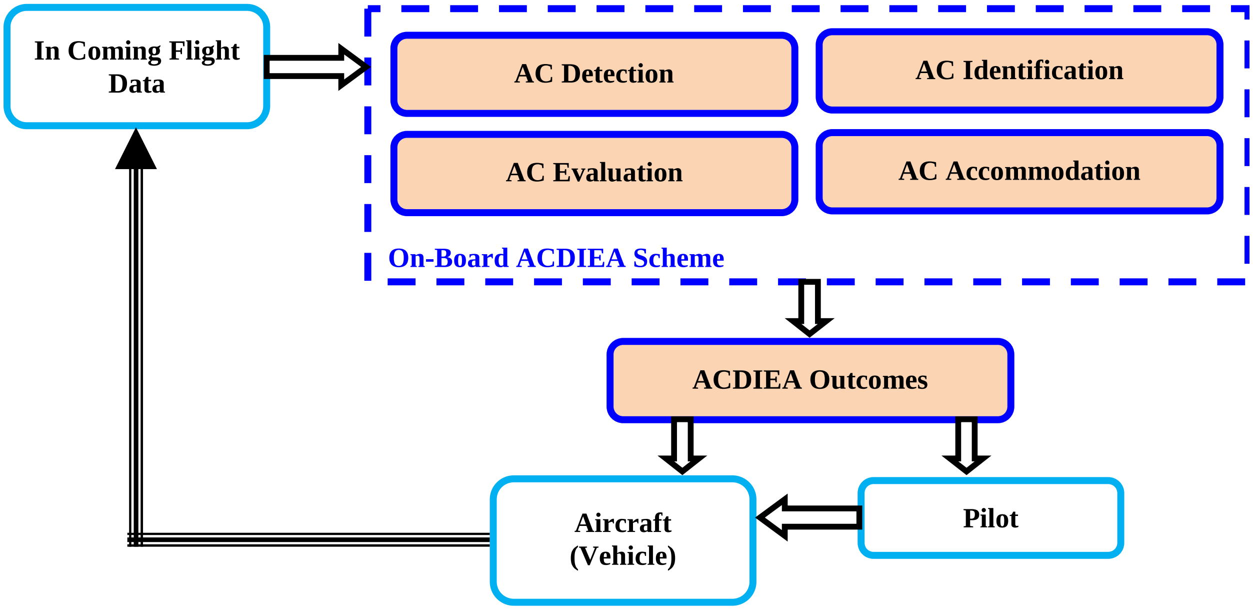


Figure 1.9. On-Line AC Detection, Identification, Evaluation, and Accommodation

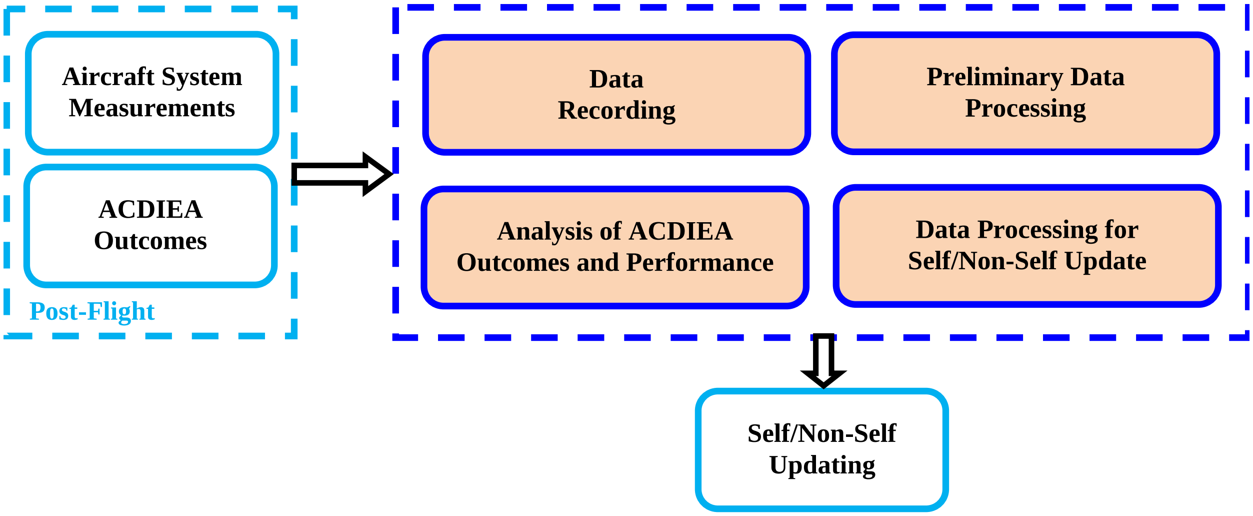

Aircraft System Measurements
ACDIEA Outcomes
Post-Flight
Data Recording
Preliminary Data Processing
Analysis of ACDIEA Outcomes and Performance
Data Processing for Self/Non-Self Update
Self/Non-Self Updating

Figure 1.10. Post-Processing of Flight Data and ACDIEA Outcomes

1.5. Definition of the Self and Non-Self

1.5.1. General Algorithm

One of the most critical elements of the design of the ACM system is the selection of features. They must be relevant to all four processes (ACDIEA). Their number and nature depend on the aircraft sub-systems that are targeted, the types of AC, the failure severity scale, the flight envelope variables, and the nature and level of passive and active accommodation. For a comprehensive and integrated approach, all states, inputs, and variable parameters of all subsystems considered must be taken into account. It should also be noted that a complete detection of an *N*-degree failure requires *N* relevant features, therefore, an *N*-dimensional self. If the number of subsystems considered is Ns, then the states Xi, inputs Ui, and other relevant parameters Pi corresponding to sub-system *i* $i = 1,2,\ldots N_S$ can be expressed as, respectively:

$$Xi = \begin{bmatrix} x_{i1} & x_{i2} & \ldots & x_{iNsi} \end{bmatrix}^T \tag{1.17}$$

$$Ui = \begin{bmatrix} u_{i1} & u_{i2} & \ldots & u_{iNui} \end{bmatrix}^T \tag{1.18}$$

$$Pi = \begin{bmatrix} p_{i1} & p_{i2} & \ldots & p_{iNpi} \end{bmatrix}^T \tag{1.19}$$

For any sub-system *i*, the maximum order an AC can have is:

$$Ni = N_{si} + N_{ui} + N_{pi} \tag{1.20}$$

Therefore, for a complete detection of all AC from an exhaustive set *F*, the number of necessary features to build the self/non-self is:

$$N = \sum_{i=1}^{N_S} \left(N_{si}^* + N_{ui}^* + N_{pi}^* \right) \tag{1.21}$$

where the *-variables exclude duplication among the sub-systems. If all states, inputs, and parameters are distinct, then:

$$N_{si}^* = N_{si}, \quad N_{ui}^* = N_{ui}, \text{ and } \quad N_{pi}^* = N_{pi} \tag{1.22}$$

otherwise, for example:

$$N_{si}^* = \begin{cases} N_{s1} & for\ i = 1 \\ N_{si} - n & for\ i > 1 \end{cases} \tag{1.23}$$

where *n* is the number of duplicate states for each sub-system, that is the states with the property that $x_{ij} \in X_i$ and $x_{ij} \in X_{i-1}$. It can be easily noticed that the number of necessary features *N* and hence the dimensionality of the self/non-self can become excessively large and create significant computational problems[9]. These problems can be avoided under certain conditions by using lower dimensional sub-selves, or projections, within a hierarchical multi-self (HMS) strategy[10].

Let the maximum order of failures in the set *F* be denoted as:

$$N_{max} = \max_{j}\left(Ord(f_j)\right), \quad j = 1,2,\ldots,N_F \tag{1.24}$$

and
$$Ord(f_m) = N_{max} \tag{1.25}$$

If $N_{max} = N$, then a complete detection for f_m can only be obtained, in general, with an *N*-dimensional self. If the self/non-self has particular shapes, then at least on one of the *(N-1)*-dimensional projections, the projection of the feature points at AC will fall inside the non-self. For example, in Figure 1.11, a 3-dimensional case is presented in which an AC feature point, AC1, is projected outside all three 2-dimensional projections. Feature point AC2 is projected outside one of the *N-1* dimensional sub-selves. For specific shapes of the self/non-self, it is possible that some AC feature points are projected inside all sub-selves, as illustrated by feature point AC3.

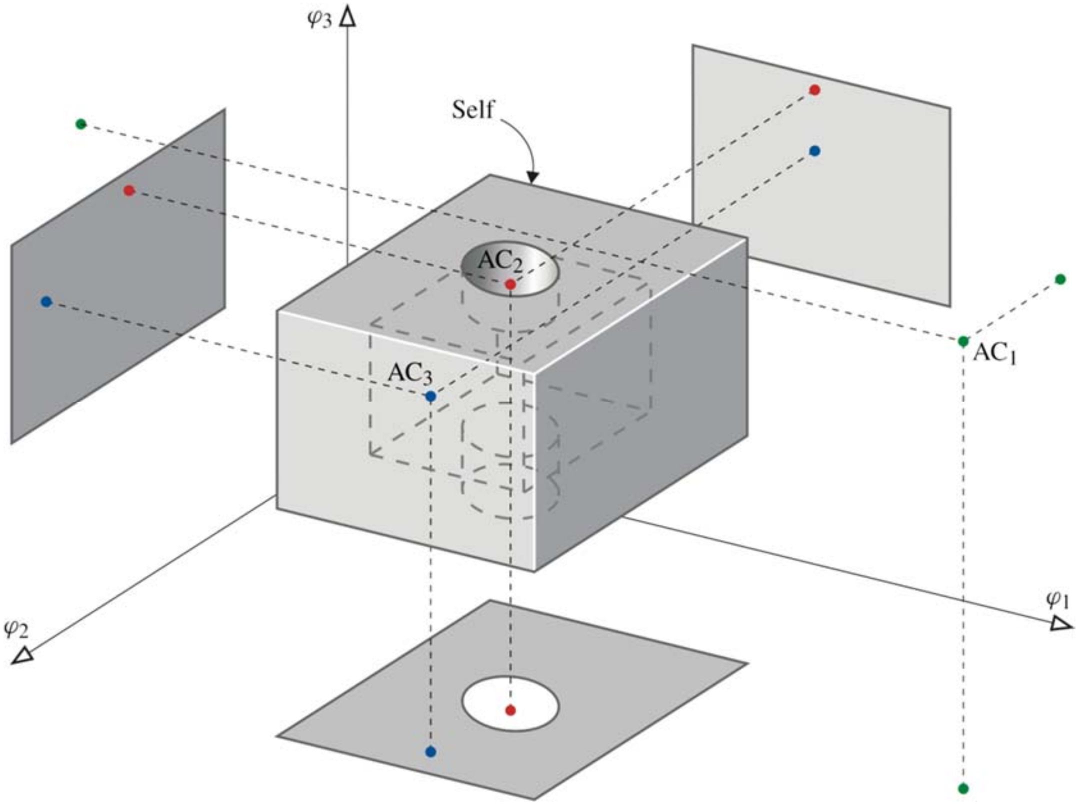


Figure 1.11. Potential of Sub-Selves to Capture the Abnormal Conditions

If $N_{max} < N$, then a complete detection for f_m can be obtained by using an N_{max}-dimensional sub-self with proper features. In this situation, the *N*-dimensional self does not need to be considered. A complete detection for all failures can be obtained by using all possible

N_{max}-dimensional projections or sub-selves. The approach is conservative and some of the projections may be ignored or replaced by lower dimensional projections without loss of performance.

Determining the order of an AC may be difficult in many cases or even impossible; therefore, N_{max} may be unknown. In this situation, a low value for N_{max} may be assumed and the self/non-self built as the set of all possible N_{max}-dimensional sub-selves, with the observation that the higher the assumed value of N_{max}, the higher the likelihood of better performance. The total number of possible N_{max}-dimensional sub-selves for *N* features is given by:

$$N_{ss} = C_N^{N_{max}} = \frac{N!}{N_{max}!(N - N_{max})!} \tag{1.26}$$

The generation of the self/non-self requires large amounts of measured data at normal conditions, ideally covering the entire flight envelope. These *N*-dimensional feature points are clustered and the self $\mathscr{S}$ is represented as a set *S* of hyper-bodies including these clusters, which will be referred to as *self clusters*. Similar hyper-bodies are used to represent the non-self $\hat{\mathscr{S}}$ as well and will be referred to as *detectors*. The geometry of these hyper-bodies can potentially have an impact on the efficiency of the detector generation process and on the detection itself. They determine how well the non-self is covered, how many detectors are necessary, and how intensive the computational process is. The following shapes for the self/non-self representation can typically be considered:

- Hyper-cubes – determined by an *N*-dimensional center and one value for the side;
- Hyper-rectangles – determined by an *N*-dimensional center and *N* values for the sides;
- Hyper-spheres – determined by an *N*-dimensional center and one value for the radius;
- Hyper-ellipsoid of rotation – determined by an *N*-dimensional center and two values for the axes;
- Generalized hyper-ellipsoid – determined by an *N*-dimensional center and *N* values for the axes.

For all shapes (except hyper-spheres) variable orientation can be considered as determined by an additional *N*-dimensional vector. For example, for the hyper-spherical representation with *Nc* clusters c_i, the self and the self clusters can be expressed as:

$$S = \{c_1 \quad c_2 \quad \dots \quad c_{Nc}\}, \quad c_i = [C_i \quad Rc_i] = [\varphi_{1i} \quad \varphi_{2i} \quad \dots \quad \varphi_{Ni} \quad Rc_i] \tag{1.27}$$

where C_i is the center and Rc_i is the radius of the cluster. For the same hyper-spherical representation with *Nd* detectors d_j, the non-self and the detectors can be expressed as:

$$\hat{S} = \{d_1 \quad d_2 \quad \dots \quad d_{Nd}\}, d_j = \lfloor D_j \quad Rd_j \rfloor = \lfloor \varphi_{1j} \quad \varphi_{2j} \quad \dots \quad \varphi_{Nj} \quad Rd_j \rfloor \tag{1.28}$$

where D_j is the center and Rd_i is the radius of the detector.

When generating the self and the non-self, the following optimization criteria should be considered:

- no overlapping among detectors and self clusters;
- minimum empty space inside the self clusters;
- minimum un-covered areas inside the non-self;
- minimum overlapping among self clusters;
- minimum overlapping among detectors;
- minimum number of detectors for a desirable resolution.

The block diagram of the self/non-self generation process is presented in Figure 1.12. A set of computational tools have been developed at WVU for the generation, optimization, and verification of detector sets within the AIS paradigm[16-18]. The WVU Immunity-based Failure Detector Optimization and Testing tool relies primarily on evolutionary computation providing a wide selection of algorithms and options as well as capabilities for testing and tuning[11,15].

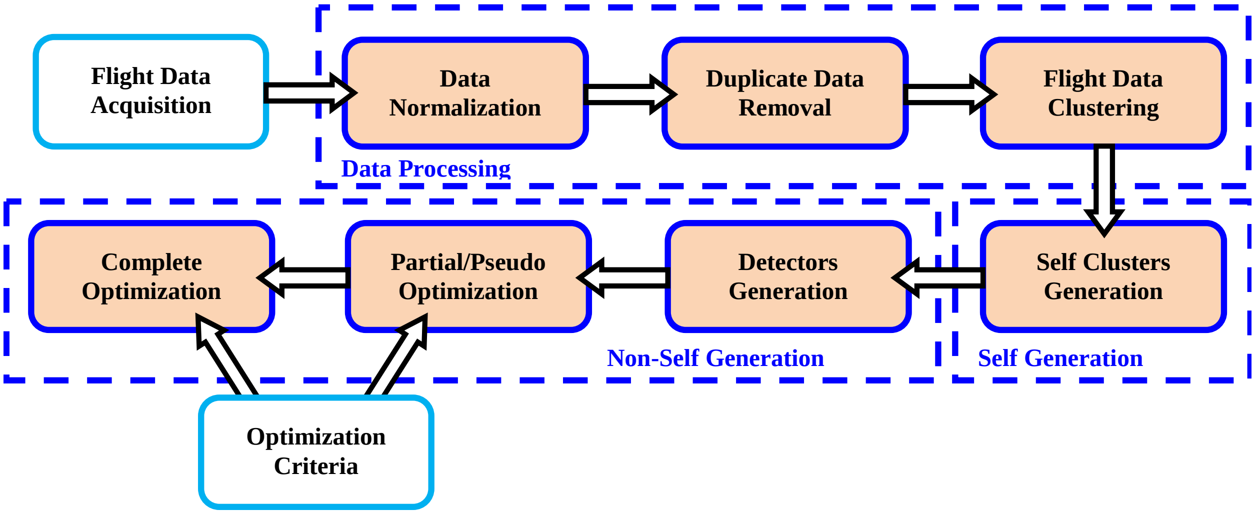


Figure 1.12. Self/Non-Self Generation Process

To support the AC identification and evaluation processes, the non-self must be structured to provide information regarding the sub-system affected, the type of the failure, and the failure severity. This can be achieved by attaching to each detector d_j a set of structuring tags dS_j:

$$dS_j = \{dSys_j \quad dFt_j \quad dFs_j\} \tag{1.29}$$

where, in general, $dSys_j \subset \{1 \quad 2 \quad \dots \quad N_S\}$, $dFt_j \subset \{1 \quad 2 \quad \dots \quad N_F\}$, and dFs_j consists of a set of failure severity metrics associated to each failure type in dFt_j and $size(dSys_j) = size(dFt_j) = size(dFs_j)$. The implication here is that each sub-system may be affected by only one failure at a time. The case in which a combination of failures occurs can be handled within this framework by defining in the set F a distinct AC with such multiplicity characteristics. Alternatively, for each sub-system represented by $dSys_j$, the elements of dFt_j and dFs_j may be defined as sub-sets of size larger than 1. In this case, $size(dSys_j) < size(dFt_j) = size(dFs_j)$. The three sub-sets in Eq. (1.29) will each have only one element in the ideal situation when the N features have been defined perfectly and completely, one *N*-dimensional self is used, and the size of the detectors provides perfect resolution. In other words, any given detector can be associated to only one sub-system, one failure type, and one level of failure severity.

The process of generating the antibodies requires adequate numerical representations of the self/non-self and adequate data processing such that they are manageable given the computational resources and storage limitations of the hardware. In this project, 2-dimensional projections were used for self/non-self representation for which antibodies were generated using two alternative approaches that are described next. In the first approach, all raw test data available are collected in one file before a set of antibodies is generated. In the second approach, the processing of smaller individual sets of data is performed by clustering and then combining the clusters in a single set for detector generation. Using the two approaches, data for each combination of features corresponding to a particular projection or sub-self are processed separately to produce a set of antibodies by covering the respective 2-dimensionmal non-self. The block diagram of the AIS antibody generation methods is presented in Figure 1.13.

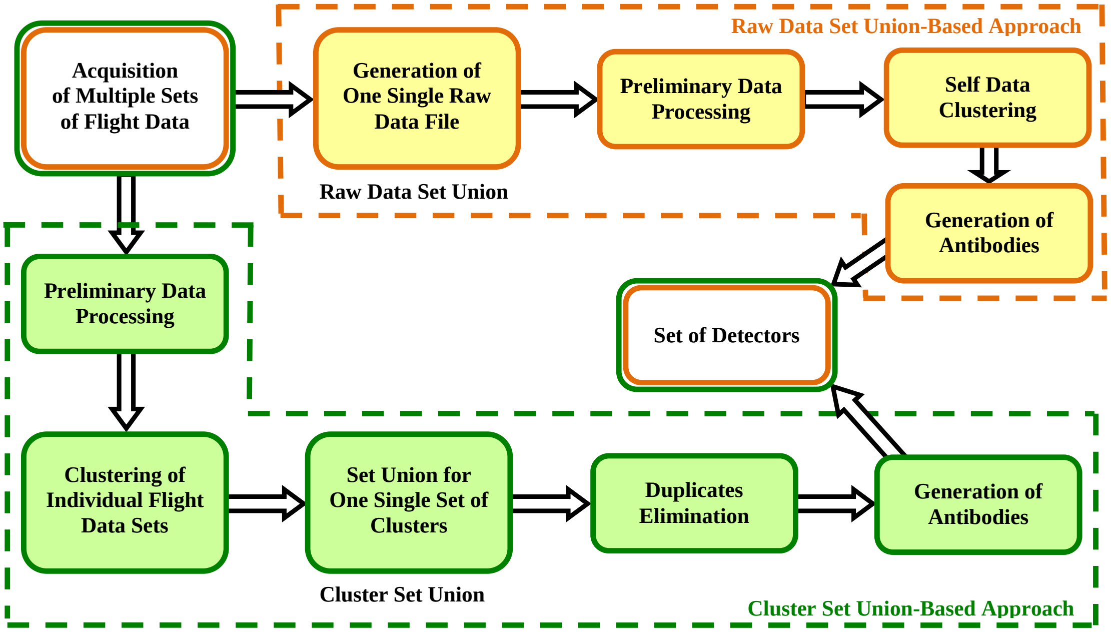

Raw Data Set Union-Based Approach
Acquisition of Multiple Sets of Flight Data
Generation of One Single Raw Data File
Preliminary Data Processing
Self Data Clustering
Raw Data Set Union
Generation of Antibodies
Preliminary Data Processing
Set of Detectors
Clustering of Individual Flight Data Sets
Set Union for One Single Set of Clusters
Duplicates Elimination
Generation of Antibodies
Cluster Set Union
Cluster Set Union-Based Approach

Figure 1.13. Two Methods for AIS Antibodies Generation

1.5.2. Raw Data Set Union Method

The raw data set union method (RDSUM) processes experimental data at normal conditions in four main phases or modules as presented in Figure 1.13.

Generation of Single Data File. Raw data from different flights or simulator tests are combined in one single data file. The data are left intact and no further processing is performed here.

Data Preprocessing. Preprocessing of the data includes two steps: normalization and duplicate elimination. The raw data received from the data fusion component are normalized between 0 and 1. Therefore, the feature space becomes a unit hypercube. The normalization factor for each dimension is determined as the span of the flight data plus a percentage margin. Duplicate points of the normalized data are then eliminated to reduce the size of the data file.

This process decreases the amount of storage and computing resources needed, while preserving the information content of the data. Note that implicit duplicate point elimination may also take place during the clustering process, which follows.

Self Data Clustering. The data produced from the previous processing component define self points that need to be represented by a definite number of geometric hyper-bodies, referred to as clusters. This can be done by using the "*k*-means" clustering algorithm. A modified version of this algorithm[17] was used to represent the clusters as hyper-spheres, but they could also be represented as hyper-rectangles or hyper-ellipsoids.

Generation of Antibodies. Self clusters are used to generate antibodies by covering the non-self hyper-space with hyper-bodies similar to the clusters. An enhanced negative selection algorithm for real-valued representation with variable non-self radius[17] was used within this project. The algorithm ensures that there is no overlapping with the self and that the non-self is covered to a desired predetermined level. It should be noted that the algorithm requires a number of specific parameters that must be carefully selected and correlated. The antibodies generation process can be stopped after a prescribed number of iterations when a preset maximum number of acceptable detectors is reached or when a desired coverage of the non-self is achieved.

Figure 1.14 shows sample 2-dimensional self clusters along with the AIS antibodies generated for a projection or sub-self defined by the non-dimensional roll rate neural network output feature NN_p and the reference roll rate p_{ref} using the raw data set union method described above.

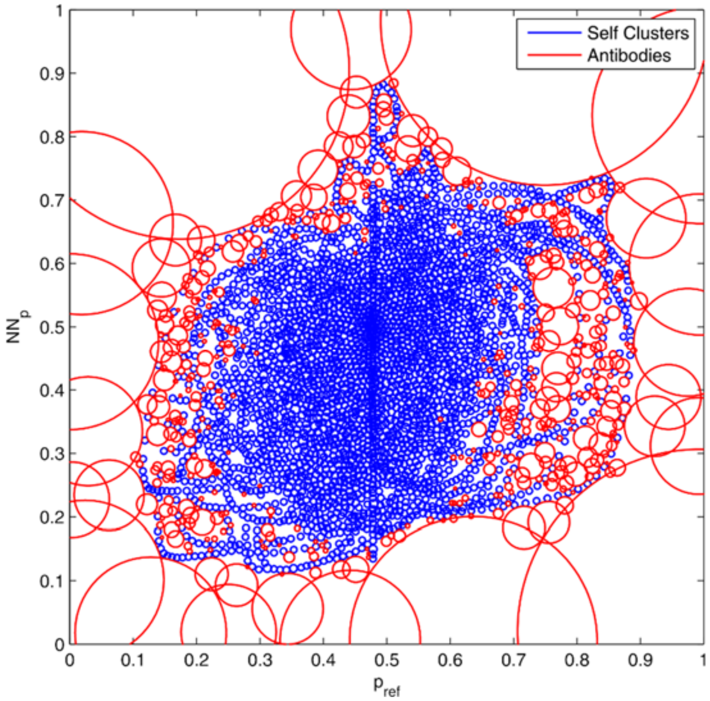


Figure 1.14. 2-D Self Clusters with AIS Antibodies Generated Using the Raw Data Set Union Method

1.5.3. Cluster Set Union Method

The mechanism to generate antibodies using the cluster set union method (CSUM) is based on a 5-phase/modules process that uses an optimized algorithm to fusion different sets of clusters generated from single sets of experimental flight or simulation data. The main components of this methodology, according to Figure 1.13, are described next.

Preliminary Data Processing. If the amount of experimental is large enough to exceed the available computer memory, within this approach, the data can be split in sub-sets and the following steps in the process can be applied to the individual smaller data sets. Pre-processing of the data includes two steps: normalization and data preparation for clustering. As a result of the normalization, the values of each measured feature are scaled to values between 0 and 1 and as is the case with the previous approach, the feature space becomes a unit hypercube. The normalization factor for each dimension is determined as the span of the flight data plus a percentage margin. Alternatively, desired maximum and minimum values can be specified in the computation of the normalization factor. Note that when multiple sets of experimental data are used for antibodies generation, the same normalization factors must be used for all data.

Clustering of Individual Data Sets. This module is similar to the previous approach; however, the clustering algorithms are applied to the individual smaller sets of experimental data. Note that parallel computation may be used to perform this phase.

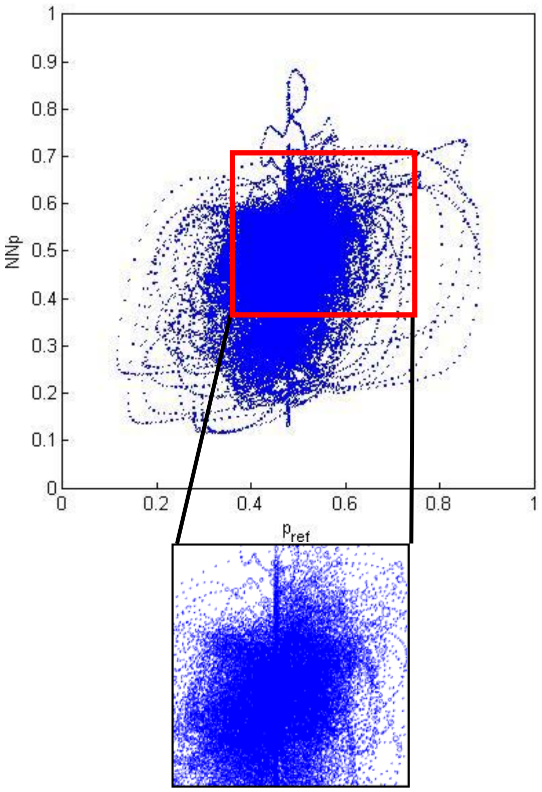
Clusters Set Union. Once several sets of clusters have been generated, a fusion process is performed that consists of set union accompanied by overlapping elimination.

Clusters Duplicate Elimination. The overlapping between clusters is estimated in a similar way as between the detectors, where a minimum overlapping threshold of a detector with respect to the others is allowed during the process. Since the radius of each cluster is known, the overlapping between a current cluster and the nearest one(s) can be determined. The distance between centers must be greater than or equal to the sum of the radii of the clusters minus the permitted overlapping threshold. This approach favors clusters with bigger radii and will preserve for the final self-representation those clusters with more efficient coverage. The approach allows the update of the database when new flight tests are available by clustering only the newly acquired data and then putting old and new clusters together and eliminating any duplication.

Figures 1.15 and 1.16 present an example of the clustering reduction using the fusion algorithm. The 2-dimensional self space in this case is defined by the non-dimensional roll rate neural network output feature NN_p and the reference roll rate p_{ref}. Figure 1.15 shows the union of different sets of clusters generated from 11 single simulator test data before the fusion process is applied. The union set has a total of 22000 clusters. Figure 1.16 shows the reduced set of 6343 clusters after the complete fusion process. The fusion algorithm is able to reduce down the number of clusters to approximately one third with an acceptable overlapping among them. The allowed overlap value has to be selected in such a way that the balance between the holes and the covered space is adequate.

Generation of Antibodies. The same enhanced negative selection algorithm for real-valued representation with variable detector radius is applied to the cluster union set generated in the previous module. The algorithm ensures no overlapping between detectors and self clusters and minimizes the un-covered areas in the non-self.

In Figure 1.17, an example of the clusters and generated antibodies is presented for a 2-dimensional self projection defined by a non-dimensional roll rate neural network output feature NN_p and the reference roll rate p_{ref} using CSUM.



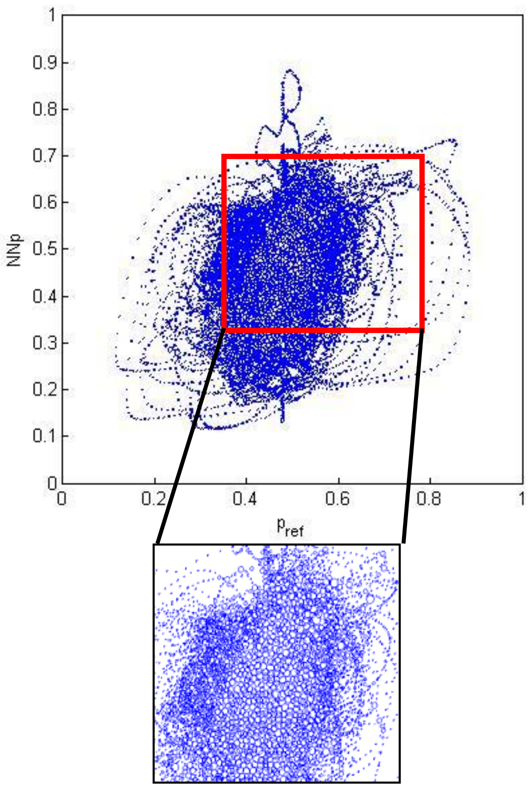


Figure 1.15. 2-D Self with 22000 Clusters

Figure 1.16. 2-D Self with 6343 Clusters

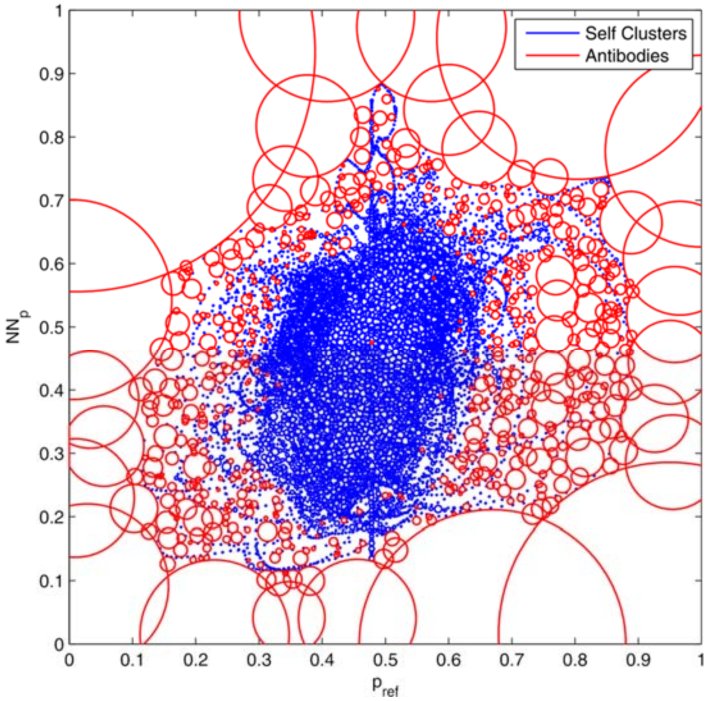


Figure 1.17. Sample 2-D Self Clusters with AIS Antibodies Generated Using the Cluster Set Union Method

An comparative analysis of the two proposed methods has been performed in terms of computational time and detection performance of the resulting sets of detectors. Numerical results and additional details are presented in journal paper #1 in Appendix A.

1.5.4. Comparison of the Proposed Methods for Antibody Generation

The computational time needed by RDSUM is consistently lower than the one needed by CSUM. The difference varies quite largely depending on the self. The fewer the number of distinct data is, the lower the computational time RDSUM needs for clustering. Typically, the raw data file reduces size significantly after the elimination of duplicate points. The clustering module is invoked only one time per self for RDSUM. The CSUM clusters the same number of flight data points 11 times (the number of flight tests used for this analysis) without any duplicate point removal. Only then, duplicate clusters are eliminated. For some of the selves generated using RDSUM, the number of unique data was around 60k (which is comparable to the number of records in each of the 11 flight files). It should be noted that the single file including all flight data used for detector generation with RDSUM has about 600,000 records. In most cases, this results in a still large data set even after duplicates are eliminated, which cannot be handled by the *k*-means clustering method on computers with less than 8GB RAM.

The detection performance in terms of percentage detection rate and false alarms of the two sets of detectors obtained with the different methods has been compared for a sub-set of relevant 2-dimensional projections or sub-selves under several types of failure. The performance is similar and the two methods can be considered equivalent from this point of view. In Figure 1.18, typical results in terms of detection rate are presented for a stabilator failure.

The RDSUM requires large computer memory, but the total computation time is lower. The CSUM can be implemented on lower memory computers; however, the overall computation time increases, unless parallel computation is used.

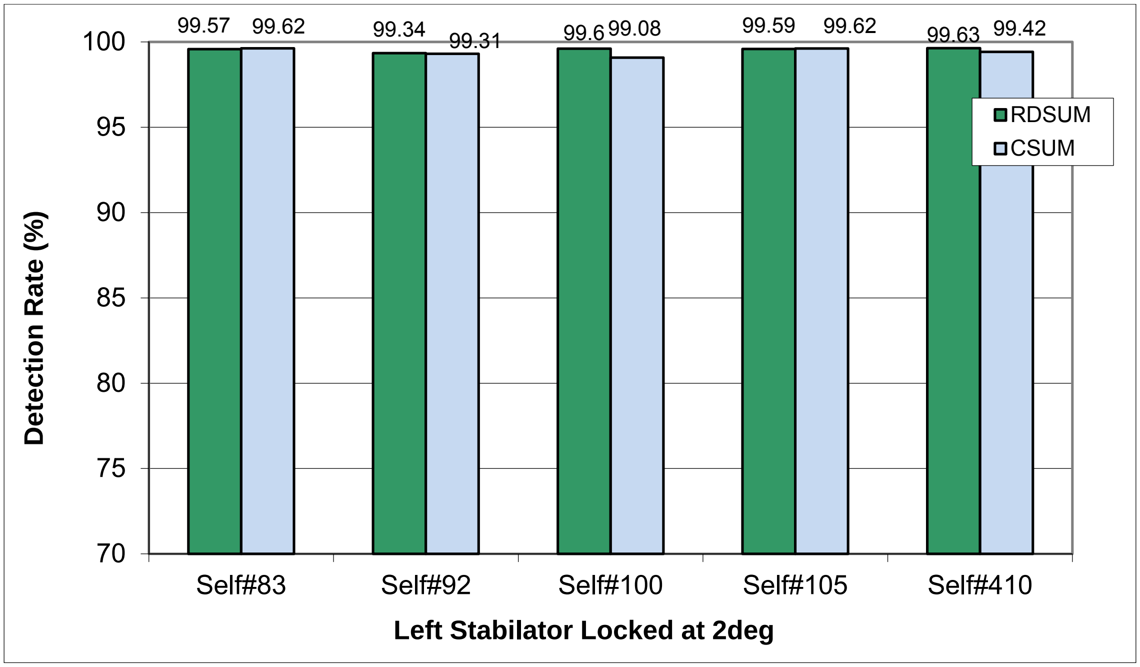


Figure 1.18. Detection Performance Comparison of Detectors Obtained with CSUM and RDSUM for a Stabilator Failure

1.6. Hierarchical Multi-Self Strategy

For a comprehensive and integrated solution to the ACDIEA problem using the AIS paradigm, the number of system features for any system of aircraft complexity ends up to be very large. Handling the resulting hyper-space as a whole produces critical issues related to computational resources and effort and to the characteristics of hyper-spaces relative to distances and thresholds.

The issues related to computational resources and effort are obvious considering that in order to achieve similar high resolution of the feature space – a critical element especially for the identification and evaluation phases – exponentially larger computational effort is required with unitary increase of the space dimensionality.

The implications of the characteristics of hyper-spaces relative to distances and thresholds are more subtle. It should be noted that when the dimensionality of the hyper-space goes to infinity the volume of the unit hyper-cube (the "*Universe*") remains equal to one, while the volume of the inscribed hyper-sphere goes to zero. This means that our intuition in establishing thresholds and assessing distances, which is built in lower dimensional spaces (actually in the 3-D physical space), becomes in-operational. This counter-intuitive effect becomes significant once the number of features goes beyond 10.

To eliminate or at least mitigate these effects, the HMS strategy was proposed[10] based on the observation that, while the entire set of features is necessary to capture the dynamic fingerprint of "all" ACs, only limited subsets may be necessary to capture the dynamic fingerprint of any individual AC. Under certain conditions (as discussed in Section 1.5) and with proper ACDIE logic, subsets of features may be used to build projections of the self that eventually yield similar ACDIE performance as when considering the entire multi-dimensional self as a whole. This concept is illustrated in the block diagram of Figure 1.19.

Simulation results[10] have shown that different low-dimensional self configurations can be selected and integrated to achieve low number of false alarms and high detection rates for a variety of subsystem abnormal conditions. In other words, the ACDIE performance achieved by a complete set of features of dimension N_s may be matched by a collection of sets i, each of dimension n_i , where $ni<<N_s$ if they are properly integrated. As a result, the issues with the multi-dimensionality of the hyperspace can be mitigated or even eliminated. This approach involves determining the capabilities of different sets of features relative to the detection of specific types of ACs. Careful analysis is needed and the selves with best detection results must be identified, organized, and integrated within a hierarchical scheme such that a high detection rate, low number of false alarms, and correct identification are achieved for all ACs. The process is performed off-line and can be customized continually until a configuration is obtained, which ensures the desirable performance of the AIS scheme. The outcome of this process consists of a reduced-number set of sub-selves (projection) of specific size and feature content.

Previous research[10] by the authors of this report has demonstrated the merits of the approach. Sets of lower dimensional selves were obtained through analysis and testing that yielded excellent ACDIE performance. However, the approach requires extensive information and/or investigation of ACFDIE capabilities of different features (equivalently, sub-selves or projections).

In this project, special focus was on developing a methodology that could avoid the need for significant prior knowledge or investigation of selected feature subsets (projections or sub-selves) capability for AC fingerprint capture. In other words, the number of projections, their dimension, and feature content is expected to be the result of an automated algorithm that does not use prior knowledge. A first target of this investigation was to consider a complete set of lower-dimensional projections of uniform size. Considering Eq. (1.26), it can be seen that the number of projections may become very large. For example, for 32 features we obtain a complete set of 2-D projections of 496 elements. This number of projections becomes 4960 if 3-D projection are considered fort he same number of features. Each projection would yield a detection outcome and all outcomes must be assessed and integrated properly to obtain a global detection outcome. This need led to the development of the dendritic cell mechanism, which will be discussed later.

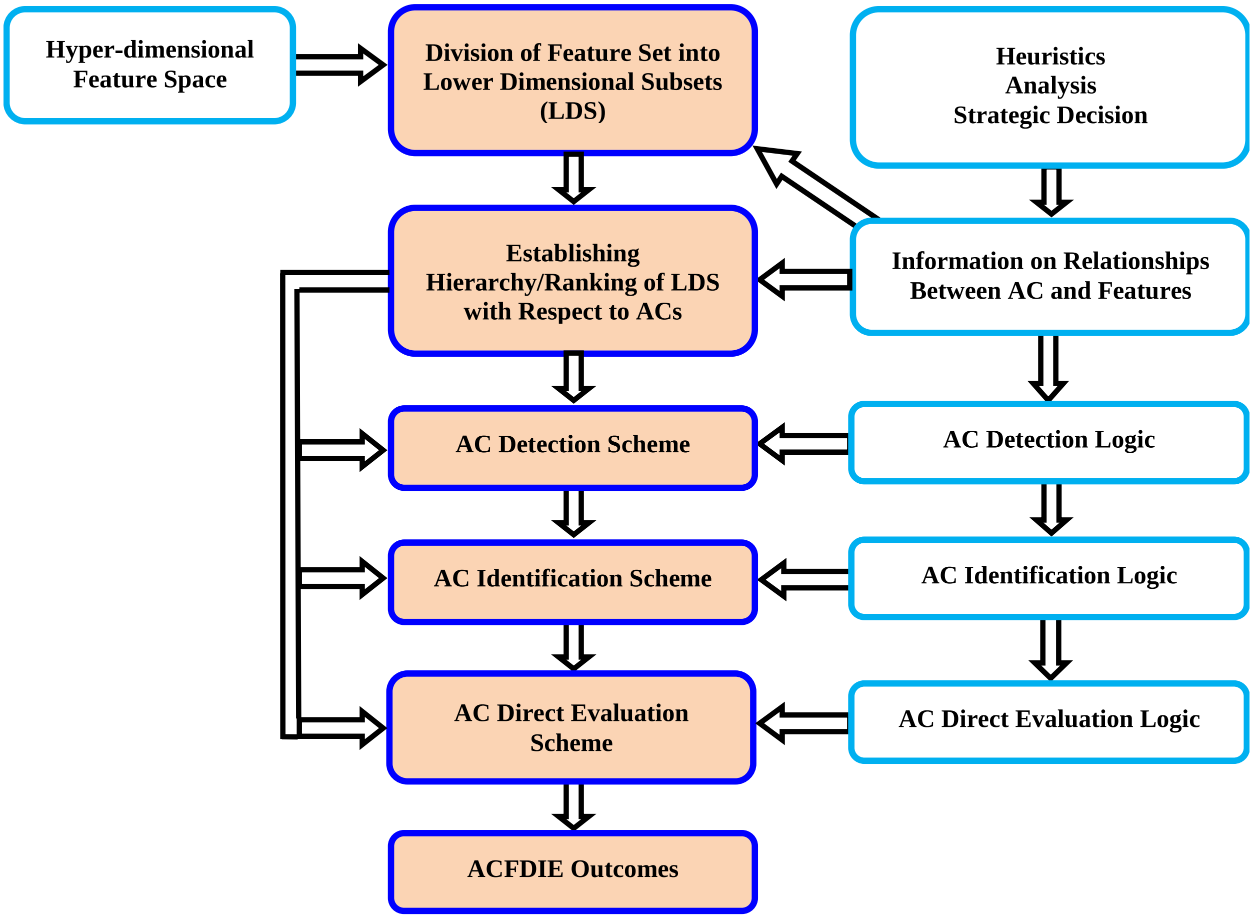

Hyper-dimensional Feature Space
Division of Feature Set into Lower Dimensional Subsets (LDS)
Heuristics Analysis Strategic Decision
Establishing Hierarchy/Ranking of LDS with Respect to ACs
Information on Relationships Between AC and Features
AC Detection Scheme
AC Detection Logic
AC Identification Scheme
AC Identification Logic
AC Direct Evaluation Scheme
AC Direct Evaluation Logic
ACFDIE Outcomes

Figure 1.19. The Hierarchical Multi-Self Strategy for ACDIE

1.7. Selection of Development and Demonstration Test Cases

1.7.1. Aircraft Subsystems

Four major aircraft subsystems with components (adding up to 19 individual subsystems) were considered for the purpose of this research effort:

- Actuators:left and right (L-R) stabilator (S), L-R aileron (A), L-R rudder (R), L-R throttle
- Sensors: pitch, roll, and yaw gyro
- Structure: L-R wing (W), L-R horizontal tail (H), L-R vertical tail (V)
- Propulsion: L-R engine

Total number of individual subsystems is:

$$Ns = 8 + 3 + 6 + 2 = 19 \tag{1.30}$$

1.7.2. Types of Subsystem Abnormal Conditions

For each subsystem, a list of targeted ACs must be established. Each list has $Nft_k, k = 1,2,\dots Ns$, failures. The following ACs have been considered within this project:

$k = 1-8$, $Nft_k = 2$ (actuators)

locked at current position (trim)
moving and locking at a non-trim position

$k = 9-11$, $Nft_k = 2$ (sensors)

sensor bias
constant sensor output

$k = 12-17$, $Nft_k = 1$ (structure)

missing structure

$k = 18-19$, $Nft_k = 1$ (propulsion)

reduced effectiveness

The total number of failures is:

$$N_F = \sum_{k=1}^{Ns} Nft_k = 2\times 8 + 2\times 3 + 1\times 6 + 1\times 2 = 30 \tag{1.31}$$

1.7.3. Flight Envelope Analysis Parameters

In Table 1.2 below, example lists of DIV, EDIV, and envelope relevant variables are presented. The resultant set of all features considered in Table 1.2 for evaluation purposes is:

$$\Im_E = \{H\ M\ V\ p\ q\ r\ a_x\ a_y\ a_z\ \dot{p}\ \dot{q}\ \dot{r}\ \alpha\ \beta\ \theta\ \varphi\} = \bigcup_i v_{Ei} \quad i = 1,2,\dots,N_F \tag{1.32}$$

The total number of features considered for evaluation purposes in Table 1.2 (i.e. flight envelope reduction assessment) is $N_E = 16$. It should be noted that subsets of these variables have been considered within the project for methodology development and demonstration.

Table 1.2. Parameters for AC Indirect Evaluation Analysis

k_F	k	Type of Failure	DIV v_δ	EDIV v_ε	Envelope var. v_E	Notes
1	1	L S locked	L S defl	longitudinal stick	a_x, a_z, H, M, q, p, $\dot{q}$, α, θ	
2	1	L S mv/lkd	L S defl	longitudinal stick	a_x, a_z, H, M, q, p, $\dot{q}$, α, θ	
3	2	R S locked	R S defl	longitudinal stick	a_x, a_z, H, M, q, p, $\dot{q}$, α, θ	

4	2	R S mv/lkd	R S defl	longitudinal stick	a_x, a_z, $H, M, q, p,$ $\dot{q}$, α, θ	
5	3	L A locked	L A defl	lateral stick	$a_y, p, r, \dot{p}$, $\dot{r}$, β, φ	
6	3	L A mv/lkd	L A defl	lateral stick	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
7	4	R A locked	R A defl	lateral stick	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
8	4	R A mv/lkd	R A defl	lateral stick	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
9	5	L R locked	L R defl	pedals	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
10	5	L R mv/lkd	L R defl	pedals	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
11	6	R R locked	R R defl	pedals	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
12	6	R R mv/lkd	R R defl	pedals	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
13	7	L T locked	L T defl	pilot throttle	a_x, $a_z, r, H, M, V,$ α, θ	
14	7	L T mv/lkd	L T defl	pilot throttle	a_x, $a_z, r, H, M, V,$ α, θ	
15	8	R T locked	R T defl	pilot throttle	a_x, $a_z, r, H, M, V,$ α, θ	
16	8	R T mv/lkd	R T defl	pilot throttle	a_x, $a_z, r, H, M, V,$ α, θ	
17	9	p bias	p meas	lateral stick	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
18	9	p const	p meas	lateral stick	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
19	10	q bias	q meas	longitudinal stick	a_x, $a_z, H, M, q,$ $\dot{q}$, α, θ	
20	10	q const	q meas	longitudinal stick	a_x, $a_z, H, M, q,$ $\dot{q}$, α, θ	
21	11	r bias	r meas	pedals	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
22	11	r const	r meas	pedals	$a_y, p, r,$ $\dot{p}$, $\dot{r}$, β, φ	
23	12	L W damage	L W lift L W pitch moment	longitudinal stick, lateral stick, pedals, pilot throttle	a_x, a_y, $a_z, H, M, p, q, r,$ $\dot{p}$,$\dot{q}$,$\dot{r}$,α, β, θ, φ	
24	13	R W damage	R W lift R W pitch moment	longitudinal stick, lateral stick, pedals, pilot throttle	a_x, a_y, $a_z, H, M, p, q, r, \dot{p}$,$\dot{q}$,$\dot{r}$, α, β, θ, φ	
25	14	L H damage	L H lift	longitudinal stick, lateral stick,	a_x, a_y, $a_z, H, M, p, q, r, \dot{p}$,$\dot{q}$,$\dot{r}$, α, β, θ, φ	
26	15	R H damage	R H lift	longitudinal stick, lateral stick,	a_x, a_y, $a_z, H, M, p, q, r, \dot{p}$,$\dot{q}$,$\dot{r}$, α, β, θ, φ	
27	16	L V damage	L V lift	lateral stick, pedals, pilot throttle	a_x, a_y, $a_z, H, M, p, r,$ $\dot{p}$, $\dot{r}$, β	
28	17	R V damage	R V lift	lateral stick, pedals, pilot throttle	a_x, a_y, $a_z, H, M,$ $p, r,$ $\dot{p}$, $\dot{r}$, β	
29	18	L E red eff	L E thrst	pilot throttle	a_x, $a_z, r, H, M,$ α, θ	
30	19	R E red eff	R E thrst	pilot throttle	a_x, $a_z, r, H, M,$ α, θ	

1.7.4. Features Selection

Based on the subsystems considered, the nature of ACs, and the extent of the flight envelope prediction attempted, a set of variable were selected to represent features for self/non-self definition, including states, states derivatives, inputs, control system variables, and artificial neural network (ANN) estimations. These variables are listed in Table 1.3.

Table 1.3. Features for Self/Non-self Definition

H	=	altitude	d_T	=	pilot throttle
V	=	aircraft ground speed	p_{ref}	=	roll rate command
M	=	Mach number	q_{ref}	=	pitch rate command
a_x	=	longitudinal acceleration	r_{ref}	=	yaw rate command
a_y	=	lateral acceleration	NN_{outp}	=	roll acceleration error
a_z	=	vertical acceleration	NN_{outq}	=	pitch acceleration error
α	=	angle of attack	NN_{outr}	=	yaw acceleration error
β	=	sideslip angle	$MQEE$	=	main quadratic estimation error
ϕ	=	roll attitude angle	$OQEE$	=	output quadratic estimation error
θ	=	pitch attitude angle	$DQEE_p$	=	decentralized quadratic roll rate estimation error
ψ	=	yaw attitude angle	$DQEE_q$	=	decentralized quadratic pitch rate estimation error
p	=	roll rate	$DQEE_r$	=	decentralized quadratic yaw rate estimation error
q	=	pitch rate			
r	=	yaw rate			
$\dot{p}$	=	roll acceleration			
$\dot{q}$	=	pitch acceleration			
$\dot{r}$	=	yaw acceleration			
d_e	=	longitudinal stick displacement			
d_a	=	lateral stick displacement			
d_r	=	pedal displacement			

Note that the parameters based on ANN estimates of angular rates are computed as follows[19]. The main quadratic estimation error (MQEE) is defined as:

$$MQEE(k)=\frac{1}{2}\left[\left(p(k)-\hat{p}_{MNN}(k)\right)^2+\left(q(k)-\hat{q}_{MNN}(k)\right)^2+\left(r(k)-\hat{r}_{MNN}(k)\right)^2\right] \tag{1.33}$$

where *p(k)*, *q(k),* and *r(k)* are measurements of angular rates at sample k and $\hat{p}_{MNN}(k)$, $\hat{q}_{MNN}(k)$, and $\hat{r}_{MNN}(k)$ are neural estimates of the angular rates based on sensor measurements including the respective gyro, over a specified time window. The output quadratic estimation error (OQEE) is defined as:

$$OQEE(k)=\frac{1}{2}\left[\left(\hat{p}_{DNN}(k)-\hat{p}_{MNN}(k)\right)^2+\left(\hat{q}_{DNN}(k)-\hat{q}_{MNN}(k)\right)^2+\left(\hat{r}_{DNN}(k)-\hat{r}_{MNN}(k)\right)^2\right] \tag{1.34}$$

where $\hat{p}_{DNN}(k)$, $\hat{q}_{DNN}(k)$, and $\hat{r}_{DNN}(k)$ are neural estimates of the angular rates based on sensor measurements that do NOT include the respective gyro, over a specified time window. Finally, the decentralized quadratic angular rate estimation error (DQEE) is defined as:

$$DQEE_x(k) = \frac{1}{2}\left(\hat{x}_{DNN}(k) - x(k)\right)^2, \quad x = p,q,r \tag{1.35}$$

1.8. WVU Simulation Environment

For the purpose of this project, experimental data were collected from the WVU 6-DOF flight simulator system[23] shown in Figure 1.20. The simulator relies on a motion platform driven by electrical induction motors to provide adequate 6-DOF translational and rotational motion cues. Laminar research X-Plane flight simulation software is used to provide external visual cues through a six-monitor system. The motion platform is interfaced with an external computer on which an aircraft model can run within the Matlab/Simulink environment driving the entire simulator system.

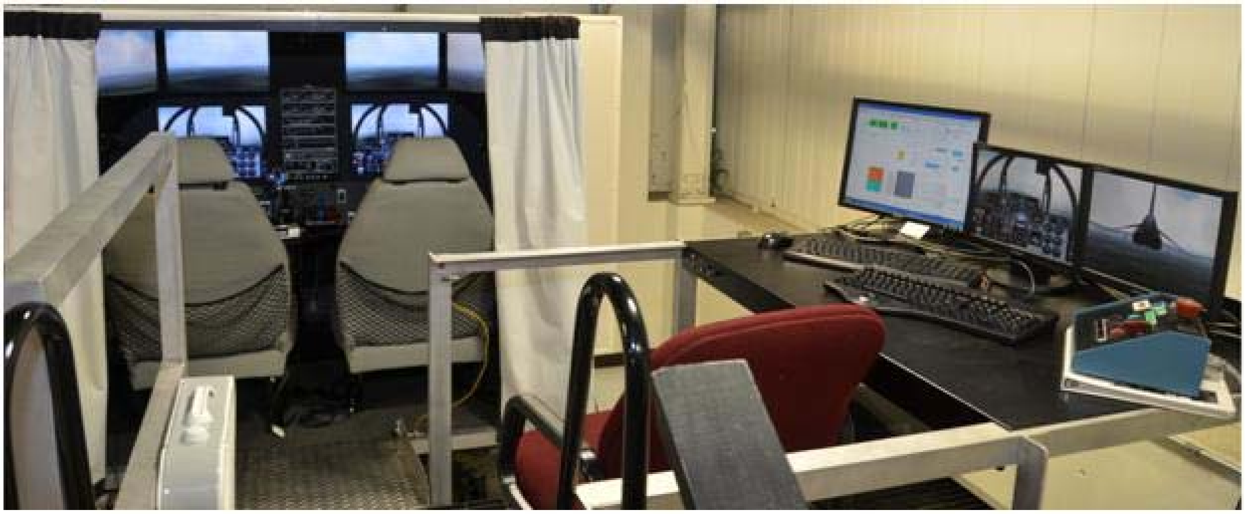


Figure 1.20. The WVU 6-DOF Motion-based Flight Simulator

Figure 1.21 shows the top level Simulink diagram of the model interfaced with the WVU Flight Simulator. The model includes the nonlinear dynamics of a supersonic fighter including models of failures/damages of actuator, sensors, structural components, and engines. The three large blocks at the bottom of the figure include the computation of specific variables to be provided to the flight simulator to drive the generation of visual and aural cues as well as the motion of the simulator platform.

To define the AIS self/non-self as completely and accurately as possible, adequate coverage of the feature space must be achieved. Different flight scenarios are considered over a wide range of the flight envelope, which is first defined based on the nine reference points shown in Figure 1.22 for Mach numbers between 0.6 and 0.9 and altitudes between 9,000 ft and 31,000 ft. All flight tests start at steady state flight condition at point 1 and continue to cover the nine points as described by the arrows. For example, one flight test starts at point 1, the aircraft is accelerated at constant altitude to point 4, descended at constant speed to point 5, and then returned to points 4 and 1. A total of eight such tests are necessary to cover the testing flight envelope. The data collected from flight tests under normal conditions with these points were used to build the self/non-self of the aircraft. Additional intermediate points (A, B, C, and D in Figure 1.22) were used to provide validation data. The set of flight scenarios, lasting between 10 and 20 minutes each, are designed to include steady-state flight conditions, transitions between steady-state conditions, and mild to moderate maneuvers. These flight scenarios are simulated under normal flight conditions. They are repeated under various failure scenarios for both design/development and validation purposes. Only one failure at a time is considered to capture/isolate the dynamic fingerprint of each type of failure and generate antibodies appropriately. The data acquisition rate from the simulator is 50 Hz.

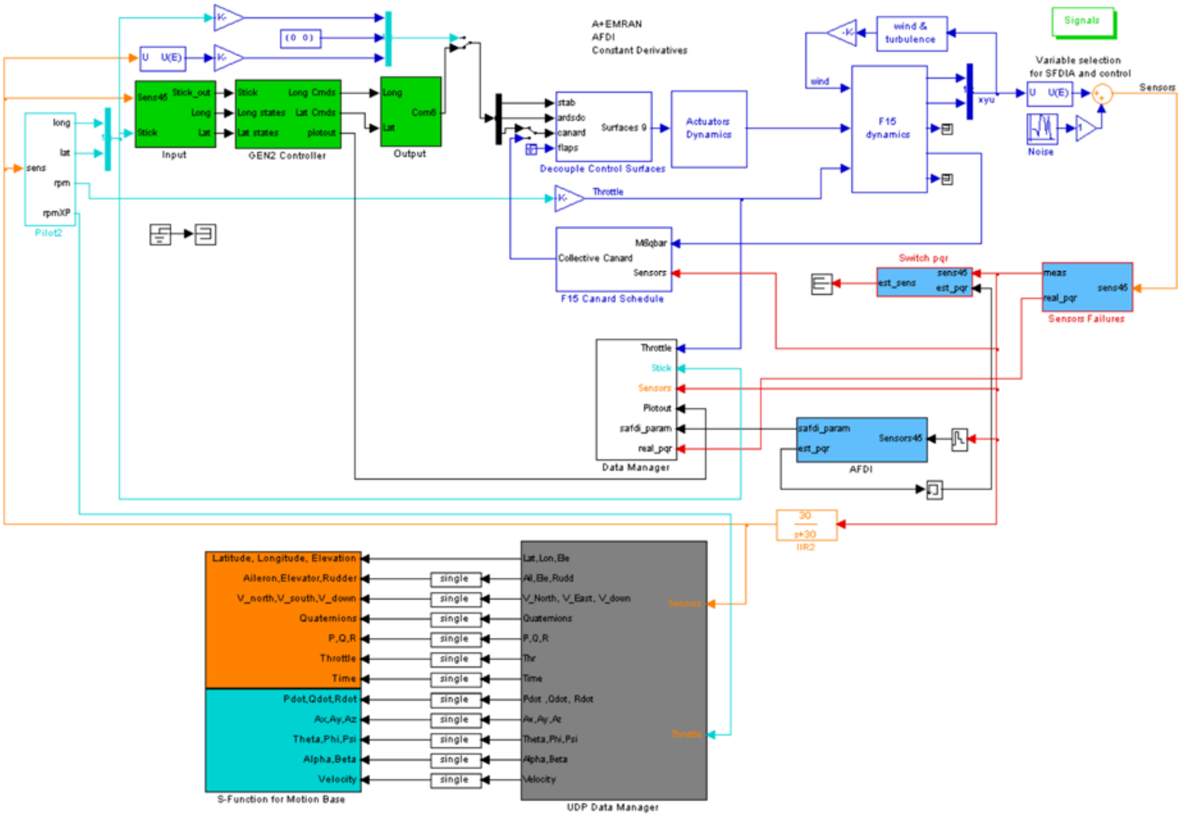


Figure 1.21. Simulink Supersonic Fighter Aircraft Model

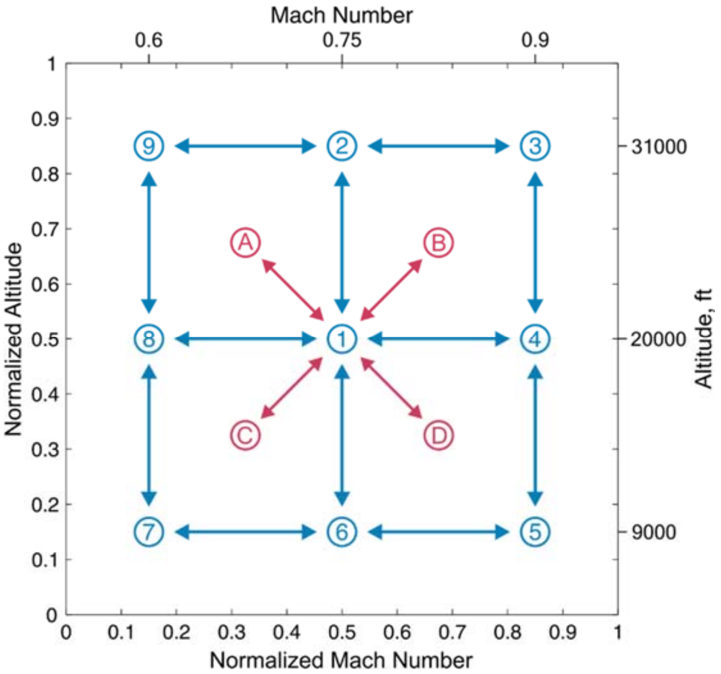
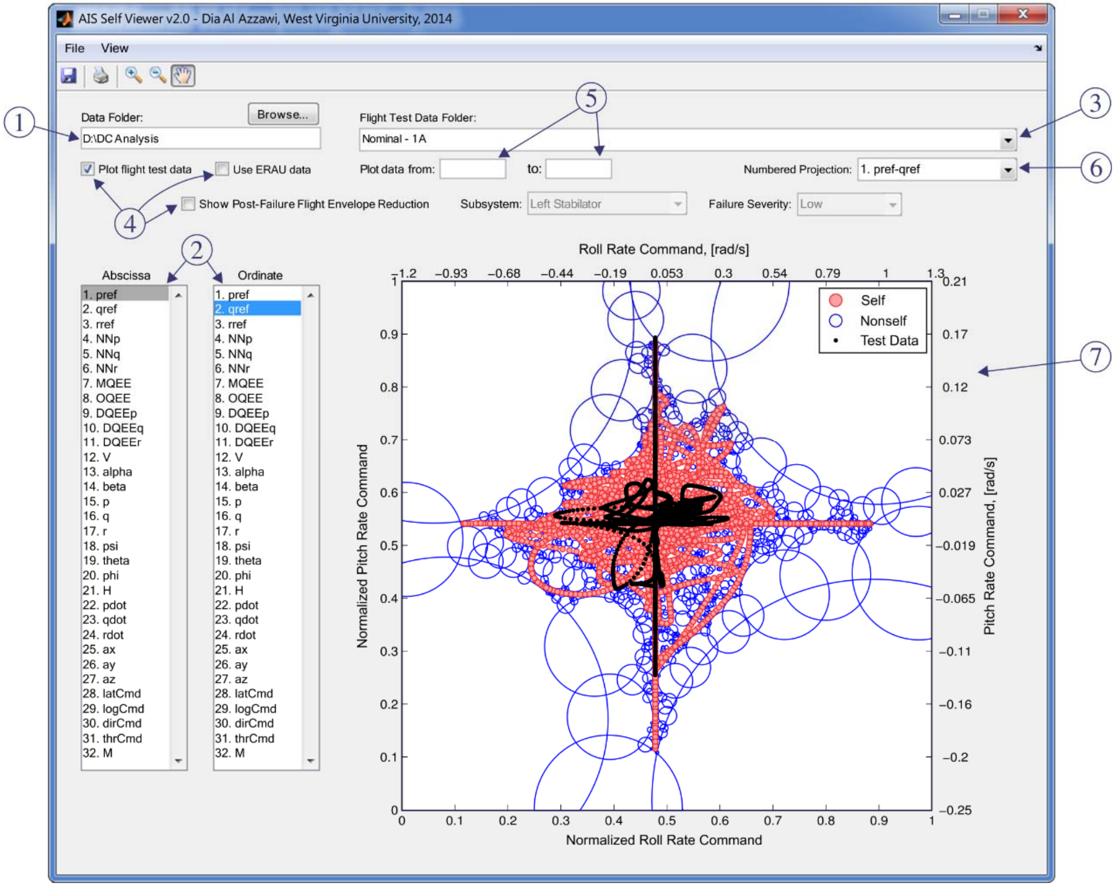


Figure 1.22. Testing Flight Envelope

1.9. WVU Self/Non-self Analysis Tool

Visualizing the 2-dimensional sub-selves is very helpful in analyzing how well the self clusters and detectors are generated using one of the methods discussed in the previous sections. A simple interactive tool (see Figure 1.23) was developed to analyze all the generated 2-D subselves and project a given flight test data on these sub-selves in order to validate them before adopting them as a basis to the ACDIE schemes.



AIS Self Viewer v2.0 - Dia Al Azzawi, West Virginia University, 2014

Data Folder: D:\DC Analysis

Flight Test Data Folder: Nominal - 1A

☑ Plot flight test data

□ Use ERAU data

Plot data from: to:

Numbered Projection: 1. pref-qref

□ Show Post-Failure Flight Envelope Reduction

Subsystem: Left Stabilator

Failure Severity: Low

Abscissa: 1. pref

Ordinate: 2. qref

Figure 1.23. The WVU Interactive Tool for Self/Non-self Analysis

The visualization tool is user friendly and has the following main components referenced in Figure 1.23:

1. “Data folder” textbox to specify the fully-qualified path of the main folder containing all the flight tests and the self/nonself data generated.

2. Two list boxes of features containing the all features considered for self/non-self definition. Any feature selected from the left list box represents the abscissa of the 2-D plot and any feature selected from the right list box represents the ordinate.

3. “Flight test” dropdown menu containing all available flight tests.

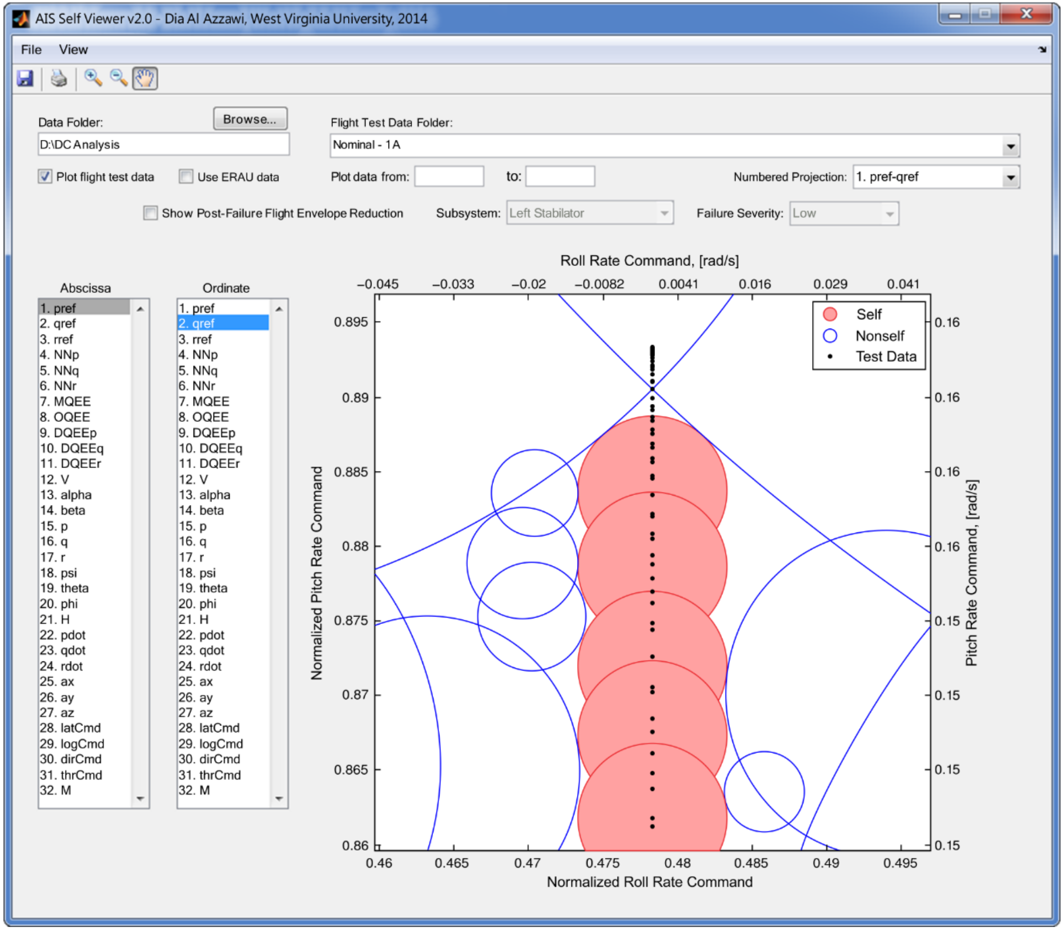
4. Three check boxes providing options for plotting flight test data points on the figure, plotting self/non-self generated with the CSUM method (use ERAU data checkbox), and plotting post- failure flight envelope ranges.

5. Two optional textboxes to specify the starting and ending time steps used in plotting flight test data points. If these textboxes are empty, all data points in the selected flight test are to be plotted.

6. An optional drop down menu to quickly plot the selected 2-D projection.

7. An X-Y plot area onto which the self clusters, non-self shapes, and (optional) flight test data are plotted.

The zooming feature of the analysis tool facilitates the validation of the self/non-self generation process such as how well the detectors cover the non-self space, whether the method has generated self clusters within the specified margins or not, whether detectors overlapping is reasonable or not, whether there are issues with the self generation method (or its underlying processes) or with the flight data, etc. Figure 1.23 illustrates the zoom feature.



AIS Self Viewer v2.0 - Dia Al Azzawi, West Virginia University, 2014

Data Folder: D:\DC Analysis

Flight Test Data Folder: Nominal - 1A

☑ Plot flight test data

☐ Use ERAU data

Plot data from: to:

Numbered Projection: 1. pref-qref

☐ Show Post-Failure Flight Envelope Reduction

Subsystem: Left Stabilator

Failure Severity: Low

Abscissa: 1. pref

Ordinate: 2. qref

Figure 1.24. The Zooming Action of the Interactive Tool for Self/Non-self Analysis

Chapter 2

Immunity-based Detection of Aircraft Subsystem Abnormal Conditions

2.1. Detection through Direct Self/Non-self Discrimination

2.1.1. General Aspects

Let the self and non-self be defined as in Eqs. (1.27) and (1.28). Let the current sample *t* of the measured feature point be:

$$\Phi_t = \begin{bmatrix} \varphi_{1t} & \varphi_{2t} & \cdots & \varphi_{Nt} \end{bmatrix} \tag{2.1}$$

The AC detection can be performed through mechanisms that can be of two types. The *NS-type* of detection mechanism involves comparing the measured feature point against the detectors, while the *PS-type* of mechanism involves comparing the measured feature point against the self clusters. If one *N*-dimensional self/non-self is used, then the negative selection-type detection can be modeled as:

$$Det_t = \begin{cases} 0 & if \ \left| \vec{r}^{\,O\Phi_t} - \vec{r}^{\,OD_j} \right| > Rd_j \quad for\ all\ \ j = 1,2,...,Nd \\ 1 & otherwise \end{cases} \tag{2.2}$$

while the PS-type of mechanism can be expressed as:

$$Det_t = \begin{cases} 1 & if \ \left| \vec{r}^{\,O\Phi_t} - \vec{r}^{\,OC_i} \right| > Rc_i \quad for\ all\ i = 1,2,...,Nc \\ 0 & otherwise \end{cases} \tag{2.3}$$

If N_{ss} sub-selves are used, then Eq. (2.2) or (2.3) is applied for each sub-self to obtain sets of values Det_i, $i = 1,2,...,N_{ss}$. Then:

$$Det_t = \max_{i=1,N_{ss}} (\, Det_{ti} \,) \tag{2.4}$$

Note that a more complicated function than *max(*)* can be considered such as one describing a dendritic cell mechanism.

To obtain a “normal condition” outcome with the negative selection approach, all detectors must be checked at each sample *t*. With the positive selection approach, the testing of the current feature point is stopped as soon as a matching self cluster is found. Therefore, at normal conditions, the positive selection-based detection is less computationally intensive. On the other hand, an AC will be declared as soon as a matching detector has been found with the negative selection approach instead of testing the entire set of self clusters as is required by the positive selection approach. Therefore, an AC may be detected faster with the negative selection approach. Depending on the number of self clusters and detectors in conjunction with the system update rate, these differences may be relevant or not. However, note that building and structuring the detectors is critical for the failure identification process if the structured non-self approach is used.

At each moment, a detection outcome can be simply obtained as:

$$Det = Det_t \tag{2.5}$$

However, in order to prevent excessive false alarms, instead of using Eq. (2.5) directly, an additional detection logic must be used to build the detection phase outcome as a function of current and past outcomes and/or a composition of sub-selves-based detection outcomes.

Within the HMS strategy, sets of 2-dimensional and 3-dimensional projections were investigated for the detection through direct self/non-self discrimination.

2.1.2. Generation of Detectors

The process of generating detectors, or antibodies, is an exhaustive and lengthy procedure that requires adequate computational data processing capabilities and adequate numerical representations of the self/non-self. However, it should be noted that the process is performed off-line and does not affect the real time operation of the on-board detection scheme.

CSUM was used to process extensive experimental data necessary to properly define the nominal "hyper-space" representation. For each combination of features corresponding to a particular projection, segments of data are processed separately to produce sets of detectors that cover the non-self hyperspace. The CSUM is a 5 phase process (see Figure 2.1) that uses an optimized algorithm to fuse different sets of clusters generated from single sets of flight or simulation data. This methodology is particularly convenient when the computer characteristics used for pre-processing are not powerful enough for an efficient application of the algorithms.

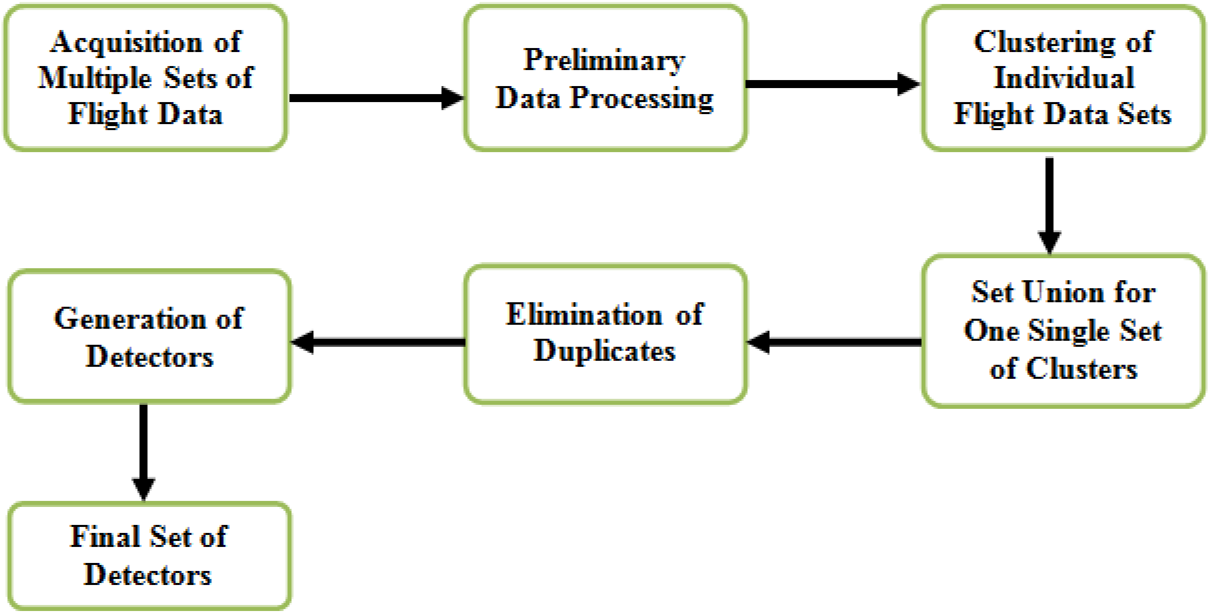

Acquisition of Multiple Sets of Flight Data
Preliminary Data Processing
Clustering of Individual Flight Data Sets
Set Union for One Single Set of Clusters
Elimination of Duplicates
Generation of Detectors
Final Set of Detectors

Figure 2.1. Cluster Set Union Based Logic

2.1.3. Analysis of Detection Performance of Individual Selves

A total amount of 496 2-D projections of sub-selves were generated. Out of the 4960 possibilities for 3-D sub-selves, only 45 were generated. The performance of each projection or sub-self was analyzed and compared based on the detection rate (DR) and false alarm (FA) rate. The detection function output is a binary signal that determines if a sub-system failure has been detected (output=1) or if the aircraft is flying at nominal conditions (output=0). The binary output can be categorized in four types as follows:

- True Positive (TP): a failure is detected and declared as failure;
- True Negative (TN): nominal conditions are declared as nominal;
- False Positive (FP): nominal conditions are declared as failures;
- False Negative (FN): failure condition is not detected.

These categories can be used for a quantitative evaluation of the DR and FA. The following equations can be used to calculate the DR of a specified projection.

$$DR = \frac{TP}{TP + FN} \times 100 \tag{2.2}$$

Similarly, the FA rate can be calculates as follows:

$$FA = \frac{FP}{TN + FP} \times 100 \tag{2.3}$$

The detection rates and false alarms were calculated for 26 different ACs varying in affected subsystem, type, and severity. Table 2.1 presents a sample set of projections among the 496 generated sub-selves that were selected as an example of the relevant combination of features utilized in this effort.

Table 2.1. Features of Selected Projections

Self	Features	Self	Features	Self	Features	Self	Features
Self#3	p_{ref}, NN_p	**Self#94**	NN_p, r	**Self#85**	$NN_p, OQEE$	**Self#105**	NN_p, d_T
Self#30	q_{ref}, NN_p	**Self#95**	NN_p, ψ	**Self#86**	$NN_p, DQEE_p$	**Self#106**	NN_p, M
Self#42	q_{ref}, q	**Self#96**	NN_p, θ	**Self#87**	$NN_p, DQEE_q$	**Self#224**	$DQEE_q, \psi$
Self#53	q_{ref}, d_r	**Self#97**	NN_p, φ	**Self#89**	NN_p, V	**Self#233**	$DQEE_q, d_r$
Self#82	NN_p, NN_q	**Self#98**	NN_p, H	**Self#90**	NN_p, α	**Self#410**	$NN_p, \dot{p}$
Self#83	NN_p, NN_r	**Self#99**	NN_p, a_x	**Self#92**	N_p, p	**Self#441**	$NN_p, \dot{q}$
Self#84	$NN_p, MQEE$	**Self#100**	NN_p, a_y	**Self#93**	NN_p, q	**Self#471**	$NN_p, \dot{r}$

Table 2.2 presents the performance analysis of those selves under four different failures: left aileron locked at 2.5 deg., left stabilator locked at 2 deg., 6% left wing loss, and 5 deg bias in the pitch rate sensor output. A sample set of 20 projections is presented. The detection rate as well as the false alarm rate is shown. These results demonstrate that certain selves favor the detection of certain types of failures while showing poor detection rates for others. This fact has been used in previous research efforts to support the formulation of the HMS startegy. The presented mechanism uses low order projections to build sub-selves using a specific hierarchy of features relevance with respect to each type of failure. Figures 2.2 through 2.5 present the individual detection rates obtained for different projections and four different failures.

Figure 2.6 presents the detection rate that a single projection can attain when tested against four different failures. It should be noted how a single projection is able to detect three different failures successfully while showing poor performance for a fourth one. This is due to the fact that the dynamics of such failure do not have great impact on the features corresponding to Self #87.

Table 2.2. Detection Performance of Selected Individual Selves

Self	**L Aileron Locked at 2 deg**		**L Stabilator Locked at 2 deg**		**6% Loss of Left Wing**		**Pitch Rate Sensor Bias (5 deg/sec)**	
	DR	**FA**	**DR**	**FA**	**DR**	**FA**	**DR**	**FA**
S3	82	1	99.3	1.85	99	1.1	2.27	1
S30	83.5	0.0	99.2	0.0	99.5	0.0	4.16	0.0
S42	0.0	0.0	8.22	0.0	0.1	0.0	99.9	0.0
S53	22.9	0.0	24.8	0.0	25.1	0.0	86.7	0.0
S82	92.3	0.0	99.7	0.0	99.2	0.0	4.63	0.0
S83	88.1	0.0	99.6	0.0	99	0.0	0.22	0.0
S84	86.2	0.0	99.5	0.0	99	0.0	0.05	0.0
S85	88.7	0.0	99.4	0.0	98.9	0.0	26.5	0.0
S86	84.2	0.0	99.4	0.0	98.9	0.0	24.2	0.0
S87	80.4	0.0	99.4	0.0	99.1	0.0	46.8	0.0
S89	90	0.0	99.6	0.0	99.2	0.0	1.46	0.0
S90	91.1	0.0	99.5	0.0	99.5	0.0	0.68	0.0
S92	80.8	0.0	99.3	0.0	99	0.0	0.09	0.0
S93	82.9	1.25	99.4	1.55	99.4	0.0	1.3	0.0
S94	85.8	2.25	99.5	2.25	99	2.4	0.51	2.25
S95	90.9	0.0	99.6	0.0	99.1	0.0	0.87	0.0
S96	89.5	0.0	99.5	0.0	99.3	0.0	3.18	0.0
S97	86.3	0.0	99.5	0.0	99	0.0	0.08	0.0
S98	94.6	0.0	99.6	0.0	99.2	0.0	11.2	0.0
S99	86	0.0	99.4	0.0	99.2	0.0	2.98	0.0
S100	86.9	0.0	99	0.0	97.7	0.0	0.16	0.0
S105	91.1	0.0	99.6	0.0	99	0.0	12.2	0.0
S106	93.1	0.0	99.5	0.0	99.1	0.0	6.76	0.0
S224	0.36	0.0	1.85	0.0	0.9	0.0	63.7	0.0
S233	5.51	0.0	8.04	0.0	12.4	0.0	87.8	0.0
S410	81.4	0.0	99.4	0.0	99.1	0.0	0.06	0.0
S441	80.9	0.0	99.4	0.0	99.1	0.0	0.64	0.0
S471	80.5	0.0	99.3	0.0	98.9	0.0	0.05	0.0

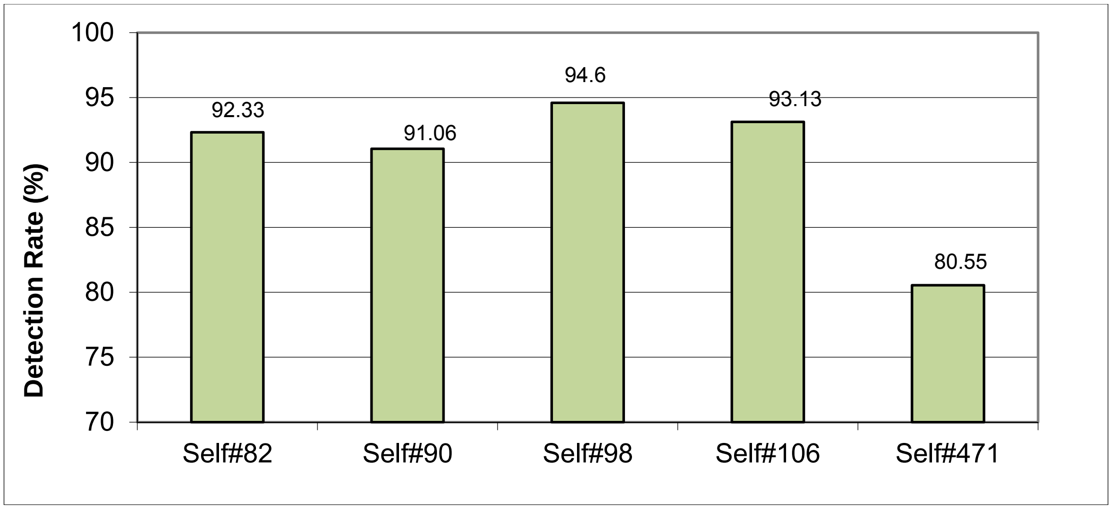


Figure 2.2. Detection Rate of Left Aileron Locked at 2.5 deg Failure

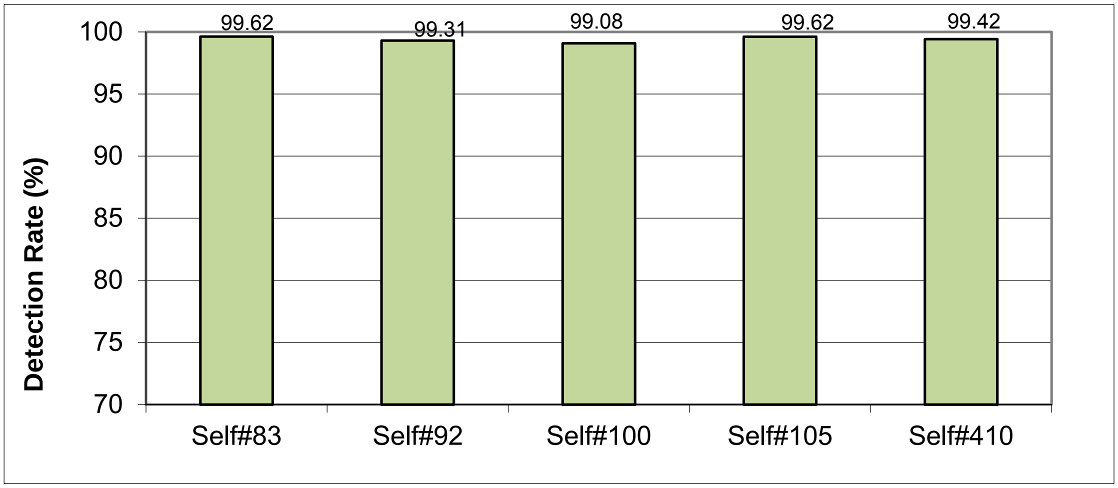


Figure 2.3. Detection Rate of Left Stabilator Locked at 2.0 deg Failure

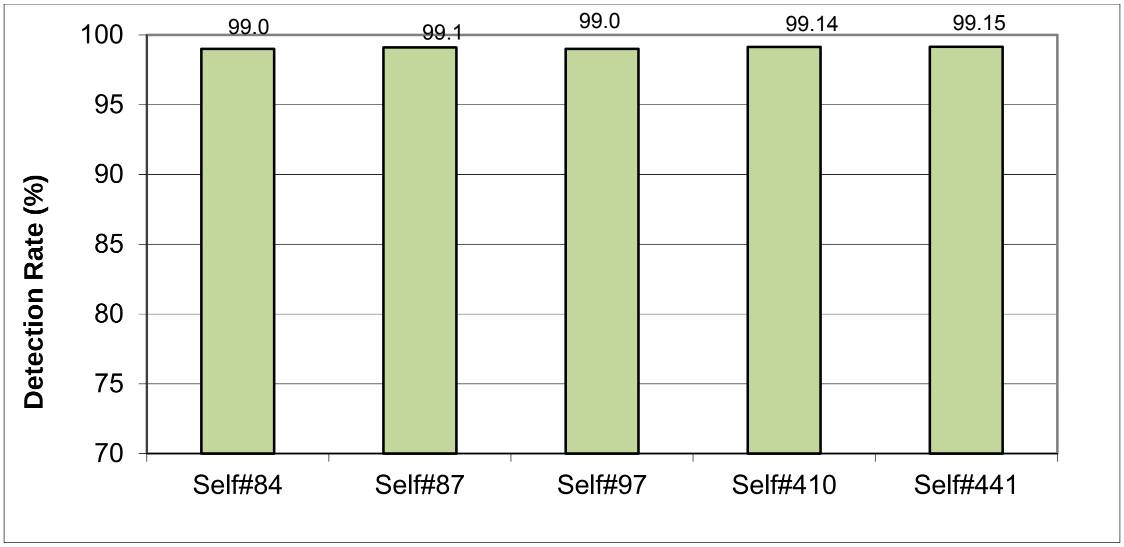


Figure 2.4. Detection Rate of 6% Left Wing Loss Failure

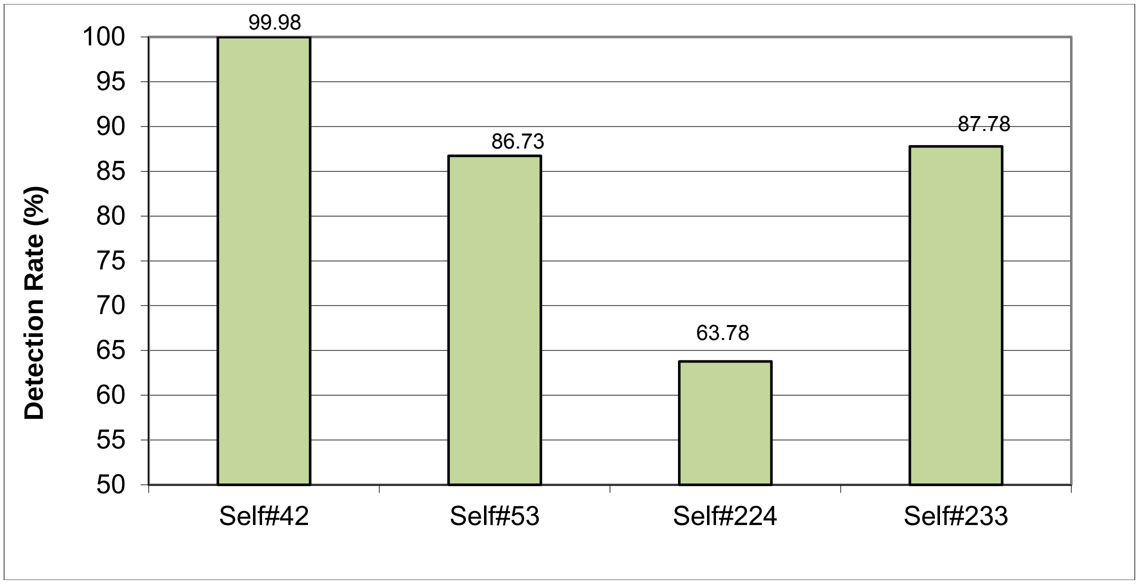


Figure 2.5. Detection Rate of Pitch Sensor Bias of 5 deg/sec Failure

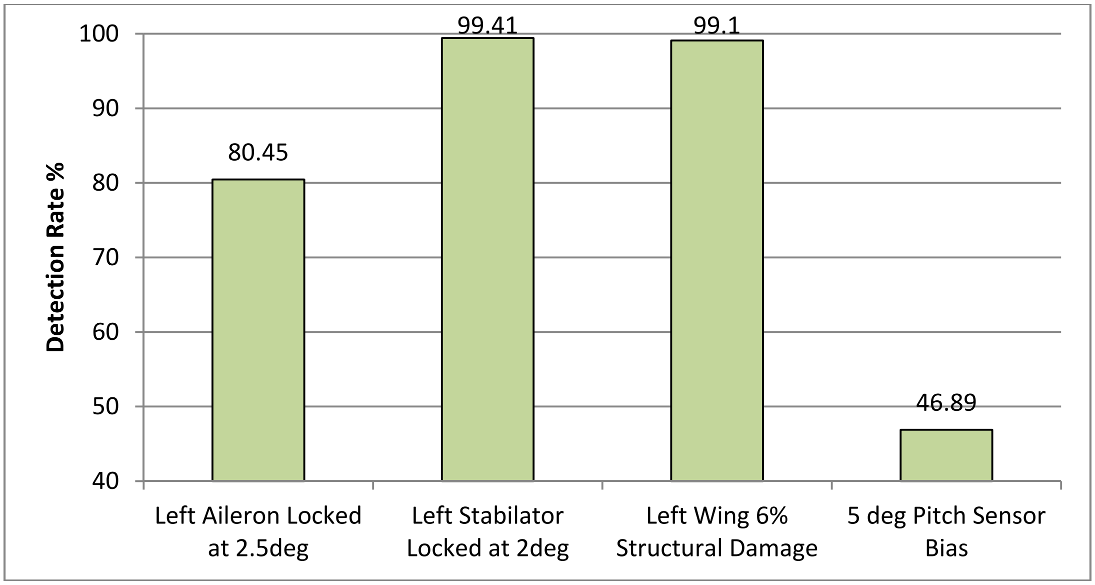


Figure 2.6. Detection Rate of Self 87 for 4 Different Failures

2.1.4. Detection Logic for Direct Self/Non-self Discrimination Scheme

Based on the sensitivity analysis that establishes the capability of each projection to capture the dynamic fingerprint of the ACs considered, a sub-set of projections of size *n* is selected for detection. A weight is associated to each projection depending on their detection potential. During the detection phase, sets of current values of the features measured in flight at a certain sampling rate are compared against the detectors that have been generated for each of the *n* 2D-projections selected for detection. A detection parameter ζ is calculated, which represents the number of consecutive points over a time window of size ω that trigger detectors, summed over all sub-selves and affected by the weights. If ζ is within a certain range (less than a threshold *Thr₁*), a failure warning is issued, but if ζ exceeds the upper bound of the range (larger than a threshold Thr2), a failure is declared and the identification phase starts. The flow chart of the SNSA detection logic is presented in Figure 2.7.

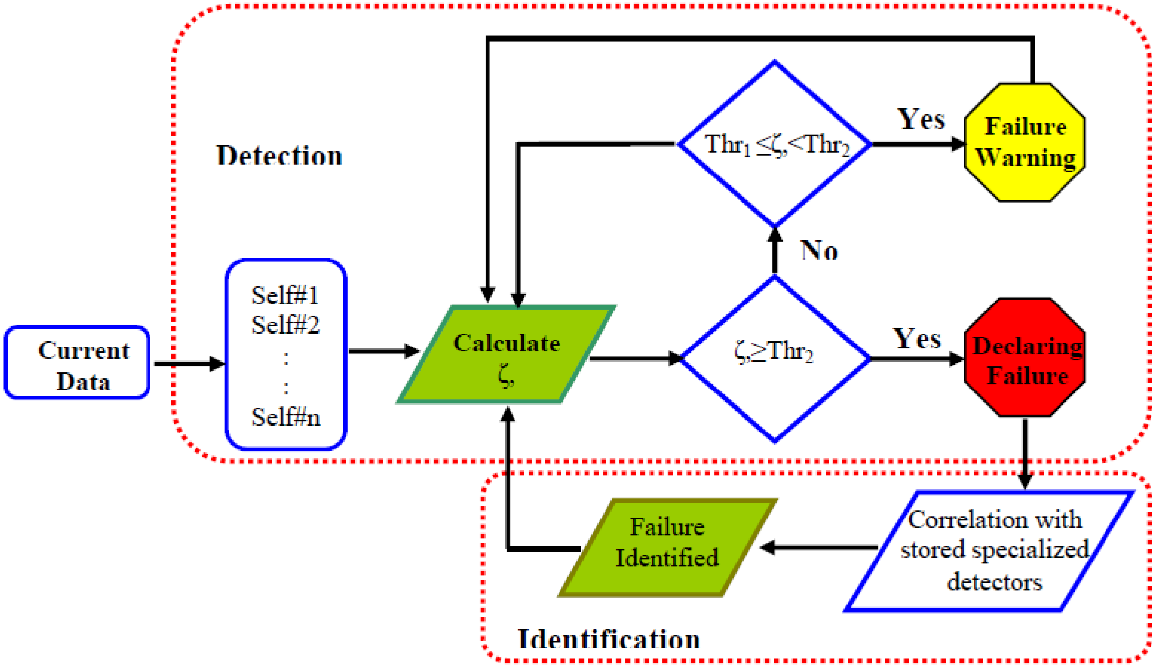


Figure 2.7. On-Line Failure Detection Logic with SNSA

2.1.5. Detection Performance Using Self/Non-self Discrimination

The set of aircraft features from Table 1.3 were used to build the AIS self and non-self. Several ACs were tested on the WVU 6-DOF motion-based flight simulator, as listed in Table 2.3. Different flight scenarios were considered over a wide range of the flight envelope for Mach numbers between 0.6 and 0.9 and altitudes between 9,000 ft and 31,000 ft. Each flight test lasted between 15 and 20 minutes. The data acquisition rate was 50 Hz. The two performance metrics used are the percentage detection rate and the false alarm rate, as defined in section 2.1.3.

Table 2.3. Aircraft Subsystems Considered for AC Detection

Category	*k*	**Subsystem**	**Tests Performed**
Actuators	1	Left Stabilator	Control surface locked at 2° and 8°
	2	Right Stabilator	Control surface locked at 2° and 8°
	3	Left Aileron	Control surface locked at 2.5° and 8°
	4	Right Aileron	Control surface locked at 2.5° and 8°
	5	Left Rudder	Control surface locked at 8°
	6	Right Rudder	Control surface locked at 8°
Structure	7	Left Wing	6% and 15% loss of the wing
	8	Right Wing	6% and 15% loss of the wing
Sensors	9	Roll Gyro Sensor	5 deg/sec and 10 deg/sec Fast Drifting Bias
	10	Pitch Gyro Sensor	5deg/sec and 10deg/sec Fast Drifting Bias
	11	Yaw Gyro Sensor	1deg/sec and 3deg/sec Fast Drifting Bias
Engine	12	Left Engine	100% efficiency reduction
	13	Right Engine	100% efficiency reduction

Table 2.4 presents the average detection performance for the SNSA-based failure detection schemes. As it can be seen from the results, the proposed methods exhibit very good performance with high detection rates and practically zero false alarms for the variety of ACs considered.

Table 2.4. Average Detection Performance of the SNSA Scheme

k	**Subsystem Affected by AC**	**Detection Rate [%]**	**False Alarms [%]**
1	LeftStabilator	99.78	0.00
2	RightStabilator	100	0.00
3	LeftAileron	97.5	0.00
4	RightAileron	97.48	0.00
5	LeftRudder	65.86	0.00
6	RightRudder	75.04	0.00
9	Left Wing	100	0.00
10	Right Wing	100	0.00
15	RollRateSensor	99.98	0.00
16	PitchRateSensor	99.98	0.00
17	YawRateSensor	97.19	0.00
18	LeftEngine	94.77	0.00
19	Right Engine	72.54	0.00

2.2. Dendritic Cell Mechanism for Abnormal Condition Detection

2.2.1. The Artificial Dendritic Cell

The DC mechanism-based approach for failure detection is inspired by the functionality of the biological DCs in the tissue and their interaction with the adaptive immune system. A detection outcome is produced based on the combined discrimination outcomes from all 2-D projections (possibly augmented with the outcomes from higher dimensional projections).

Let the time sample be denoted as *t*. Define the *detection matrix* D_t as:

$$D_t = \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & \dots & S_N \end{array} \\ \begin{bmatrix} Det_{11} & Det_{12} & \dots & Det_{1N_{SS}} \\ Det_{21} & Det_{22} & \dots & Det_{2N_{SS}} \\ \vdots & \vdots & \ddots & \vdots \\ Det_{w1} & Det_{w2} & \dots & Det_{wN_{SS}} \end{bmatrix} \end{array} \tag{2.3}$$

where $$Det_{ij} = \begin{cases} 0 & if \;\; feature \;\; po\,int \; is \; inside \; S_j \\ 1 & if \;\; feature \;\; po\,int \; is \; outside \; S_j \end{cases}, \quad i = 1,2,\dots,w \quad j = 1,2,\dots,N_{SS} \tag{2.4}$$

A moving time window of size *w* is assumed for detection. Each column of matrix D_t is a set of detection outcomes of a particular sub-self over the entire time window. Ideally, all elements of this matrix are equal to zero under normal conditions and to 1 under abnormal conditions. In practice, D_t usually contains both zeros and ones whether the time window is under normal or abnormal conditions due to imperfections in designing and building the sub-selves and/or due to the shape of the self that may project abnormal condition feature points into the self area. This situation may lead to false alarms and missed detections if not properly handled.

Let the complement of matrix D_t be:

$$\overline{D}_t = \left\{ \overline{Det}_{ij} \;\; | \;\; \overline{Det}_{ij} = non(\,Det_{ij}\,) \right\} \tag{2.5}$$

Cells from a pool of DCs are randomly selected to process the input matrices D_t and $\overline{D}_t$. For each DC the "*triggered features matrix*" F_1 and the "*non-triggered features matrix*" F_0 are built.

$$F_1 = \begin{array}{cc} \begin{array}{cccc} S_1 & S_2 & \dots & S_{N_{SS}} \end{array} & \\ \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1N_{SS}} \\ F_{21} & F_{22} & \dots & F_{1N_{SS}} \\ \vdots & \vdots & \ddots & \vdots \\ F_{N1} & F_{N2} & \dots & F_{NN_{SS}} \end{bmatrix} & \begin{array}{c} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{array} \end{array} \tag{2.6}$$

where $F_{qj} = F_{qj} + 1$ if φ_q belongs to the <u>triggered</u> sub-self S_j.

$$F_0 = \begin{array}{c} \\ \left[\begin{array}{cccc} S_1 & S_2 & \dots & S_{N_{SS}} \\ F_{11} & F_{12} & \dots & F_{1N_{SS}} \\ F_{21} & F_{22} & \dots & F_{1N_{SS}} \\ \vdots & \vdots & \ddots & \vdots \\ F_{N1} & F_{N2} & \dots & F_{NN_{SS}} \end{array}\right] \end{array} \begin{array}{c} \\ \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N \end{array} \tag{2.7}$$

where $F_{qj} = F_{qj} + 1$ if φ_q belongs to the <u>not-triggered</u> sub-self S_j.

For each sub-self S_j, a "confidence" factor ($w_{0i} \in [0,\ 1],\quad w_{1i} \in [0,\ 1]$) for accurately capturing the self and non-self is defined, which is used to define the following weighting vectors:

$$W_0 = \left[w_{01},\ \ w_{02},\ \ \dots,\ \ w_{0N_{SS}}\right]^T \tag{2.8}$$

represents the level of confidence that an output of "*0*" is trustworthy.

$$W_1 = \left[w_{11},\ \ w_{12},\ \ \dots,\ \ w_{1N_{SS}}\right]^T \tag{2.9}$$

represents the level of confidence that an output of "*1*" is trustworthy.

When a DC processes an input, the concentration of the secreted proteins, known as "co-stimulatory molecules," denoted by *CSM*, on the surface of the cell increases, regardless of whether the input is self or non-self (i.e., antigen). It is important to note that $\mathrm{T_h}$ -cells cannot be activated unless *both* the antigen and the *CSM* coexist. If the input is an antigen, the cell produces a special cytokine called "interleukin-12," denoted by $IL12$. Otherwise, it produces another type of cytokine called "interleukin-10," denoted by $IL10$. Thus, the presence of $IL12$ is an indication of abnormal condition and that of $IL10$ is a normal one. Any DC cannot migrate to the lymph node until its *CSM* reaches a predefined threshold, $\mathcal{M}$, which is *randomly* assigned to each cell $C_\ell \in \mathbb{C}$ during initialization. Like any other cells, some DCs experience "apoptosis," the healthy programmed cell death. This is expressed by assigning a *random* life, *L*, to each artificial cell $C_\ell \in \mathbb{C}$ during initialization.

Some cells migrate to the lymph node (the adaptive immune system) with an indication of abnormal conditions and stimulate the adaptive immune system to generate cytotoxic T-cells while other cells migrate with an indication of normal conditions and stimulate the adaptive immune system to generate suppressor T-cells to regulate the generated cytotoxic T-cells. This stimulation/suppression of the cytotoxic T-cells determines the resultant response of the adaptive immune system which indicates whether the system is under normal or abnormal conditions.

At each time sample, the selected DCs update their internal parameters based on detection outcome from all projections according to the following relationships:

$$IL10_t = IL10_{t-1} + I_{N_{SS}} \cdot \overline{D_t} \cdot W_0 \tag{2.10}$$

where $I_{N_{SS}} = [1,\ \ 1,\ \ \dots,\ \ 1],\ \ size(I_{N_{SS}}) = N_{SS}$

$$IL12_t = IL12_{t-1} + I_{N_{SS}} \cdot D_t \cdot W_1 \tag{2.11}$$

$$CSM_t = CSM_{t-1} + 1 \tag{2.12}$$

$$L_t = L_{t-1} - 1 \tag{2.13}$$

If $L = 0$, a cell is considered dead and it is replaced by a new cell. This can be done by reinitializing the dead cell with the default properties. Any cell with $CSM \geq \mathcal{M}$ should migrate from the tissue to the lymph node to present the processed input and must be replaced by a new cell with default properties. If these two conditions are not met, the cell continues processing the next element of D_t and $\overline{D}_t$, if the cell belongs to the randomly selected subset. Any migrated DC with $IL12 \geq IL10$ is called “stimulatory” since it activates cytotoxic T-cells; otherwise it is called “regulatory” since it activates suppressor (regulatory) T-cells.

Let the mature DC be denoted by DC_k, $k = 1,2,\ldots,N_{DC}$, where N_{DC} is the total number of mature DCs. Let the set of activated cytotoxic T-cells be $K = \{K_q \mid q = 1,2,\ldots,N\}$, where K_q is the number of activated cytotoxic T-cells corresponding to feature φ_q, then:

$$K_q = \sum_{m=1}^{N_{DC}} \sum_{j=1}^{N_{SS}} F_{1qj} \text{ for all } m \text{ for which } IL12 \geq IL10 \tag{2.14}$$

Let the set of activated suppressor T-cells be:

$$R = \{R_q \mid q = 1,2,\ldots,N\} \tag{2.15}$$

where R_q is the number of activated suppressor T-cells corresponding to feature φ_q, then:

$$R_q = \sum_{m=1}^{N_{DC}} \sum_{j=1}^{N_{SS}} F_{0qj} \text{ for all } m \text{ for which } IL12 < IL10 \tag{2.16}$$

The role of the suppressor T-cells is to regulate the adaptive immune response by suppressing a corresponding number of activated cytotoxic T-cells. This results in a set of residual cytotoxic T-cells given by:

$$\widetilde{K} = \{\widetilde{K}_q = K_q - R_q \mid q = 1,2,\ldots,N\} \tag{2.17}$$

The detection outcome at sample time *t* is finally defined as:

$$Det_t = \begin{cases} 0 & if \ \sum_{q=1}^{N} \widetilde{K}_q \leq 0 \\ 1 & otherwise \end{cases} \tag{2.18}$$

The block diagram of the proposed algorithm is illustrated in Figure 2.8. A set $\mathbb{C}$ of N_{DC} dendritic cells is initialized with each cell $C_\ell \in \mathbb{C}$ ($\ell = 1, 2, \cdots, N_{DC}$) having the data structure shown in Figure 2.9.

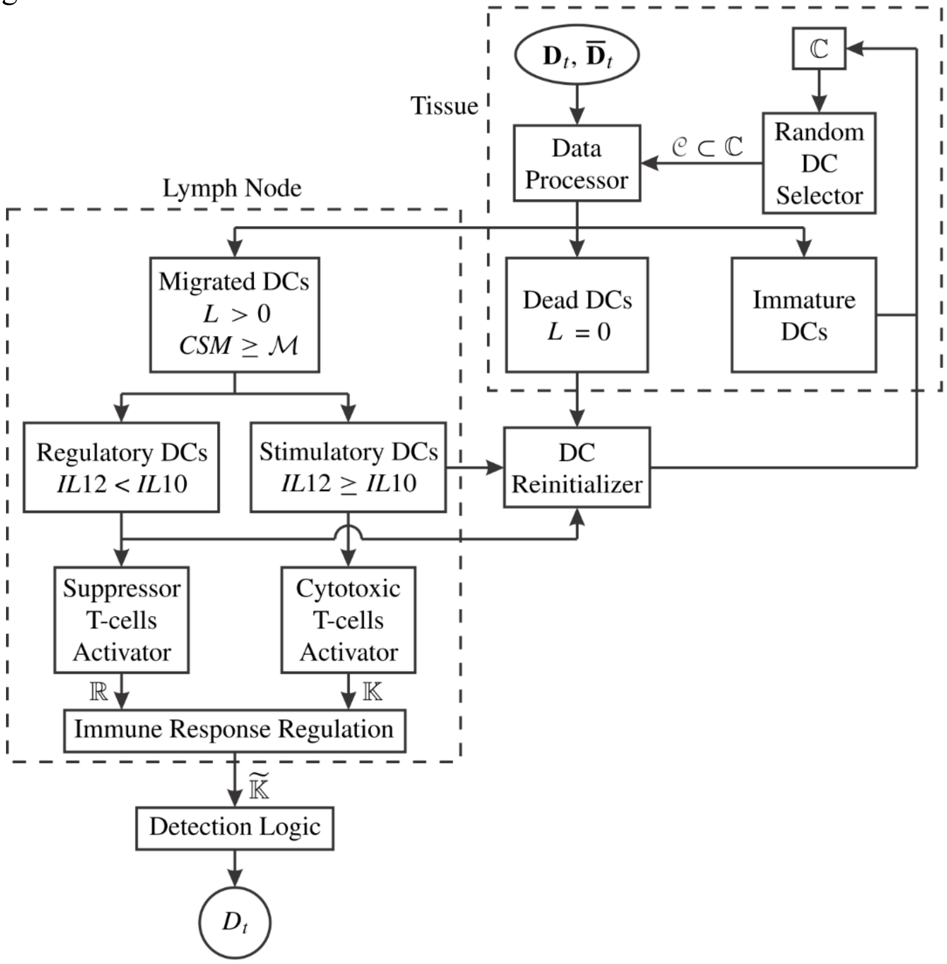

$\mathbf{D}_t, \overline{\mathbf{D}}_t$
$\mathbb{C}$
Tissue
Data Processor
$\mathcal{C} \subset \mathbb{C}$
Random DC Selector
Lymph Node
Migrated DCs $L > 0$ $CSM \geq \mathcal{M}$
Dead DCs $L = 0$
Immature DCs
Regulatory DCs $IL12 < IL10$
Stimulatory DCs $IL12 \geq IL10$
DC Reinitializer
Suppressor T-cells Activator
Cytotoxic T-cells Activator
$\mathbb{R}$
$\mathbb{K}$
Immune Response Regulation
$\widetilde{\mathbb{K}}$
Detection Logic
D_t

Figure 2.8. Block Diagram of the DC Algorithm for Failure Detection

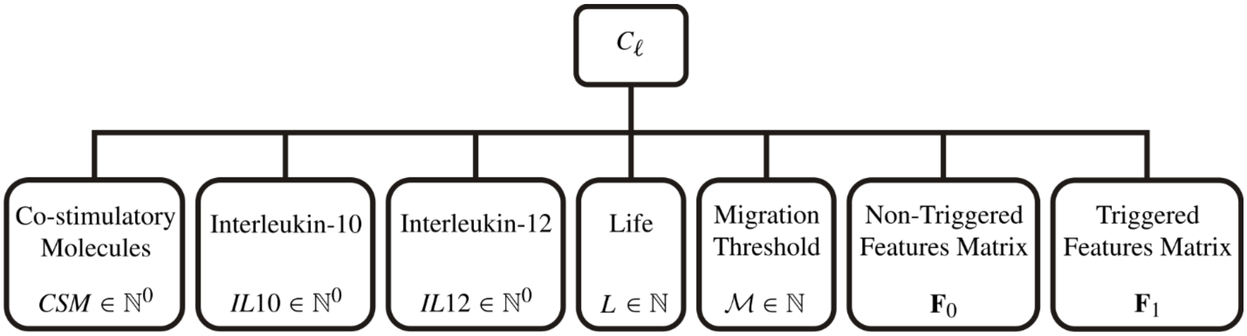

C_ℓ
Co-stimulatory Molecules $CSM \in \mathbb{N}^0$
Interleukin-10 $IL10 \in \mathbb{N}^0$
Interleukin-12 $IL12 \in \mathbb{N}^0$
Life $L \in \mathbb{N}$
Migration Threshold $\mathcal{M} \in \mathbb{N}$
Non-Triggered Features Matrix $\mathbf{F}_0$
Triggered Features Matrix $\mathbf{F}_1$

Figure 2.9. Data Structure of an Artificial Dendritic Cell

2.2.2. Detection Performance Using the DC Mechanism

The artificial DC mechanism for detection was implemented in C# for computational speed considerations. The code was compiled with Microsoft® .NET™ Framework version 4.5 and integrated with the WVU simulation environment for the fighter aircraft. Validation tests, along with the information of the generated sub-selves, were used as input data to the DC algorithm. The algorithm was initialized with a pool of 100 DCs and a moving time window of 50 time steps (i.e., 1 s). The sum of the residual cytotoxic T-cells $\sum_{i=1}^{N} \tilde{k}_i$ in Eq. 2.18 versus time is shown in Figures 2.10 - 2.13 for different tests. These figures show that the sum of the residual cytotoxic T-cells produced by the DC mechanism is negative and remains in a narrow band for normal conditions. When an AC occurs, this parameter experiences an abrupt increase and remains positive, capturing the occurrence of the AC and providing a reliable detection criterion.

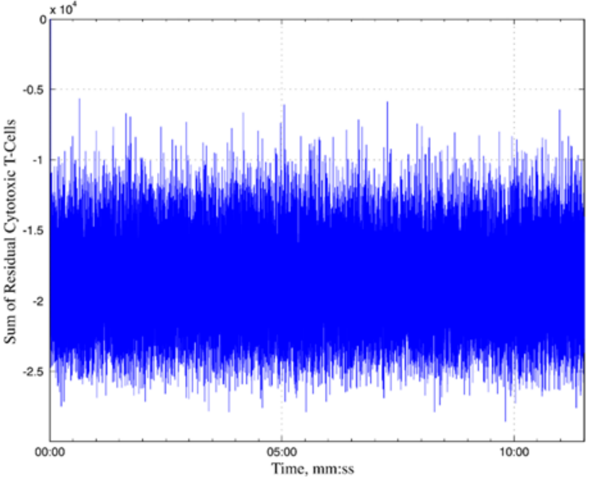


Figure 2.10. Sum of Residual Cytotoxic T-cells vs. Time of a Nominal Flight Test

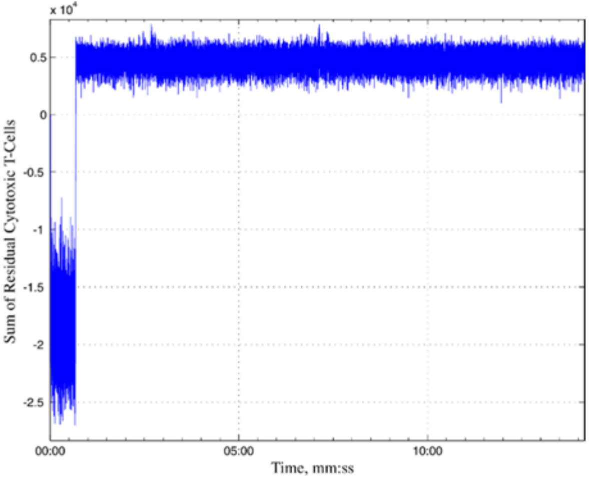


Figure 2.11. Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: Right Aileron Locked at 8°. Actual Failure Time = 40 s; Detection Time = 40.76 s

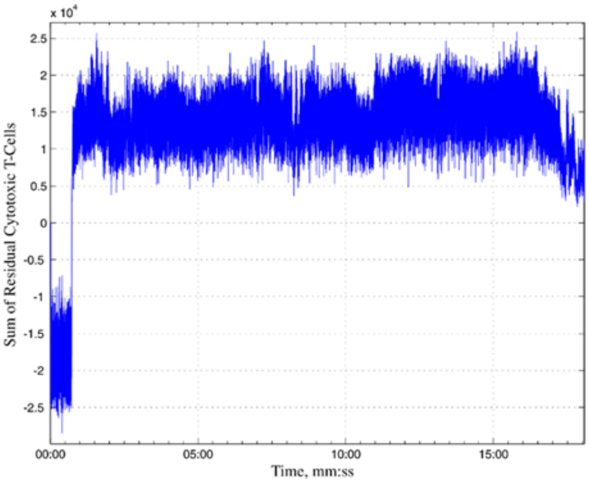


Figure 2.12. Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: 3° LFDB in Yaw Rate Sensor. Actual Failure Time = 40 s; Detection Time = 43.38 s

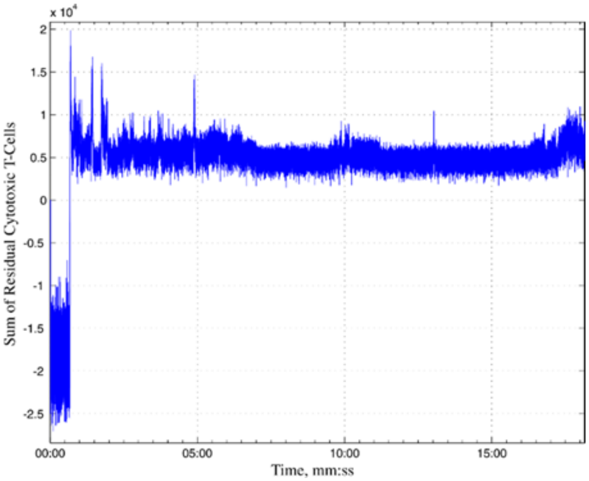


Figure 2.13. Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: 15% Loss of the Left Wing. Actual Failure Time = 40 s; Detection Time = 40.52 s

Table 2.5. summarizes the detection rate for different subsystems. The percentage detection rate is computed based on the ratio between the number of samples detected as AC and total number of samples under AC. Note that these rates are the averages of detection rates of the corresponding number of different tests with different failure magnitudes and/or flight paths in the third column of the table. Also note that each flight test lasts for about 20 min. on average and that all failures were introduced at 40 s. Failures can be introduced at any time during the flight test without affecting the performance of the algorithm. Samples during the first 40 s were used to determine the false alarm rate as the ratio between the number of samples detected as failure and the total number of samples under normal conditions. In all tests, the false alarm rate was zero. Without specific tuning of the parameters, the average detection time of the algorithm for the failures considered was 2 s, with a standard deviation of 2.8 s.

Table 2.5. Average DC-based Detection Rate for Different Subsystems

k	**Subsystem**	**Detection Rate,%**	**No.Tests**
1	Left Stabilator	99.93	3
2	Right Stabilator	99.93	3
3	Left Aileron	99.46	4
4	Right Aileron	98.94	3
5	Left Rudder	93.09	3
6	Right Rudder	80.47	2
7	Left Throttle	54.76	2
8	Right Throttle	54.72	2
9	Left Wing	99.93	4
10	Right Wing	99.94	4
11	Left Horizontal Tail	99.82	2
12	Right Horizontal Tail	99.91	2
13	Left Vertical Tail	24.95	3
14	Right Vertical Tail	16.66	1
15	Roll Rate Sensor	96.46	6
16	Pitch Rate Sensor	99.86	5
17	Yaw Rate Sensor	99.78	5
18	Left Engine	59.72	4
19	Right Engine	59.22	4

Chapter 3

Immunity-based Identification of Aircraft Subsystem Abnormal Conditions

3.1. Structured Non-self Approach for Abnormal Condition Identification

3.1.1. Principles of AC Identification Using the Structured Non-self Approach

Structuring the non-self for AC identification and converting the detectors into identifiers by attaching the $dSys_j$ tags involves the identification of detector sub-sets that correspond to AC that affect specific sub-systems. This implies the prior generation of the non-self with adequate resolution and knowledge of AC characteristics. Such information can be obtained from tests, simulation, or analysis. The concept is illustrated in Figure 3.1 for the 2-dimensional case with the assumption that the sub-sets $dSys_j$ have only one element each.

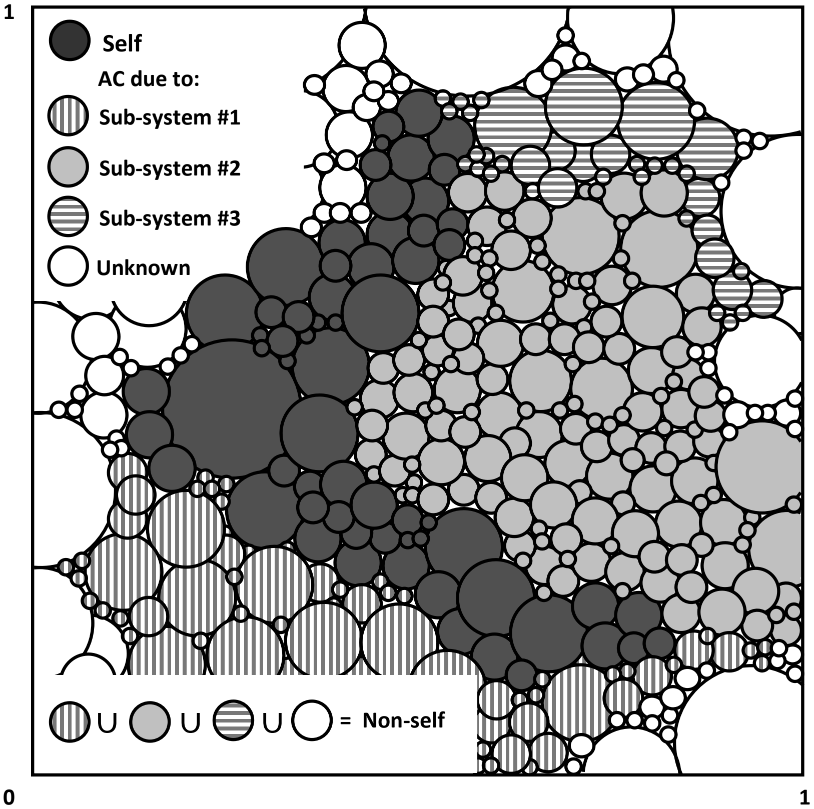


Figure 3.1. Failed Subsystem Identification Using the Structured Non-self Approach

If $Det_t = 1$, then the triggering detector is checked for structural parameters and a current identification outcome is defined using Eq. (1.2), for example, as:

$$Idt_t = \begin{bmatrix} id_1 & id_2 & \cdots & id_{Ns} \end{bmatrix}, \qquad \begin{array}{ll} id_i = 0 & for\ i \notin dSys_j \\ id_i = 1 & for\ i \in dSys_j \end{array} \tag{3.1}$$

If Eq. (1.3) is used then the current identification outcome is defined as:

$$Idt_t = \lfloor id_1 \quad id_2 \quad \cdots \quad id_{Nsf} \rfloor, \qquad id_{\hat{i}} = i \quad for\ i \in dSys_j \tag{3.2}$$

If one N-dimensional non-self is used with perfect feature definition, then the identification process outcome can be defined as:

$$Idt = Idt_t \tag{3.3}$$

If the HMS strategy is used, then Eq. (3.1) or (3.2) is applied to each sub-self and a composition logic $\mathscr{C}$ must be designed to obtain the current identification outcome as:

$$Idt_t = \mathscr{C}(\, Idt_{ti}\,), \qquad i = 1,2,\ldots,N_{ss} \tag{3.4}$$

then Eq. (3.3) can be applied. However, as was the case with the detection process, using Eq. (3.3) directly may produce a large number of incorrect identifications. This can be mitigated by reprocessing the current outcomes Idt_t over a moving time window. The composition logic $\mathscr{C}$ can also rely on the DC mechanism.

3.1.2. Identification Algorithm Using the Structured Non-self Approach

The identification algorithm is only activated when the detection algorithm declares that an abnormal condition is present. The identification phase assumes that the output of the detection phase is correct.

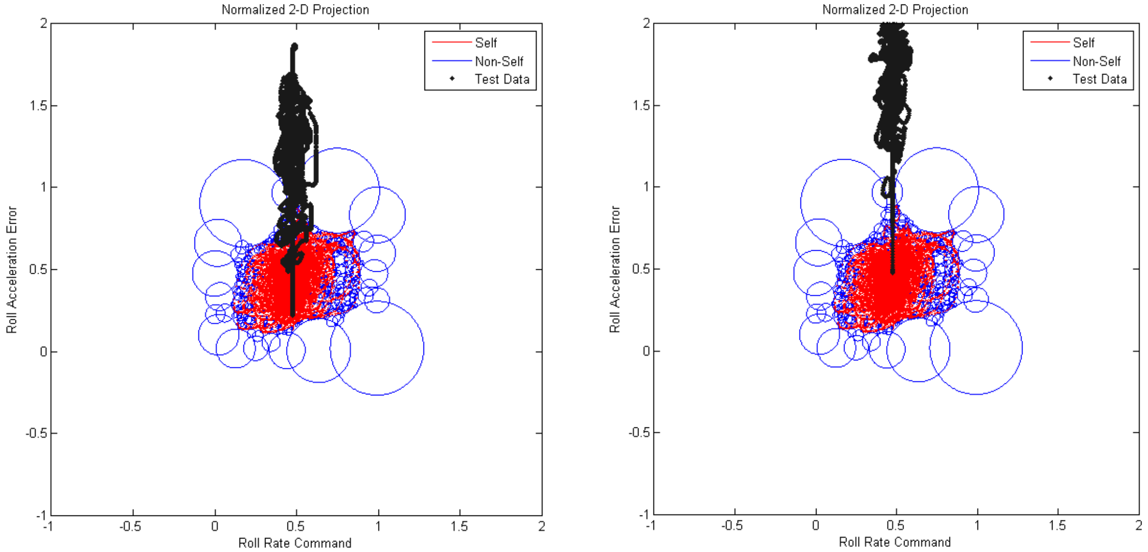
In this research effort a novel structured non-self approach (SNSA) has been developed within a HMS strategy. This approach is based on a structuring process, or arrangement, of non-self projections and intends to reduce the computational effort required and facilitate the real-time application of the AIS approach without compromising the ACFDIE performance. The SNSA consists of a dual- phase algorithm where 2-dimensional self/non-self projections, previously generated using a negative selection mechanism and tested in simulation under several abnormal conditions, are selected according to the ability to detect failures at a predefined detection rate percentage. Then, by using a positive selection-type mechanism, the resulting projections are processed in order to generate identifiers capable of differentiating similar dynamic prints among several abnormal conditions and declaring correct failure types, and magnitudes. For example, within a first phase of the SNSA, a total of 496 self/non-self projections were generated based on the availability of 32 features to capture the dynamic print of abnormal conditions. After extensive experimentation it was possible to reduce the number of self/non-self projections to 183 possible candidates with a DR equal or larger than 70%. These projections are considered to possess the ability to capture the dynamic print of several sub-system failures and, more importantly, facilitate the process of characterizing the projections that perform better during the identification of specific failures. Table 3.1 presents a sample set of 2-dimensional projections investigated within the identification phase.

It is important to note that the dynamic fingerprint of several failures may produce a very similar effect on the features of self/non-self projections. This characteristic presents a more complex problem in which incorrect identification may be produced if the identification problem is not defined appropriately. For example, let us assume that an identification algorithm, only consisting of *Self#3* (p_{ref}, NN_p), is tested for two sub-system failures (i.e. right wing structural failure and left aileron stuck failure). This particular pair of failures will produce an undesired roll rate that can be successfully perceived and detected by *Self#3*. The dynamic fingerprint produced by both abnormal conditions in the selected projection may look very similar, increasing the complexity of the identification problem. Now, let us assume that the same identification algorithm is augmented with *Self#30* (q_{ref}, NN_p), which can also capture the abnormal condition dynamic print of the mentioned failures. Due to the fact that *Self#30* also captures dynamic changes in pitch rate, it is possible to identify and distinguish between the two mentioned failures. Within a second phase of the SNSA, positive selection applied to the 183 self/non-self projections is performed in order to address the mentioned identification problem. For illustration, Figures 3.2 and 3.3 present the similarity of the dynamic print of two different failures in a 2-dimensional projection.

Table 3.1. Self/Non-self 2-D Projections for SNSA Approach

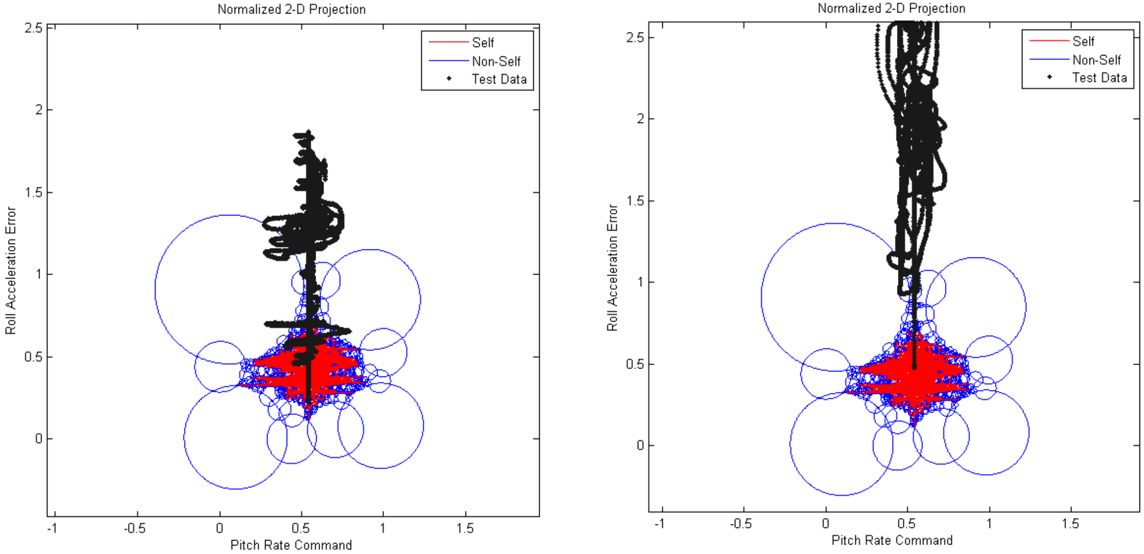
Self	**Features**	**Self**	**Features**
Self#3	p_{ref}, NN_p	**Self#56**	r_{ref}, NN_p
Self#4	p_{ref}, NN_q	**Self#57**	r_{ref}, NN_q
Self#7	$p_{ref}, OQEE$	**Self#60**	$r_{ref}, OQEE$
Self#8	$p_{ref}, DQEE_p$	**Self#61**	$r_{ref}, DQEE_p$
Self#9	$p_{ref}, DQEE_q$	**Self#62**	$r_{ref}, DQEE_q$
Self#30	q_{ref}, NN_p	**Self#69**	r_{ref}, r
Self#31	q_{ref}, NN_q	**Self#82**	NN_p, NN_q
Self#34	q_{ref}, NN_p	**Self#83**	NN_p, NN_r
Self#35	$q_{ref}, DQEE_p$	**Self#84**	$NN_p, MQEE$
Self#36	$q_{ref}, DQEE_q$	**Self#85**	$NN_p, OQEE$
Self#42	p_{ref}, NN_p	**Self#86**	$NN_p, DQEE_p$
Self#52	q_{ref}, d_e	**Self#87**	$NN_p, DQEE_q$
Self#53	q_{ref}, d_r	**Self#88**	$NN_p, DQEE_r$

The combined identification capabilities of the projections utilized within the two phases of the SNSA (see Figure 3.4) provides a more robust system capable of not only correctly identifying the detected failure but also providing information regarding the magnitude of the investigated failures. With the correct combination of projections and their corresponding identifiers, it is possible to discard incorrect identifications and ultimately determine which abnormal condition is affecting the system.



(a) (b)

Figure 3.2. (a) Self#3 with Left Aileron Failure; b) Self#3 with Right Wing Structural Damage



(a) (b)

b)

Figure 3.3. (a) Self#30 with L Aileron Failure; (b) Self#30 with R Wing Structural Damage

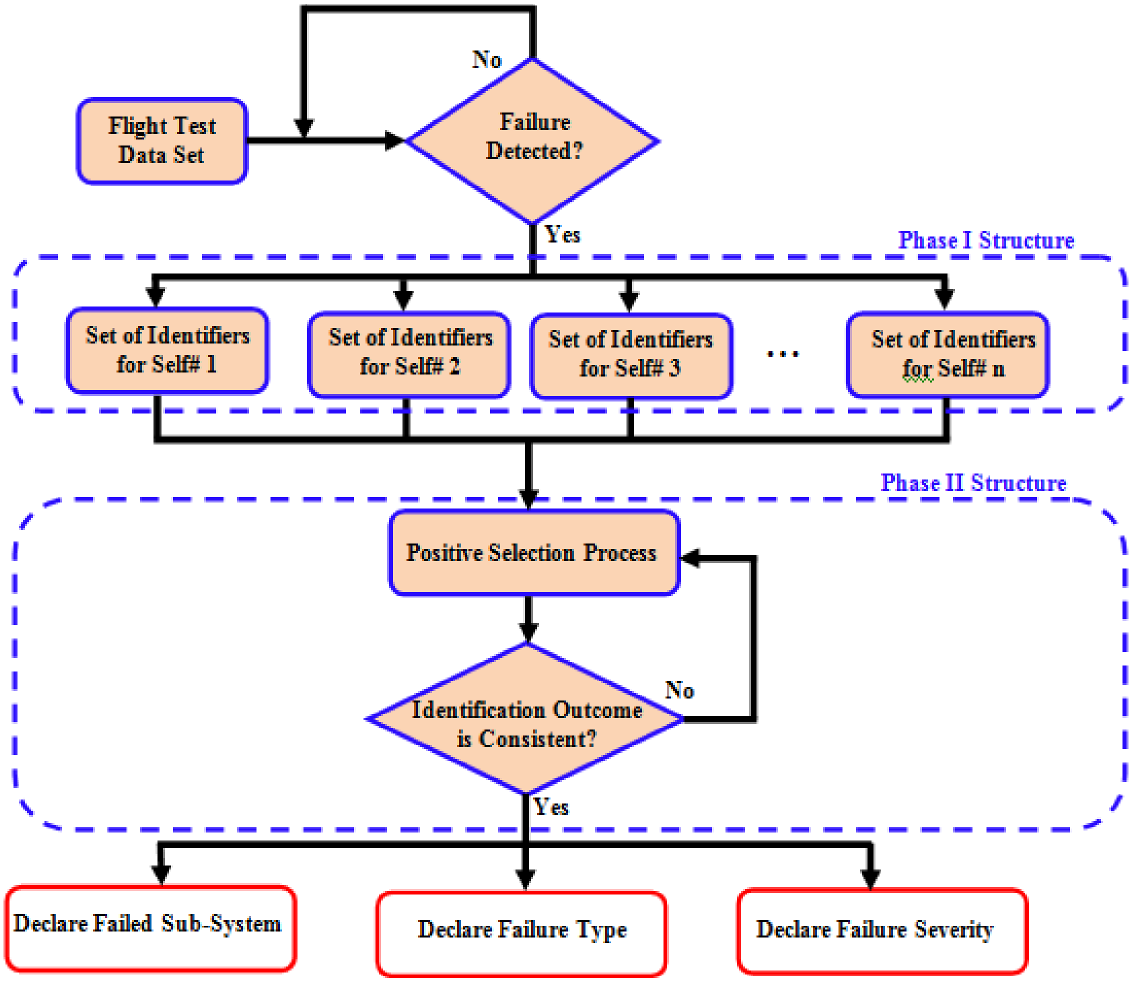

No
Flight Test Data Set
Failure Detected?
Yes
Phase I Structure
Set of Identifiers for Self# 1
Set of Identifiers for Self# 2
Set of Identifiers for Self# 3
. . .
Set of Identifiers for Self# n
Phase II Structure
Positive Selection Process
Identification Outcome is Consistent?
No
Yes
Declare Failed Sub-System
Declare Failure Type
Declare Failure Severity

Figure 3.4. Structured Non-Self Approach Logic

3.1.3. Generation of Identifiers

The process of generating identifiers is very similar to the detector generation. However, the generation of identifiers is based on a positive selection-type of algorithm, in which failure flight test data are used to expose the dynamic fingerprint of a failure throughout the entire flight envelope. The generation of identifiers consists of a multi-step process that optimizes the set . The radii of the identifiers depend mostly on their distance to the self. The flow chart of the identifier generation is presented in Figure 3.5. The main components of the process are described next.

Abnormal Flight Tests. Flight tests at different abnormal conditions throughout the entire flight envelope must be performed. Previously selected features corresponding to the self/non-self definition as shown in Table 3.1 must be recorded for future processing and identifier definition.

Normalization. The sets of raw data received from the flight tests are recorded and their values are normalized between 0 and 1. The normalization factor of each projection is determined by the range of the flight data plus a percent margin. The normalization factor is the same used for the self/non-self projections during the antibodies generation. Therefore, the normalized data points of failure data correspond to the correct hypercube projection of each specific feature combination.

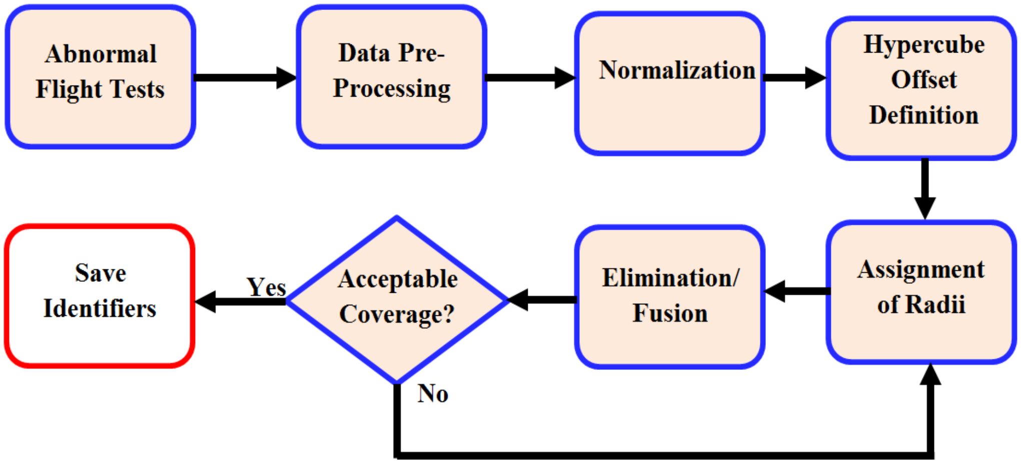


Figure 3.5. Flowchart of the Identifier Generation Process

Offset Hypercubes. The unit hypercube determined during the normalization process delimits the hyperspace of the nominal conditions. High magnitude failures may contain data points that lay far away from the unit hypercube of the self/non-self projection. Therefore, outward concentric hypercubes are defined in order to determine the distance of the abnormal condition point from the self (see Figure 3.6), which subsequently allows the algorithm to determine the magnitude of the corresponding failure as part of the direct quantitative evaluation. Note that these phases defined, in general, subsequently may be performed simultaneously.

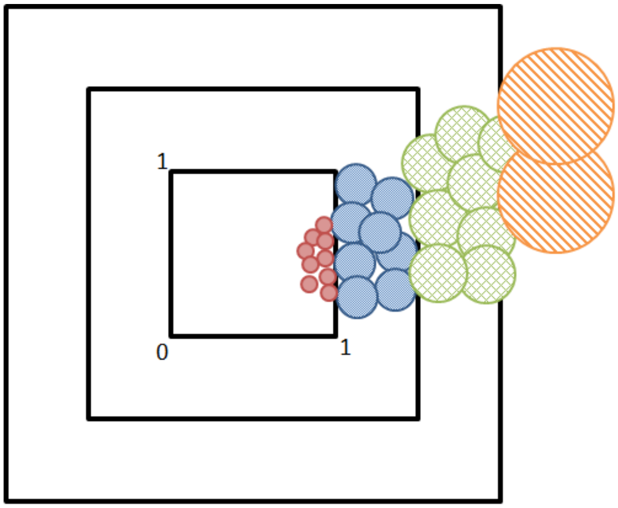


Figure 3.6. Concentric Hypercubes and Radii Variation With Respect to Distance from the Self

Radii Assignment. The radius of any identifier is predetermined and it is assigned depending on the location of its center with respect to the offset hypercubes. The radius of an identifier increases as the position of its center lies within an outward hybercube. In other words, the radii of all identifiers increase as their distance to the self increases as illustrated in Figure 3.6.

Identifiers Elimination/Fusion. The amount of initial identifiers depends on the number of data points obtained from the flight tests. This may yield an enormous number of identifiers which will produce a degradation of the computer processing capability. A simple elimination algorithm is implemented in order to reduce the number of identifiers. Identifiers that lay inside the radius of another identifier plus a tolerance are eliminated. Finally, a fusion process is performed. The fusion process consists of a set union accompanied with overlapping elimination. After this step is concluded, the final number of identifiers is reduced considerably.

3.1.4. The Two Phases of the SNSA Approach

The first phase of the SNSA is the result of the failure detection testing within the HMS strategy and consists in the selection of lower dimensional projections. Only 2-D projection have been considered in the phase of the project. As mentioned previously, 496 2-dimensional self/non-self projections were generated for failure detection algorithm experimentation. These projections were then tested against over 20 different failures. Different magnitudes/severities of the ACs were also considered. Extensive experimentation was required in order to determine which projections could substantially detect a failure with good detection rates and minimum false alarms within a NS approach. It was determined that a total of 183 projections were able to fulfill the objectives of a DR equal or higher than 70%. This process is referred to as the Phase I Non-Self Structuring. The selected projections were chosen as potential candidates for identification included sensor outputs, state estimates, and statistical parameters among other features. The set of abnormal conditions involved sensor failures, structural damage on the wings, engine failures, and control surface failures. Table 3.2 presents a list of the failures investigated in this research effort.

Table 3.2. Phase I SNSA Subsystem Failures

Failure #	**Failure Type**	**Failure #**	**Failure Type**
1	Left Aileron 2deg	**9**	Left Wing Loss 6%
2	Right Aileron 2deg	**10**	Right Wing Loss 6%
3	Left Aileron 8deg	**11**	Left Wing Loss 15%
4	Right Aileron 8deg	**12**	Right Wing Loss 15%
5	Left Stabilator 2deg	**13**	Left Engine Out
6	Right Stabilator 2deg	**14**	Right Engine Out
7	Left Stabilator 8deg	**15**	Roll Sensor Bias 5deg/sec
8	Right Stabilator 8deg	**16**	Roll Sensor Bias 10deg/sec

Several failures presented similar dynamic prints on several 2D-projections, which subsequently led to the repetition of several projections with the ability to detect multiple failures. On the other hand, certain failures that are difficult to detect, such as rudder failure, only resulted in the activation of a small number of projections. The negative selection logic behind the Phase I Structuring resulted in the reduction of the number of the original projections into a smaller set, reducing the complexity and the hardware requirements for its implementation. Table 3.3 presents a sample of the projections that are considered to be adequate for abnormal condition identification based on a detection performance equal or higher than 70%. The detection capability of this sample set of projections is presented for five types of failures. Various projections present the ability to capture the dynamic fingerprint of several abnormal conditions while others can only capture the dynamics of a small set or just a single abnormal condition. For example, *Self#3* demonstrated its ability to capture the dynamic fingerprint of a left aileron locked, a right wing structural damage and a left stabilator locked failure. On the other hand, *Self#4* only demonstrated the ability to capture the dynamic fingerprint of a left stabilator locked type of failure.

Table 3.3. Detection Sensitivity of a Sample Set of Projections (%DR)

Failure **Self**	**Left Aileron stuck at 2.5 deg**	**Right Wing 6% Structural Damage**	**Left Stabilator Stuck at 8 deg.**	**Right Engine Out**	**Pitch Sensor 10deg/sec Bias**
Self# 3	82.02	99.84	99.96	10.51	3.42
Self# 4	1.45	3.87	99.85	1.42	4.08
Self# 30	83.49	99.83	99.96	10.57	30.02
Self# 31	0.85	1.94	99.82	0.52	60.68
Self# 52	0.99	0.76	1.56	1.10	71.52
Self# 56	86.85	99.94	99.88	12.94	0.59
Self# 82	92.33	99.96	99.97	21.32	15.01
Self# 83	88.06	99.93	99.97	14.13	0.74
Self# 84	86.23	99.94	99.98	12.05	0.30
Self# 85	88.76	99.91	99.96	12.80	37.42
Self# 100	86.92	99.45	99.25	15.20	0.46
Self# 142	0.06	29.35	56.08	72.42	0.99
Self# 233	5.51	7.47	7.02	5.49	92.07
Self# 259	13.44	54.05	77.33	72.54	9.41
Self# 350	15.39	33.10	60.70	72.47	22.27
Self# 351	26.07	50.30	67.13	71.49	14.23
Self# 433	1.39	1.44	6.76	2.33	77.7

Within this analysis it was possible to isolate the projections that can be used for identification purposes. From the Phase I analysis, it was possible to determine which specific projections correspond to every specific failure investigated. Furthermore, it is also possible to determine how many projections capture the dynamic fingerprint of an abnormal condition. Table 3.4 below presents the number of projections from the 183 total initially selected that have the potential to be used for successful identification purposes.

The outcome of the Phase I self/non-self structuring reduces the total amount of projections needed to perform ACFDIE. Its outcomes also allow for adequate design of the mapped-based positive selection algorithm utilized in the second phase of the SNSA by reducing the number of possible projections for the generation of identifiers as well as the reduction of identifiers required in the identification algorithm for each individual projection.

Phase II of SNSA includes a positive selection process where flight failure test data are used to generate higher resolution non-self detectors called identifiers. Resulting projections from Phase I are processed in order to generate identifiers capable of differentiating similar dynamic fingerprints among several abnormal conditions and declaring correct failure types, and magnitudes. In order to obtain correct identification results, the identification logic must be carefully formulated and the generation and selection of identifiers must be adequate. Subsets of antibodies or identifiers must be generated with sufficient resolution to avoid incorrect outputs.

Table 3.4. Total Number of Projections Activated per Failure

Actuator						**Engine**		**Structural**	
Aileron Stuck (8 deg)		**Stabilator Stuck (8 deg)**		**Rudder Stuck (8 deg)**		**Engine Out**		**Wing Damage (15%)**	
L	**R**	**L**	**R**	**L**	**R**	**L**	**R**	**L**	**R**
31	31	72	62	9	11	31	4	31	31

The identifiers generated during Phase I and II are loaded into an identification function and organized in a single array such that the index of each identifier corresponds to a failure type and magnitude. The arrangement of the identifiers is inspired by a mapping-based algorithm, which simplifies the selection scheme. The positive selection process is performed in parallel by all the projections included in the identification algorithm. Each projection outputs a single index that corresponds to a type and magnitude of failure. The outputs of all projections are compared among each other and the most frequent value is determined. If a specific failure index is constant throughout the majority of the projections' outputs, its value is selected and a proper identification is declared.

3.1.5. AC Identification Performance Using Self/Non-self Discrimination

Since the SNSA covers not only general identification logic, but also a qualitative and quantitative evaluation logic integrated into a single less complex algorithm, the identification results are presented in Section 4.1 integrated with classification results of magnitude and category of failure. This novel approach intends to reduce the computational processing for real time application of the solution to the ACDIE problem. The proposed mapping-based positive selection logic targets a multi-dimensional problem by means of a simpler but effective logic that can result in a more efficient real time algorithm.

3.2. Dendritic Cell Mechanism for Abnormal Condition Identification

3.2.1. Dendritic Cell Mechanism for AC Identification

The DC mechanism-based approach for AC identification infers the subsystem affected based on the topography of the detection outcomes from self/non-self projections, within the HMS strategy. Therefore, with the DC approach, the AC identification is formulated as a pattern recognition problem. The AC identification consists of determining which one of the N_S subsystems has been affected by the AC and it starts once an AC has been detected. Patterns for AC identification purposes may be defined based on the F_1 matrix. Note that the F_0 matrix may also be introduced into the process. N_S different patterns must be established, one associated to each subsystem. The process of establishing the library of pattern is presented in Figure 3.7. There are three possible alternatives for defining the patterns:

Feature-pattern (F-P) approach
Projection-pattern (P-P) approach
Matrix-pattern (M-P) approach

A). Feature-pattern (F-P) approach for AC identification

For each AC, define the vector of membership values of each feature to the set of equivalent directly involved variables:

$$FP_{ACj} = \left[m_1 \quad m_2 \quad \dots \quad m_N \right]^T, \ m_i \in \left[0, \quad 1 \right] \tag{3.5}$$

For each subsystem *k*, define the *FP* vector as:

$$FP_k = \frac{1}{\max\limits_i \left(\sum\limits_{j=1}^{N_{ftk}} FP_{ACj} \right)} \sum_{j=1}^{N_{ftk}} FP_{ACj} = \left[\overline{m}_1 \quad \overline{m}_2 \quad \dots \quad \overline{m}_N \right]^T \tag{3.6}$$

where $$\overline{m}_i \in \left[0, \quad 1 \right], \ k = 1,2,\dots,N_S \tag{3.7}$$

All these membership values can be determined based on tests and heuristics. Note that binary logic can be used instead of fuzzy logic. In this case: $m_i \in \{0 \quad 1\}$ and $\overline{m}_i \in \{0 \quad 1\}$.

At each sample time, after an AC is detected, use the F_1 matrices of the mature DCs to compute the current AC FP as:

$$F_{1\varphi} = \left[\sum_{q=1}^{N_{DC}} F_{1q} \right] \cdot I_{N_{SS}}, \ I_{N_{SS}} \in \{1\}^{N_{SS} \times 1} \tag{3.8}$$

Use a matching algorithm to determine which one of the N_S FP vectors best matches $F_{1\varphi}$.

B). Projection-pattern (P-P) approach for AC identification

For each AC, define the vector of membership values of each projection to the set of AC-relevant projections (projections that can capture the dynamic fingerprint of the AC):

$$PP_{ACj} = \left[n_1 \quad n_2 \quad \dots \quad n_{N_{SS}} \right]^T , \; n_i \in [0, \quad 1] \tag{3.9}$$

For each sub-system *k*, define the *P-P* vector as:

$$PP_k = \frac{1}{\max\limits_{i}\left(\sum\limits_{j=1}^{N_{ftk}} PP_{ACj} \right)} \sum_{j=1}^{N_{ftk}} PP_{ACj} = \left[\bar{n}_1 \quad \bar{n}_2 \quad \dots \quad \bar{n}_{N_{SS}} \right]^T \tag{3.10}$$

where $$\bar{n}_i \in [0, \quad 1], \; k = 1,2,\dots,N_S \tag{3.11}$$

All these membership values can be determined based on tests and heuristics. Note that binary logic can be used here as well.

At each sample time, after an AC is detected, use the F_1 matrices of the mature DCs to compute the current AC PP as:

$$F_{1P} = I_N \cdot \left[\sum_{q=1}^{N_{DC}} F_{1q} \right], \; I_N \in \{1\}^{1 \times N_{SS}} \tag{3.12}$$

Use a matching algorithm to determine which one of the N_S PP vectors best matches F_{1P}.

C). Matrix-pattern (M-P) approach for AC identification

For each sub-system k, define the FP vector FP_k and the PP vector PP_k as described above. Use them to build the MP matrix MP_k.

$$MP_k = \begin{bmatrix} m_{11}^* & m_{12}^* & \dots & m_{1N_{SS}}^* \\ m_{21}^* & m_{22}^* & \dots & m_{2N_{SS}}^* \\ \vdots & \vdots & \ddots & \\ m_{N1}^* & m_{N2}^* & \dots & m_{NN_{SS}}^* \end{bmatrix}, \; m_{ij}^* \in [0, \quad 1], \; i = 1,2,\dots,N , \; j = 1,2,\dots,N_{SS} \tag{3.13}$$

where $$m_{ij}^* = \begin{cases} \max(\bar{m}_i , \bar{n}_j) & \text{if } \varphi_i \text{belongs to } S_j \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

At each sample time, after the AC is detected, compute the sum of all mature DC F_1 matrices as:

$$F_{1M} = \sum_{q=1}^{N_{DC}} F_{1q} , \; q = 1,2,\dots,N_{DC} \tag{3.15}$$

Finally, use a matching algorithm to determine which one of the N_S M-P matrices best matches F_{1M}.

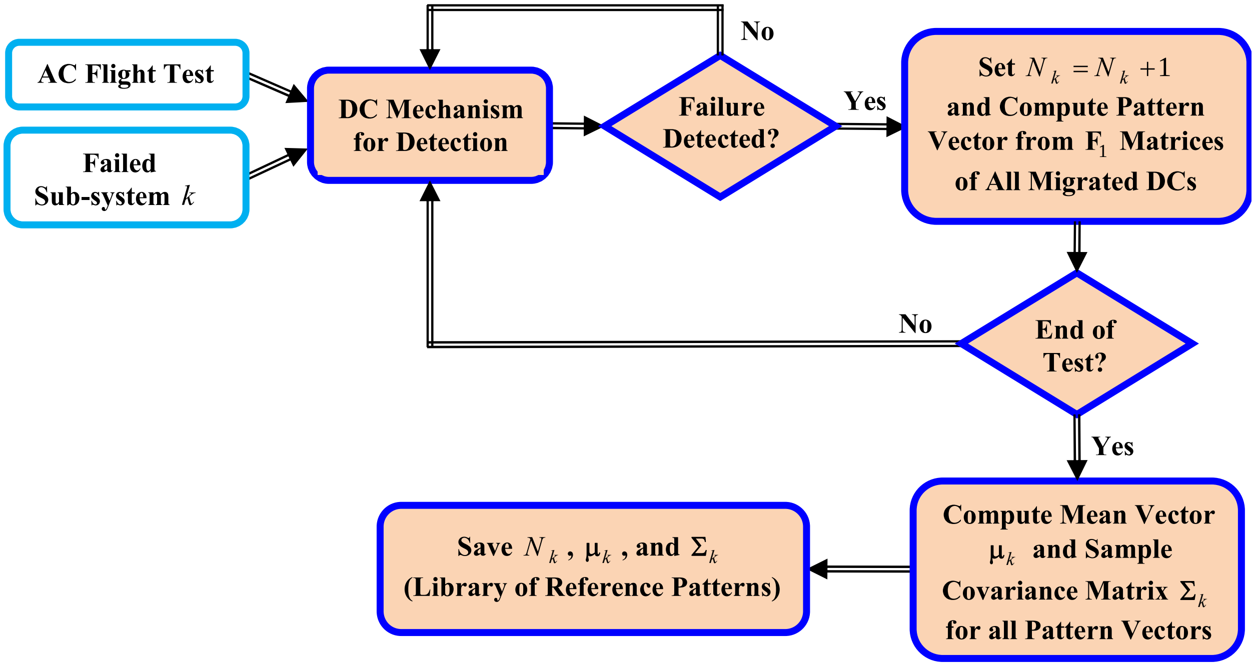

AC Flight Test
Failed Sub-system k
DC Mechanism for Detection
Failure Detected?
No
Yes
Set $N_k = N_k + 1$ and Compute Pattern Vector from F_1 Matrices of All Migrated DCs
End of Test?
No
Yes
Compute Mean Vector μ_k and Sample Covariance Matrix Σ_k for all Pattern Vectors
Save N_k, μ_k, and Σ_k (Library of Reference Patterns)

Figure 3.7. Generation of Library of Reference Patterns for Identification Using the DC Mechanism for Identification and the Naïve Bayes Classifier

3.2.2. Performance of AC Identification Using a Dendritic Cell Mechanism

Among the three pattern approaches, the F-P approach was selected since it requires less computational resources as compared to the P-P and M-P approaches. The artificial DC mechanism for identification with the F-P approach was implemented in C#. The naïve Bayes classifier was trained with the current F-P vectors $F_{1\varphi}$ from a set of training failure tests for each subsystem k to implicitly define the reference features-pattern FP_k of that subsystem in terms of the corresponding mean vector and covariance matrix. Figure 3.8 illustrates the variation of the current F-P vectors $F_{1\varphi}$ over the entire test time of a failed right wing subsystem (k = 13) in one of these tests. Figures 3.9 through 3.11 depict the reference FP_k vectors of the right wing (k = 13), left aileron (k = 3), and the yaw rate sensor (k = 11), respectively. Once a failure is declared by the DC mechanism, the current $F_{1\varphi}$ vector is passed to the naïve Bayes classifier to select the closest FP_k to the current $F_{1\varphi}$ vector, and thus identify the failed subsystem. A sample current $F_{1\varphi}$ is shown in Figure 3.12 for a validation failure test with right wing damage.

Listed in Table 3.5 are the average identification rates for different subsystems. These results show the capability of the proposed AC identification scheme in isolating the failed subsystem with very high identification rates. Note that the first column in Table 3.4 represents the subsystem affected by the failure and the first row represents the subsystem identified as the failed one. The off-diagonal numbers in the table represent the incorrect identification (misclassification) rates. For example, 99.13% of the samples under failure of the right stabilator were correctly identified, whereas for 0.81% of these samples, the failed subsystem was incorrectly identified as the left stabilator; for 0.05% of these samples, the failed subsystem was incorrectly identified as the left rudder; and for 0.01% of these samples, the failed subsystem was incorrectly identified as the pitch rate sensor. The small variations in performance between subsystems may be typically attributed to imperfect coverage of the self, imperfect detector generation, errors introduced by the projections, and variations of the classification algorithm used in conjunction with the DC mechanism.

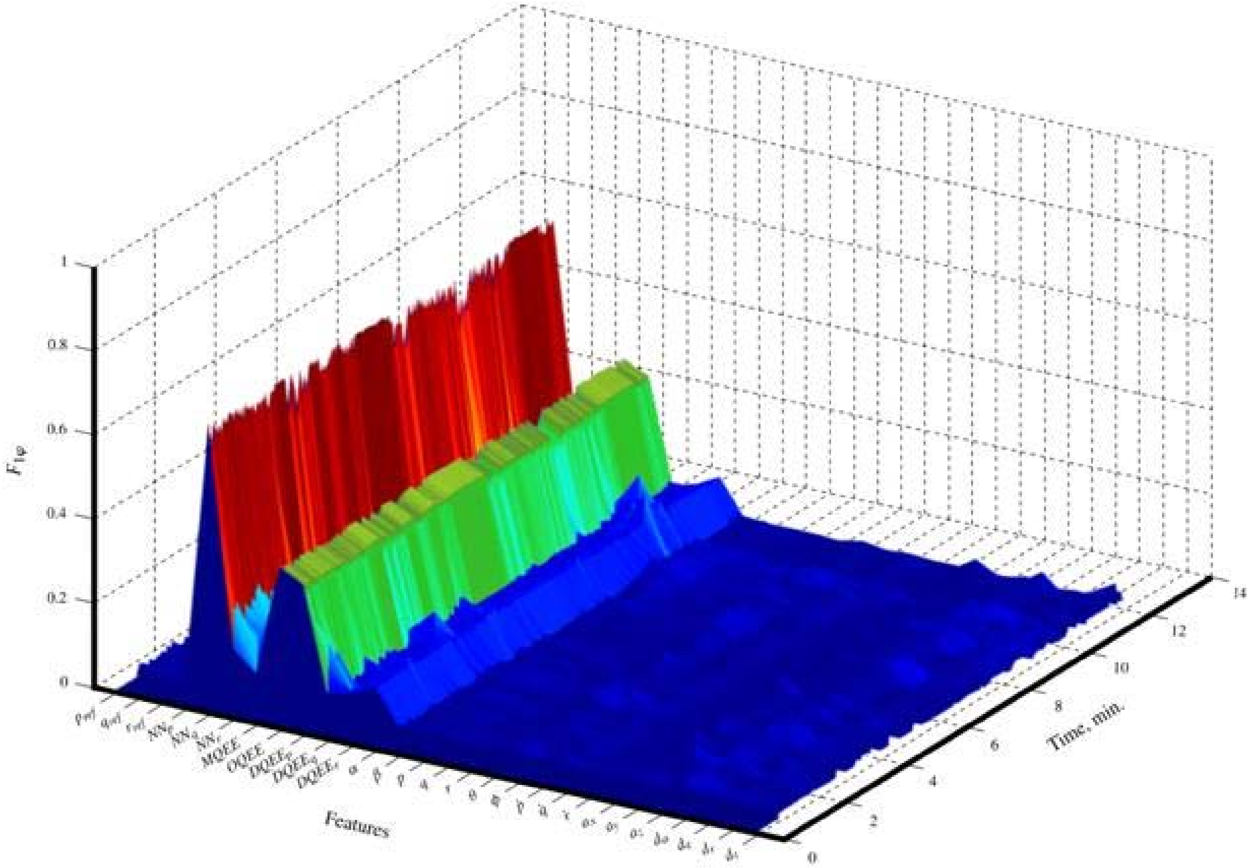


Figure 3.8. Variation of the Feature-Pattern Vector with Time of a Failed Right Wing Subsystem

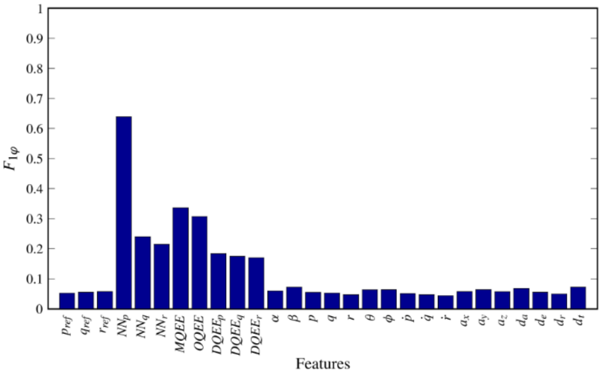


Figure 3.9. Reference Features-Pattern of a Failed Right Wing

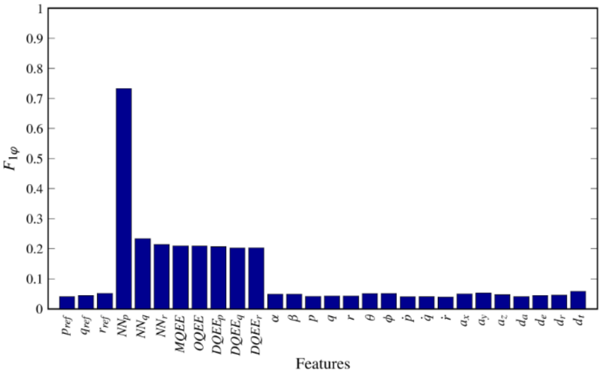


Figure 3.10. Reference Features-Pattern of a Failed Left Aileron

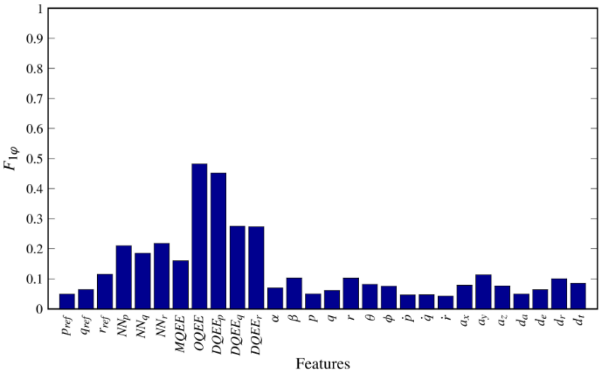


Figure 3.11. Reference Features-Pattern of a Failed Yaw Rate Sensor

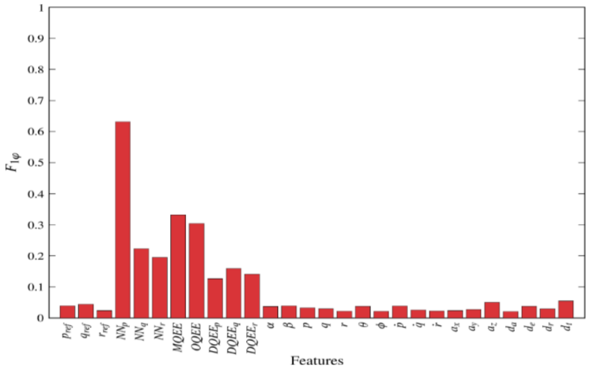


Figure 3.12. Sample Current Feature-Pattern of a Failed Right Wing

Table 3.5. Average Identification Rates Using the DCMechanism

Sub-system	Left Stab.	Right Stab.	Left Aileron	Right Aileron	Left Rudder	Right Rudder	Left Wing	Right Wing	Roll Rate	Pitch Rate	Yaw Rate
Left Stab.	**100.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Right Stab.	0.81	**99.13**	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.01	0.00
Left Aileron	0.00	0.00	**99.78**	0.14	0.01	0.00	0.04	0.00	0.03	0.00	0.00
Right Aileron	0.00	0.00	0.00	**99.89**	0.08	0.00	0.00	0.00	0.01	0.00	0.02
Left Rudder	0.00	0.00	0.00	0.00	**99.95**	0.05	0.00	0.00	0.00	0.00	0.00
Right Rudder	0.00	0.00	0.00	0.00	0.04	**99.96**	0.00	0.00	0.00	0.00	0.00
Left Wing	0.00	0.00	0.00	0.00	0.00	0.00	**99.95**	0.05	0.00	0.00	0.00
Right Wing	0.00	0.00	0.00	0.00	0.00	0.00	0.02	**99.98**	0.00	0.00	0.00
Roll Rate	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	**99.99**	0.00	0.00
Pitch Rate	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**	0.00
Yaw Rate	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	**99.99**

Chapter 4

Immunity-based Direct Evaluation of Aircraft Subsystem Abnormal Conditions

AC direct evaluation consists of two parts: qualitative direct evaluation and quantitative direct evaluation. The qualitative evaluation refers to determining the nature or the type of the AC, while the quantitative direct evaluation refers to determining the severity of the AC or the actual values of parameters that are a direct consequence of the AC. The AC direct evaluation within the AIS paradigm can be approached in a similar way as the identification phase.

4.1. Structured Non-self Approach for Abnormal Condition Direct Evaluation

The Structured Non-self approach can be used not only to perform general AC identification, but also a qualitative and quantitative AC evaluation within an integrated logical framework, such that the overall algorithm is less complex. This novel approach intends to reduce the computational processing for real time application of the solution to the ACFDIE problem. The proposed mapping-based positive selection-type of approach targets a multi-dimensional problem by means of a simpler but effective logic that can result in a more efficient real time algorithm. Within this method, the determination of the type and severity of an AC, also known as direct evaluation, is performed simultaneously with the identification stage as a single process, leaving the indirect evaluation as an individual stage in the SNSA. This architecture is presented in Figure 4.1.

4.1.1. Structured Non-self Approach for Qualitative Direct Evaluation

The qualitative evaluation can be performed in a manner similar to the identification by structuring the non-self and attaching to each detector d_j the tags dFt_j. This process involves the identification of detector sub-subsets within each subset corresponding to one single subsystem. As was the case with the identification process, prior generation of the non-self with adequate resolution and knowledge of AC characteristics is required. This concept is illustrated in Figure 4.2 for the 2-dimensional case with the assumption that the sub-sets dFt_j have only one element each.

If $Det_t = 1$, then the triggering detector is checked for structural parameters and a current qualitative evaluation outcome for the case when only one sub-system is affected can be defined as:

$$Ev1_t = \lfloor ev1_1 \quad ev1_2 \quad \cdots \quad ev1_{\overline{Nft}} \rfloor, \text{ with } \overline{Nft} = Nft_{dSys_j} \text{ and } \begin{matrix} ev1_i = 0 & for\ i \notin Ft_j \\ ev1_i = 1 & for\ i \in Ft_j \end{matrix} \tag{4.1}$$

Note that $Ev1_t$ can be formulated to consist of non-zero elements only, case in which:

$$Ev1_t = dFt_j \tag{4.2}$$

If several subsystems may be affected (that is $size(dSys_j) > 1$), then $Ev1_t$ from Eq. (4.1) becomes a matrix with $size(dSys_j)$ rows and $\max_{k=1,\dots N_s}(Nft_k)$ columns. Note that some rows of this matrix must be completed with zeros to account for the different values of Nft_k.

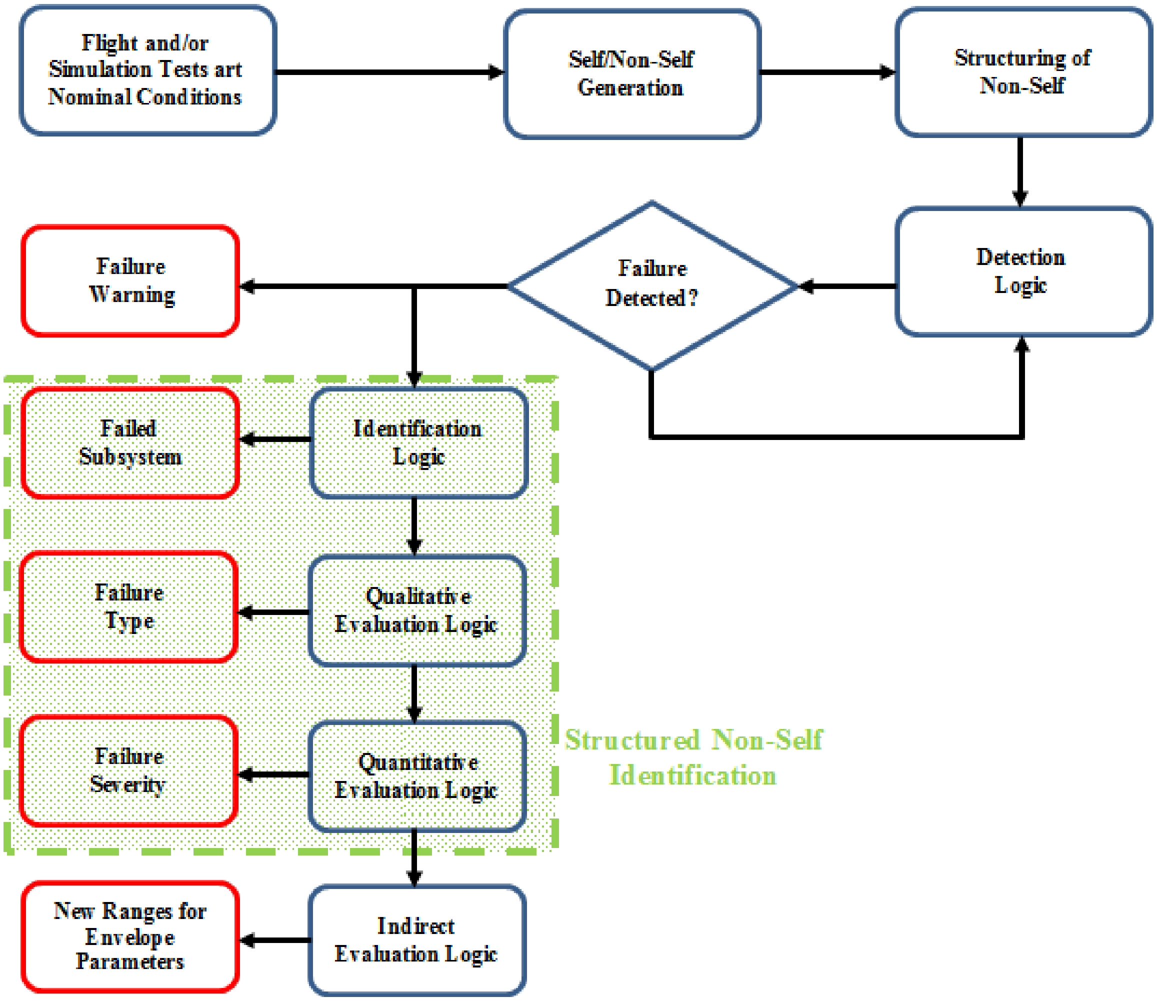

Flight and/or Simulation Tests art Nominal Conditions
Self/Non-Self Generation
Structuring of Non-Self
Detection Logic
Failure Detected?
Failure Warning
Identification Logic
Failed Subsystem
Qualitative Evaluation Logic
Failure Type
Quantitative Evaluation Logic
Failure Severity
Structured Non-Self Identification
Indirect Evaluation Logic
New Ranges for Envelope Parameters

Figure 4.1. Integration of Direct Evaluation and Identification Within SNSA

If one N-dimensional non-self is used with perfect feature definition, then the qualitative evaluation outcome can be defined as:

$$Ev1 = Ev1_t \tag{4.3}$$

If the HMS strategy is used, then Eq. (4.1) or (4.2) is applied to each sub-self and a composition logic $\overline{\mathscr{C}}$ must be designed to obtain the current qualitative evaluation outcome as:

$$Ev1_t = \overline{\mathscr{C}}(Ev1_{ti}), \qquad i = 1,2,\dots,N_{ss} \tag{4.4}$$

then Eq. (4.3) can be applied. However, the current qualitative evaluation outcomes $Ev1_t$ can be further processed over a moving time window to improve performance. As was the case with the identification process, structuring of the non-self can be avoided within the HMS strategy if an adequate composition logic $\overline{\mathscr{C}}$ can be formulated.

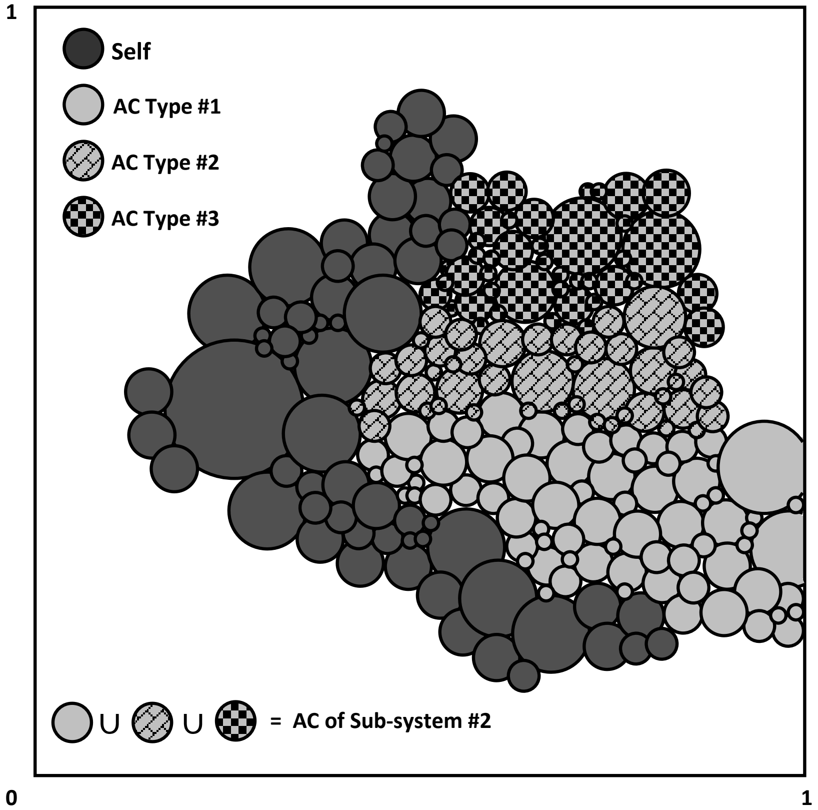


Figure 4.2. Failure Qualitative Evaluation Using the Structured Non-Self Approach

4.1.2. Structured Non-self Approach for Quantitative Direct Evaluation

The direct quantitative evaluation relies on correct qualitative evaluation and must be approached in different ways depending on the specific definition of the process outcome. If a definition of $Ev2$ as illustrated by Eq. (1.8) is used, then the dFs_j tags in Eq. (1.29) must represent the values of some ad-hoc defined metric that can capture the severity of the failure. For example, for many types of failure, the minimum distance from the center of the detector to the nearest self cluster can be used as illustrated in Figure 4.3. If a definition illustrated by Eq. (1.9) or (1.10) is used instead, then the dFs_j tags represent quantitative attributes (for Eq. (1.9)) or membership function values associated to a set of quantitative attributes (for Eq. (1.10)), as illustrated in Figure 4.4. In any case, the quantitative evaluation outcome *Ev2* must correspond to *Ev1*, that is for each scalar element of *Ev1* there will be a scalar element of *Ev2* if Eq. (1.8) or (1.9) are used, or a vector element if Eq. (1.10) is used. Therefore, at each sampling time *t*:

$$Ev2_t = dFs_j \tag{4.5}$$

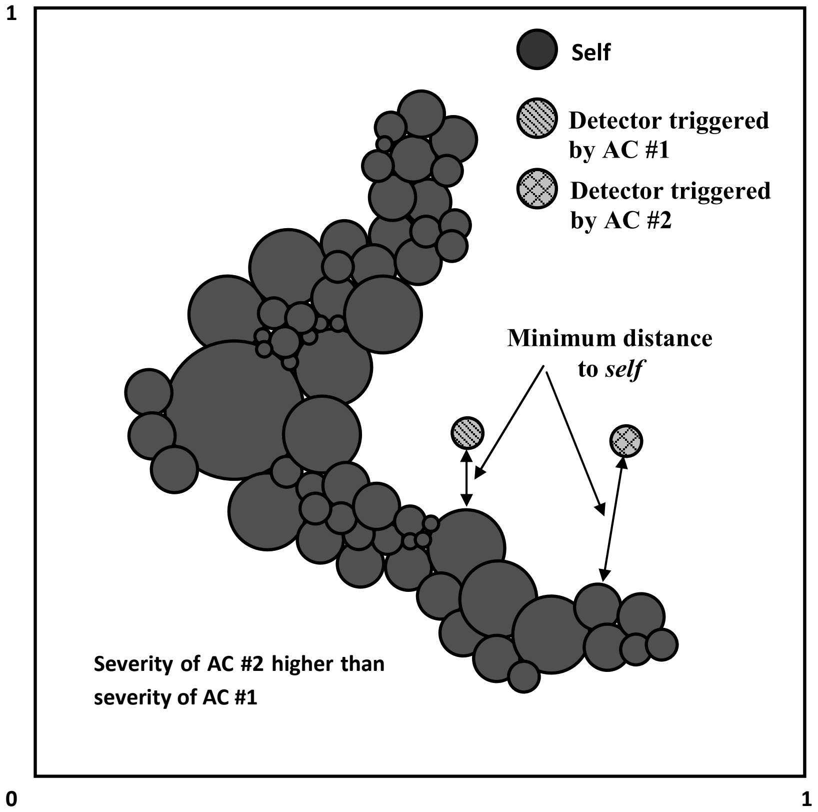

1
Self
Detector triggered by AC #1
Detector triggered by AC #2
Minimum distance to *self*
Severity of AC #2 higher than severity of AC #1
0
1

Figure 4.3. Failure Direct Quantitative Evaluation Based on Distance to Self

If one *N*-dimensional non-self is used with perfect feature definition, then the overall direct quantitative evaluation outcome can be defined as:

$$Ev2 = Ev2_t \tag{4.6}$$

If the HMS strategy is used, then Eq. (4.6) is applied to each sub-self and a composition logic $\tilde{\mathscr{C}}$ must be designed to obtain the current qualitative evaluation outcome as:

$$Ev2_t = \tilde{\mathscr{C}}(Ev2_{ti}), \qquad i = 1,2,\ldots,N_{ss} \tag{4.7}$$

then Eq. (4.6) can be applied. As in the previous phases, the current outcomes $Ev2_t$ can be further processed over a moving time window to improve performance.

It should be noted that associating type and/or severities of ACs to regions of the non-self, that is to specific detectors (they become thus "evaluators") requires tests at abnormal conditions of different types and severities and identification of the corresponding areas in the non-self (detectors/identifiers). The approach may prove resource prohibitive for a comprehensive solution. The distance-to-self approach for severity assessment, on the other hand, does not require extensive specific tests for development; however, distance thresholds must be established for the discrete set of linguistic values typically used for direct quantitative evaluation.

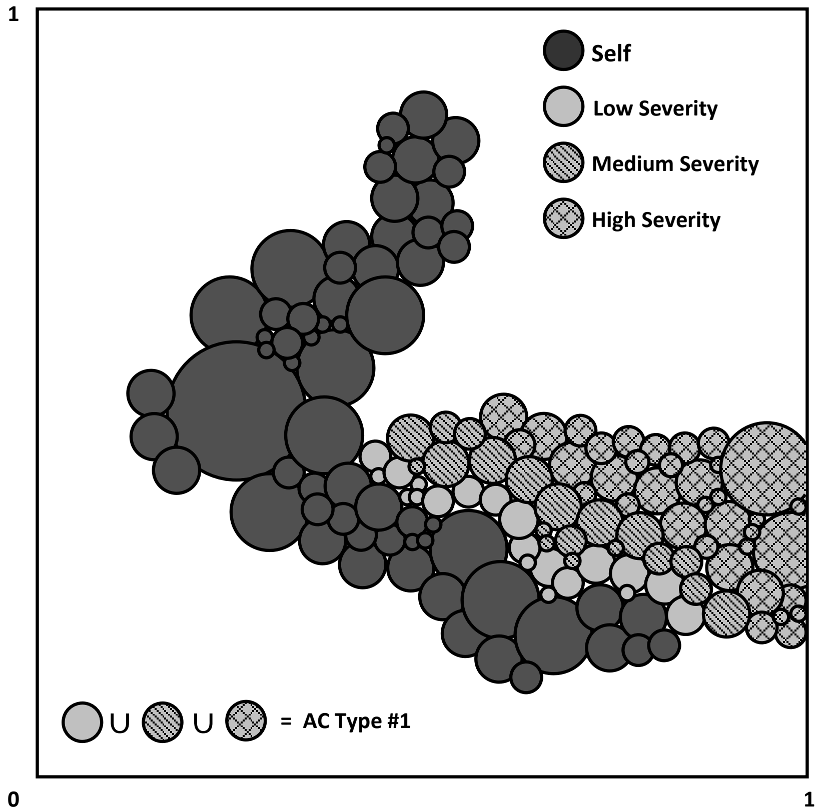


Figure 4.4. Failure Direct Quantitative Evaluation Using the Structured Non-Self Approach

4.1.3. Performance of Direct Evaluation Using the Structured Non-self Approach

The identification of the affected system, the determination of the type of AC, and its severity have been performed simultaneously using the SNSA. The performance evaluation was performed using similar metrics as for the detection phase. A correct identification percentage and a miss-identification percentage are calculated depending on an accurate declaration of subsystem failure for every time step in which an upset condition is present. The identifier generation algorithm proposed in this research effort was implemented for 9 different failures considered to be high magnitude using the 183 selected projections. Based on the assumption that lower magnitude failures of the same type of failure generate similar dynamic fingerprints with a close proximity to the self, the set of identifiers was subdivided into two groups. The first set corresponds to high magnitude and the second set to low magnitude failures (i.e. closer to the self). This approach increases the total amount of equivalent AC that can be identified to 18 instead of the original 9. A total of 1647 different cases for identifier generation were implemented in order to cover all the possible failure outputs investigated.

Each set of identifiers generated per failure contains on average 36 identifiers. Considering that every set of identifiers for all failures is integrated into each projection, an approximate total of 324 identifiers per projection are used for the identification through this positive selection-type of process. After an initial analysis, the algorithm was optimized and it was determined that out of the 183 projections, a total of 93 projections were enough to correctly identify the investigated failures. The reduction of the total number of projections required for identification has reduced the computational complexity of the algorithm considerably. Table 4.1 presents a sample set of projections used for AC identification and direct evaluation.

Table 4.1. Projections Used for Simultaneous Identification and Direct Evaluation

Self#	**Features**		**Self#**	**Features**		**Self#**	**Features**		**Self#**	**Features**	
3	p_{ref}	NN_p	**57**	r_{ref}	NN_q	**110**	NN_q	$DQEEp$	**121**	NN_q	φ
4	p_{ref}	NN_q	**60**	r_{ref}	$OQEE$	**111**	NN_q	$DQEEq$	**123**	NN_q	ax
7	p_{ref}	$OQEE$	**82**	NN_p	NN_q	**113**	NN_q	v	**124**	NN_q	ay
9	p_{ref}	$DQEEq$	**83**	NN_p	NN_r	**114**	NN_q	α	**125**	NN_q	az
31	q_{ref}	NN_q	**84**	NN_p	$MQEE$	**115**	NN_q	β	**126**	NN_q	d_a
34	q_{ref}	$OQEE$	**86**	NN_p	$DQEEp$	**116**	NN_q	p	**127**	NN_q	d_e
35	q_{ref}	$DQEEp$	**87**	NN_p	$DQEEq$	**117**	NN_q	q	**128**	NN_q	d_r
42	q_{ref}	q	**88**	NN_p	$DQEEr$	**118**	NN_q	r	**129**	NN_q	d_T
56	r_{ref}	NN_p	**107**	NN_q	NN_r	**120**	NN_q	θ	**130**	NN_q	M

Further analysis was carried out to reduce the number of projections required to produce desirable identification outputs. In some cases, the use of a single projection was enough to obtain favorable identification rates. On the other hand, other failures require more projections in order to obtain desirable identification results and also to reduce misidentification rates. Table 4.2 presents the number of projections required for a correct failure identification output.

Table 4.2. Total Number of Projections Used for Simultaneous Identification and Direct Evaluation

Failure #	**Failure Type**	**Projections Used**	**Failure #**	**Failure Type**	**Proj. Used**
1	Left Aileron Stuck at 2deg	14	9	Left Wing Loss of 6%	2
2	Right Aileron Stuck at 2deg	7	10	Right Wing Loss of 6%	1
3	Left Aileron Stuck at 8deg	8	11	Left Wing Loss of 15%	1
4	Right Aileron Stuck at 8deg	8	12	Right Wing Loss of 15%	2
5	Left Stabilator Stuck at 2deg	18	13	Left Engine Out	1
6	Right Stabilator Stuck at 2deg	2	14	Right Engine Out	18
7	Left Stabilator Stuck at 8deg	9	15	Roll Sensor Bias of 5deg/sec	1
8	Right Stabilator Stuck at 8deg	31	16	Roll Sensor Bias of 10deg/sec	7

The identification and direct evaluation algorithm was tested under 16 different failures (refer to Table 4.2). Table 4.3 shown below presents the results for the performance rate analysis. Note that in Table 4.3 the first column represents the actual failure and the corresponding row represents how it was identified/evaluated. For example, failure #1 is identified correctly 99.7% of the time but presents confusion with failures 3, 4 and 14 for 0.1% of the time, respectively.

Table 4.3. Performance of AC Identification and Direct Evaluation

		Identified Failure #															
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Actual Failure #	**1**	**99.7**	0	0.1	0.1	0	0	0	0	0	0	0	0	0	0.1	0	0
	2	9.9	**87.3**	0	0	0	0	0	0	0	0	0	0	0	2.8	0	0
	3	0.4	0	**95.6**	0.6	0	0	0	0	0	0	0	0	0	3.4	0	0
	4	0	0	0.1	**97.2**	0	0	0	0	0	0	0	0	0.3	2.4	0	0
	5	0.5	1.7	1.5	0	**92.5**	1.5	1.1	0	0	0	0	0	0	0	1.2	0
	6	9.9	0	1.2	0	0	**86.8**	0	2.1	0	0	0	0	0	0	0	0
	7	0.2	1	0	0	0	0	**96.1**	1.5	0	0	0	1.2	0	0	0	0
	8	0.5	0	0	0	0	0	4.8	**93.8**	0.9	0	0	0	0	0	0	0
	9	1.2	2	0.2	0	0	0.1	0	0.9	**95.6**	0	0	0	0	0	0	0
	10	0	0	2.1	0	0	0	0	0	1.1	**94.5**	0	0	0	1.1	0	1.2
	11	0	0	0	0	0	0	0	0.1	7.6	0.1	**92.2**	0	0	0	0	0
	12	0	0	0	0	0	0	1.1	0	0	1.1	0	**97.5**	0.3	0	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**92.6**	7.4	0	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0.1	**99.9**	0	0
	15	0	0	0	0	0	0.3	0	0.2	0.3	0	0.2	0	2.1	1.3	**95.6**	0
	16	0	0	0	0.3	0	0	0	0	0	0	2.7	0	0	2.1	0	**94.9**

4.2. Dendritic Cell Mechanism for Abnormal Condition Direct Evaluation

4.2.1. Dendritic Cell Approach for Qualitative Direct Evaluation

The AC qualitative evaluation assumes that correct detection and identification have been performed. At this point it is known that a failure is affecting at least one sub-system *k*, $k = 1,2,\ldots,N_S$. With the DC approach, the AC qualitative evaluation is approached as a specific identification problem, using the same algorithms. While the AC identification presented previously is a pattern recognition problem, where one must distinguish between N_S different patterns each corresponding to one of the N_S sub-systems considered, the AC qualitative evaluation is equally a pattern recognition problem, where one must distinguish between N_{tfk} different types of failures corresponding to sub-system *k*. In a similar manner to the SNSA, the AC qualitative evaluation based on the DC mechanism may be viewed as an identification process where the target is the "AC", which has the type of the failure as defining element in association with the affected sub-system. Patterns for qualitative evaluation are similar to the ones defined for identification and rely on the F_1 matrix. Note that the F_0 matrix may also be introduced into the process. The flowchart of the process of establishing qualitative evaluation patterns is presented in Figure 4.5.

For each sub-system, N_{tfk} different patterns must be established, one associated to each of the AC considered.

As is the case with AC identification, three possible alternatives exist for the definition of the patterns:

Feature-pattern (FP) approach
Projection-pattern (PP) approach
Matrix-pattern (MP) approach

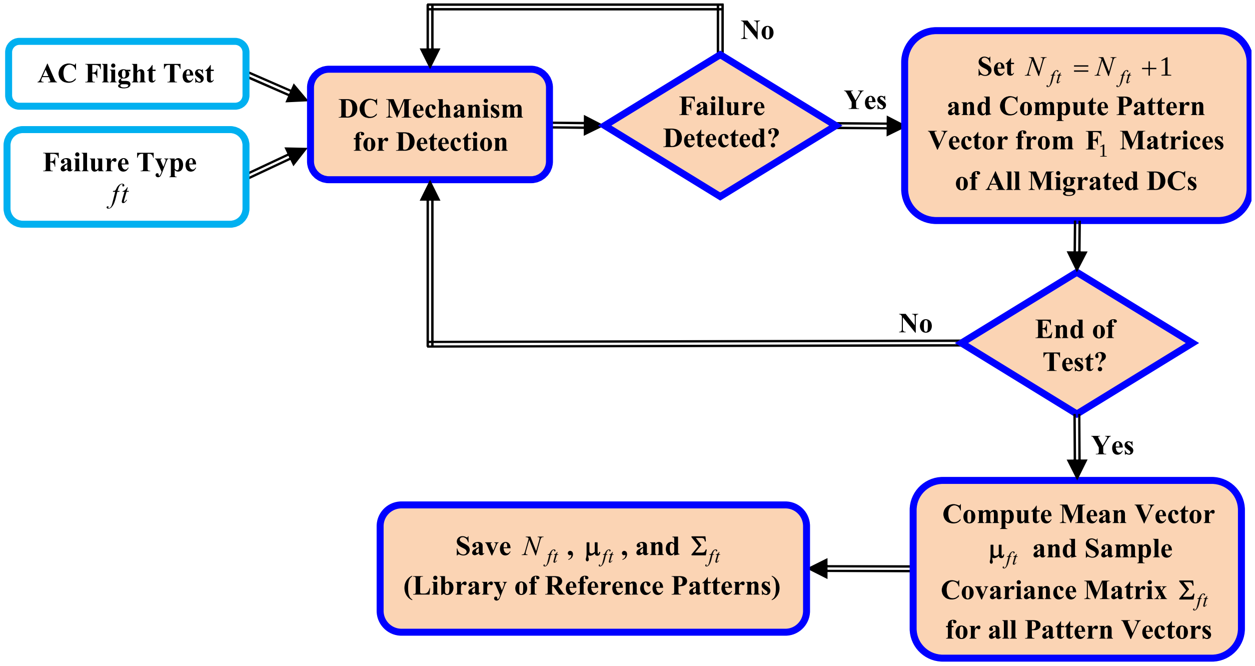

AC Flight Test
Failure Type
ft
DC Mechanism for Detection
Failure Detected?
No
Yes
Set $N_{ft} = N_{ft} + 1$ and Compute Pattern Vector from F_1 Matrices of All Migrated DCs
End of Test?
No
Yes
Compute Mean Vector μ_{ft} and Sample Covariance Matrix Σ_{ft} for all Pattern Vectors
Save N_{ft}, μ_{ft}, and Σ_{ft} (Library of Reference Patterns)

Figure 4.5. Generation of Library of Reference Patterns for Qualitative Evaluation Using the DC Mechanism and the Naïve Bayes Classifier

A). Feature-pattern (FP) approach for AC qualitative evaluation

For each sub-system *k*, define for all N_{ftk} ACs the vector of membership values of each feature to the set of equivalent directly involved variables (this is the same AC FP vector as described earlier):

$$FP_{ACj} = \begin{bmatrix} m_1 & m_2 & \dots & m_N \end{bmatrix}^T, \; m_i \in [0, \;\; 1], \;\; j = 1,2,\dots,N_{tfk} \tag{4.8}$$

All these membership values can be determined based on tests and heuristics. Note that binary logic can be used instead of fuzzy logic. In this case: $m_i \in \{0 \quad 1\}$. The N_{tfk} AC FP vectors form a library of patterns against which the current AC FP vector is compared to determine the AC type. The current AC FP vector is computed at each sample time, after an AC is detected, using the F_1 matrices of the mature DCs, as part of the identification process:

$$F_{1\varphi} = \left[\sum_{q=1}^{N_{DC}} F_{1q} \right] \cdot I_{N_{SS}}, \;\; I_{N_{SS}} \in \{1\}^{N_{SS} \times 1} \tag{4.9}$$

A matching algorithm must be used to determine which one of the N_{tfk} AC FP vectors best matches $F_{1\varphi}$. Note that AC qualitative can be performed before identification, with possibly some benefits (to be investigated).

B). Projection-pattern (PP) approach for AC qualitative evaluation

For each sub-system *k*, define for all N_{tfk} ACs the vector of membership values of each projection to the set of AC-relevant projections (projections that can capture the dynamic fingerprint of the AC, that are the projections that get triggered under AC):

$$PP_{ACj} = \begin{bmatrix} n_1 & n_2 & \dots & n_{N_{SS}} \end{bmatrix}^T, \; n_i \in [0, \;\; 1] \tag{4.10}$$

This is the same AC PP vector as for the previous method. All these membership values can be determined based on tests and heuristics. Binary logic may also be used here. At each sample time, after an AC is detected, use the F_1 matrices of the mature DCs to compute the current ACPP vector as:

$$F_{1P} = I_N \cdot \left[\sum_{q=1}^{N_{DC}} F_{1q} \right], \;\; I_N \in \{1\}^{1 \times N_{SS}} \tag{4.11}$$

A matching algorithm must be used to determine which one of the N_{tfk} AC PP vectors from the library best matches F_{1P}.

C). Matrix-pattern (MP) approach for AC qualitative evaluation

For each subsystem k, define for all N_{tfk} ACs the AC FP and the AC PP vectors as described previously. Build then the MP matrix as:

$$MP_{\bar{k}} = \begin{bmatrix} m^*_{11} & m^*_{12} & \cdots & m^*_{1N_{SS}} \\ m^*_{21} & m^*_{22} & \cdots & m^*_{2N_{SS}} \\ \vdots & \vdots & \ddots & \\ m^*_{N1} & m^*_{N2} & \cdots & m^*_{NN_{SS}} \end{bmatrix}, \tag{4.12}$$

where $m^*_{ij} \in [0, \;\; 1], \;\; i = 1,2,\ldots,N \,, \;\; j = 1,2,\ldots,N_{SS} \,, \; \bar{k} = 1,2,\ldots,N_{ftk}$ (4.13)

and
$$m^*_{ij} = \begin{cases} max(m_i, n_j) & if \;\; \varphi_i belongs \; to \; S_j \\ 0 \quad otherwise \end{cases} \tag{4.14}$$

At each sample time, after the AC is detected, compute the sum of all mature DC F_1 matrices as:

$$F_{1M} = \sum_{q=1}^{N_{DC}} F_{1q} \,, \quad q = 1,2,\ldots,N_{DC} \tag{4.15}$$

Use a matching algorithm to determine which one of the N_{tfk} MP matrices in the library best matches F_{1M}.

Similarly to the SNSA, the AC identification and qualitative evaluation can be performed simultaneously by redefining ACs to include both affected subsystem and AC type as category designations. The pattern generation process for the joint approach is summarized in Figure 4.6.

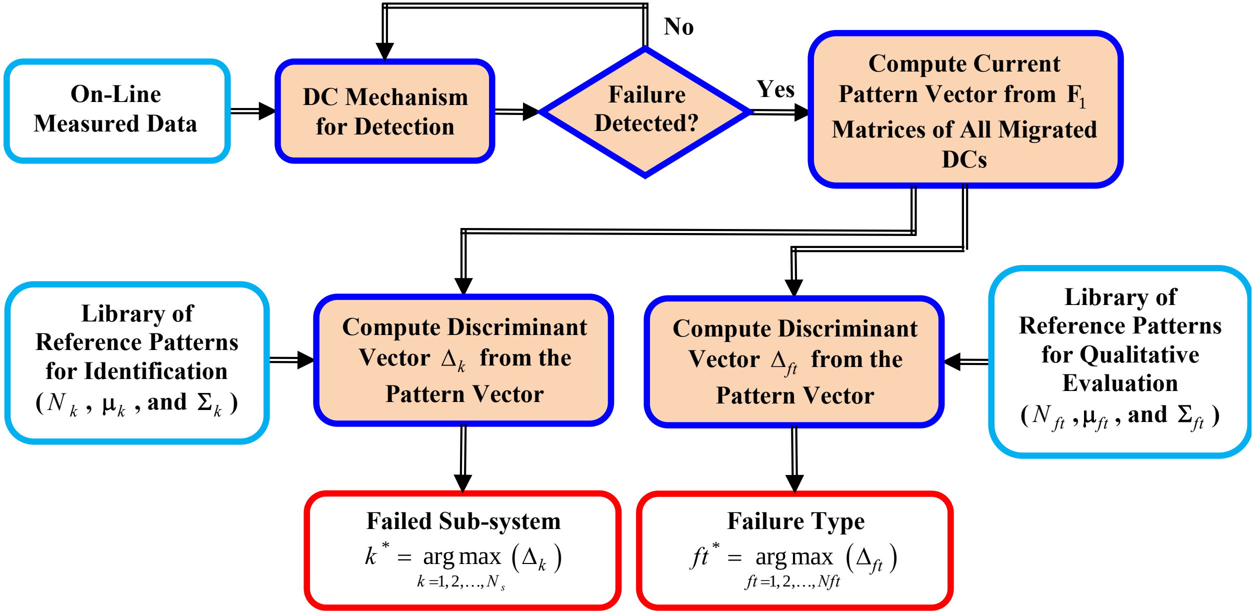

No
On-Line Measured Data
DC Mechanism for Detection
Failure Detected?
Yes
Compute Current Pattern Vector from F_1 Matrices of All Migrated DCs
Library of Reference Patterns for Identification (N_k, μ_k, and Σ_k)
Compute Discriminant Vector Δ_k from the Pattern Vector
Compute Discriminant Vector Δ_{ft} from the Pattern Vector
Library of Reference Patterns for Qualitative Evaluation (N_{ft}, μ_{ft}, and Σ_{ft})
Failed Sub-system
$k^* = \underset{k=1,2,\ldots,N_s}{\arg\max} (\Delta_k)$
Failure Type
$ft^* = \underset{ft=1,2,\ldots,Nft}{\arg\max} (\Delta_{ft})$

Figure 4.6. DC Mechanism for Simultaneous AC Identification and Qualitative Evaluation

4.2.2. Dendritic Cell Approach for Quantitative Direct Evaluation

Similar to the AC qualitative evaluation, patterns for each failure severity can be defined based on the detection outcome of all subselves based on the F1 matrices of the migrated DCs. If the number of severity scales of failure *ACj* is *Nfs*, then *Nfs different* reference patterns must be established: one associated to each severity scale. The reference F-P vector corresponding to the failure severity $AC_{j\varsigma}$ can be expressed as:

$$FP_{AC_{j\varsigma}} = \begin{bmatrix} \tilde{m}_1 & \tilde{m}_2 & \cdots & \tilde{m}_N \end{bmatrix}^T , \qquad (4.16)$$

where $\tilde{m}_i \in [0,1]$ are fuzzy membership values of each feature with respect to the set of directly involved variables. The reference F-P of Eq. (4.16) can be determined through a training process with experimental or simulation data under failure exposed to the artificial DC mechanism. The naïve Bayes classifier described previously can be used to construct these reference patterns by training the classifier offline against samples from a set of training tests of known failure severity scales. Training the classifier for simultaneous detection and qualitative evaluation is very similar to training it for AC qualitative evaluation as shown in the schematic diagram of Figure 4.7.

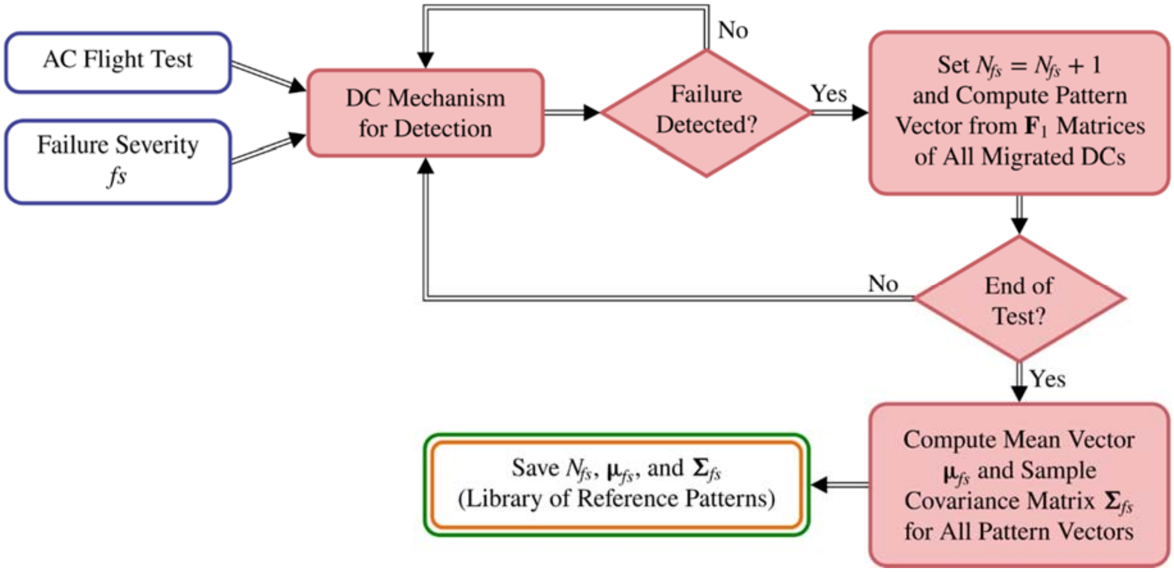

AC Flight Test
Failure Severity fs
DC Mechanism for Detection
No
Failure Detected?
Yes
Set $N_{fs} = N_{fs} + 1$ and Compute Pattern Vector from $\mathbf{F}_1$ Matrices of All Migrated DCs
No
End of Test?
Yes
Compute Mean Vector $\boldsymbol{\mu}_{fs}$ and Sample Covariance Matrix $\boldsymbol{\Sigma}_{fs}$ for All Pattern Vectors
Save N_{fs}, $\boldsymbol{\mu}_{fs}$, and $\boldsymbol{\Sigma}_{fs}$ (Library of Reference Patterns)

Figure 4.7. Training the Naïve Bayes Classifier for AC Detection and Qualitative Evaluation

The current features-pattern $F_{1\varphi}$ is computed and compared to the library of reference patterns to determine the discriminant $AC_{j\varsigma}$ for each severity scale from:

$$\mathit{\Delta}_{fs}(\mathbf{x}) = \ln N_{fs} - \frac{1}{2}\ln\left|\mathbf{\Sigma}_{fs}\right| - \frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_{fs}\right)^T \mathbf{\Sigma}_{fs}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_{fs}\right), \qquad (4.17)$$

where N_{fs} is the number of samples in tests with failure severity *fs*, $\mathbf{x} = F_{1\varphi}$, and $\boldsymbol{\mu}_{fs}$ and $\mathbf{\Sigma}_{fs}$ are the mean vector and sample covariance matrix given by:

$$\boldsymbol{\mu}_{fs} = \frac{1}{N_{fs}} \sum_{m=1}^{N_{fs}} \mathbf{x}_{fs}^{(m)}, \tag{4.18}$$

and
$$\boldsymbol{\Sigma}_{fs} = \frac{1}{N_{fs}-1} \sum_{m=1}^{N_{fs}} \left(\mathbf{x}_{fs}^{(m)} - \boldsymbol{\mu}_{fs}\right)\left(\mathbf{x}_{fs}^{(m)} - \boldsymbol{\mu}_{fs}\right)^{T}, \tag{4.19}$$

respectively. The block diagram of AC detection and qualitative evaluation scheme is presented in Figure 4.8.

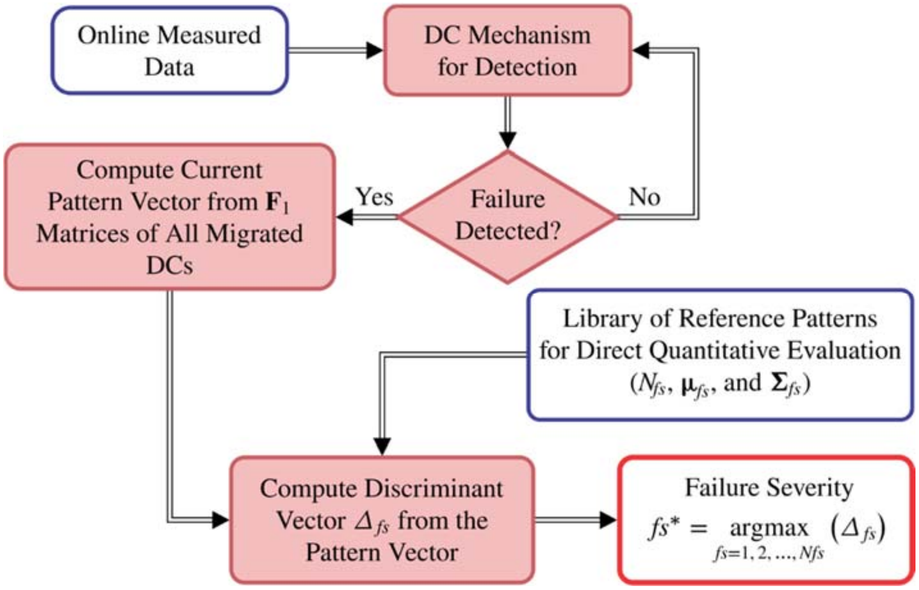

Online Measured Data
DC Mechanism for Detection
Failure Detected?
Yes
No
Compute Current Pattern Vector from $\mathbf{F}_1$ Matrices of All Migrated DCs
Library of Reference Patterns for Direct Quantitative Evaluation (N_{fs}, $\boldsymbol{\mu}_{fs}$, and $\boldsymbol{\Sigma}_{fs}$)
Compute Discriminant Vector Δ_{fs} from the Pattern Vector
Failure Severity $fs^* = \underset{fs=1,2,\ldots,Nfs}{\operatorname{argmax}} (\Delta_{fs})$

Figure 4.8. AC detection and Qualitative Evaluation Scheme Using the DC Mechanism

4.2.3. Performance of Direct Evaluation Using the Dendritic Cell Approach

The F-P approach was also used to implement the AC detection and qualitative evaluation scheme with the DC mechanism. The naïve Bayes classifier was trained using two sets of training tests under failure. The first training set provides the reference features-pattern FP_{fs} for each failure severity fs. Figures 4.9 through 4.14 illustrate the reference features patterns for subsystems under different levels of failure severity.

Table 4.4 presents the AC detection and qualitative evaluation rates for the four main subsystems considered in this study. The rate is defined as the number of time steps for which the failure severity is correctly evaluated divided by the total number of time steps after failure. The second column in this table represents the “known” failure severity of the corresponding subsystem, whereas the first row represents the evaluated failure severity. Note that only “low” and “high” severity levels were simulated for the sensors and engine failures. These results show the capability of the proposed scheme in isolating the failure severity with very high rates.

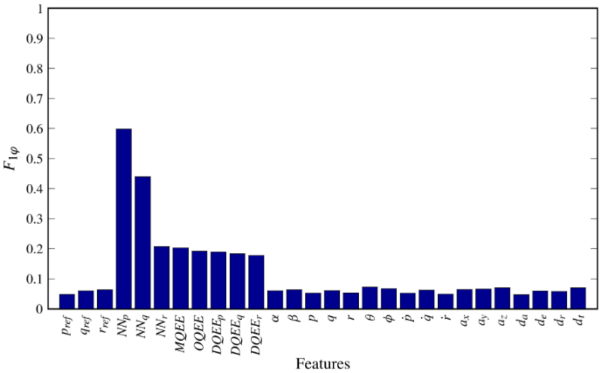


Figure 4.9. Reference Feature Patterns for a Stabilator Under Low Severity Failure

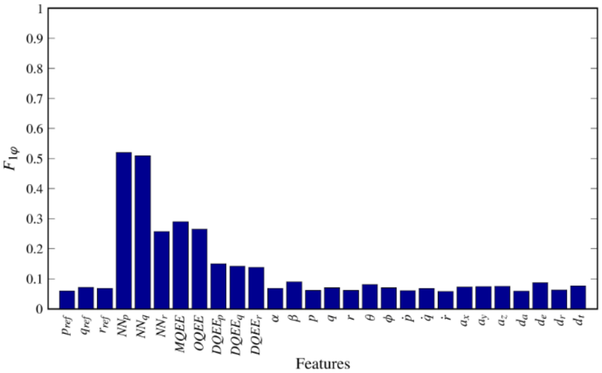


Figure 4.10. Reference Features-Pattern for a Stabilator Under High Severity Failure

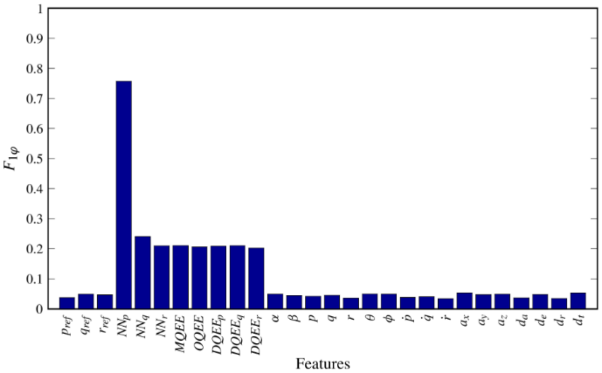


Figure 4.11. Reference Features-Pattern for a Wing Under Low Severity Failure

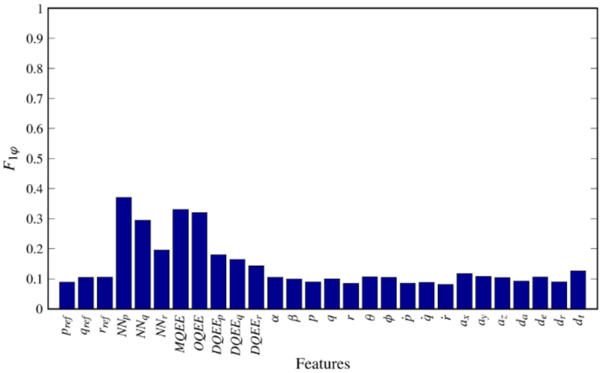


Figure 4.12. Reference Features-Pattern for a Wing Under High Severity Failure

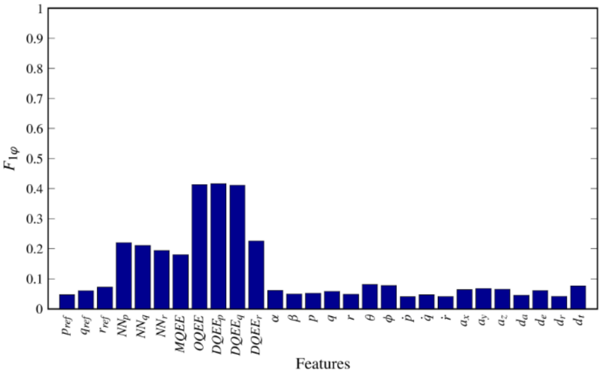


Figure 4.13. Reference Features-Pattern for a Roll Rate Sensor Under Low Severity Failure

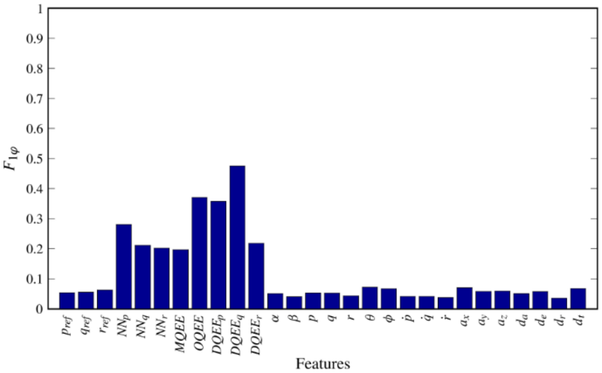


Figure 4.14. Reference Features-Pattern for a Roll Rate Sensor Under High Severity Failure

Table 4.4. AC Detection and Qualitative Evaluation Rates of the DC Mechanism Scheme

Subsystem	Failure Severity	Low	Medium	High
Stabilator	Low	98.72	1.28	0.00
	Medium	0.04	99.96	0.00
	High	0.05	0.00	99.95
Aileron	Low	100.00	0.00	0.00
	Medium	3.10	95.47	1.43
	High	0.11	0.00	99.89
Rudder	Low	100.00	0.00	0.00
	Medium	0.01	99.99	0.00
	High	0.00	0.00	100.00
Wing	Low	100.00	0.00	0.00
	Medium	0.01	99.99	0.00
	High	0.06	0.00	99.94
HorizontalTail	Low	100.00	0.00	0.00
	Medium	0.00	99.99	0.01
	High	0.01	0.00	99.99
VerticalTail	Low	100.00	0.00	0.00
	Medium	0.01	99.99	0.00
	High	0.00	0.00	100.00
Roll Rate Sensor	Low	99.99	—	0.01
	High	0.00	—	100.00
Pitch Rate Sensor	Low	100.00	—	0.00
	High	0.00	—	100.00
Yaw Rate Sensor	Low	100.00	—	0.00
	High	0.01	—	99.99
Engine	Low	100.00	—	0.00
	High	0.00	—	100.00

Chapter 5

Flight Envelope Estimation Under Abnormal Conditions Using Artificial Immune System

5.1. Problem Formulation

The AC indirect quantitative evaluation requires that the direct evaluation is successful. While for detection, identification, qualitative evaluation, and direct quantitative evaluation consistent general design algorithms can be applied as presented so far that are valid for a variety of systems and AC types without significant changes, the indirect quantitative failure evaluation process or altered flight envelope evaluation requires specific customization depending on the subsystem, nature of the AC, and the affected parameter of the flight envelope addressed by the evaluation. However, a general framework can be still formulated up to a point. A comprehensive indirect evaluation or altered flight envelope prediction must rely on a combined strategy based on analytical flight envelope assessment and AIS-based approaches for parameter space alteration assessment. The analytical methods require accurate modeling of the failures and significant on-line computational capabilities. The AIS method implies that all pertinent parameters to the flight envelope – considering its generalized meaning – are part of the feature sets that define the "self".

Let us assume that the self S and non-self $\hat{S}$ are defined as sets of N-dimensional hyperspheres according to Eq. (1.27) and (1.28). Some main concepts previously defined for the AC evaluation process will be repeated here for completeness. A ***directly involved variable*** (DIV) in the AC is a variable whose alteration or abnormal variation is directly and significantly the result of the AC. Typically, DIV are used to define/characterize the AC. They may be part of the feature set or not. If they are not, then a relationship between the DIV and some other variable(s) in the feature set must be established. This process will define ***equivalent directly involved variables*** (EDIV), which must be part of the feature set. For example, consider the case of the left stabilator locked failure. The DIV can be defined as the left stab deflection δ_{eL}. It obviously defines the failure, but let us assume that it is not part of the feature set. A relationship between the left stab deflection δ_{eL} and the longitudinal stick displacement d_e, which is a feature, can be established:

$$d_e = f(\delta_{eL}) \tag{5.1}$$

Therefore, the EDIV is in this case d_e. The self can be viewed as a generalized flight envelope based on features φ_i. Assume that each failure $f_i \in F$, $i = 1,2,\ldots,N_F$, produces a set of $N_{\Gamma i}$ constraints Γ_i on a set of known variables X_i, where:

$$\Gamma_i = \{\gamma_{i1} \quad \gamma_{i2} \quad \ldots \quad \gamma_{iN_{\Gamma i}}\} \tag{5.2}$$

$$\gamma_{ij} = \gamma_{ij}(X_i) = 0 \,,\; i = 1,2,\ldots,N_F \text{ and } j = 1,2,\ldots,N_{\Gamma i} \tag{5.3}$$

The variables X_i must be part of the feature set, that is:

$$X_i = \{x_{i1} \quad x_{i2} \quad \ldots \quad x_{iN_X}\} \subset \Im \tag{5.4}$$

Let us assume that the alteration of the flight envelope is assessed in terms of a set $Y \subset \Im$ of N_Y variables y_i, that is:

$$Y = \{y_1 \quad y_2 \quad \ldots \quad y_{N_Y}\} \subset \Im \tag{5.5}$$

Typically the constraints Γ_i are not completely known, otherwise they would specify any alteration of the variables y_i and the problem would be solved. The solution is equivalent to determining a new self at post-failure conditions based on a set of features *Y*. If $Y = \Im$ then a solution is found if perfect structuring of the non-self is possible, which in the most general case is not. If $N_Y < N$, then the flight envelope alteration assessment is equivalent to obtaining the N_Y-dimensional projection of the *N*-dimensional non-self region corresponding to the targeted failure f_i. Again, if structuring of the non-self for f_i is available, then the problem is solved. If not, we can still obtain the N_Y-dimensional projection of the self. In many cases, depending on the failure, the constraints Γ_i, and the set *Y* we can infer the relative position of the N_Y-dimensional projections of the non-self and the self. For example, let us consider a stuck elevator failure (the left or right elevator is locked at a constant deflection δ_{e0}). The effects on aircraft pitch rate *q* must be assessed. Therefore, $x_1 = \delta_e$, $y_1 = q$, and $\gamma_1 \equiv \delta_e - \delta_{e0} = 0$. Due to the nature of this failure, it can be inferred that the values of the pitch rate at post failure conditions will be in the range of normal conditions corresponding to δ_{e0}. In other words, the projection of the non-self on the 2-dimensional plane (x_1, y_1) falls inside the projection of the self. For such situations, a “new” self $\bar{S}$ or reduced envelope at post failure conditions can then be defined as:

$$\bar{S} = \{c_i \mid c_i \text{ satisfies constraints } \Gamma_i\} \tag{5.6}$$

The concept is illustrated in Figure 5.1 for the 2-dimensional case.

The development of schemes for indirect quantitative failure evaluation (IQE) is based on Table 1.2. Specific algorithms must be developed for all targeted variables ($N_E = 16$) in conjunction with the nature and type of the failure.

It is assumed that failure detection, identification, and direct evaluation are successfully completed at this point. Therefore, the sets of DIV (v_δ), EDIV (v_ε), and envelope relevant variables (v_E) are determined. Note that, in general:

$$\begin{aligned}
v_\delta &= \lfloor v_{\delta 1} \quad v_{\delta 2} \quad \cdots \quad v_{\delta N_\delta} \rfloor \\
v_\varepsilon &= \lfloor v_{\varepsilon 1} \quad v_{\varepsilon 2} \quad \cdots \quad v_{\varepsilon N_\varepsilon} \rfloor, \quad v_\varepsilon \subset \Im \\
v_E &= \lfloor v_{E1} \quad v_{E2} \quad \cdots \quad v_{EN_E} \rfloor, \quad v_E \subset \Im_E \subset \Im
\end{aligned} \tag{5.7}$$

For IQE, a relationship between v_ε and v_δ must be known:

$$v_\varepsilon = f_{\varepsilon\delta}(v_\delta) \tag{5.8}$$

where

$$f_{\varepsilon\delta} = \left\lfloor f_{\varepsilon\delta 1} \quad f_{\varepsilon\delta 2} \quad \dots \quad f_{\varepsilon\delta N_\varepsilon} \right\rfloor \tag{5.9}$$

such that

$$v_{\varepsilon i} = f_{\varepsilon\delta i}(v_\delta), \quad i = 1,2,\dots,N_\varepsilon \tag{5.10}$$

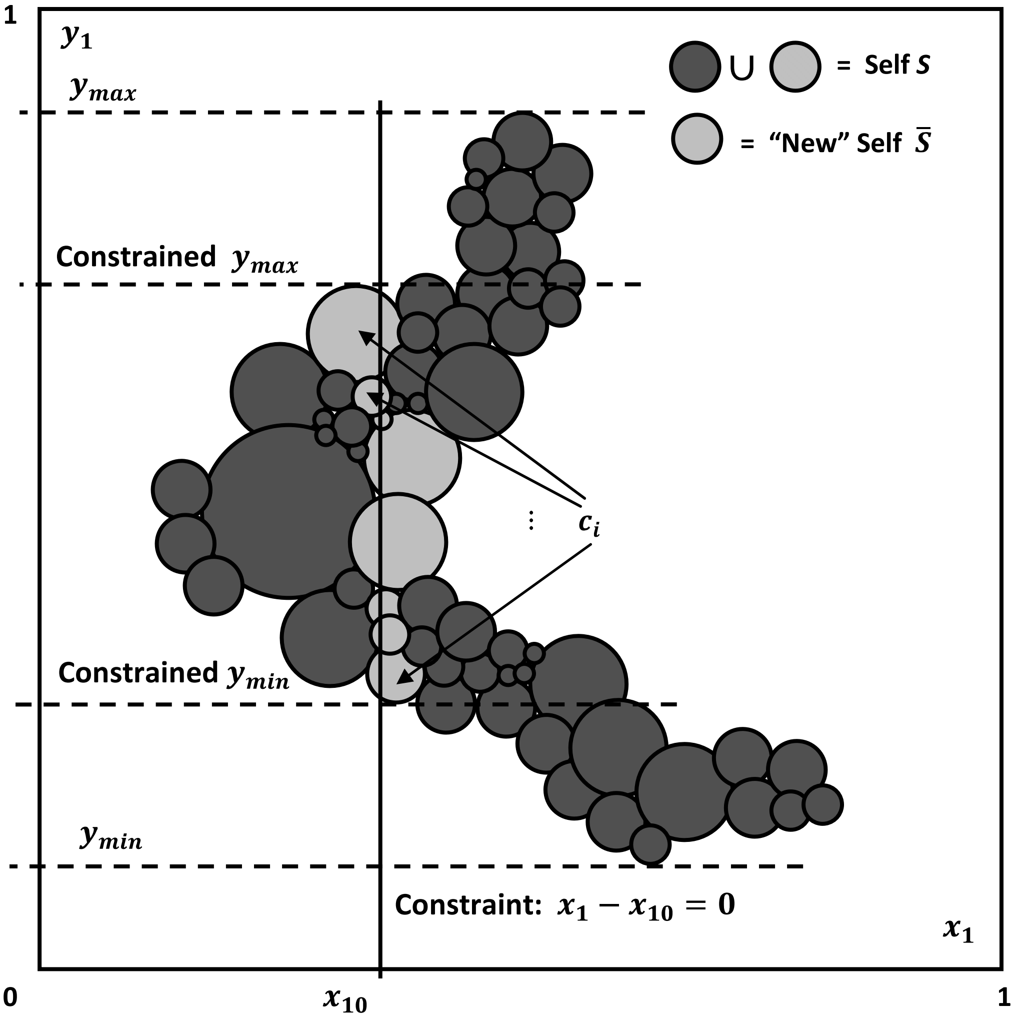


Figure 5.1. Post-Failure Flight Envelope Estimation Using the AIS Paradigm

5.2. Estimation Algorithms for Actuator AC

5.2.1. Stabilator Failure

This discussion covers failures $k_F = 1$ through $k_F = 4$ of Table 1.2, since, from the point of view of the IQE, they are equivalent. The discussion will continue with reference to the left stab; however, everything applies similarly to the right surface failure as well. For $k_F = 1 \div 4$, the following parameters can be specified:

$$\begin{aligned}
&v_\delta = [v_{\delta 1}] = [\delta_{eL}], \text{ left (or right) stab deflection, positive downwards} \\
&v_\varepsilon = [v_{\varepsilon 1}] = [d_e], \text{ longitudinal stick} \\
&v_E = [a_x \quad a_z \quad H \quad M \quad V \quad q \quad p \quad \dot{q} \quad \alpha \quad \theta]
\end{aligned} \tag{5.11}$$

The relationship between DIV and EDIV is:

$$d_e = f_{\varepsilon\delta 1}(\delta_{eL}) \tag{5.12}$$

where

$$d_e = k_e \delta_e = k_e \frac{\delta_{eL} + \delta_{eR}}{2} \tag{5.13}$$

If the left stab is locked, then:

$$\delta_{eL}(t) = \delta_{eLF} \tag{5.14}$$

where δ_{eLF} is a value determined through direct quantitative evaluation. The nominal range for d_e is (typically $d_{e\,max} = -d_{e\,min}$):

$$d_{e\,max} = k_e \frac{\delta_{eL\,max} + \delta_{eR\,max}}{2} \tag{5.15}$$

$$d_{e\,min} = k_e \frac{\delta_{eL\,min} + \delta_{eR\,min}}{2} \tag{5.16}$$

After failure, the range for d_e becomes:

$$d_{e\,maxF} = k_e \frac{\delta_{eLF} + \delta_{eR\,max}}{2} \tag{5.17}$$

$$d_{e\,minF} = k_e \frac{\delta_{eLF} + \delta_{eR\,min}}{2} \tag{5.18}$$

Note that this is a rather virtual or equivalent range, since the stick in the cockpit is assumed to still be capable of moving within the nominal range, only the effects are different. Note also that the range at post failure conditions ($\left[d_{eminF},\ \ d_{emaxF}\right]$) is invariable with respect to the surface failed (L or R) and that $\left[d_{eminF},\ \ d_{emaxF}\right] \subset \left[d_{emin},\ \ d_{emax}\right]$. With the new range after failure, new ranges for all v_E may be obtained from the 2-D projections, if there is a mutual constraint (cause/effect relationship) between the two features involved in the projection. If there is not, it means that one variable will reach all its values in the nominal range irrespective of the other variable values and the self projection will be a square. In this cases, if it is known that the variable involved is part of v_E, special consideration must be given. For example, a locked stab failure will have an effect on the capability to produce rolling rate. The asymmetry induced by a locked stab failure produces undesired roll moments and rates that must be typically compensated by aileron deflections. These deflections affect the control authority on the roll channel and thus the production of roll rate. This effect is rather indirect as the self projection defined by d_e and p is a square. To determine the effects on p produced by a locked stab failure, it will be stipulated that the locked stab failure induces an aileron pseudo-failure, from the point of view of producing roll rate. This pseudo-failure involves a non-zero aileron trim deflection at a position δ_{aPF} necessary to compensate for the rolling effect of the locked stab failure. The compensating aileron deflection can be approximated using the balance of the rolling moment:

$$C_{l\delta_{eL}} \cdot \delta_{eLF} + C_{l\delta_{eR}} \cdot \delta_{eR} = -C_{l\delta_a} \cdot \delta_{aPF} \tag{5.19}$$

$$C_{l\delta_{eL}} \cdot (\delta_{eLF} - \delta_{eR}) = -C_{l\delta_a} \cdot \delta_{aPF} \tag{5.20}$$

$$\delta_{aPF} = -\frac{C_{l\delta_{eL}}}{C_{l\delta_a}} (\delta_{eLF} - \delta_{eR}) \tag{5.21}$$

The following convention for the definition of the generic aileron deflection will be assumed, which is the one used for the aerodynamic model of the aircraft within the simulation environment:

$$\delta_a = \delta_{aL} - \delta_{aR} \tag{5.22}$$

With linearity assumption, the relationship with the EDIV is:

$$d_a = k_a \delta_a \tag{5.23}$$

The nominal lateral control range is:

$$d_{a\,max} = k_a \delta_{a\,max} \tag{5.24}$$

$$d_{a\,min} = k_a \delta_{a\,min} \tag{5.25}$$

Under stab failure, the new range for the lateral control is:

$$d_{a\,max\,F} = k_a(\delta_{a\,max} - \delta_{aPF}) \quad ..(5.26)$$

$$d_{a\,min\,F} = k_a(\delta_{a\,min} - \delta_{aPF}) \quad (5.27)$$

Note that this is equivalent to shifting the range of d_a in one direction or the other (depending on the sign of the stab failure), while maintaining the same span:

$$[d_{aminF},\ \ d_{amaxF}] \not\subset [d_{amin},\ \ d_{amax}] \quad (5.28)$$

$$d_{a\,max\,F} - d_{a\,min\,F} = d_{a\,max} - d_{a\,min} \quad (5.29)$$

Now the new range of d_a can be used with the (d_a, p) projection to determine the effects on p of the stab failure. Note that extrapolation on the projection is needed, since the new range ends up outside the old one. This range can be expressed in terms of the failed surface as follows:

$$d_{a\,max\,F} = k_a[\delta_{a\,max} + \frac{C_{l\delta_{eL}}}{C_{l\delta_a}}(\delta_{eLF} - \delta_{eR})] \quad (5.30)$$

$$d_{a\,min\,F} = k_a[\delta_{a\,min} + \frac{C_{l\delta_{eL}}}{C_{l\delta_a}}(\delta_{eLF} - \delta_{eR})] \quad (5.31)$$

If the right stab fails, then the range can be expressed as:

$$d_{a\,max\,F} = k_a[\delta_{a\,max} + \frac{C_{l\delta_{eR}}}{C_{l\delta_a}}(\delta_{eRF} - \delta_{eL})] \quad (5.32)$$

$$d_{a\,min\,F} = k_a[\delta_{a\,min} + \frac{C_{l\delta_{eR}}}{C_{l\delta_a}}(\delta_{eRF} - \delta_{eL})] \quad (5.33)$$

Note that $C_{l\delta_{eL}} = -C_{l\delta_{eR}}$. Therefore, the aileron pseudo-failure range is the same if the opposite stab is locked in the opposite direction relative to the trim. Span is preserved.

5.2.2. Aileron Failure

This discussion covers failures $k_F = 5$ through $k_F = 8$ of Table 1.2, since, from the point of view of the indirect quantitative evaluation, they are equivalent. The discussion will continue with reference to the left aileron; however, everything applies similarly to the right surface failure as well, unless otherwise noted. For $k_F = 5 \div 8$, one can specify:

$$\begin{aligned} &v_\delta = [v_{\delta 1}] = [\delta_{aL}],\ \text{left (or right) aileron deflection, positive downwards} \\ &v_\varepsilon = [v_{\varepsilon 1}] = [d_a],\ \text{lateral stick} \\ &v_E = [a_y \quad p \quad r \quad \dot{p} \quad \dot{r} \quad \beta \quad \varphi] \end{aligned} \tag{5.34}$$

The relationship between DIV and EDIV is:

$$d_a = f_{\varepsilon\delta 1}(\delta_{aL}) \tag{5.35}$$

$$d_a = k_a(\delta_{aL} - \delta_{aR}) \tag{5.36}$$

If the left aileron is locked, then:

$$\delta_{aL}(t) = \delta_{aLF} \tag{5.37}$$

where δ_{aLF} is a value determined through direct quantitative evaluation. The nominal range for d_a is (typically $d_{a\,max} = -d_{a\,min}$):

$$d_{a\,max} = k_a(\delta_{aL\,max} - \delta_{aR\,min}) \tag{5.38}$$

$$d_{a\,min} = k_a(\delta_{aL\,min} - \delta_{aR\,max}) \tag{5.39}$$

After failure, the range for d_a becomes:

$$d_{a\,max\,F} = k_a(\delta_{aLF} - \delta_{aR\,min}) \tag{5.40}$$

$$d_{a\,min\,F} = k_a(\delta_{aLF} - \delta_{aR\,max}) \tag{5.41}$$

The same observation must be made as in the case of the stab, that is that this is a rather virtual or equivalent range, since the stick in the cockpit is assumed to still be capable of moving within the nominal range, only the effects are different. Note also that the span of the range at post failure conditions is reduced ($[d_{a\,minF},\ \ d_{a\,maxF}] \subset [d_{a\,min},\ \ d_{a\,max}]$); however, the post failure range is *not* invariable with respect to the surface failed (L or R). Assume that the range of the generic aileron deflection at normal conditions is ±30 deg. The ranges for left and right aileron deflections are ±15. If the left aileron is locked at 5 deg, then the post failure range is:

$$d_{a\,max\,F} = k_a[5 - (-15)] = 20k_a \tag{5.42}$$

$$d_{a\,min\,F} = k_a(5 - 15) = -10k_a \tag{5.43}$$

If the right aileron is locked at 5 deg, then the post failure range is:

$$d_{a\,max\,F} = k_a(15-5) = 10k_a \tag{5.44}$$

$$d_{a\,min\,F} = k_a(-15-5) = -20k_a \tag{5.45}$$

Note that this range will be the same if the opposite surface is locked in the opposite direction. With the new range after failure, new ranges for all v_E may be obtained from the 2-D projections.

5.2.3. Rudder Failure

This discussion covers failures $k_F = 9$ through $k_F = 12$ of Table 1.2, since, from the point of view of the indirect quantitative evaluation, they are equivalent. The discussion will continue with reference to the left rudder; however, everything applies similarly to the right surface failure as well, unless otherwise noted. For $k_F = 9 \div 12$, the following parameters can be specified:

$v_\delta = \left[v_{\delta 1}\right] = \left[\delta_{rL}\right]$, left (or right) rudder deflection, positive to left of pilot such that it produces positive lift on the vertical tail in body axes

$v_\varepsilon = \left[v_{\varepsilon 1}\right] = \left[d_r\right]$, pedals (5.46)

$v_E = \left[a_y \quad p \quad r \quad \dot{p} \quad \dot{r} \quad \beta \quad \varphi\right]$

The relationship between DIV and EDIV is:

$$d_r = f_{\varepsilon\delta 1}(\delta_{rL}) \tag{5.47}$$

$$d_r = k_r \frac{\delta_{rL} + \delta_{rR}}{2} \tag{5.48}$$

If the left rudder is locked, then:

$$\delta_{rL}(t) = \delta_{rLF} \tag{5.49}$$

where δ_{rLF} is a value determined through direct quantitative evaluation. The nominal range for d_r is (typically $d_{r\,max} = -d_{r\,min}$):

$$d_{r\,max} = k_r \frac{\delta_{rL\,max} + \delta_{rR\,max}}{2} \tag{5.50}$$

$$d_{r\,min} = k_a \frac{\delta_{rL\,min} + \delta_{rR\,min}}{2} \tag{5.51}$$

After failure, the range for d_r becomes:

$$d_{r\,maxF} = k_r \frac{\delta_{rLF} + \delta_{rR\,max}}{2} \tag{5.52}$$

$$d_{r\,minF} = k_r \frac{\delta_{rLF} + \delta_{rR\,min}}{2} \tag{5.53}$$

As was the case with the other control channels, this is a rather virtual or equivalent range, since the pedals in the cockpit are assumed to still be capable of moving within the nominal range, only the effects are different. Note also that the range at post failure conditions ($[d_{r\,minF},\ d_{r\,maxF}]$) is invariable with respect to the surface failed (L or R) and that $[d_{r\,minF},\ d_{r\,maxF}] \subset [d_{r\,min},\ d_{r\,max}]$. The span of the range at post failure conditions is reduced ($[d_{r\,minF},\ d_{r\,maxF}] \subset [d_{r\,min},\ d_{r\,max}]$). Because of its collective deflection, the case of the rudder is similar, in many respects, to the stabilator, if the aircraft is equipped with dual rudder, as is the case with the aircraft used for demonstration in this project. With the new range after failure, new ranges for all v_E may be obtained from the 2-D projections.

5.2.4. Throttle Failure

This discussion covers failures $k_F = 13$ through $k_F = 16$ of Table 1.2, since, from the point of view of the indirect quantitative evaluation, they are equivalent. The discussion will continue with reference to the left (engine) throttle; however, everything applies similarly to the right actuator failure as well. For $k_F = 13 \div 16$, the following key parameters can be specified:

$$\begin{aligned} &v_\delta = [v_{\delta 1}] = [\delta_{TL}],\ \text{left (or right) throttle/fuel valve, positive = open} \\ &v_\varepsilon = [v_{\varepsilon 1}] = [d_T],\ \text{collective throttle (is assumed)} \\ &v_E = [a_x \quad a_z \quad r \quad H \quad M \quad V \quad \alpha \quad \theta] \end{aligned} \tag{5.54}$$

The relationship between DIV and EDIV is:

$$d_T = f_{\varepsilon\delta 1}(\delta_{TL}) \tag{5.55}$$

$$d_T = k_T \delta_T = k_T \frac{\delta_{TL} + \delta_{TR}}{2} \tag{5.56}$$

If the left throttle is locked, then:

$$\delta_{TL}(t) = \delta_{TLF} \tag{5.57}$$

where δ_{TLF} is a value determined through direct quantitative evaluation. The nominal range for d_T is typically $d_T \in [0,\ 100\%]$ with:

$$d_{T\,max} = k_T \frac{\delta_{TL\,max} + \delta_{TR\,max}}{2} = 100\% \tag{5.58}$$

and
$$d_{T\,min} = k_T \frac{\delta_{TL\,min} + \delta_{TR\,min}}{2} = 0\% \tag{5.59}$$

After failure, the range for d_T becomes:

$$d_{T\,maxF} = k_T \frac{\delta_{TLF} + \delta_{TR\,max}}{2} = k_T \frac{\delta_{TLF} + 100}{2} \tag{5.60}$$

$$d_{T\,minF} = k_T \frac{\delta_{TLF} + \delta_{TR\,min}}{2} = k_T \frac{\delta_{TLF}}{2} \tag{5.61}$$

The throttle lever in the cockpit is assumed to still be capable of moving within the nominal range, only the effects are different; therefore, the range in Eq (5.60) and (5.61) is a rather virtual or equivalent range. Note also that the range at post failure conditions ($\left[d_{T\,minF},\ \ d_{T\,maxF}\right]$) is invariable with respect to the surface failed (L or R) and that $\left[d_{T\,minF},\ \ d_{T\,maxF}\right] \subset \left[0,\ \ 100\right]$. With the new range after failure, new ranges for all v_E may be obtained from the 2-D projections, if there is a direct mutual constraint (direct cause/effect relationship) between the two features involved in the projection. It seems that this is the case for some of the envelope relevant variables not for others. For instance, *r* appears to need some special attention since it is clear that the effects on *r* are different whether the L or R engine fails and the difference in L or R failure is not captured by the virtual control range after the failure. Due to asymmetry, a throttle failure will create an undesired yawing moment (depending on the distance of the engines with respect to the aircraft centerline), which needs to be compensated for by using rudder deflection. This will reduce the overall control authority on the yaw channel. A similar approach as in the case of the rolling moment compensated using ailerons, in the case of a stab failure, will be applied next. To determine the effects on *r* produced by a locked throttle failure, it will be stipulated that the locked throttle failure induces a rudder pseudo-failure, from the point of view of producing yawing rate. This pseudo-failure consists of a shift of the trim position of the rudder at δ_{rPF} necessary to compensate for the yawing effect of the locked throttle. The compensating rudder deflection can be approximated using the balance of the yawing moment:

$$C_{n\delta_{TL}} \cdot (\delta_{TLF} - \delta_{TR}) = -C_{n\delta_r} \cdot \delta_{rPF} \tag{5.62}$$

$$\delta_{rPF} = -\frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(\delta_{TLF} - \delta_{TR}) \tag{5.63}$$

Note that $C_{n\delta_{TL}} > 0$ and $C_{n\delta_r} < 0$. Considering that $\delta_{TLF} \in [0, 100]$ then:

$$\delta_{rPF} \in [\frac{C_{n\delta_{TL}}}{C_{n\delta_r}}\delta_{TR}, \quad -\frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(100-\delta_{TR})] \tag{5.64}$$

For full range of δ_{TR} we get:

$$\delta_{rPF} \in [\frac{C_{n\delta_{TL}}}{C_{n\delta_r}}100, \quad -\frac{C_{n\delta_{TL}}}{C_{n\delta_r}}100] \tag{5.65}$$

Note that $-\frac{C_{n\delta_{TL}}}{C_{n\delta_r}}100 > 0$. If the right engine fails, then:

$$\delta_{rPF} = \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(\delta_{TL} - \delta_{TRF}) \tag{5.66}$$

Note that $C_{n\delta_{TR}} < 0$ and $C_{n\delta_r} < 0$. Considering that $\delta_{TLR} \in [0,100]$ then:

$$\delta_{rPF} \in [\frac{C_{n\delta_{TR}}}{C_{n\delta_r}}\delta_{TL}, \quad \frac{C_{n\delta_{TR}}}{C_{n\delta_r}}(\delta_{TL} - 100)] \tag{5.67}$$

For full range of δ_{TL} we get:

$$\delta_{rPF} \in [-\frac{C_{n\delta_{TR}}}{C_{n\delta_r}}100, \quad \frac{C_{n\delta_{TR}}}{C_{n\delta_r}}100] \tag{5.68}$$

Note that $-\frac{C_{n\delta_{TR}}}{C_{n\delta_r}}100 < 0$. The generic rudder deflection is (for dual rudder aircraft), of course:

$$\delta_r = \frac{\delta_{rL} + \delta_{rR}}{2} \tag{5.69}$$

and the relationship with the EDIV is:

$$d_r = k_r\delta_r \tag{5.69}$$

The nominal directional control range is:

$$d_{r\,max} = k_r\delta_{r\,max} \tag{5.70}$$

$$d_{r\,min} = k_r\delta_{r\,min} \tag{5.71}$$

Under throttle failure, the new range for the directional control is:

$$d_{r\,max\,F} = k_r(\delta_{r\,max} - \delta_{rPF}) \tag{5.72}$$

$$d_{r\,min\,F} = k_r(\delta_{r\,min} - \delta_{rPF}) \tag{5.73}$$

Note that this is equivalent to shifting the range of d_r in one direction or the other (depending on the side of the failed engine and the throttle of the healthy engine), while maintaining the same span:

$$\left[d_{r\,minF},\quad d_{r\,maxF}\right] \not\subset \left[d_{r\,min},\quad d_{r\,max}\right] \tag{5.74}$$

$$d_{r\,maxF} - d_{r\,minF} = d_{r\,max} - d_{r\,min} \tag{5.75}$$

Now the new range of d_r can be used with the (d_r, r) projection to determine the effects on r of the throttle failure. This range can be expressed in terms of the failed surface and the throttle of the healthy engine as follows:

$$d_{r\,max\,F} = k_r\,[\,\delta_{r\,max} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(\delta_{TLF} - \delta_{TR})] \tag{5.76}$$

$$d_{r\,min\,F} = k_r\,[\,\delta_{r\,min} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(\delta_{TLF} - \delta_{TR})] \tag{5.77}$$

Therefore, for $\delta_{TR} = 100$

$$max(d_{r\,max\,F}) = k_r\,[\,\delta_{r\,max} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(\delta_{TLF} - 100)] \tag{5.78}$$

and for $\delta_{TR} = 0$

$$min(d_{r\,min\,F}) = k_r(\delta_{r\,min} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}\delta_{TLF}) \tag{5.79}$$

If the right engine fails, then the range can be expressed as:

$$d_{r\,max\,F} = k_r\,[\,\delta_{r\,max} - \frac{C_{n\delta_{TR}}}{C_{n\delta_r}}(\delta_{TL} - \delta_{TRF})] \tag{5.80}$$

$$d_{r\,min\,F} = k_r\,[\,\delta_{r\,min} - \frac{C_{n\delta_{TR}}}{C_{n\delta_r}}(\delta_{TL} - \delta_{TRF})] \tag{5.81}$$

Therefore, for $\delta_{TL} = 0$

$$max(d_{r\,max\,F}) = k_r(\delta_{r\,max} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}\delta_{TRF})$$

(5.82)

and for $\delta_{TL} = 100$

$$min(d_{r\,min\,F}) = k_r[\delta_{r\,min} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}}(\delta_{TRF} - 100)] \tag{5.83}$$

Note that $C_{n\delta_{TL}} = -C_{n\delta_{TR}}$. Therefore, the rudder pseudo-failure range shifts towards positive deflections for a left engine failure, and towards negative deflections for a right engine failure. The span remains the same.

5.3. Estimation Algorithms for Sensor AC

5.3.1. Roll Rate Sensor Bias

The analysis of the sensor failures relies on the specifics of the control system that uses the sensor information. Based on stick displacement, the control laws determine first a "desired" angular rate response as determined by reference models (established using handling qualities criteria). Then, they track the desired angular rate response. For $k_F = 17$, one can specify:

$$\begin{aligned} v_\delta &= [v_{\delta 1}] = [p_{meas}], \text{ measured roll rate} \\ v_\varepsilon &= [v_{\varepsilon 1}] = [d_a], \text{ lateral stick} \\ v_E &= [a_y \quad p \quad r \quad \dot{p} \quad \dot{r} \quad \beta \quad \varphi] \end{aligned} \qquad ..(5.84)$$

At nominal conditions, the desired or reference roll rate is:

$$p_{ref} = F_1(d_a) \qquad (5.85)$$

where F_1 represents the roll channel reference model, a first order transfer function. If the roll rate measurement is assumed to be perfect:

$$p_{meas} = p_{act} \qquad (5.86)$$

and the commanded roll rate is given by a control law of the form:

$$p_{cmd} = F_2(p_{ref} - p_{act}) \qquad (5.87)$$

At failure condition, the reference is unchanged; however:

$$p_{measF} = p_{act} + p_{bias} \qquad (5.88)$$

$$p_{cmdF} = F_2(p_{ref} - p_{act} - p_{bias}) \qquad (5.89)$$

As a result of the failure, an undesired additional roll rate command is produced, which can be expressed with some simplifying assumptions at steady state as:

$$\Delta p_{cmd} = p_{cmdF} - p_{cmd} = -p_{bias} \qquad (5.90)$$

In other words, the control laws will rotate the aircraft in the opposite direction of the sensor bias. This effect must be compensated using aileron command; therefore, the control authority on the roll channel will be affected by restricting the excursion of the cockpit controls by the amount d_{aF} equal to what is necessary to command a roll rate equal to the bias:

$$d_{aF} = F_1^{-1}(p_{bias}) \qquad (5.91)$$

The new equivalent range of d_a is:

$$d_{a\,max\,F} = d_{a\,max} - d_{aF} \tag{5.92}$$

$$d_{a\,min\,F} = d_{a\,min} - d_{aF} \tag{5.93}$$

With the new range of lateral stick after failure, corresponding new ranges for all v_E may be obtained from the 2-D projections.

5.3.2. Pitch Rate Sensor Bias

For $k_F = 19$, one can specify:

$v_\delta = [v_{\delta 1}] = [q_{meas}]$, measured pitch rate
$v_\varepsilon = [v_{\varepsilon 1}] = [d_e]$, longitudinal stick
(5.94)
$v_E = [a_x \quad a_z \quad H \quad M \quad V \quad q \quad \dot{q} \quad \alpha \quad \theta]$

The analysis of the pitch sensor failures follows the same approach as the roll sensor failures. At nominal conditions, the desired or reference pitch rate is:

$$q_{ref} = F_1(d_e) \tag{5.95}$$

where F_1 represents the pitch channel reference model, a second order transfer function. The pitch rate measurement is assumed to be perfect:

$$q_{meas} = q_{act} \tag{5.96}$$

and the commanded pitch rate is given by a control law of the form:

$$q_{cmd} = F_2(q_{ref} - q_{act}) \tag{5.97}$$

At failure condition, the reference is unchanged; however:

$$q_{measF} = q_{act} + q_{bias} \tag{5.98}$$

$$q_{cmdF} = F_2(q_{ref} - q_{act} - q_{bias}) \tag{5.99}$$

As a result of the failure, an undesired additional pitch rate command is produced, which can be expressed with some assumptions, at steady state, as:

$$\Delta q_{cmd} = q_{cmdF} - q_{cmd} = -q_{bias} \tag{5.100}$$

In other words, the control laws will rotate the aircraft in the opposite direction of the sensor bias. This effect must be compensated using stabilator/elevator command; therefore, the control authority on the pitch channel will be affected by restricting the excursion of the cockpit controls by the amount d_{eF} equal to what is necessary to command a pitch rate equal to the bias:

$$d_{eF} = F_1^{-1}(q_{bias}) \tag{5.101}$$

The new equivalent range of d_e is:

$$d_{e\,max\,F} = d_{e\,max} - d_{eF} \tag{5.102}$$

$$d_{e\,min\,F} = d_{e\,min} - d_{eF} \tag{5.103}$$

With the new range of longitudinal stick after failure, corresponding new ranges for all v_E may be obtained from the 2-D projections.

5.3.3. Yaw Rate Sensor Bias

For $k_F = 21$, the following parameters can be specified:

$$\begin{aligned} v_\delta &= [v_{\delta 1}] = [r_{meas}],\ \text{measured yaw rate} \\ v_\varepsilon &= [v_{\varepsilon 1}] = [d_r],\ \text{pedals} \\ v_E &= [a_y \quad p \quad r \quad \dot{p} \quad \dot{r} \quad \beta \quad \varphi] \end{aligned} \tag{5.104}$$

At nominal conditions, the desired or reference yaw rate is:

$$r_{ref} = F_1(d_r) \tag{5.105}$$

where F_1 represents the yaw channel reference model, a second order transfer function. The yaw rate measurement is assumed to be perfect:

$$r_{meas} = r_{act} \tag{5.106}$$

and the commanded yaw rate is given by a control law of the form:

$$r_{cmd} = F_2(r_{ref} - r_{act}) \tag{5.107}$$

At failure condition, the reference is unchanged; however:

$$r_{measF} = r_{act} + r_{bias} \tag{5.108}$$

$$r_{cmdF} = F_2(r_{ref} - r_{act} - r_{bias}) \qquad (5.109)$$

As a result of the failure, an undesired additional yaw rate command is produced, which can be expressed with some assumptions, at steady state, as:

$$\Delta r_{cmd} = r_{cmdF} - r_{cmd} = -r_{bias} \qquad (5.110)$$

In other words, the control laws will rotate the aircraft in the opposite direction of the sensor bias. This effect must be compensated using rudder command; therefore, the control authority on the yaw channel will be affected by restricting the excursion of the cockpit controls by the amount d_{rF} equal to what is necessary to command a yaw rate equal to the bias:

$$d_{rF} = F_1^{-1}(r_{bias}) \qquad (5.111)$$

The new equivalent range of d_r is:

$$d_{r\,max\,F} = d_{r\,max} - d_{rF} \qquad (5.112)$$

$$d_{r\,min\,F} = d_{r\,min} - d_{rF} \qquad (5.113)$$

With the new range of pedals after failure, corresponding new ranges for all v_E may be obtained from the 2-D projections.

5.4. Estimation Algorithms for Structural Damage

5.4.1. Wing Damage

For $k_F = 23 \div 24$, the key parameters for AC indirect evaluation can be specified as:

$v_\delta = [v_{\delta 1} \quad v_{\delta 2}] = [\Delta C_{L_{WL}} \quad \Delta C_{m_{WL}}]$, lift and pitching moment alteration due to left wing failure
$v_\varepsilon = [v_{\varepsilon 1} \quad v_{\varepsilon 2} \quad v_{\varepsilon 3} \quad v_{\varepsilon 4}] = [d_a\ d_e\ d_r\ d_T]$, lateral stick, longitudinal stick, pedals, throttle lever
$v_E = \lfloor H\ M\ V\ \ p\ q\ r\ a_x\ a_y\ a_z\ \dot{p}\ \dot{q}\ \dot{r}\ \alpha\ \beta\ \theta\ \varphi \rfloor$ (5.114)

At normal trim conditions, the external forces acting on the aircraft are presented in Figure 5.2.

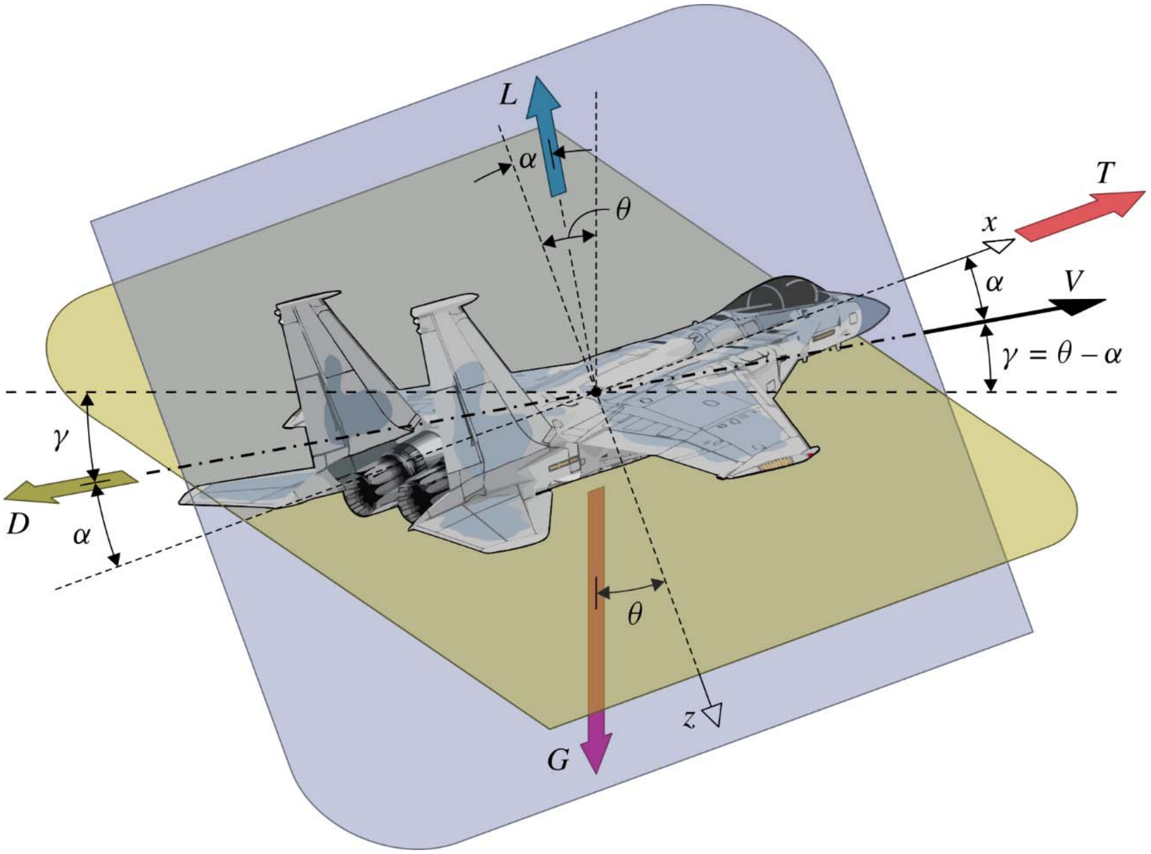


Figure 5.2. Aircraft Forces in the Plane of Symmetry at Trim

The balance equation can be expressed as:

$$\begin{cases} T + L\sin(\alpha) - D\cos(\alpha) - G\sin(\theta) = 0 \\ -L\cos(\alpha) - D\sin(\alpha) + G\cos(\theta) = 0 \\ M_{WF} = -M_{HT} \\ C_L = C_L(C_D) \\ C_{lWF} = 0 \end{cases} \tag{5.115}$$

$$\begin{cases} T(M,H,d_T) + L(M,\alpha)sin(\alpha) - D(M,\alpha)cos(\alpha) - G\,sin(\theta) = 0 \\ -L(M,\alpha)cos(\alpha) - D(M,\alpha)sin(\alpha) + G\,cos(\theta) = 0 \\ M_{WF}(M,\alpha) = -M_{HT}(M,\alpha,\delta_e) \\ C_L(M,\alpha) = C_L(C_D(M,\alpha)) \\ C_{lWF} = 0 \end{cases} \tag{5.116}$$

Note that WF stands for "wing and fuselage" and HT for "horizontal tail". Assuming that at post failure conditions the same flight condition is maintained, then:

$$M_F = M\,,\ H_F = H\,,\ \gamma_F = \gamma \tag{5.117}$$

The effect of the failure will consist of producing variations Δ's of all other variables. Therefore:

$$\begin{cases} T + \Delta T + (L + \Delta L)sin(\alpha + \Delta\alpha) - (D + \Delta D)cos(\alpha + \Delta\alpha) - G\,sin(\theta + \Delta\theta) = 0 \\ -(L + \Delta L)cos(\alpha + \Delta\alpha) - (D + \Delta D)sin(\alpha + \Delta\alpha) + G\,cos(\theta + \Delta\theta) = 0 \\ M_{WF} + \Delta M_{WF} = -M_{HT} - \Delta M_{HT} \\ C_{LF} = C_{LF}(C_D) \\ \Delta C_{lWF} = -\Delta C_{LWL} \cdot l_{AL} \neq 0 \end{cases} \tag{5.118}$$

where l_{AL} is the distance to the plane of symmetry of the left wing aerodynamic center at post-failure conditions. Note that $l_{AL} < 0$, while $l_{AR} > 0$. The non-zero rolling moment must be compensated using aileron deflection, thus reducing the authority of the aileron command. The same concept of aileron "pseudo-failure" may be considered here as in the case of the stabilator failure. This pseudo-failure involves a non-zero aileron trim deflection at a position δ_{aPF} necessary to compensate for the rolling effect of the damaged wing. This compensating aileron deflection can be approximated using the balance of the rolling moment:

$$-\Delta C_{LWL} \cdot l_{AL} = -\varepsilon_a b C_{l\delta_a} \cdot \delta_{aPF} \tag{5.119}$$

$$\delta_{aPF} = \frac{\Delta C_{LWL} \cdot l_{AL}}{\varepsilon_a b C_{l\delta_a}} \tag{5.120}$$

where b is the wingspan and ε_a is an alteration factor depending on how the left aileron is affected by the wing failure ($\varepsilon_a = 1$ if the aileron is not affected by the wing failure and $\varepsilon_a = 0.5$ if the aileron is completely out of work). For a linear relationship with the EDIV is:

$$d_a = k_a \delta_a \tag{5.121}$$

then the nominal lateral control range is:

$$d_{a\,max} = k_a \delta_{a\,max} \tag{5.122}$$

$$d_{a\,min} = k_a \delta_{a\,min} \tag{5.123}$$

With the damaged wing, the new range for the lateral control can be assumed to be:

$$d_{a\,max\,F} = k_a(\delta_{a\,max} - \delta_{aPF}) \tag{5.124}$$

$$d_{a\,min\,F} = k_a(\delta_{a\,min} - \delta_{aPF}) \tag{5.125}$$

Note that this is equivalent to shifting the range of d_a in one direction or the other, while maintaining the same span:

$$\left[d_{aminF},\quad d_{amaxF}\right] \not\subset \left[d_{amin},\quad d_{amax}\right] \tag{5.126}$$

$$d_{a\,maxF} - d_{a\,minF} = d_{a\,max} - d_{a\,min} \tag{5.127}$$

The direction of the shift depends on the failed wing side and the sign of the lift alteration. However, it is reasonable to assume that $\Delta C_{LWL} < 0$ since one would expect that wing damage would decrease the lift produced. Now the new range of d_a can be used with the respective projections to determine the effects of the wing failure on lateral-directional variables. This range can be expressed in terms of the failed surface as follows:

$$d_{a\,max\,F} = k_a\left(\delta_{a\,max} - \frac{\Delta C_{LWL} \cdot l_{AL}}{\varepsilon_a b C_{l\delta_a}}\right) \tag{5.128}$$

$$d_{a\,min\,F} = k_a\left(\delta_{a\,min} - \frac{\Delta C_{LWL} \cdot l_{AL}}{\varepsilon_a b C_{l\delta_a}}\right) \tag{5.129}$$

Note that ΔC_{LWL}, l_{AL}, and ε_a must be known. How to adequately determine them must be further investigated. ΔC_{LWL} could be a measure of wing failure severity and determined as an outcome of the quantitative direct failure evaluation. The same observation applies to ΔC_{mWL} as well. If this alteration is known, then the reduction of longitudinal control authority can be determined using the pitching moment balance, as follows:

$$\Delta M_{WF} = -\Delta M_{HT} \tag{5.130}$$

$$\Delta C_{mWL} = -C_{m\delta_e} \cdot \Delta\delta_{ePF} \tag{5.131}$$

$$\Delta\delta_{ePF} = -\frac{\Delta C_{mWL}}{C_{m\delta_e}} \tag{5.132}$$

With the same previous assumptions, the new equivalent range for d_e is:

$$d_{e\,max\,F} = d_{e\,max} - k_e \cdot (\Delta\delta_{ePF} + \delta_{etrim}) \tag{5.133}$$

$$d_{e\,min\,F} = d_{e\,min} - k_e \cdot (\Delta\delta_{ePF} + \delta_{etrim}) \tag{5.134}$$

The new equivalent range for d_e can be used to obtain, from the respective projections, the new ranges for longitudinal variables. To determine effects on H, M, and V, we can consider the equations for the longitudinal force, the vertical force, and the polar:

$$\begin{cases} T + \Delta T + (L + \Delta L) sin(\alpha) cos(\Delta\alpha) + (L + \Delta L) cos(\alpha) sin(\Delta\alpha) - \\ -(D + \Delta D) cos(\alpha) cos(\Delta\alpha) + (D + \Delta D) sin(\alpha) sin(\Delta\alpha) - \\ -G\, sin(\theta) cos(\Delta\theta) - G\, cos(\theta) sin(\Delta\theta) = 0 \\ -(L + \Delta L) cos(\alpha) cos(\Delta\alpha) + (L + \Delta L) sin(\alpha) sin(\Delta\alpha) - \\ -(D + \Delta D) sin(\alpha) cos(\Delta\alpha) - (D + \Delta D) cos(\alpha) sin(\Delta\alpha) + \\ +G\, cos(\theta) cos(\Delta\theta) - G\, sin(\theta) sin(\Delta\theta) = 0 \\ \Delta D = C_{LF}^{-1}(\Delta L) \end{cases} \tag{5.135}$$

For keeping the same trajectory slope γ, then $\Delta\alpha = \Delta\theta$. ΔL is known. If the polar $C_{LF}(C_D)$ is known, then a system of 3 equations and 3 unknowns ΔT, ΔD, and $\Delta\alpha$ is obtained. Solving for ΔT, allows determining new virtual ranges for d_T. From respective projections, new ranges for *M*, *H*, and *V* can then be obtained. Note that the method derived here relies on several strong assumption. Therefore, the method applies for classes of ACs that satisfy these assumptions.

5.4.2. Horizontal Tail Damage

For $k_F = 25 \div 26$, one can specify:

$$\begin{aligned} &v_\delta = [v_{\delta 1}] = [\Delta C_{L_{HL}}], \text{ lift alteration due to left horizontal tail damage} \\ &v_\varepsilon = [v_{\varepsilon 1}\ v_{\varepsilon 2}] = [d_a\ d_e], \text{ lateral stick and longitudinal stick} \\ &v_E = [H\ M\ V\ p\ q\ r\ a_x\ a_y\ a_z\ \dot{p}\ \dot{q}\ \dot{r}\ \alpha\ \beta\ \theta\ \varphi] \end{aligned} \tag{5.136}$$

The assumption is made that the left elevator (stabilator) can be still deflected; however, the respective control derivative may be affected, such that:

$$\Delta C_{L_{HL}} = \Delta C_{L_{HL0}} + \Delta C_{L\delta_e}\delta_e \tag{5.137}$$

It is assumed that both $\Delta C_{L_{HL0}}$ and $\Delta C_{L\delta_e}$ are negative representing a reduction of the horizontal tail capability to produce lift. For longitudinal balance, this lift variation must be compensated by an additional elevator deflection $\Delta\delta_e > 0$ such that:

$$-\Delta C_{L_{HL}} = \left(\frac{C_{L\delta_e}}{2} - \Delta C_{L\delta_e}\right)\Delta\delta_e + \frac{C_{L\delta_e}}{2}\Delta\delta_e \tag{5.138}$$

$$-\Delta C_{L_{HL}} = \left(C_{L\delta_e} - \Delta C_{L\delta_e}\right)\Delta\delta_e \tag{5.139}$$

$$\Delta\delta_e = -\frac{\Delta C_{L_{HL}}}{C_{L\delta_e} - \Delta C_{L\delta_e}} \tag{5.140}$$

For lateral balance, a compensatory rolling moment must be produced using aileron, such that:

$$\Delta C_{L_{HL}} \cdot \frac{l_{LHT}}{b} = C_{l\delta_a}\Delta\delta_a \tag{5.141}$$

where $l_{LHT} < 0$ is the distance from the left horizontal tail aerodynamic center to the plane of symmetry. Note that $l_{RHT} > 0$. Then:

$$\Delta\delta_a = \frac{\Delta C_{L_{HL}} \cdot l_{LHT}}{bC_{l\delta_a}} \tag{5.142}$$

To estimate $\Delta C_{L_{HL}}$ consider that:

$$\begin{aligned}\Delta C_{L_{HL}} &= \Delta C_{L_{HL\_BeforeFailure}} - \Delta C_{L_{HL\_AfterFailure}} = \\ &= \Delta C_{L_{HL\_BeforeFailure}} - sddeL \cdot \Delta C_{L_{HL\_BeforeFailure}} = \\ &= (1 - sddeL)\Delta C_{L_{HL\_BeforeFailure}} = \\ &= (1 - sddeL)\frac{C_{L\delta_e}}{2}\delta_e\end{aligned} \tag{5.143}$$

Therefore:

$$\varDelta\delta_e = -\frac{\varDelta C_{L_{HL}}}{C_{L\delta_e} - \varDelta C_{L\delta_e}} = -\frac{(1-sddeL)\frac{C_{L\delta_e}}{2}\delta_e}{C_{L\delta_e} - (1-sddeL)\frac{C_{L\delta_e}}{2}} =$$

$$= -\frac{(1-sddeL)\frac{C_{L\delta_e}}{2}\delta_e}{C_{L\delta_e}\ (1-\frac{1}{2}+\frac{sddeL}{2})} = -\frac{(1-sddeL)\delta_e}{1+sddeL} \tag{5.144}$$

The value of l_{LHT} can be approximated as:

$$l_{LHT} = 2\frac{C_{l\delta_e L}}{C_{L\delta_e}} \tag{5.145}$$

The relationships between DIV and EDIV are:

$$d_e = f_{\varepsilon\delta 1}(\delta_e) \quad \text{and} \quad d_a = f_{\varepsilon\delta 2}(\delta_a) \tag{5.146}$$

or $$d_e = k_e\delta_e \quad \text{and} \quad d_a = k_a\delta_a \tag{5.147}$$

After failure, the ranges for d_e and d_a become:

$$d_{e\,max\,F} = d_{e\,max} - k_e\varDelta\delta_e \ \text{and} \quad d_{e\,min\,F} = d_{e\,min} - k_e\varDelta\delta_e \tag{5.148}$$

$$d_{a\,max\,F} = d_{a\,max} - k_a\varDelta\delta_a \text{and} \quad d_{a\,min\,F} = d_{a\,min} - k_a\varDelta\delta_a \tag{5.149}$$

It is postulated that new ranges for longitudinal variables may be obtained from corresponding 2-D projections of d_e. Similarly, new ranges for lateral-directional variables may be obtained from corresponding 2-D projections of d_a.

5.4.3. Vertical Tail Damage

For $k_F = 27 \div 28$, the key parameters can be specified as:

$$\begin{aligned} &v_\delta = [v_{\delta 1}] = [\varDelta C_{Y_{VL}}], \ \text{lift (lateral force) alteration due to left vertical tail damage} \\ &v_\varepsilon = [v_{\varepsilon 1}\ v_{\varepsilon 2}] = [d_a\ d_r], \ \text{lateral stick and pedals} \\ &v_E = [p\ r\ a_y\ \dot{p}\ \dot{r}\ \beta\ \varphi] \end{aligned} \tag{5.150}$$

It is assumed that the left rudder can be still deflected; however, the respective control derivative may be affected, such that:

$$\Delta C_{Y_{VL}} = \Delta C_{Y_{VL0}} + \Delta C_{Y\delta_r}\delta_r \tag{5.151}$$

For directional balance, this lift variation must be compensated by an additional rudder deflection $\Delta\delta_r$ such that:

$$-\Delta C_{Y_{VL}} = \left(\frac{C_{Y\delta_r}}{2} - \Delta C_{Y\delta_r}\right)\Delta\delta_r + \frac{C_{Y\delta_r}}{2}\Delta\delta_r \tag{5.152}$$

$$-\Delta C_{Y_{VL}} = \left(C_{Y\delta_r} - \Delta C_{Y\delta_r}\right)\Delta\delta_r \tag{5.153}$$

$$\Delta\delta_r = -\frac{\Delta C_{Y_{VL}}}{C_{Y\delta_r} - \Delta C_{Y\delta_r}} \tag{5.154}$$

For roll balance, a compensatory rolling moment must be produced using aileron, such that:

$$\Delta C_{Y_{VL}} \cdot \frac{h_{VT}}{b} = C_{l\delta_a}\Delta\delta_a \tag{5.155}$$

where $h_{VT} < 0$ is the distance from the vertical tail aerodynamic center to the body horizontal plane. Note that h_{VT} is normally the same for L and R. Then:

$$\Delta\delta_a = \frac{\Delta C_{Y_{VL}} \cdot h_{VT}}{bC_{l\delta_a}} \tag{5.156}$$

The relationships between DIV and EDIV are:

$$d_a = f_{\varepsilon\delta 1}(\delta_a) \quad \text{and} \quad d_r = f_{\varepsilon\delta 2}(\delta_r) \tag{5.157}$$

or

$$d_a = k_a\delta_a \quad \text{and} \quad d_r = k_r\delta_r \tag{5.158}$$

After failure, the ranges for d_r and d_a become:

$$d_{r\,maxF} = d_{r\,max} - k_r\Delta\delta_r \quad \text{and} \quad d_{r\,minF} = d_{r\,min} - k_r\Delta\delta_r \tag{5.159}$$

$$d_{a\,maxF} = d_{a\,max} - k_a\Delta\delta_a \quad \text{and} \quad d_{a\,minF} = d_{a\,min} - k_a\Delta\delta_a \tag{5.160}$$

It is postulated that new ranges for directional channel variables may be obtained from corresponding 2-D projections of d_r. Similarly, new ranges for lateral channel variables may be obtained from corresponding 2-D projections of d_a. This hypothesis requires further investigation given the typical coupling between the two dynamic channels.

5.5. Estimation Algorithms for Propulsion AC

This discussion covers failure $k_F = 29$ and $k_F = 30$ of Table 1.2. It makes reference to the left engine; however, everything applies similarly to the right engine failure. For $k_F = 29 \div 30$, it can be specify that:

$$\begin{aligned} &v_\delta = [v_{\delta 1}] = [T_L],\ \text{left (or right) engine thrust} \\ &v_\varepsilon = [v_{\varepsilon 1}] = [d_T],\ \text{collective throttle (is assumed)} \\ &v_E = [v_{E1}] = [a_x \quad a_z \quad r \quad H \quad M \quad V \quad \alpha \quad \theta] \end{aligned} \tag{5.161}$$

The relationship between DIV and EDIV is:

$$d_T = f_{\varepsilon\delta 1}(T_L + T_R) \tag{5.162}$$

The nominal range for d_T is typically $d_T \in [0, \quad 100\%]$, with:

$$d_{T\,max} = f_{\varepsilon\delta 1}(T_{Lmax} + T_{Rmax}) = f_{\varepsilon\delta 1}(2T_{max}) = 100\% \tag{5.163}$$

and

$$d_{T\,min} = f_{\varepsilon\delta 1}(T_{Lmin} + T_{Rmin}) = f_{\varepsilon\delta 1}(2T_{min}) = 0\% \tag{5.164}$$

The failure consists of changing the range of T_L to T_{LminF} and T_{LmaxF} and changing the function $f_{\varepsilon\delta 1}$ accordingly. After failure, the equivalent range of d_T can be defined as:

$$d_{T\,maxF} = f_{\varepsilon\delta 1}(T_{L\,maxF} + T_{R\,max}) \tag{5.165}$$

$$d_{T\,minF} = f_{\varepsilon\delta 1}(T_{LminF} + T_{Rmin}) \tag{5.166}$$

Most often, $T_{LminF} = 0$ and $T_{LmaxF} < T_{max})$. The values of $T_{L\,minF}$ and T_{LmaxF} must be determine through direct quantitative evaluation. Also, $f_{\varepsilon\delta 1}$ is assumed known; it can be approximated as a linear relationship. Note also that the equivalent range at post failure conditions ($[d_{T\,minF}, \quad d_{T\,maxF}]$) is invariable with respect to the surface failed (L or R) and that $[d_{T\,minF}, \quad d_{T\,maxF}] \subset [0, \quad 100]$. With the new range after failure, new ranges for all v_E may be obtained from the 2-D projections, except for *r* (see discussion for throttle failure). An undesired yawing moment (depending on the distance of the engines with respect to the aircraft centerline) may be produced, which needs to be compensated for by using rudder deflection. This will reduce the overall control authority on the yaw channel. A similar approach as in the case of the locked throttle can be applied to handle this case. However, this time, the pseudo-failure of the rudder consists of a variable trim deflection depending on d_T.

Let us assume linear relationships at normal conditions between pilot throttle input and total thrust as well as between pilot throttle input and fuel valve deflection. Therefore, the total yawing moment is:

$$C_{n\delta_{TL}} \cdot \delta_{TL} + C_{n\delta_{TR}} \cdot \delta_{TR} = 0 \tag{5.167}$$

or
$$(C_{n\delta_{TL}} + C_{n\delta_{TR}}) \cdot d_T = 0 \tag{5.168}$$

where, of course, $C_{n\delta_{TL}} = -C_{n\delta_{TR}} > 0$. At post failure conditions, the healthy engine control derivative remains unchanged, while the failed engine control derivative is affected:

$$(C_{n\delta_{TLF}} + C_{n\delta_{TR}}) \cdot d_T \neq 0 \tag{5.169}$$

For $l_{eL} = -l_{eR}$:

$$(C_{n\delta_{TLF}} + C_{n\delta_{TR}}) \cdot d_T = \frac{-l_{eL}(T_L - T_R)}{\frac{\rho}{2} SbV^2} \tag{5.170}$$

$$C_{n\delta_{TLF}} \cdot d_T = \frac{-l_{eL} T_L}{\frac{\rho}{2} SbV^2} \tag{5.171}$$

Assuming a linear relationship and $T_{LminF} = 0$:

$$C_{n\delta_{TLF}} = -\frac{l_{eL} \cdot T_{LmaxF}}{\frac{\rho}{2} SbV^2 d_{T\,max}} \tag{5.172}$$

where l_{eL} is the distance from the aircraft center of gravity to the axis of the left engine (negative for L, positive for R). The yawing moment is compensated by the rudder such that:

$$(C_{n\delta_{TLF}} + C_{n\delta_{TR}}) \cdot d_T = -C_{n\delta_r} \cdot \delta_{rPF} \tag{5.173}$$

and
$$\delta_{rPF}(d_T) = -\frac{C_{n\delta_{TLF}} + C_{n\delta_{TR}}}{C_{n\delta_r}} d_T \tag{5.174}$$

The relationship with the EDIV for this pseudo-failure is:

$$d_r = k_r \delta_r \tag{5.175}$$

The nominal directional control range is:

$$d_{r\,max} = k_r \delta_{r\,max} \tag{5.176}$$

$$d_{r\,min} = k_r \delta_{r\,min} \tag{5.177}$$

Under the loss of efficiency failure, the new range for the directional control is:

$$d_{r\,max\,F}(d_T) = k_r(\delta_{r\,max} - \delta_{rPF}(d_T)) \tag{5.178}$$

$$d_{r\,min\,F}(d_T) = k_r(\delta_{r\,min} - \delta_{rPF}(d_T)) \tag{5.179}$$

Note that this effect is similar to the throttle failure. The new range of d_r can be used with the (d_r, *r*) projection to determine the effects on *r* of the engine loss of efficiency failure.

5.6. Performance of Flight Envelope Assessment Under AC

5.6.1. Evaluation Metrics for Flight Envelope Assessment

Assumed nominal ranges for flight envelope relevant variables (ERV) have been determined based on simulator tests. Verification tests under AC have been performed for each specific subsystem failure attempting to achieve the same tasks and level of maneuverability as in the nominal tests.

To achieve desirable performance of the post-failure envelope prediction methodology, data points acquired through properly designed verification tests under abnormal conditions must stay within the predicted range and be close to the predicted limits. Therefore, for performance evaluation, three metrics have been defined attempting to capture the level of prediction confidence, the level of possible range exceedance, and the level of predicted range conservativeness. The first evaluation criterion is expressed by a prediction rate (PR) calculated as the percentage of all verification data points ERV_{val} that fall inside the predicted range. If the predicted range of any *ERV* is denoted as $R_{ERV} = [\, ERV_{min} \, , \; ERV_{max} \,]$, then the total number of validation points N_V is the sum of points inside (N_R) and outside ($N_{\overline{R}}$) the predicted range:

$$N_V = N_R + N_{\overline{R}} \tag{5.180}$$

The prediction rate is then calculated as:

$$PR = \frac{N_R}{N_V} \cdot 100 \tag{5.181}$$

The level of possible range exceedance is assessed as the relative maximum amount by which points in the verification data set exceed the predicted range. The predicted range exceedance index (REI) is defined as:

$$REI = \frac{\max\limits_{N_{\overline{R}}} [min(|ERV_{val} - ERV_{max}|, |ERV_{val} - ERV_{min}|)]}{|ERV_{max} - ERV_{min}|} \cdot 100 \tag{5.182}$$

Obviously, it is desirable to have large values of *PR* and low values of *REI*. However, it should be noted that, if the predicted range is exceedingly/unrealistically large, then *PR = 100%* and *REI = 0%* without representing good performance. Therefore, the values of the previous two evaluation metrics must be considered in conjunction with a margin index (MI) that is expected to capture the level of predicted range conservativeness. MI is defined as:

$$MI = \frac{\max\limits_{N_R} [min(|ERV_{val} - ERV_{max}|, |ERV_{val} - ERV_{min}|)]}{|ERV_{max} - ERV_{min}|} \cdot 100 \tag{5.183}$$

It should be also noted that, if the verification test does not cover the nominal range of the tests used for self generation, then high values of MI do not necessarily indicate a flaw in the post-failure range prediction.

5.6.2. Flight Envelope Assessment Under Actuator Failure

Table 5.1 presents the predicted post-failure flight envelope ranges under an elevator failure along with the corresponding nominal ranges and the performance metrics defined above. Figure 5.3 shows an example of a 2-D projection in the AIS WVU Self Viewer (the 2-D analysis tool developed for the purpose of this project) with the verification test points and the corresponding predicted post-failure ranges represented by the vertical and horizontal lines. The failure considered consists of left stab locked at 8 degrees. Two different verification tests have been performed covering the altitude range of the nominal tests for Mach between 0.7 and 0.9. The results show alterations of the ranges of pitch rate, pitch acceleration, and roll rate as expected. Note that the reduced range of Mach and altitude considered when building the self is not affected by this elevator failure. The verification points are well within predicted ranges for all variables considered. Only very few points exceed the predicted ranges by small relative amounts, as captured by REI, except for pitch acceleration. The variation of this variable experiences a short-duration sudden peak at the failure occurrence moment. The duration of the peak is very short as shown by the values of the pitch rate, which remain practically within predicted range. The low values of MI show that the predicted limits of the respective variables are not unrealistically large. It should be noted that the large MI value recorded for Mach number is due to the fact that the verification tests do not cover the range between 0.5 and 0.7, which was included in the self. This is one example of a situation when large values of MI reflect the incompleteness of the verification tests rather than flaws in the range prediction process.

Table 5.1. Performance of Flight Envelope Prediction Under Stabilator Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*,%	*REI*,%	*MI*,%
H	[2012,9929]	[2012,9929]	100.0	0.0	8.0
M	[0.526,0.95]	[0.526,0.95]	99.9	1.6	32.9
q	[–0.225,0.183]	[–0.141,0.105]	100.0	2.6	0.0
$\dot{q}$	[–0.383,0.386]	[–0.23,0.32]	99.7	24.6	0.0
p	[–0.94,1.08]	[–0.90,0.79]	100.0	0.0	1.0

The predicted post-failure flight envelope ranges under the aileron failure are presented in Table 5.2. Corresponding nominal ranges and values of the three performance metrics are included. The failure consists of a right aileron locked at 8°. Two different verification tests have been performed covering an altitude range between 2,000 m and 10,000 m and a Mach number range between 0.7 and 0.9. The results show alterations of the ranges of roll rate and roll acceleration, as expected. The verification points are all within predicted ranges for both variables considered with only insignificant exceedence as indicated by *REI*. The low values of *REI* and *MI* indicate that the predicted ranges are not unrealistically large.

The predicted post-failure flight envelope ranges under the rudder failure are presented in Table 5.3. The failure consists of a right rudder locked at 8°. Two different verification tests have been performed covering the altitude and Mach ranges of the nominal tests. The results show alterations of the ranges of yaw rate and yaw acceleration, as expected. The verification points are all within predicted ranges for both variables considered. The relatively low values of *MI* lead to the conclusion that the predicted limits of the respective variables are not unrealistically large.

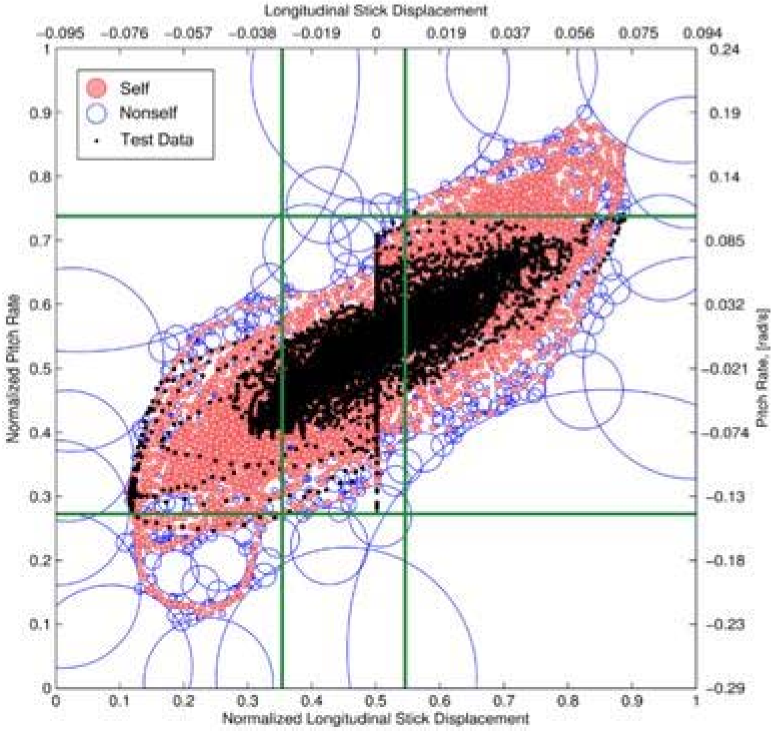


Figure 5.3. Post-Failure Range of the Pitch Rate Under Stabilator Failure

Table 5.2. Performance of Flight Envelope Prediction Under Aileron Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
p	[–0.94, 1.08]	[–0.77, 1.01]	99.94	5.6	1.2
$\dot{p}$	[–2.95, 3.1]	[–2.98, 2.94]	99.98	1.5	0.0

Table 5.3. Performance of Flight Envelope Prediction Under Rudder Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
r	[–0.064, 0.062]	[–0.047, 0.047]	100.0	0.0	7.8
$\dot{r}$ P	[–0.117, 0.126]	[–0.114, 0.104]	100.0	0.0	10.4

5.6.3. Flight Envelope Assessment Under Sensor Failure

Table 5.4 presents the predicted post-failure flight envelope ranges under the roll rate sensor failure. The failure consists of a 5°/sec bias in the roll rate sensor output. Two different verification tests have been performed covering the entire altitude and Mach ranges of the nominal tests. The results show alterations of the ranges of roll rate and roll acceleration, as expected. The verification points are all within predicted ranges for all variables considered with only insignificant exceedance in the roll attitude angle, as reflected by the corresponding *REI* value. The relatively low values of *MI* show that the predicted limits of the respective variables are not unrealistically large.

Table 5.4. Performance of Flight Envelope Prediction Under Roll Rate Sensor Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
p	[–0.94, 1.08]	[–0.82, 1.1]	100.0	0.0	15.3
$\dot{p}$ P	[–2.95, 3.1]	[–1.8, 3.13]	100.0	0.0	15.5

Listed in Table 5.5 are the predicted post-failure flight envelope ranges under the pitch rate sensor failure. The failure consists of a 5°/sec bias in the pitch rate sensor output. Four different verification tests have been performed covering the entire altitude and Mach ranges of the nominal tests. The results show alterations of the ranges of pitch rate and pitch acceleration, as expected. The high values of *PR* show that the majority of the verification points are within the predicted ranges for all variables considered. Although the values of *REI* for the pitch acceleration and pitch attitude angle are relatively high, very few verification points exceed the corresponding predicted limits, as indicated by the corresponding *PR* values. The relatively low values of *MI* lead to the conclusion that the predicted limits of the respective variables are not unrealistically large.

Table 5.5. Performance of Flight Envelope Prediction Under Pitch Rate Sensor Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
q	[–0.225, 0.183]	[–0.364, 0.153]	99.8	8.7	13.0
$\dot{q}$ P	[–0.383, 0.386]	[–0.417, 0.32]	99.9	30.9	6.0

The predicted post-failure flight envelope ranges under the yaw rate sensor failure are listed in Table 5.6. The failure consists of a 5°/sec bias in the yaw rate sensor output. Four different verification tests have been performed covering the entire altitude and Mach ranges of the nominal tests. The results show alterations of the ranges of yaw rate and yaw acceleration, as expected. Although a rudder failure would be expected to affect sideslip capability, this effect is not captured because the sideslip range considered for normal operation was already limited within the capability under failure of such low severity. The verification points are all within predicted ranges for all variables considered with only insignificant exceedance as reflected by *REI*. The simultaneous occurrence of non-zero values of *REI* and *MI* reflect the situation when one limit of the predicted range is exceeded and the other is not reached during the verification tests. The relatively low values of *MI* lead to the conclusion that the predicted limits of the respective variables are not unrealistically large.

Table 5.6. Performance of Flight Envelope Prediction Under Yaw Rate Sensor Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
r	[–0.064, 0.062]	[–0.08, 0.0477]	100.0	1.0	10.7
$\dot{r}$ P	[–0.117, 0.126]	[–0.093, 0.128]	100.0	4.8	5.8
β	[–9.6, 9.8]	[–9.6, 9.8]	100.0	0.7	16.9

5.6.4. Flight Envelope Assessment Under Structural Damage

The predicted post-failure flight envelope ranges under a wing damage affecting 15% of the left surface are listed in Table 5.7. Two verification tests have been performed covering the altitude and Mach number ranges of the nominal tests. The results show alterations of the ranges of pitch rate, pitch acceleration, roll rate, and roll acceleration, as expected. Note that the reduced range of Mach and altitude considered when building the self is not affected by this failure. The verification points are well within the predicted ranges for all variables considered. Only very few points exceed the predicted ranges by small relative amounts, as captured by *REI*. The relatively low values of *MI* lead to the conclusion that the predicted limits of the respective variables are not unrealistically large.

Table 5.7. Performance of Flight Envelope Prediction Under Wing Damage

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
H	[2012, 9929]	[2012, 9929]	99.7	1.8	9.0
M	[0.526, 0.95]	[0.526, 0.95]	99.9	2.2	10.3
p	[–0.94, 1.08]	[–1.12, 0.51]	99.9	4.3	7.6
$\dot{p}$ P	[–2.95, 3.1]	[–2.97, 2.5]	99.9	2.5	0.0
q	[–0.225, 0.183]	[–0.05, 0.064]	100.0	0.0	1.2
$\dot{q}$ P	[–0.383, 0.386]	[–0.091, 0.072]	99.8	9.7	3.7

Table 5.8. presents the predicted post-failure flight envelope ranges under a horizontal tail damage affecting 60% of the left surface. One verification test has been performed covering an altitude range between 6,000 m and 10,000 m and a Mach number range between 0.7 and 0.9. The results show alterations of the ranges of pitch rate, pitch acceleration, roll rate, and roll acceleration, as expected. Note that the reduced range of Mach and altitude considered when building the self is not affected by this horizontal tail damage. The verification points are well within the predicted ranges for all variables considered. Only very few points exceed the predicted ranges by small relative amounts, as captured by *REI*. Relatively large values of *MI* are recorded, which seem to indicate that the maneuvers performed were not as aggressive as for the nominal conditions. Note that the ranges of Mach and altitude for the verification test were limited, explaining the large values of *MI* for these two variables.

Table 5.8. Performance of Flight Envelope Prediction Under Horizontal Tail Damage

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
H	[2012, 9929]	[2012, 9929]	99.0	2.0	40.3
M	[0.526, 0.95]	[0.526, 0.95]	98.5	0.8	35.5
q	[–0.225, 0.183]	[–0.327, 0.165]	100.0	0.0	38.2
$\dot{q}$ P	[–0.383, 0.386]	[–0.37, 0.355]	100.0	0.0	9.1
p	[–0.94, 1.08]	[–0.925, 1.09]	100.0	0.0	18.1
$\dot{p}$ P	[–2.95, 3.1]	[–2, 3.13]	100.0	0.0	24.5

5.6.5. Flight Envelope Assessment Under Propulsion System Failure

Table 5.9 presents the predicted post-failure ranges of relevant variables with left engine throttle locked at idle. Corresponding nominal ranges and the three performance metrics are also included. One verification test has been performed covering an altitude range between 6,000 m and 10,000 m for Mach numbers between 0.7 and 0.9. The results show alterations of the ranges of the longitudinal and vertical acceleration, and roll rate, as expected. Note that the reduced range of Mach and altitude considered when building the self is not affected by this failure. The verification points are well within predicted ranges for all variables considered. Only few points exceed the predicted ranges by small relative amounts, as captured by *REI*. The relatively large values of *MI* are the result of the verification test not completely covering the maneuver range of the self-generating data. In particular, the verification test does not include maneuvers between 2,000 m and 6,000 m or for Mach numbers between 0.5 and 0.7.

Table 5.9. Performance of Flight Envelope Prediction Under Engine Failure

ERV	**Nominal Range**	**Post-Failure Range**	*PR*, %	*REI*, %	*MI*, %
a_x	[–0.344, 0.597]	[–0.344, 0.353]	99.9	4.8	20.4
a_z	[–2.46, 3.05]	[–2.46, 2.16]	99.6	10.3	10.4
r	[–0.0637, 0.0621]	[–0.0637, 0.0393]	99.9	5.3	21.5
H	[2012, 9929]	[2012, 9929]	95.0	3.4	50.0
M	[0.526, 0.95]	[0.526, 0.95]	100.0	0.0	54.1

Chapter 6

Investigation of Immunity-based Accommodation of Aircraft Subsystem Abnormal Conditions

6.1. Immunity Approaches for AC Accommodation

The immunity-based AC accommodation can be approached based on two concepts. First, the biological feedback that establishes a balance between the activation and suppression of the antibodies generation can be converted into an adaptive mechanism augmenting a baseline controller[20,21]. For the biological system, it can be considered that the activation *aTh* of the helper T-cells is governed by the balance between the amount of antigens *a(t)* and the action of antibodies *u* generated previously at $t-\tau$ and whose antiseptic action is modeled by a function F_1; therefore:

$$aTh(t) = a(t) - F_1(u(t-\tau)) \tag{6.1}$$

The delay τ models the time needed for the antibodies to mature and function. The generation of the helper T-cells (T_h) depends on the activation *aTh*:

$$T_h(t) = F_2(aTh(t)),\ \frac{dF_2}{daTh} > 0 \tag{6.2}$$

The production of the B-cells is the result of the combined effects of helper T-cells and suppressor T-cells:

$$B(t) = F_3(T_h(t) - T_s(t))) \tag{6.3}$$

The suppressor T-cell generation depends on the balance between the amount of antigens and antibodies and the rate at which the currently matured antibodies were generated $u'(t-\tau)$:

$$T_s(t) = K_s F_4(u'(t-\tau))aTh(t) \tag{6.4}$$

Finally, the antibodies are generated by activated B-cell; therefore:

$$u(t) = F_5(B(t)) \tag{6.5}$$

The immune system mechanisms are not completely understood; therefore, the parameters of this model, the functions F_1 through F_5, τ and K_s must be determined heuristically, which is not trivial. The block diagram of the immune system feedback model is presented in Figure 6.1.

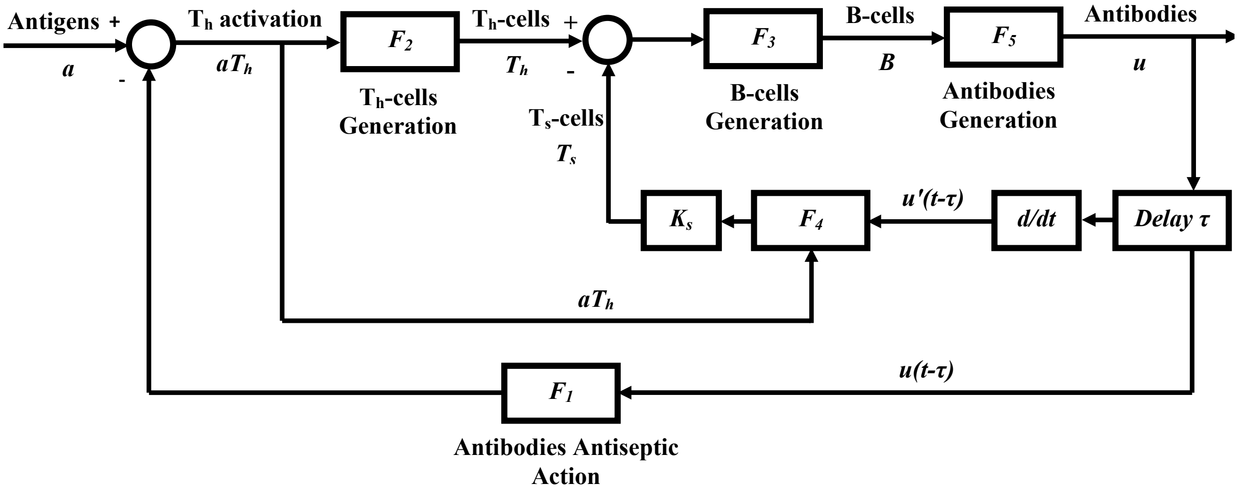


Figure 6.1. Block Diagram of the Immune System Feedback Model

The second conceptual approach for using the AIS paradigm for control purposes relies on the assumption that the classification capabilities of the AIS can be extended[22] and used not only to detect the problem (AC) but also to select or find the solution, which is mitigating the AC effects. The control action is considered as a mapping between control variables and controlled variables under specific performance requirements or constraints. Therefore, if the AIS is to be used for control purposes, both control and controlled variables must be part of the feature set. If that is the case, the AIS framework provides the relationship between the control and controlled variables for the normal conditions and for those AC for which detailed structuring of the non-self is available. Alternatively, the control variables associated with the adaptive immune feedback system and with the best compensation performance under specific upset conditions may be differentiated into memory cells, which can mount a faster and more aggressive secondary response in future encounters with the same AC.

Let us assume that we try to control variable y using the control variable x, which are both part of the feature set. Let us also assume that structuring of the non-self is available such that a mapping between x and y is established under AC. For any desired value of the controlled variable y_d, corresponding clusters c_k can be determined/extracted from the structured non-self as illustrated in Figure 6.2. Further processing of the clusters c_k is necessary to produce compensatory commands x_{cmd}. These commands can be typically obtained by considering additional dimensions of the feature hyper-space, which may eventually result in one command detector, let us say c_K. An operator may still be needed to convert the command detector/cluster into a numerical value x_{cmd}, which can be formulated as:

$$x_{cmd} = O(c_K) \tag{6.6}$$

Depending on the resolution of the non-self (in other words, the size of the detectors/compensators) the operator $O(*)$ can simply represent the center of the compensator or a more involved computation providing values inside the compensator.

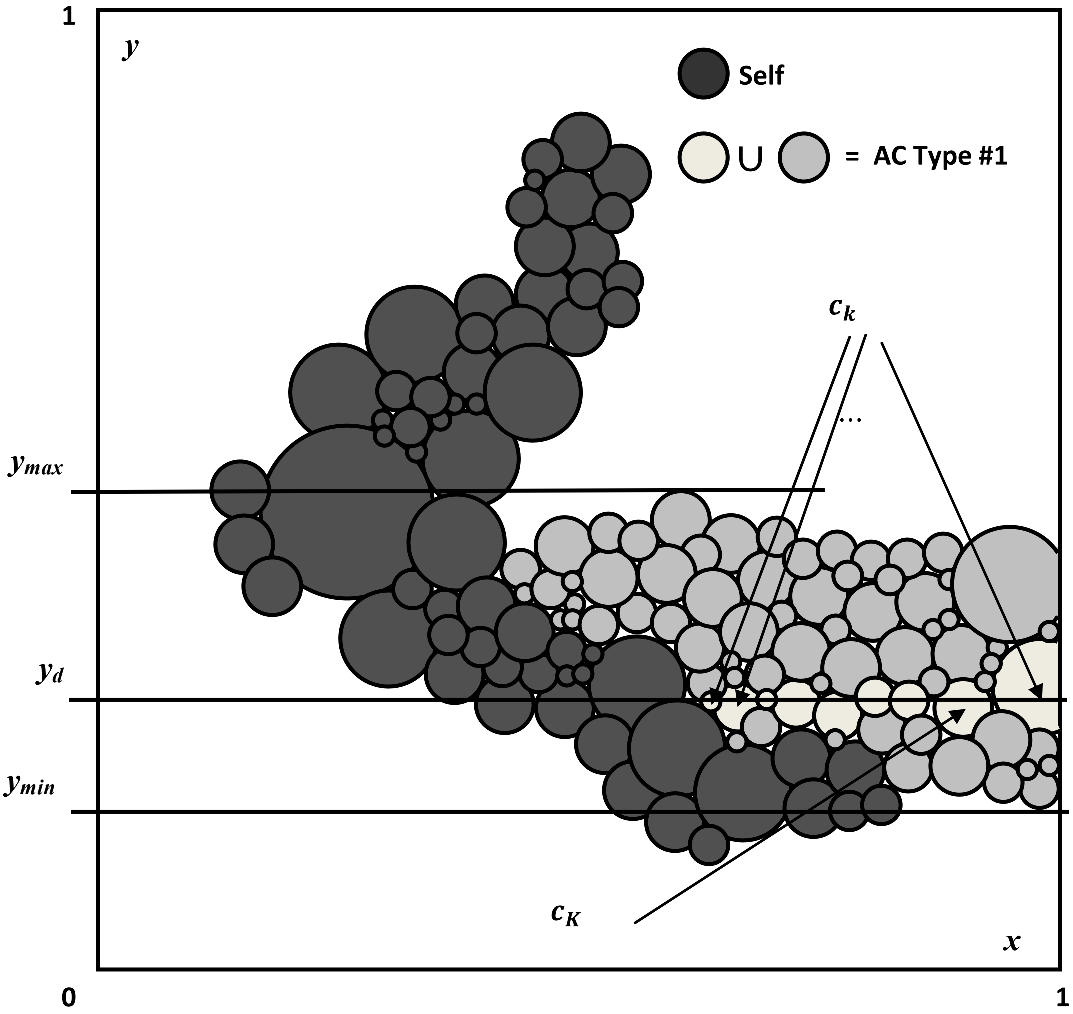


Figure 6.2. Failure Accommodation Based on Structured Non-Self

6.2. The Humoral Immune Mechanism for Abnormal Condition Accommodation

6.2.1. General Control Architecture

A novel approach for direct adaptive control laws was investigated based on the humoral immune mechanism. Specifically, the proposed adaptive element is used to augment a baseline controller providing improved capabilities for handling subsystem ACs. The general block diagram of the humoral immune mechanism is presented in Figure 6.3. Three control architectures have been considered and compared in this study: a baseline controller (without adaptation), baseline controller augmented with ANN, and the baseline controller augmented with the artificial immune system-based mechanism.

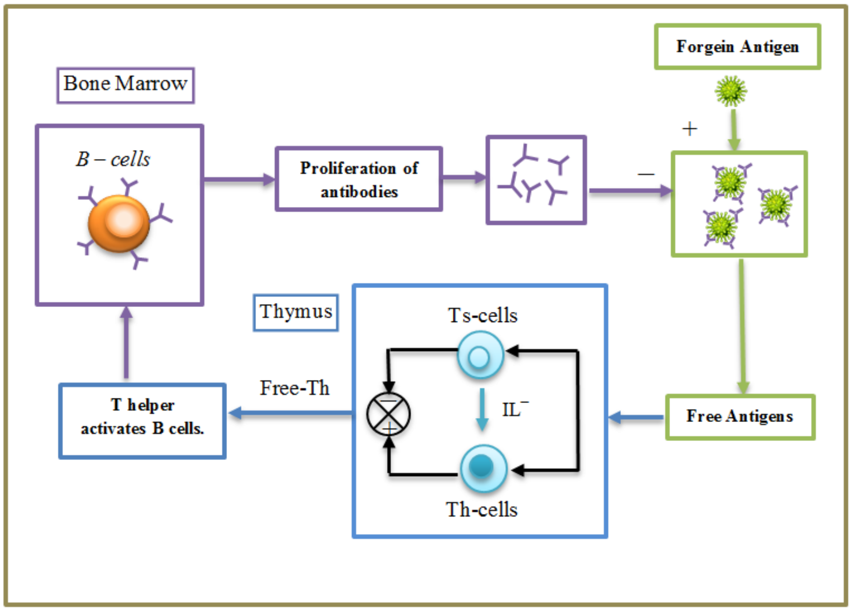


Figure 6.3. Humoral Immune System Representation

The baseline controller relies primarily on a model reference architecture that uses the pilot stick inputs to generate desired angular rate and angular acceleration commands that are then used to produce compensating commands based on a proportional-integral-derivative (PID) approach in conjunction with approximate non-linear dynamic inversion (NLDI). The structure of the baseline control architecture[24] is presented in Figure 6.4.

A second architecture considered here consists of the baseline control laws augmented with ANN. The neural networks are used in conjunction with the output from the PID compensation, the states of the aircraft and the reference angular rates and accelerations[24], as shown in the control architecture of Figure 6.5. The input to the ANN is $x=\left[V\ H\ \alpha\ \beta\ \omega\ \omega_{ref}\ \bar{U}_{\omega}\right]^{T}$, where V is the aircraft velocity, H is the altitude, α is the angle of attack, and β is the sideslip angle. These four inputs are the same on all three channels. For the longitudinal channel, $\omega=q$, while for the lateral and directional channels, $\omega=[p\ r]$. Only on-channel variables are considered for ω_{ref} and $\bar{U}_{\omega}$. Finally, $\bar{U}_{\omega}$ is defined as $\bar{U}_{\omega}=\dfrac{1-e^{(\hat{y}-U_{\omega})}}{1+e^{(\hat{y}-U_{\omega})}}$, with $\hat{y}$, the previous output of the ANN.

Finally, the AIS is used to augment the PID part of the control laws. The augmented form of the controller using the AIS paradigm leads to the following expression in its discrete version:

$$u(k) = k_p \left(1 + \frac{k_i}{z-1} + k_d \frac{z-1}{z}\right)[1 - \eta f(\Delta u(k))]e(k) \tag{6.7}$$

The function that expresses the change in concentration of the B cells will be given by:

$$f(\Delta u(k)) = 1 - \frac{2}{e^{[\Delta u(k)^2]a} + e^{-[\Delta u(k)^2]a}} \tag{6.8}$$

where a is another parameter that will modulate the adequate response of the system under abnormal conditions.

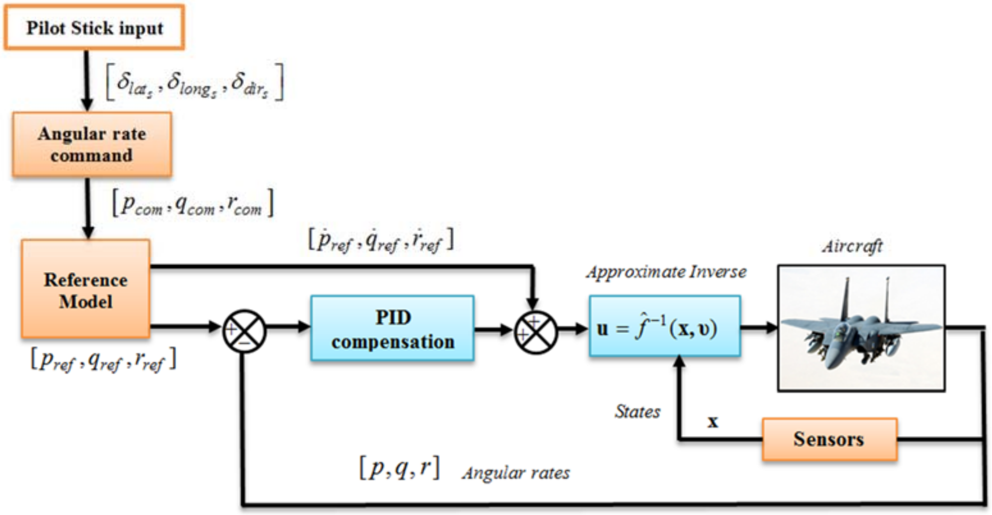

Pilot Stick input
$[\delta_{lat_s}, \delta_{long_s}, \delta_{dir_s}]$
Angular rate command
$[p_{com}, q_{com}, r_{com}]$
$[\dot{p}_{ref}, \dot{q}_{ref}, \dot{r}_{ref}]$
Aircraft
Approximate Inverse
Reference Model
PID compensation
$\mathbf{u} = \hat{f}^{-1}(\mathbf{x}, \upsilon)$
$[p_{ref}, q_{ref}, r_{ref}]$
States
x
Sensors
$[p, q, r]$
Angular rates

Figure 6.4. General Block Diagram of the Baseline Controller

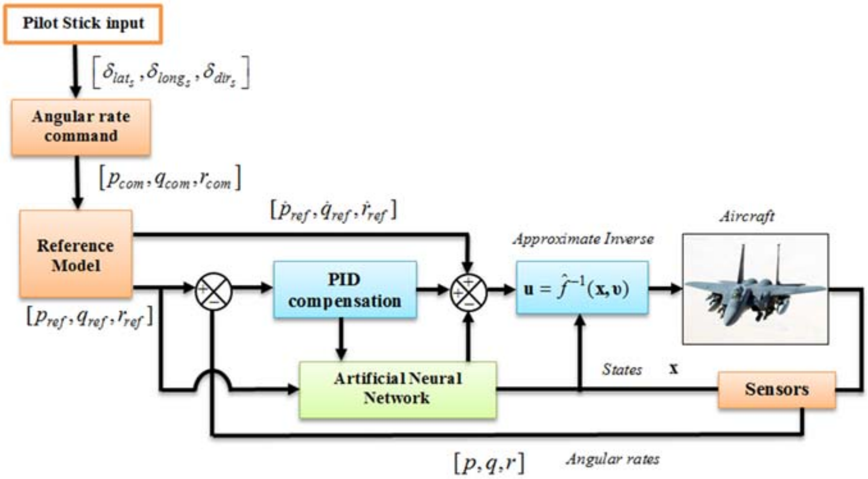

Pilot Stick input
$[\delta_{lat_s}, \delta_{long_s}, \delta_{dir_s}]$
Angular rate command
$[p_{com}, q_{com}, r_{com}]$
$[\dot{p}_{ref}, \dot{q}_{ref}, \dot{r}_{ref}]$
Aircraft
Approximate Inverse
Reference Model
PID compensation
$\mathbf{u} = \hat{f}^{-1}(\mathbf{x}, \upsilon)$
$[p_{ref}, q_{ref}, r_{ref}]$
Artificial Neural Network
States
x
Sensors
$[p, q, r]$
Angular rates

Figure 6.5 General Block Diagram of the Baseline controller (PID+NLDI) Augmented with ANN

The following relationships are satisfied by definition for the parameters k_p, k_i, and k_d and η:

$$(k_p, k_i, k_d) > 0, \text{ and } \eta \geq 0 \tag{6.9}$$

Since the control laws perform negative feedback control action if $0 < \eta f[\Delta u(k)] \leq 1$, and positive feedback control action if $1 < \eta f[\Delta u(k)]$, an upper limit of the factor η keeps the control system stable. Assuming the existence of parameters k_{p0}, k_{i0}, and k_{d0}, which ensure the stability of the control system with the conventional PID controller, the stability condition of the immune control algorithm is given by:

$$0 < k_p \leq k_{p0}, \quad 0 < k_p \leq k_{p0}, \quad 0 < k_p \leq k_{p0} \tag{6.10}$$

and
$$0 \leq \eta \leq \frac{1}{\sup f[\Delta u(k)]} \quad \text{for all } k \tag{6.11}$$

Consequently, if the parameter η satisfies Eq. (6.11), the control system is guaranteed within the stable state of the conventional PID controller. The inner loop controller scheme based on the immune augmentation is shown in the Figure 6.6.

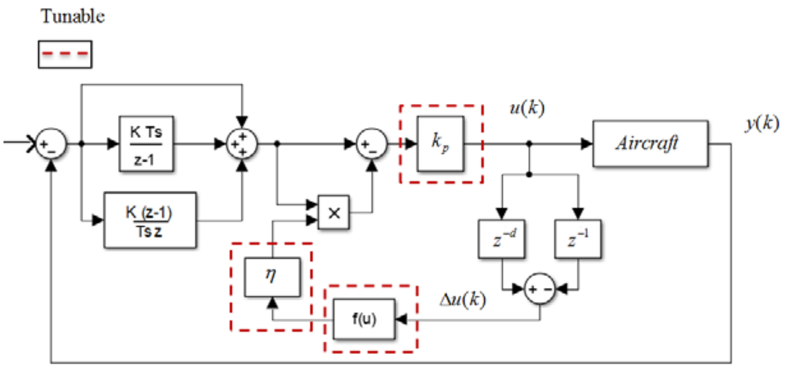


Figure 6.6. Block Diagram of the PID-AIS-based Adaptive Mechanism

Parameters in red in Figure 6.6 must be tuned to modulate a proper response of the AIS on each channel. There will be a total of nine tunable parameters for the AIS for the roll, pitch, and yaw angular rates channels:

$$\begin{aligned} K_{AISp} &= [K_p, \eta_p, a_p] \\ K_{AISq} &= [K_q, \eta_q, a_q] \\ K_{AISr} &= [K_r, \eta_r, a_r] \end{aligned} \tag{6.12}$$

Figure 6.7 illustrates the control scheme architecture of the baseline controller extended with the AIS.

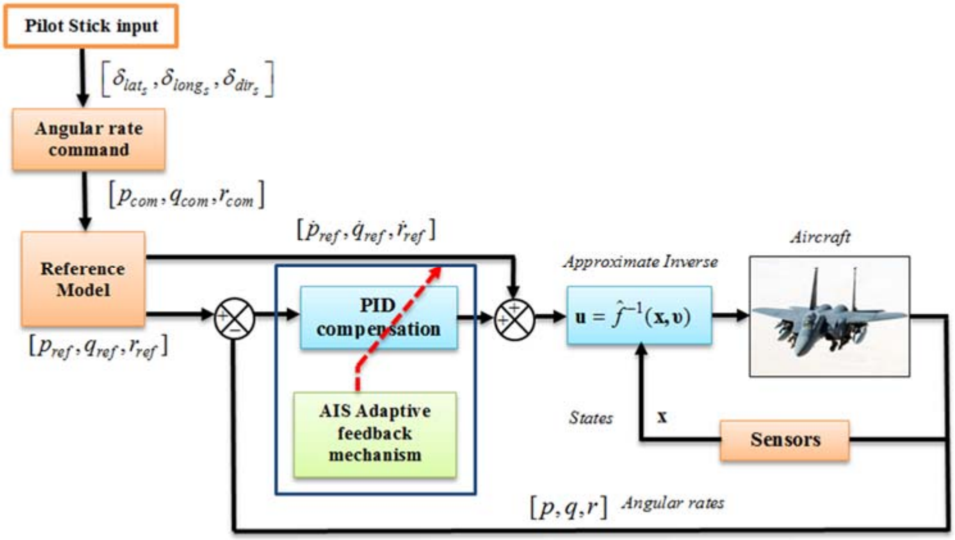

Pilot Stick input
$[\delta_{lat_s}, \delta_{long_s}, \delta_{dir_s}]$
Angular rate command
$[p_{com}, q_{com}, r_{com}]$
$[\dot{p}_{ref}, \dot{q}_{ref}, \dot{r}_{ref}]$
Reference Model
$[p_{ref}, q_{ref}, r_{ref}]$
PID compensation
AIS Adaptive feedback mechanism
Approximate Inverse
$\mathbf{u} = \hat{f}^{-1}(\mathbf{x}, \upsilon)$
Aircraft
States
x
Sensors
$[p, q, r]$ Angular rates

Figure 6.7. Baseline Controller Augmented with AIS-based Mechanism

6.2.2. Analysis of AIS Compensation Performance

In order to obtain quantitative assessments of the performance of the different control architectures, specific performance metrics were defined in terms of the total pilot input activity, angular rate tracking errors, and the total amount of work required by each of the control surfaces. Extensive simulation tests were performed using both a desktop setup as well as the motion-based flight simulator. The nine parameters of the AIS augmentation have been optimized using an evolutionary algorithm.

The comparison between the different control configurations shows that, in general, the baseline controller augmented with the AIS has better failure rejection capabilities than the baseline + ANN. Both augmented sets of control laws perform better than the baseline (see Figure 6.8).

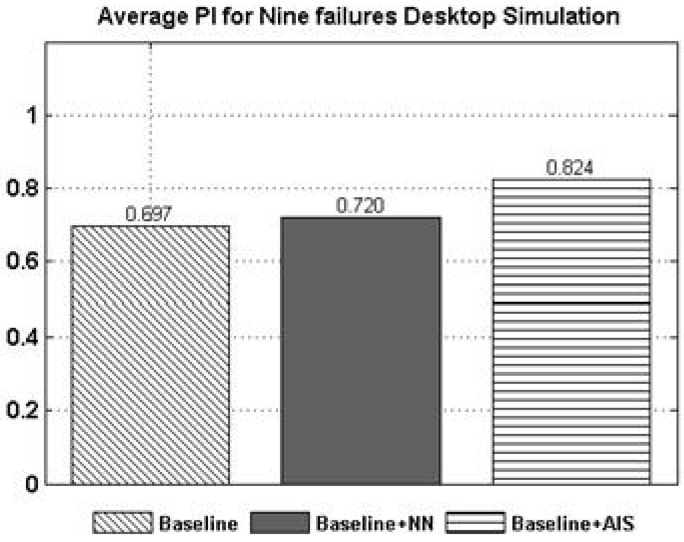


Figure 6.8. Results of Average PI for Different ACs

The level of performance improvement is illustrated in Figure 6.9 and 6.10 showing attitude responses in roll and pitch, respectively, when attempting to keep constant wing level and constant heading after a left stabilator high magnitude failure is injected at 2 seconds. These results are also supported by Figure 6.11 where the pilot activity history is plotted against time for the different control architectures considered. It can be seen that the pilot is required to exercise less activity controlling the aircraft when the AIS is part of the control architecture.

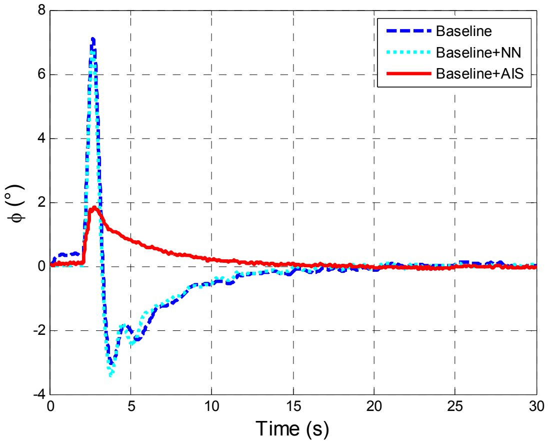


Figure 6.9. Roll Response Under High Magnitude Stabilator Failure

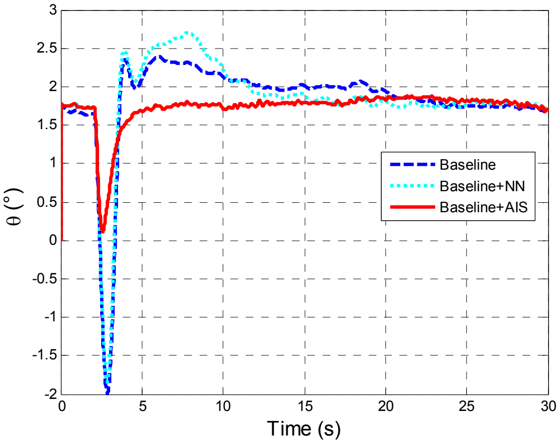


Figure 6.10. Pitch Response Under High Magnitude Stabilator Failure

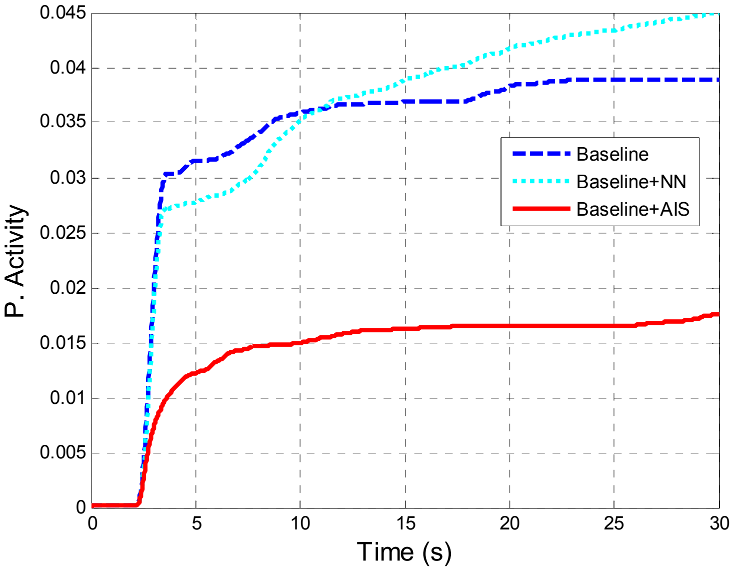


Figure 6.11. Total Pilot Activity Under High Magnitude Stabilator Failure

On the average, the AIS augmentation achieves better tracking errors with similar level of control activity. Figure 6.12 shows the average PI for angular rate tracking errors over all failures for the three control architectures.

It should be noted that the design complexity and processing requirements of the ANN makes the AIS more desirable from the point of view of implementation and efficiency in terms of hardware task execution time, as well. A simple test was designed and executed in order to compare the required hardware time for each of the approaches. The test was repeated for a thousand simulations on a computer with characteristics listed in Table 6.1. The average results are presented in Table 6.2. Figures 6.13 presents the histogram of the task execution time and Figure 6.14 presents the time per simulation versus for all simulation performed. The computational efficiency of the AIS version of the control laws can be easily inferred from these results.

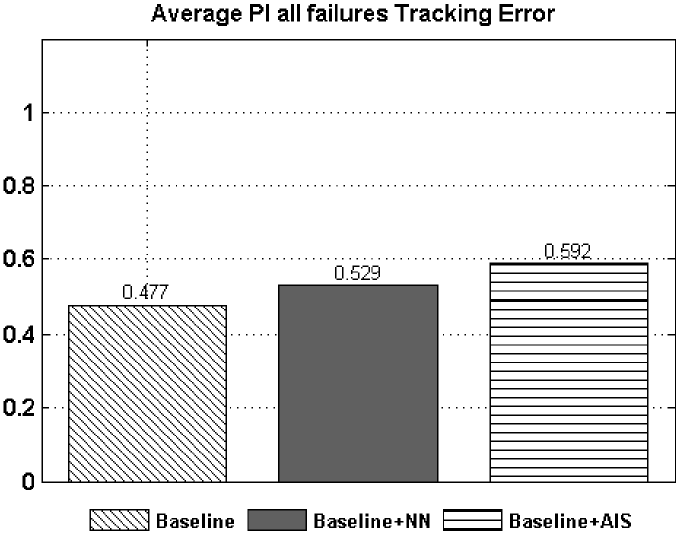


Figure 6.12. Average Performance Index for Tracking Error

Table 6.1. Computer Characteristics for Execution Time Assessment

Computer Specifications		**Simulation setting**	
Model	Dell XPS L501X	N of simulations	1000
Processor	Intel Core i5 @ 2.67GHz	Solver	Discrete, fixed step
Ram memory	6 GB	Sampling time	0.02 seconds

Table 6.2. Task Execution Time Assessment

Time to run a hundred seconds simulation	
T mean AIS	0.1652 seconds
T mean NN	0.3463 seconds
Ratio AIS:NN	2.09

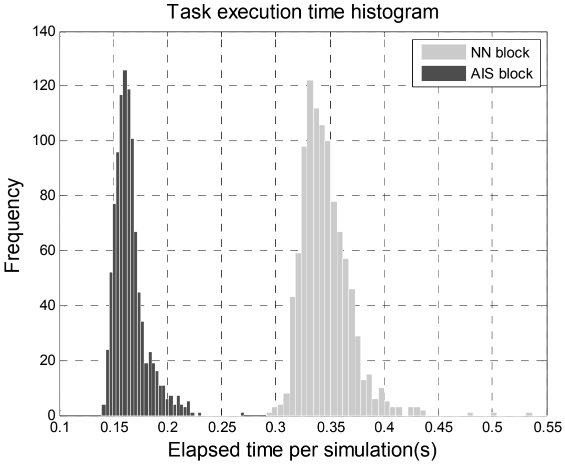


Figure 6.13. Task Execution Time Histogram

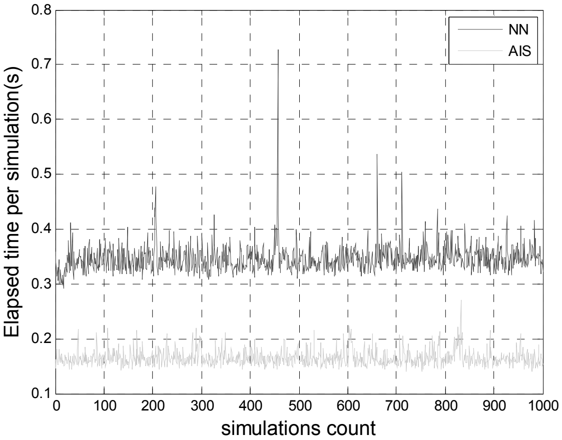


Figure 6.14. Time per Simulation vs Simulation Count

6.3. Structured Self/Non-self for Control

The SNSA for control attempts to use the AIS already built for ACDIE purposes. It should be noted that two scenarios can be envisioned within the AIS paradigm for control. The first scenario, assumes that substantial information about the failure and its compensation is available and already stored in the AIS. This can be achieved by memorizing the dynamic fingerprint of the AC during first encounter situations, by generating and recording pilot compensation during actual or simulated failures. The second scenario involves an unknown failure that requires a specific new compensation. It is expected that compensation can be produced by properly composing bits of information within the self and non-self.

Within this project, a preliminary investigation has been performed with the objective of assessing the possibility of extracting useful control/compensation information out of properly structured AIS. This general feasibility study was performed using data collected from the motion-based flight simulator at normal and failure conditions for two simple maneuvers: uniform symmetric climb/descent and coordinated turn. The failures considered were actuator locked (stab, aileron, and rudder) and pitch rate sensor bias. Several simplifying assumptions were made for this first step. The information about which subsystem failed, what part of subsystem is affected, and what is the nature and magnitude of the failure must be available. Figure 6.15 illustrates the concept of extracting pilot compensatory commands followed by implementing those commands to the same aircraft, which will be referred to as accommodation. In order to solve the accommodation problem, a set of specific steps must be accomplished as illustrated in Figure 6.16.

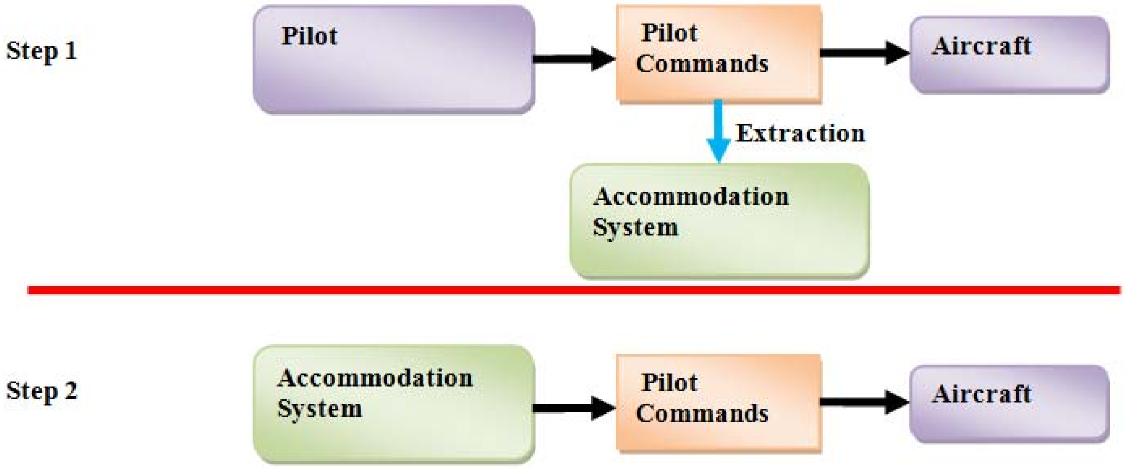


Figure 6.15. Failure Accommodation Definition

6.3.1. Formulation of Mathematical Model

A set *FF* of *flight features* must be selected such that the dynamic fingerprint of the aircraft operation is accurately captured at normal and abnormal conditions. *FF* can be defined as a vector with *Nf* components:

$$FF = \{ feat_1 \ , \ feat_2 \ , \ldots , feat_{N_f} \} \tag{6.13}$$

where *feat*$_j$ *(j=1,2, …, N*$_f$*)* represents the feature (e.g. roll rate, velocity). An input to the pilot (or mission objectives) can be defined as a matrix *REF* whose rows are represented by vectors *REF*$_v$ of length *N*$_f$:

$$REF_v = [df_1,\ df_2, \dots, df_{Nf}] \tag{6.14}$$

Each element *df*$_j$ of the vector *REF*$_v$ represents a desirable value of the corresponding feature in set *FF*. Duration of the flight is defined by the number of time samples *ts*. Consequently, matrix *REF* consists of *ts* rows.

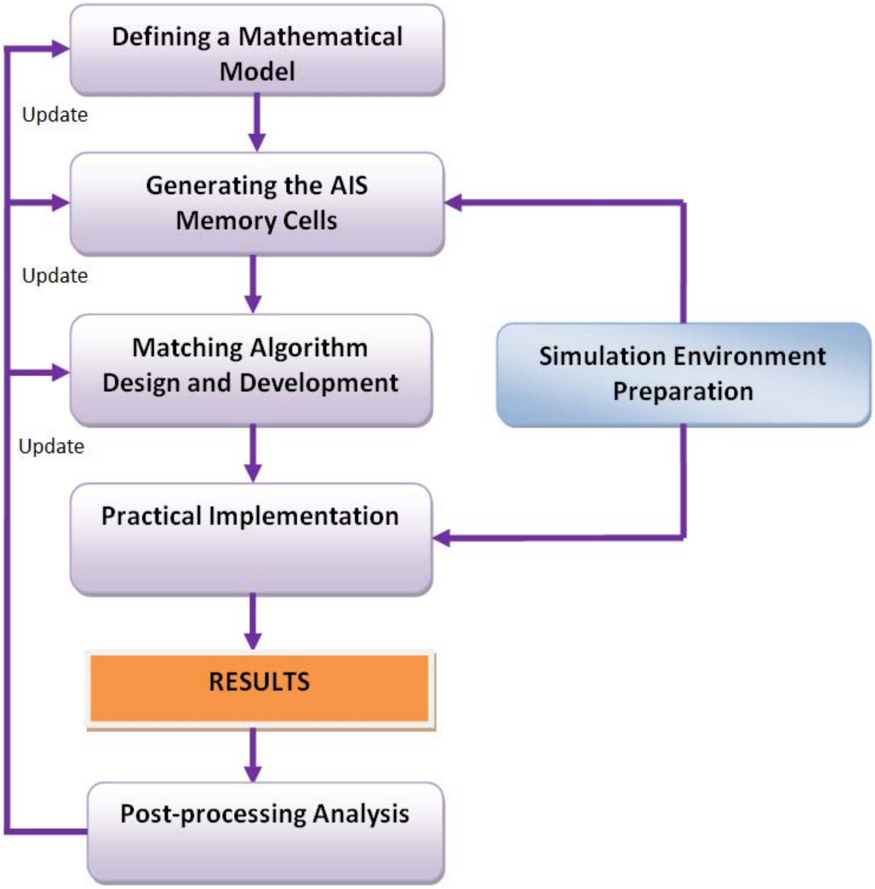

Defining a Mathematical Model
Update
Generating the AIS Memory Cells
Update
Matching Algorithm Design and Development
Simulation Environment Preparation
Update
Practical Implementation
RESULTS
Post-processing Analysis

Figure 6.16. Accommodation Problem Main Components

An outcome of pilot's performance is a matrix *PF*. Every row in this matrix is represented by a vector *PF*$_v$ of length *N*$_f$:

$$PF_v = [pf_1, pf_2, \dots, pf_{N_f}] \tag{6.15}$$

where each element *pf*$_j$ of the vector *PF*$_v$ represents a value of the corresponding feature of set *FF* obtained from pilot performance. Matrix *PF* consists of *ts* rows as well. Pilot generated control commands are defined by matrix *PC*. Every row in this matrix is represented by a vector *PC*$_v$ of length *N*$_{cc}$, where *N*$_{cc}$ is the number of control commands:

$$PC_v = [ctrl_1, ctrl_2, \dots, ctrl_{N_{cc}}] \tag{6.16}$$

where element *ctrl_j* of the vector PC_v represents a control command provided by pilot at each time sample during flight. Matrix *PC* consists of *ts* rows. The vector of measured in-coming values of features at every time sample is defined as INC_v. The length of the vector is N_f.

$$INC_v = [if_1, if_2, \dots, if_{N_f}] \tag{6.17}$$

Each element if_j of the vector INC_v represents an in-coming value of the corresponding feature of set *FF*.

6.3.2. Generating AIS Memory

There is a certain flight path that the aircraft should follow. There are also specific flight conditions that the aircraft should maintain during the flight (e.g. constant altitude, constant velocity). These parameters define the task and are considered as the input to the pilot+aircraft system. After being informed or making the decision about this flight path and about the specific flight conditions, an experienced pilot intends to execute the task as accurate as possible. An aircraft following the desirable flight path under the commanded flight conditions will be assimilated to a healthy organism. Every value of in-coming flight variable that alters from the desirable value during the pilot performance will be defined as invading entity/antigen. The control commands provided by the pilot that are trying to bring the aircraft to the desirable state will be defined as immune system antibodies. This process is an analogy to the immune system being affected by a disease for the first time. The adaptive immune system fights back and generates antibodies, which eventually eliminate the disease. The recorded and saved flight data will define a set of created artificial memory cells (B-cells, T-cells, and antibodies) and antigens.

The set of antigens is defined as the difference between in-coming data and reference input (pilot mission at each time sample *tsc*.

$$AG_v = REF_v - INC_v \tag{6.18}$$

$$AG_v = [df_1 - if_1, df_2 - if_2, \dots , df_{Nf} - if_{Nf}] \tag{6.19}$$

Depending on flight scenarios, the content of the antigen cell might be extended to include not only values of differences but actual values of incoming features as well:

$$AG_v = [df_1 - if_1, \dots , df_{Nf} - if_{Nf}, if_1, \dots, if_{Nf}] \tag{6.20}$$

These additional features are selected depending on the specific task and prescribed maneuver. The collection of AG_v vectors over the entire flight time defines the matrix *AG*. The overall number of time samples during the flight is *ts*, consequently matrix *AG* has *ts* rows.

The immune system memory has information about the T-cells and B-cells produced during the prior infection encounters, and antibodies that were used to suppress these antigens. Within the AIS paradigm, T-cells together with B-cells will be assimilated to a difference between pilot features data and reference input data. The set of *T-cells and B-cells* for a specific time sample *tsc* is a vector TB_v defined by subtraction of elements of vector PF_v from corresponding elements of vector REF_v.

$$TB_v = REF_v - PF_v \tag{6.21}$$

$$TB_v = [df_1 - pf_1, df_2 - pf_2, \dots , df_{Nf} - pf_{Nf}] \tag{6.22}$$

Similarly to *AG*$_v$ vector, a *TB*$_v$ vector might need an extension and include some actual values of features obtained from pilot performance:

$$TB_v = [df_1 - pf_1, \dots, df_{Nf} - pf_{Nf}, pf_1, \dots, pf_{Nf}] \tag{6.23}$$

The additional features for *TB*$_v$ should be the same as for *AG*$_v$. The collection of *TB*$_v$ vectors over the entire flight time defines the matrix *TB*. The overall number of time samples during the flight is *ts*, consequently, matrix *TB* has *ts* rows.

The set of *antibodies* for a specific time sample *tsc* is a vector *AB*$_v$ defined by the corresponding vector from matrix PC.

$$AB_v = PC_v \tag{6.24}$$

The collection of *AB*$_v$ vectors over the entire flight time defines matrix *AB*. Considering all definitions and notations in this section, the generic structure of the AIS memory is presented in Figure 6.17.

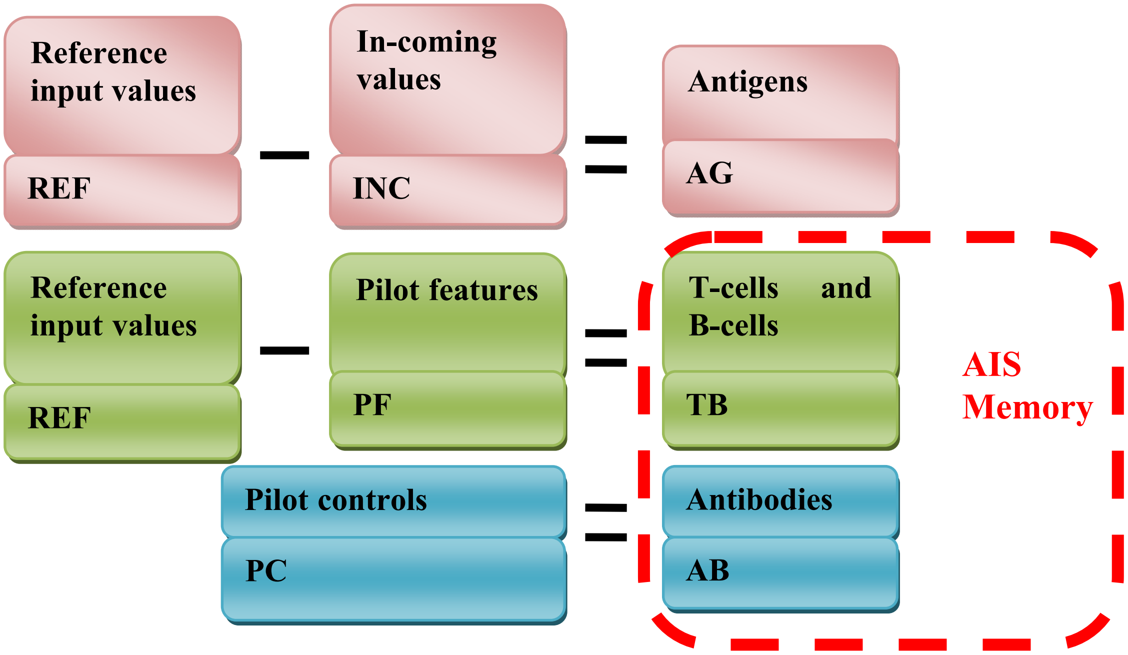


Figure 6.17. AIS Memory Structure

6.3.3. Matching Algorithm

At each time sample, an in-coming antigen AG_v is compared to the sets of T-cells and B-cells from matrix *TB*. Once the "best match" is found, a corresponding set of antibodies *AB*$_v$ is retrieved from matrix *AB*. As a result, control commands are obtained dictating the behavior of the aircraft. The "best match" is a row vector *TB*$_v$ (from matrix *TB*) which is most similar to an in-coming antigen vector AG_v. Let vector *ERR* be the difference between AG_v and *TB*$_v$:

$$ERR = AG_v - TB_v \tag{6.25}$$

Then sum *S* can be defined as follows:

$$S = \sum_{i=1}^{N_f} abs(ERR(i)) \tag{6.26}$$

where *ERR(i)* is an element of vector *ERR*. Therefore, a row vector *TB*$_v$ that leads to the lowest value of *S* is defined as the "best match". It should be noted that more sophisticated matching algorithm may be used to improve performance. For higher performance, one should take into consideration the current state and the state in which aircraft had been before (i.e. several time samples before). Once the number of the states prior to current is defined, the sum will be calculated accordingly:

$$S = \sum_{j=tsc-Nt}^{tsc} \sum_{i=1}^{N_f} abs(ERR(i))\, j \tag{6.27}$$

where *tsc* is the current time sample, and N_t is the size of the floating time window over which matching is investigated.

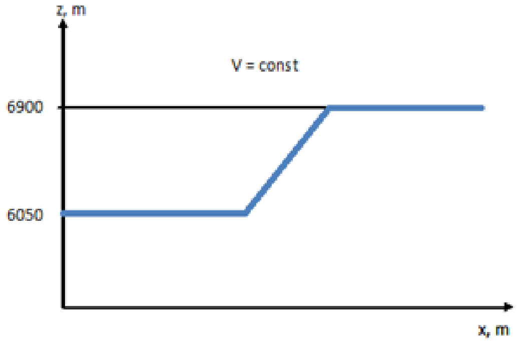
6.3.4. Example Results Using the Structured Self/Non-self for Control

A). Simulation Scenarios

The flight scenarios, lasting between 2 and 4 minutes, were designed to include steady state flight conditions and moderate maneuvers. These flight scenarios were first simulated under nominal flight conditions. They were repeated under various failure conditions for both design/development and validation purposes. Only one failure at a time is injected. Two simplified maneuvers were considered, one in the vertical plane of symmetry (climb) and one on the lateral-directional channel with longitudinal coupling (coordinated turn).

The symmetric climb maneuver (Figure 6.18) consists of steady state symmetric flight at 6050 m and Mach 0.75 for 30 seconds, uniform climb to 6900 m at Mach 0.75, and steady state symmetric horizontal flight for 1 minute at the destination altitude (6900 m) and Mach 0.75. The coordinated half turn maneuver (Figure 6.19) consists of steady state symmetric flight at 6050 m and Mach 0.75 for 30 seconds and coordinated half turn at constant bank angle while maintaining altitude and velocity.

Two types of failures were considered: left actuator locked at +4 degrees and roll rate sensor step bias. Failure injection time for all the tests is T_f*=15 s.* Sensor noise and mild atmospheric turbulence are included in the simulation.



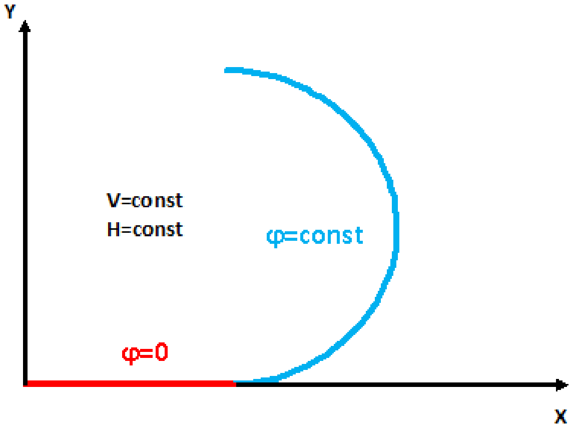


Figure 6.18. Symmetric Uniform Climb

Figure 6.19. Coordinated Half Turn

B). System Implementation

For the climbing flight scenario, the following set of flight features *FF* was selected: aircraft velocity *V*, roll rate *p*, pitch rate *q*, yaw rate *r*, altitude *H*, and pitch angle *θ*.

$$FF = \{V, p, q, r, H, \theta\} \qquad (6.28)$$

The pilot generated commands are the following: longitudinal channel stick displacement *de*, lateral channel stick displacement *da*, directional channel pedals displacement *dr*, and throttle displacement *dt*. Therefore:

$$PC = \{de, da, dr, dt\} \qquad (6.29)$$

The antigen vector AG_v will consist of elements that represent the differences between the desired and actual values of corresponding feature. However, antigen vector has to be extended by including additional elements that represent the actual values of certain features. The additional features for the extension of antigen vector are: bank angle *φ*, pitch angle *θ,* yaw angle *ψ*, pitch rate *q*, and altitude *H*. The flight duration time is *T=130s*, consequently, the overall number of time samples is *ts=6500*. The number of time samples used prior to the current moment in the matching algorithm is selected to be $N_t=4$.

For the coordinated half turn maneuver, the following set of flight features *FF* was selected: aircraft velocity *V*, roll rate *p*, pitch rate *q*, yaw rate *r*, altitude *H*, and bank angle *φ*. Therefore:

$$FF = \{V, p, q, r, H, \varphi\} \qquad (6.30)$$

The pilot generated commands are the same as in the climb maneuver. The additional features for the extension of antigen vector are: bank angle *φ*, pitch angle *θ*, yaw angle *ψ*, and roll rate *p*. The flight duration time is *T=215s*, consequently, the overall number of time samples is *ts=10750*. The number of time samples used prior to the current moment in the matching algorithm is selected to be *Nt=4*.

C). Simulation Results for Symmetric Climb

Figure 6.20 presents the variation of aircraft altitude during the symmetric flight under nominal conditions. The generated altitude was very accurately reproduced and is very close to the altitude of pilot performance. There is a little deviation at the end of the flight; however, the drifting down trend is preserved. Figure 6.21 presents the longitudinal channel control commands. The outcomes of the failure accommodation system almost coincide with the pilot performance.

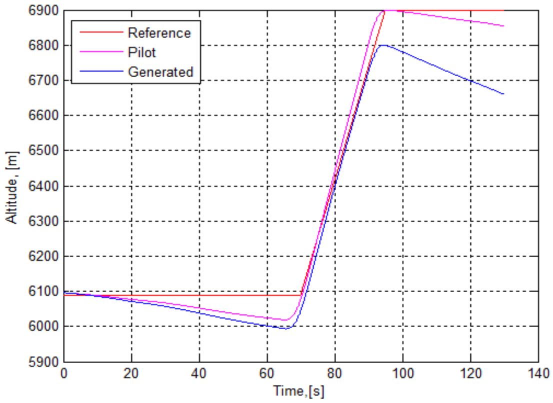


Figure 6.20. Aircraft Altitude at Nominal

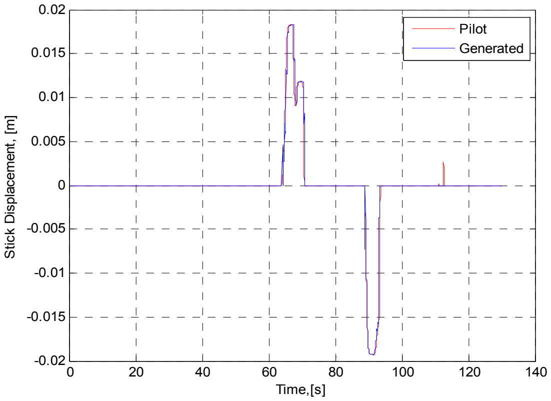


Figure 6.21. Longitudinal Channel Commands

Figure 6.22 presents the aircraft altitude variation under locked stabilator. A small decrease of altitude can be noticed at around *T=20s* due to the stabilator failure. Figure 6.23 illustrates the longitudinal channel commands. Notice that the pilot workload on the longitudinal channel has significantly increased, in comparison with nominal conditions flight. Small insignificant inaccuracies can be observed.

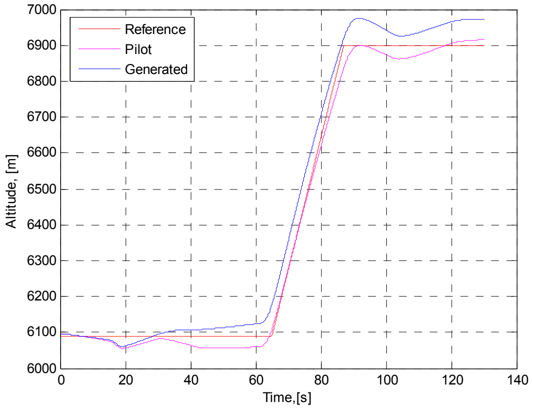


Figure 6.22. Aircraft Altitude

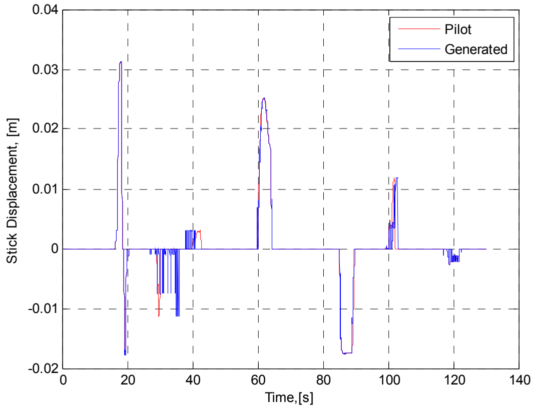


Figure 6.23. Longitudinal Channel Commands

Figure 6.24 illustrates the lateral channel commands. Although the failure affects mostly the longitudinal channel, pilot workload on the lateral channel is larger. The generated commands almost coincide with the pilot input. Figure 6.25 presents the variation of the lateral position in Earth axes. Although the stabilator failure mostly affects the longitudinal channel, a coupling with lateral channel takes place. Therefore, because of the bank angle φ alteration (see Figure 6.26), the aircraft slightly deviates from the straight forward trajectory.

Figure 6.27 illustrates the trajectory of symmetric climb under a roll rate sensor failure. Trajectory has been accurately simulated and followed the pilot's outcome. A small delay in climbing process can be noticed. Also, the altitude deviates from the desired value in the final stage of the flight. Longitudinal channel commands have been generated accurately during the sensor failure flight as presented in Figure 6.28. The generated commands followed the pilot trend, including the beginning of the flight when the failure was injected. Several inaccuracies can be observed during the climb maneuver; however, they did not have a significant effect on aircraft altitude variation.

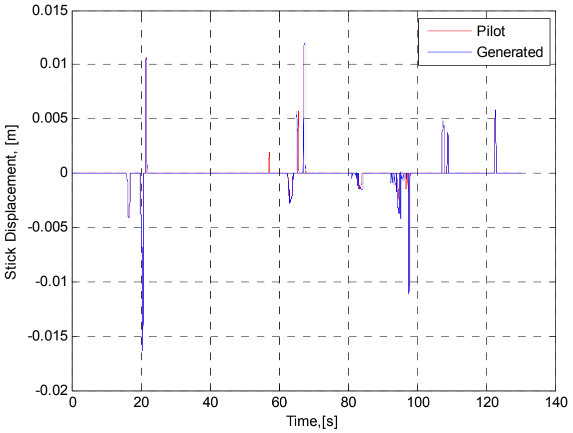


Figure 6.24. Lateral Channel Commands

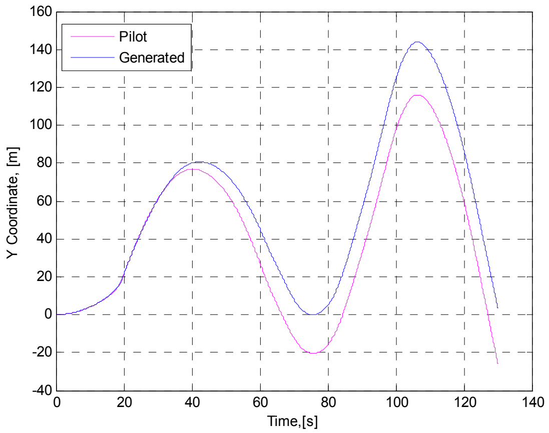


Figure 6.25. Lateral Aircraft Position Coordinate

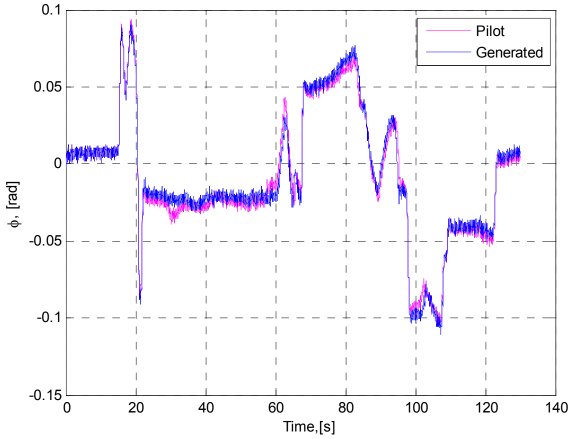


Figure 6.26. Aircraft Bank Angle

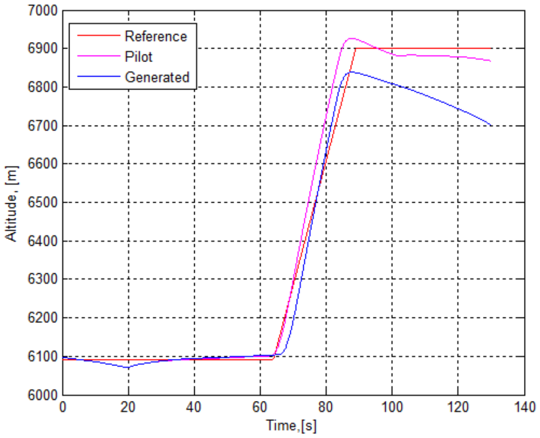


Figure 6.27. Aircraft Altitude Under Sensor Failure

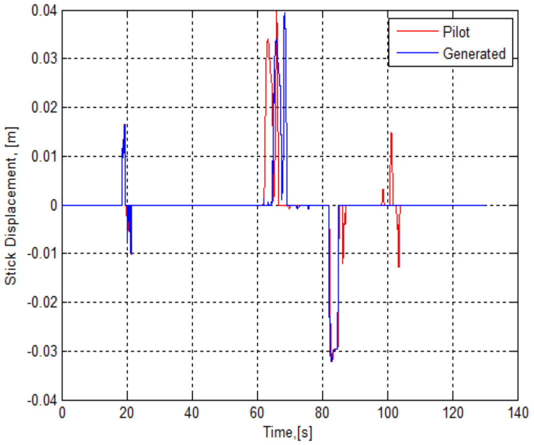


Figure 6.28. Longitudinal Channel Commands

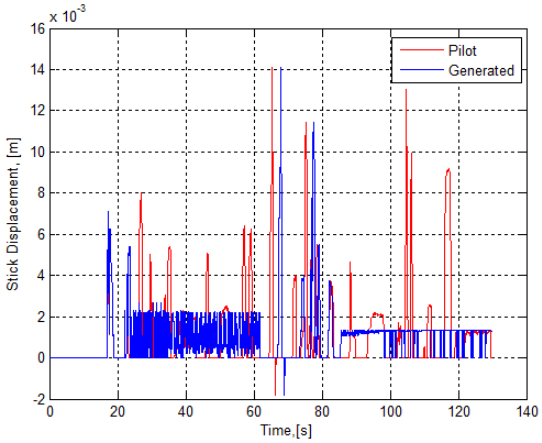


Figure 6.29. Lateral Channel Commands

Lateral channel commands extraction was not as successful as the one for the longitudinal channel (see Figure 6.29). Generally, the plot exhibits visual inaccuracies in the command extraction. However, it can be noted that at the initial stage of the flight, the commands that were dedicated to overcome the injected failure have been generated very accurately. A bank angle offset results, as presented in Figure 6.30, with an integral effect eventually producing a lateral deviation from the straight flight.

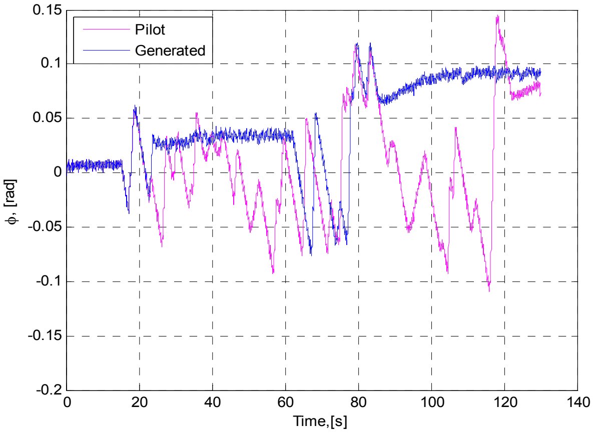


Figure 6.30. Aircraft Bank Angle

D). Simulation Results for Coordinated Half-Turn

Figure 6.31 presents the trajectory of the coordinated half turn in the horizontal plane under nominal conditions. The entire maneuver was generated successfully without any significant differences from the pilot performance outcome.

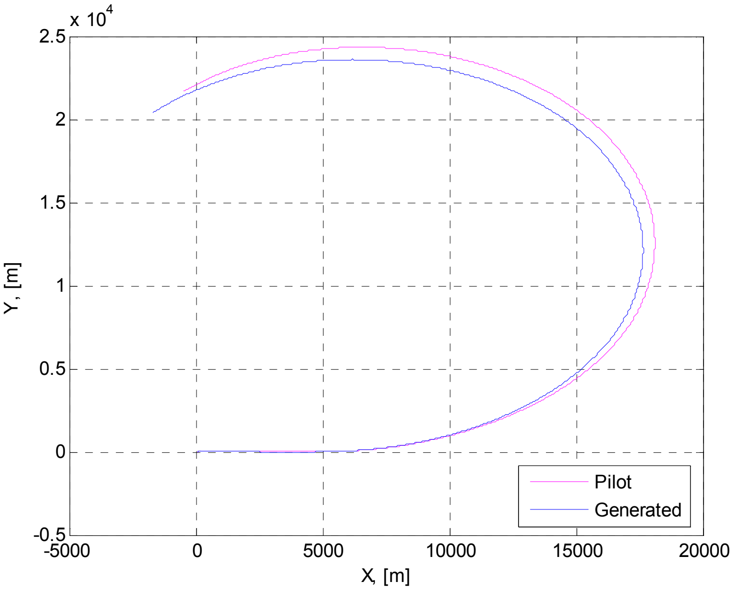
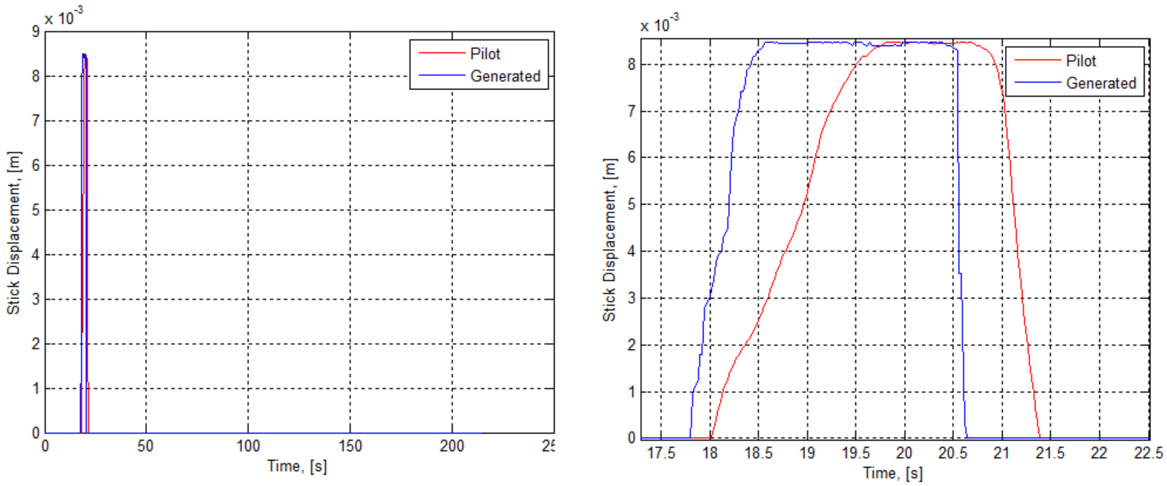


Figure 6.31. Coordinated Turn Flight Trajectory Under Nominal Conditions

Figure 6.32 presents the lateral channel commands. A short command at the beginning of the flight was produced to reach the desirable bank angle φ (see Figure 6.33). Since it is a nominal conditions flight, for the rest of the flight, no commands on this channel were needed.

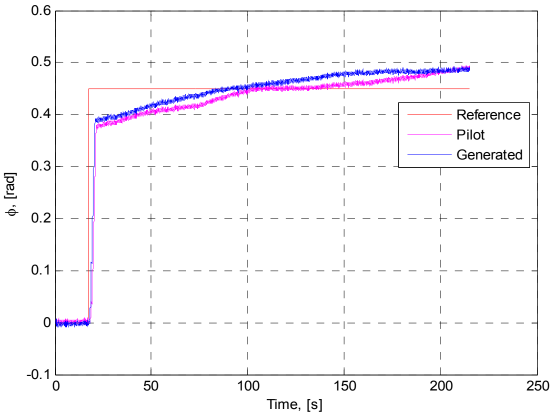


a). Entire Flight Time

b). Bank Command

Figure 6.32. Lateral Channel Commands for Coordinated Turn at Nominal Conditions

Figure 6.34 presents the longitudinal channel commands. When the aircraft is banked at a constant angle it tends to lose altitude; that is why the pilot must provide certain longitudinal commands to maintain the vertical position. This task is slightly more demanding because constant lateral input has to be preserved as well. Longitudinal command reproduction is not as neat and accurate; however, it follows the pilot trend.



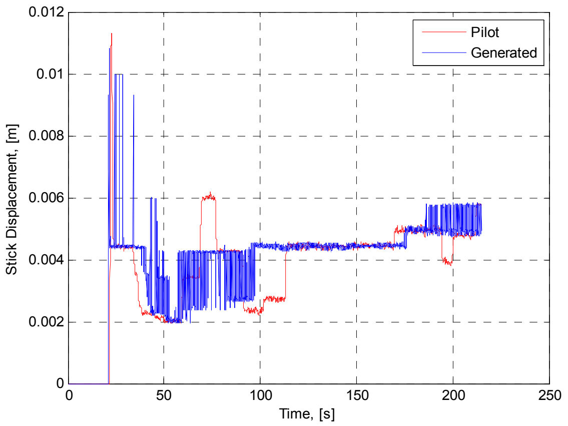


Figure 6.33. Aircraft Bank Angle

Figure 6.34. Longitudinal Channel Commands

Figure 6.35 illustrates the aircraft altitude variation. Although the generated longitudinal commands were slightly inaccurate, the aircraft altitude is well maintained and does not differ a lot from the pilot performance outcome.

Figure 6.36 presents the flight trajectory in the horizontal plane under stabilator failure when performing a coordinated turn. An excellent duplication of pilot performance can be observed. Longitudinal channel commands are presented in Figure 6.37. There is a considerable input signal at the beginning of the flight that has been provided to mitigate the failure. The generated simulation provided that signal as well. During the rest of the time, the commands are extracted very accurately, with several inaccuracies at the end.

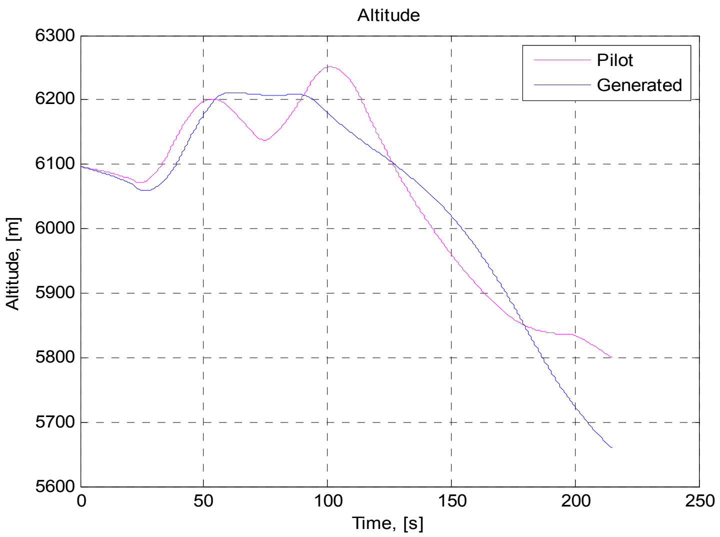


Figure 6.35: Aircraft Altitude - Nominal Conditions Coordinated Half Turn

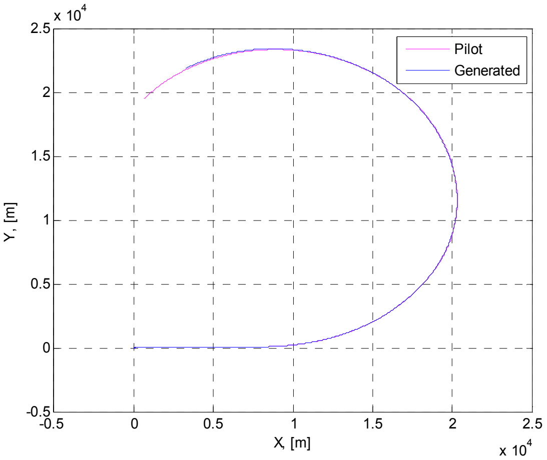


Figure 6.36. Coordinated Turn Trajectory Under Stab Failure

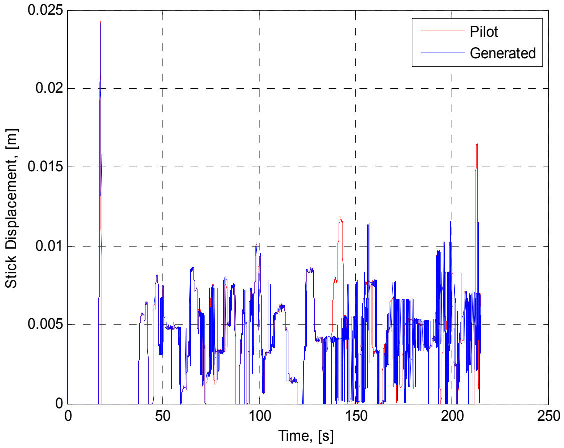


Figure 6.37. Longitudinal Channel Commands Under Stab Failure

Lateral channel control commands are presented in Figure 6.38. Because of the coupling, the pilot had to provide some commands on the lateral channel as well right after the failure was injected. Over the entire flight, the commands have been extracted precisely: the lines almost coincide. Generated bank angle φ dictated by the lateral commands also follows the pilot performance outcome accurately (Figure 6.39). little peak is observed at the beginning of the flight due to the failure; however, the value has been brought back close to zero by pilot compensation.

Figure 6.40 illustrates the flight trajectory in the horizontal plane for a coordinated turn executed under roll rate sensor bias. Generated trajectory follows the pilot trend; however, the generated curve has a smaller radius of turn as compared to the one from the pilot performance outcome due to the inaccuracies in reproducing lateral channel commands. Figure 6.41 presents the lateral channel commands provided during the flight simulation. The commands have not been extracted as accurately as for actuator failures; however, they behave similarly to pilot generated commands with a small delay and several inaccuracies of small magnitude. Figure

6.42 illustrates the commands on the longitudinal channel. Similarly to the lateral channel, the command extraction is not as accurate as for actuator failure; however, the similar trend can be observed in the behavior. These inaccuracies on longitudinal and lateral channels can be explained by excessive amount of the pilot workload and rapid change of stick displacement during the flight. For the flight with a large pilot workload and very intensive stick commands, a more advanced matching algorithm might be needed for obtaining higher performance results. Also, there is a delay between pilot input and aircraft response which becomes more critical when abrupt maneuvers are involved. Under such conditions, this delay should be better taken into account. One possible solution could be to increase the time window over which the matching algorithm is applied.

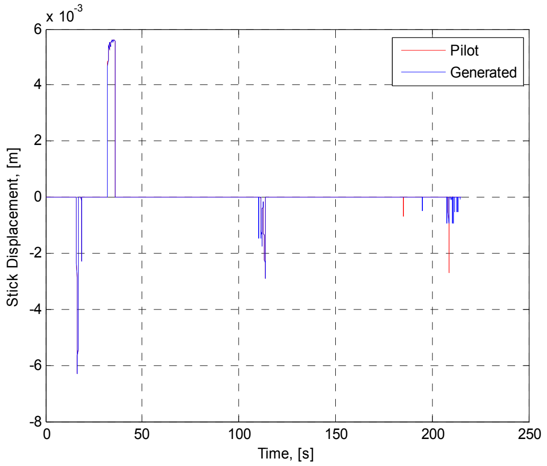


Figure 6.38. Lateral Channel Commands Under Stab Failure

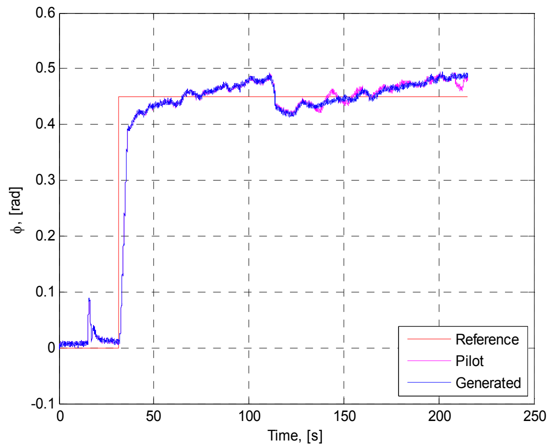


Figure 6.39. Aircraft Bank Angle Under Stab Failure

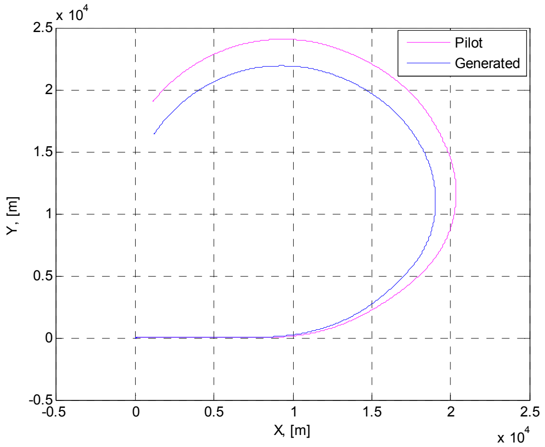


Figure 6.40. Flight Trajectory for Coordinated Turn Under Roll Rate Sensor Failure

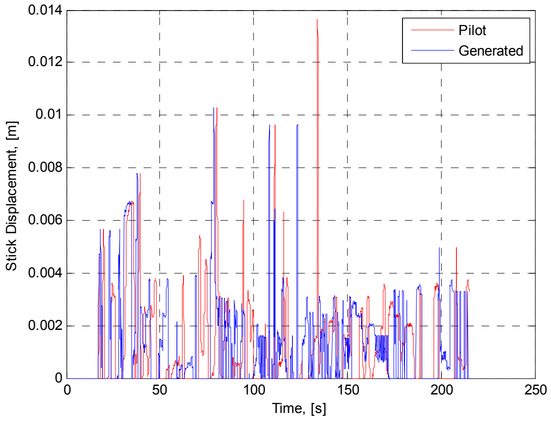


Figure 6.41. Lateral Channel Commands Under Roll Rate Sensor Failure

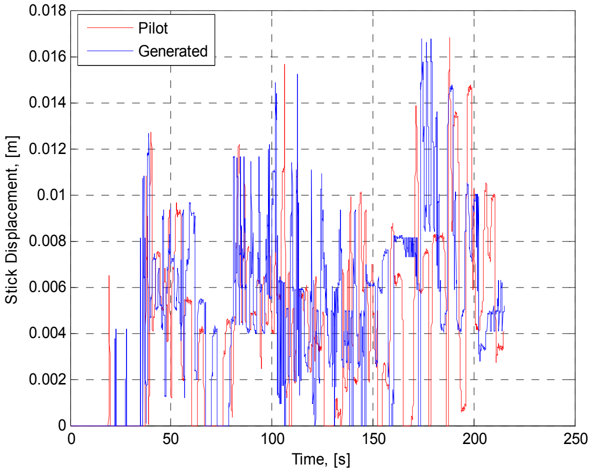


Figure 6.42. Longitudinal Channel Commands for Coordinated Turn Under Roll Rate Sensor Failure

Chapter 7

Integrated Scheme for Aircraft Subsystem Abnormal Conditions Detection, Identification, and Evaluation

7.1. Integrated ACDIE Scheme Using the Structured Self/Non-self Approach

Based on the AIS paradigm and a HMS strategy, the SNSA was developed and implemented for aircraft subsystems ACDIE. All phases of the ACDIE process have been integrated in one single architecture, that starts with the generation of antibodies (detectors) and continues with structuring of the non-self for identification and evaluation purposes, as well as providing the necessary information for AC accommodation. Flight data from the WVU 6-DOF flight simulator were initially used to define a large set of 2-dimensional self/non-self projections as well as for the generation of antibodies and identifiers designated for health assessment of an aircraft under upset conditions. As shown in Figure 7.1, this represents the first step in the complex ACDIE process. Selecting the predefined set of features or characteristic variables for the definition of the self/non-self is a critical component of the AIS-based methodology.

During "detection" phase, the process of declaring a generic failure in one or more of the aircraft sub-systems is performed. A detection logic is also designed for real time operation to ensure that high detection rates and low false alarm rates are achieved. Once a failure has been detected, the "identification" phase starts. The identification phase determines which sub-system has failed by analyzing which of the detectors has been activated through a positive selection-type scheme. In this phase, all the detectors are labeled in a previous offline process (SNSA) in order to assign specific detectors to particular categories of failures. This off-line process or "structuring" consists of outlining which non-selves are activated under a specific failure. Depending on the complexity of the targeted systems, the identification process can determine which subsystems have failed (i.e. actuator, sensor, structural, etc.). The evaluation phase can be divided into two steps, namely direct evaluation and indirect evaluation. Furthermore, the direct evaluation phase can be classified into qualitative and quantitative evaluation. The direct qualitative evaluation phase isolates and determines the specific subsystem that has failed. The quantitative evaluation phase determines the severity or magnitude of a failed subsystem. The indirect evaluation phase determines the effect on the flight envelope maneuverability and performance of the system after a failure has affected the system. Finally, once the failure has been detected and properly identified, the best solution to compensate such failure is executed using active or passive accommodation tasks that have been previously stored in the immune memory.

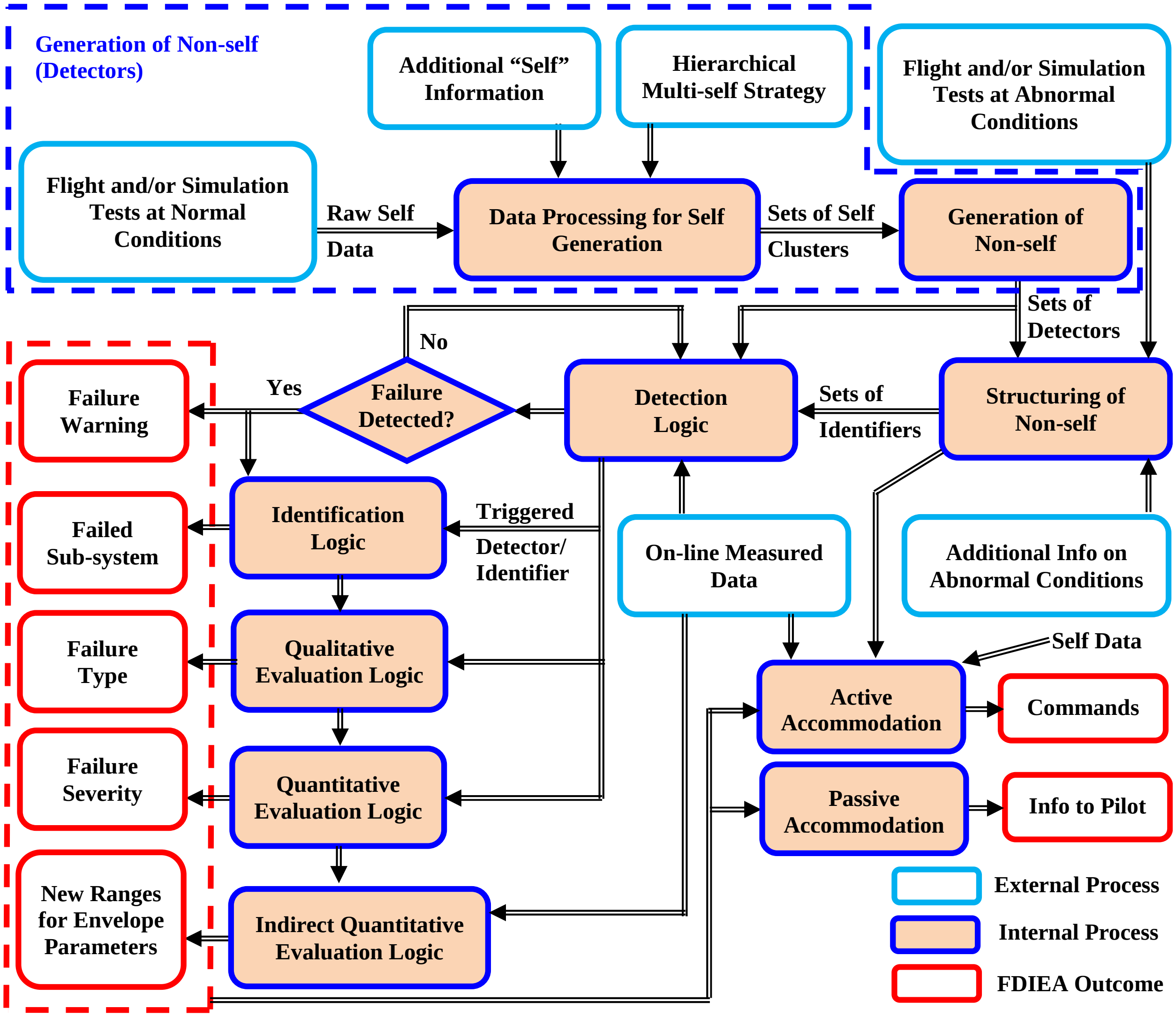

Generation of Non-self (Detectors)
Additional "Self" Information
Hierarchical Multi-self Strategy
Flight and/or Simulation Tests at Abnormal Conditions
Flight and/or Simulation Tests at Normal Conditions
Raw Self Data
Data Processing for Self Generation
Sets of Self Clusters
Generation of Non-self
Sets of Detectors
No
Yes
Failure Detected?
Detection Logic
Sets of Identifiers
Structuring of Non-self
Failure Warning
Failed Sub-system
Failure Type
Failure Severity
New Ranges for Envelope Parameters
Identification Logic
Triggered Detector/ Identifier
Qualitative Evaluation Logic
Quantitative Evaluation Logic
Indirect Quantitative Evaluation Logic
On-line Measured Data
Additional Info on Abnormal Conditions
Self Data
Active Accommodation
Passive Accommodation
Commands
Info to Pilot
External Process
Internal Process
FDIEA Outcome

Figure 7.1. ACDIE with the Structured Non-Self Approach

7.2. Integrated ACDIE Scheme Using the DC Mechanism-based Approach

Figure 7.2 shows a top-level block diagram of the online ACDIEA scheme using the artificial DC mechanism. The measured data (i.e., feature values) are first normalized over a moving time window using the same normalization used in generating the self and non-self. The normalized data are then projected on the corresponding projections (sub-selves) to generate the discrimination matrices that are used as input to the artificial DC mechanism for detection. Once a failure is detected, the features-pattern vector is computed based from the $\mathbf{F}_1$ matrices of all migrated DCs (stimulatory and regulatory DCs) and compared with the libraries of reference patterns to identify the failed subsystem and evaluate the type of the failure and its severity simultaneously.

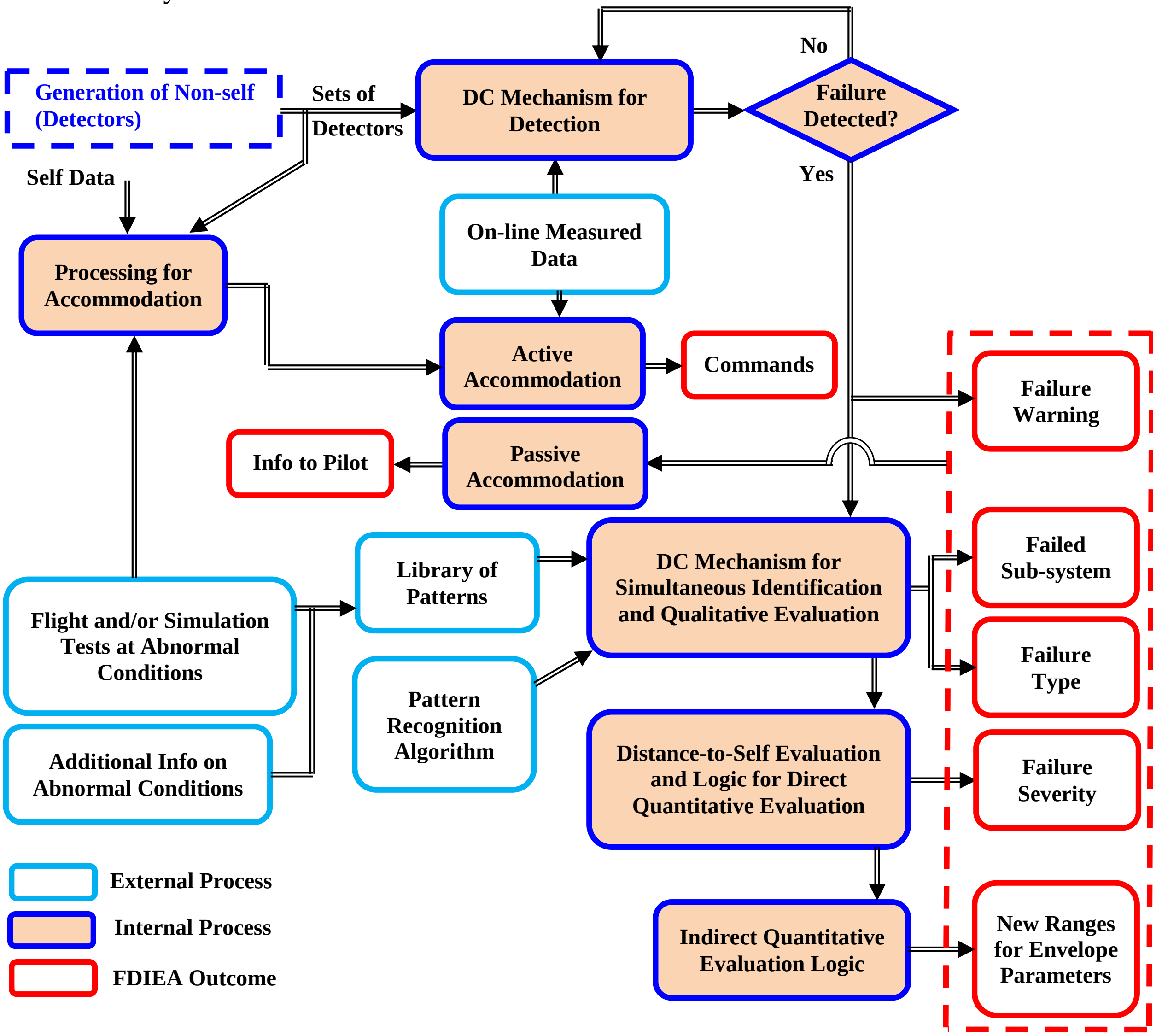

No
Generation of Non-self (Detectors)
Sets of Detectors
DC Mechanism for Detection
Failure Detected?
Yes
Self Data
On-line Measured Data
Processing for Accommodation
Active Accommodation
Commands
Failure Warning
Info to Pilot
Passive Accommodation
Library of Patterns
DC Mechanism for Simultaneous Identification and Qualitative Evaluation
Failed Sub-system
Flight and/or Simulation Tests at Abnormal Conditions
Failure Type
Pattern Recognition Algorithm
Additional Info on Abnormal Conditions
Distance-to-Self Evaluation and Logic for Direct Quantitative Evaluation
Failure Severity
External Process
Internal Process
FDIEA Outcome
Indirect Quantitative Evaluation Logic
New Ranges for Envelope Parameters

Figure 7.2. ACDIE with the DC Mechanism Approach

Chapter 8

Demonstration of the Integrated Scheme for Aircraft Subsystem Abnormal Conditions Detection, Identification, and Evaluation

8.1. Implementation of the Integrated Scheme for Aircraft Subsystem Abnormal Conditions Detection, Identification, and Evaluation

The integrated aircraft subsystem ACDIE scheme was implemented and tested with Matlab/Simulink 2010a (32-bit). Figure 8.1 shows the top level Simulink model using the SNSA, where the "immunity-HMS" block detail is shown in Figure 8.2. The top level Simulink model of the integrated scheme using the DC mechanism is illustrated in Figure 8.3 and the DC mechanism block detail is shown in Figure 8.4.

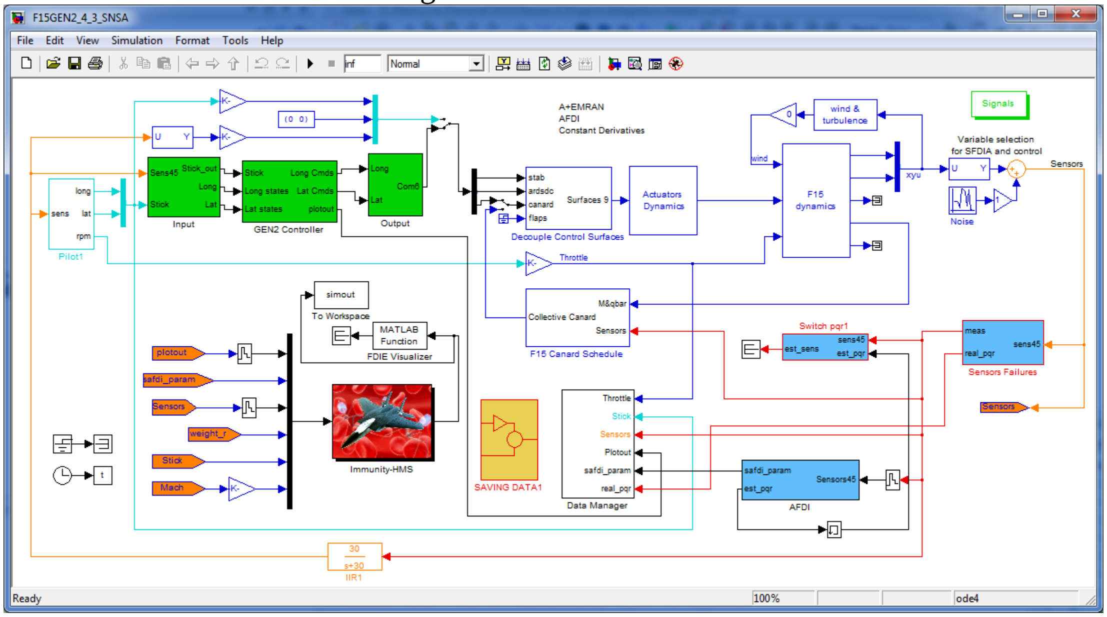


Figure 8.1. Top-level Simulink Model of the Integrated Aircraft ACDIE Using the Structured Non-Self Approach

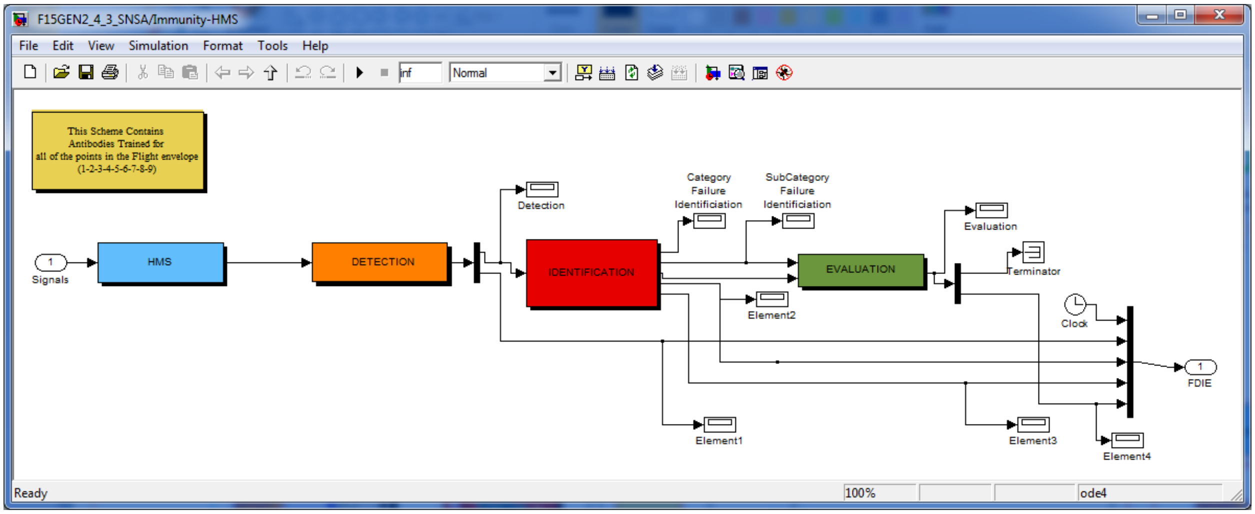

F15GEN2_4_3_SNSA/Immunity-HMS
File Edit View Simulation Format Tools Help
inf
Normal
This Scheme Contains Antibodies Trained for all of the points in the Flight envelope (1-2-3-4-5-6-7-8-9)
1
Signals
HMS
DETECTION
Detection
IDENTIFICATION
Category Failure Identificiation
SubCategory Failure Identificiation
EVALUATION
Evaluation
Terminator
Element2
Clock
1
FDIE
Element1
Element3
Element4
Ready
100%
ode4

Figure 8.2. Simulink Block Diagram of the Structured Non-Self Approach

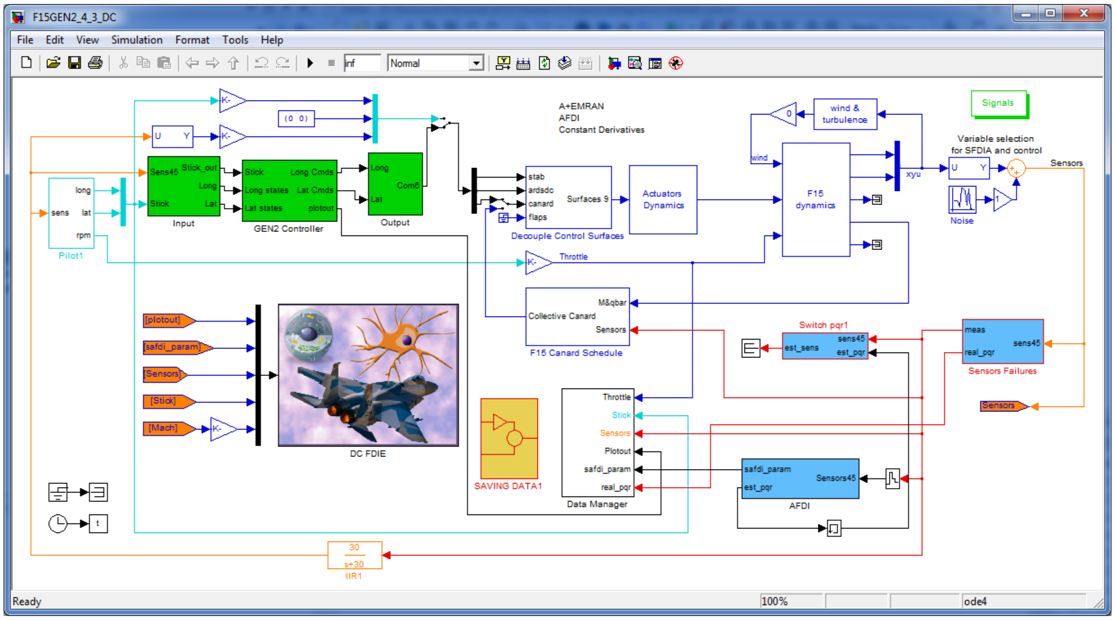


Figure 8.3. Top-level Simulink Model of the Integrated Aircraft ACDIE Scheme Using the DC Mechanism

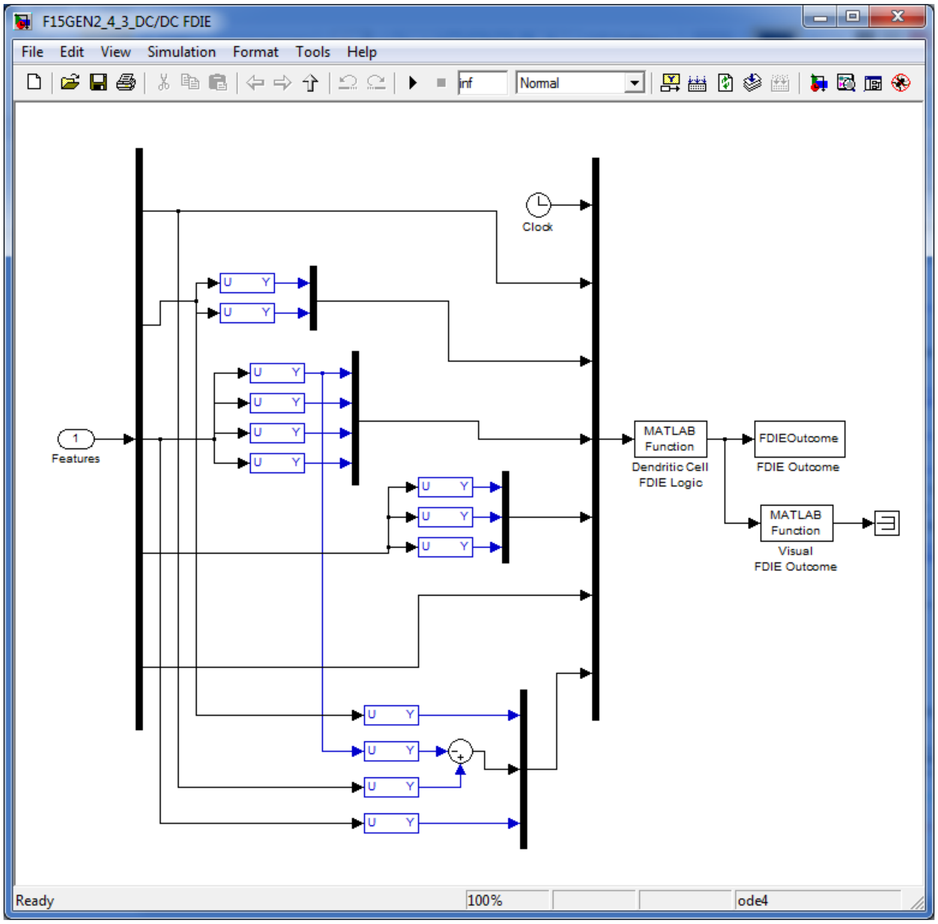
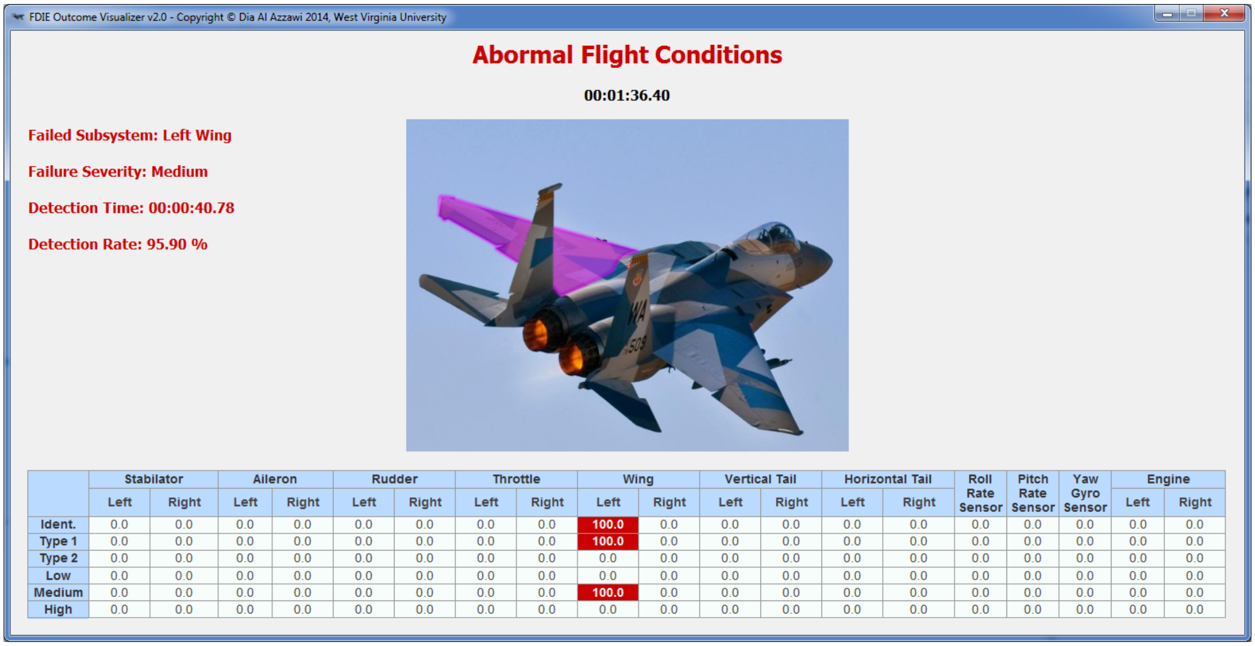


Figure 8.4. Simulink Block Diagram of the DC Mechanism

Both of the models deliver their outcome to a GUI at each time step. Figure 8.5 shows the interface displaying the current ACDIE outcome for a flight test under failure of the left wing. The implementation of the integrated scheme has two components. One allows the scheme to run in conjunction with the aircraft simulation. The other allows feeding previously obtained simulation data into the ACDIE scheme for faster testing and analysis.


FDIE Outcome Visualizer v2.0 - Copyright © Dia Al Azzawi 2014, West Virginia University

Abormal Flight Conditions

00:01:36.40

Failed Subsystem: Left Wing

Failure Severity: Medium

Detection Time: 00:00:40.78

Detection Rate: 95.90 %

	Stabilator		Aileron		Rudder		Throttle		Wing		Vertical Tail		Horizontal Tail		Roll Rate Sensor	Pitch Rate Sensor	Yaw Gyro Sensor	Engine	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right				Left	Right
Ident.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Type 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Type 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Low	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Medium	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
High	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 8.5. ACDIE Outcome Visualization Interface

The ACDIE outcome visualization interface has four components:

1. A header displaying the current flight condition and the current sample time. If the current flight condition is normal, the header will be “Normal Flight Conditions” in green. If the current flight condition is abnormal, the header will be “Abnormal Flight Conditions” in red.
2. Four-line text on the left of the interface showing brief information about the failure (failed subsystem, its severity, detection time, and detection rate).
3. An aircraft image displaying the current status of aircraft’s flight condition. When the ACDIE outcome carries information about the failed subsystem, the image dynamically highlights the failed subsystem.
4. A table displaying the percentage rates of the identified subsystems, types of the failure, and severity of the failure.

The interface computes the detection, identification, and evaluation rates for all received ACDIE outcomes. However, the interface cannot update its contents when it receives two successive outcomes in a short time interval. Therefore, a predefined “*update rate*” representing the number of graphical update milliseconds was introduced in the configuration of the interface. For example, when the interface is configured with an “update rate” of 100, it will only update its graphical contents every 100 millisecond, while it keeps computing the ACDIE rates for *any* ACDIE outcome it receives.

8.2. Demonstration of the Integrated Scheme for Aircraft Subsystem Abnormal Conditions Detection, Identification, and Evaluation Through Simulation

The integrated ACDIE scheme was tested for 10 additional demonstration flight tests with different failed subsystems, failure types, and failure magnitudes. In these tests, the both the DC mechanism approach and the SNSA were used. Table 8.1 lists the detection rates as well as the path of the flight test (points in the test envelope of Figure 1.22) and the detection time. The failure was introduced at 40 s. The false alarm rate was zero for all tests. These example results were obtained through direct recorded data input and through on-line simulation input.

Table 8.1. Detection Results for a Set of Demonstration Tests

Subsystem Affected by AC	**Failure Type**	**Failure Magnitude**	**Path**	**Detection Time**	**Detection Rate (%)**
Left Stabilator	locked	8°	145	40.68	99.9
Left Wing	damaged	15%	167	40.62	99.9
Pitch Sensor	bias	10°	189	40.72	99.9
Right Aileron	locked	8°	165	40.72	99.9
Right Wing	damaged	6%	189	40.88	99.9
Roll Sensor	bias	10°	145	41.68	98.9
Yaw Sensor	bias	3°	167	41.12	99.9
Left Stabilator	locked	6°	1B	40.28	100.0
Left Wing	damaged	10%	1B	40.28	100.0
Roll Sensor	bias	10°	1B	45.58	100.0

Table 8.2 presents the identification rates for the same tests listed in Table 8.1, whereas the corresponding failure type and severity evaluation rates are presented in Tables 8.3 and 8.4, respectively. Note that the evaluation rates are computed with respect to the number of points for which correct identification was obtained.

Table 8.2. Identification Results for the Demonstration Tests

	Identified Subsystem									
Test #	**Left Stab**	**Right Stab**	**Left Aileron**	**Right Aileron**	**Left Rudder**	**Right Rudder**	**Left Throttle**	**Right Throttle**	**Left Wing**	**Right Wing**
Left Stabilator	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Left Wing	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**98.7**	1.3
Pitch Sensor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Right Aileron	0.0	0.0	0.0	**98.8**	0.0	0.0	0.0	0.0	0.0	0.0
Right Wing	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**99.9**
Roll Sensor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Yaw Sensor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Left Stabilator	**99.9**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Left Wing	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	**99.8**	0.0
Roll Sensor	0.0	0.0	0.0	0.0	0.0	12.7	0.0	0.0	0.0	0.0

Table 8.2. - Cont'd

	Identified Subsystem								
Test	**Left V Tail**	**Right V Tail**	**Left H Tail**	**Right H Tail**	**Roll Sensor**	**Pitch Sensor**	**Yaw Sensor**	**Left Engine**	**Right Engine**
Left Stabilator	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Left Wing	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pitch Sensor	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0
Right Aileron	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	0.0
Right Wing	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Roll Sensor	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0
Yaw Sensor	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0
Left Stabilator	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
Left Wing	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Roll Sensor	0.0	0.0	0.0	0.0	**87.0**	0.3	0.0	0.0	0.0

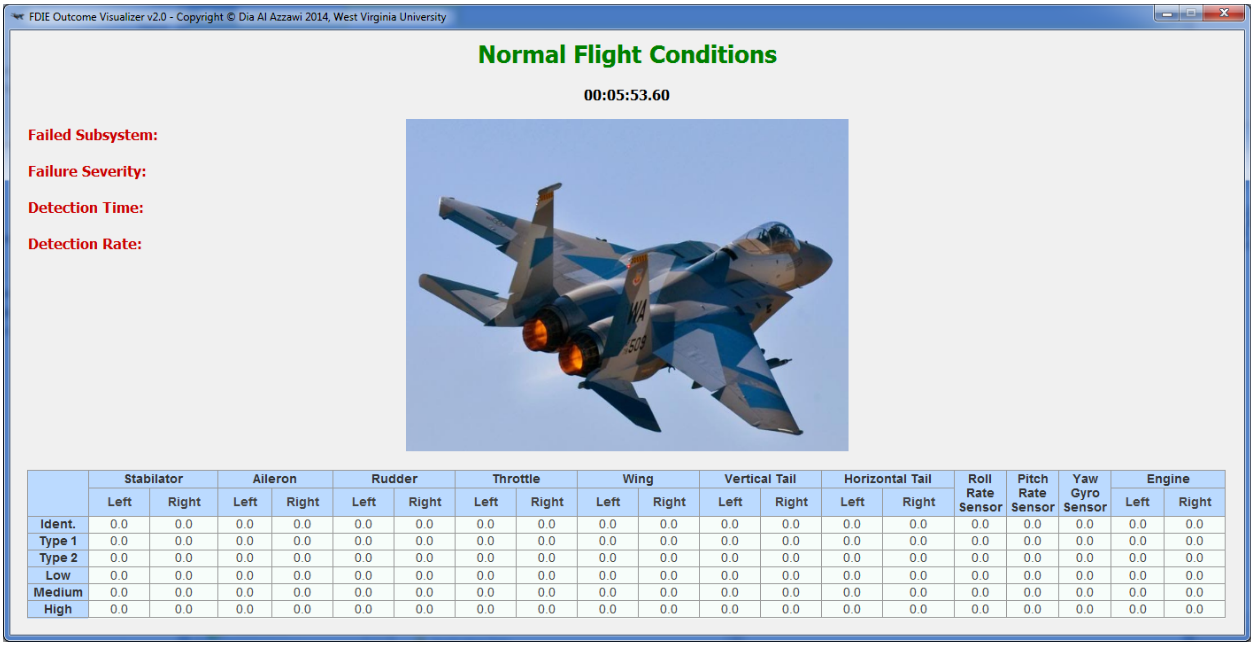
Table 8.3. Failure Type Evaluation Results for the Demonstration Tests

	Evaluated Type	
Test	**Type 1**	**Type 2**
Left Stabilator	100.0	0.0
Left Wing	100.0	0.0
Pitch Sensor	100.0	0.0
Right Aileron	100.0	0.0
Right Wing	100.0	0.0
Roll Sensor	100.0	0.0
Yaw Sensor	100.0	0.0
Left Stabilator	100.0	0.0
Left Wing	100.0	0.0
Roll Sensor	100.0	0.0

Table 8.4. Failure Severity Evaluation Results for the Demonstration Tests

	Evaluated Severity		
Test	**Low**	**Medium**	**High**
Left Stabilator	0.0	0.0	100.0
Left Wing	0.0	100.0	0.0
Pitch Sensor	0.0	0.0	100.0
Right Aileron	1.2	0.0	98.8
Right Wing	99.9	0.0	0.1
Roll Sensor	0.0	0.0	100.0
Yaw Sensor	0.0	0.0	100.0
Left Stabilator	0.0	100.0	0.0
Left Wing	0.0	100.0	0.0
Roll Sensor	0.0	0.0	100.0

Figures 8.6 and 8.7 show the ACDIE outcome visualization window for demo tests under normal flight conditions and a 15% damage of the left wing, respectively.


FDIE Outcome Visualizer v2.0 - Copyright © Dia Al Azzawi 2014, West Virginia University

Normal Flight Conditions

00:05:53.60

Failed Subsystem:

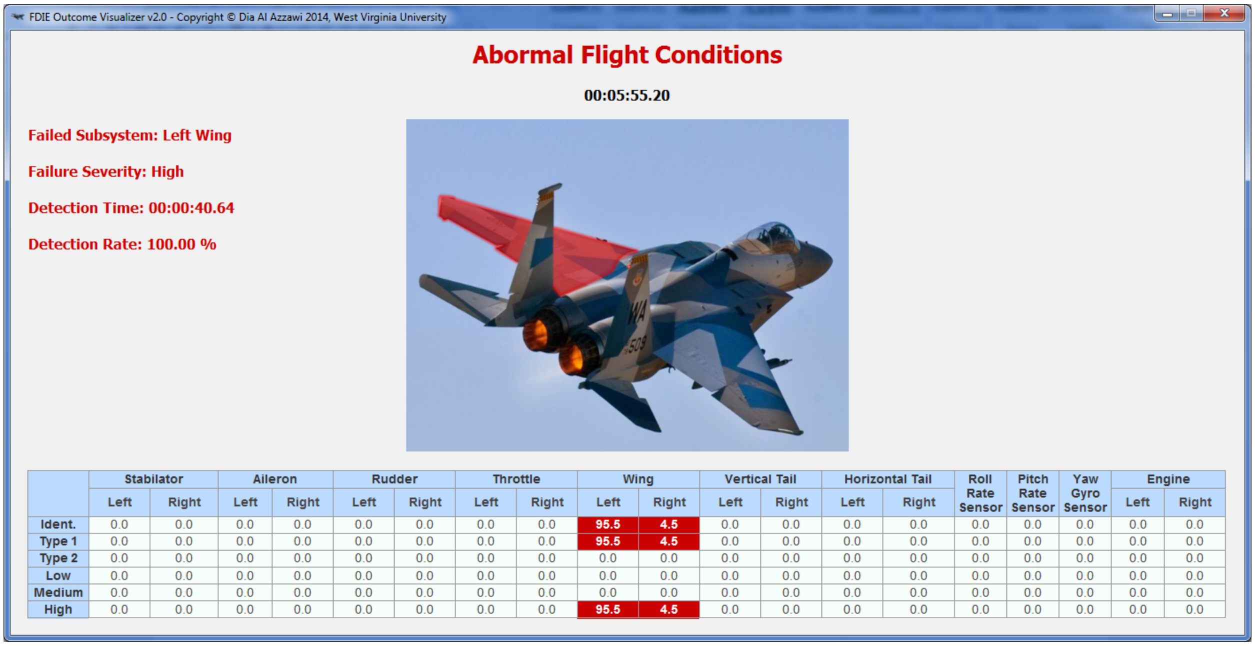
Failure Severity:

Detection Time:

Detection Rate:

	Stabilator		Aileron		Rudder		Throttle		Wing		Vertical Tail		Horizontal Tail		Roll Rate Sensor	Pitch Rate Sensor	Yaw Gyro Sensor	Engine	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right				Left	Right
Ident.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Type 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Type 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Low	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Medium	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
High	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 8.6. Demonstration Results for a Nominal Flight


FDIE Outcome Visualizer v2.0 - Copyright © Dia Al Azzawi 2014, West Virginia University

Abormal Flight Conditions

00:05:55.20

Failed Subsystem: Left Wing

Failure Severity: High

Detection Time: 00:00:40.64

Detection Rate: 100.00 %

	Stabilator		Aileron		Rudder		Throttle		Wing		Vertical Tail		Horizontal Tail		Roll Rate Sensor	Pitch Rate Sensor	Yaw Gyro Sensor	Engine	
	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right				Left	Right
Ident.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.5	4.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Type 1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.5	4.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Type 2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Low	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Medium	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
High	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	95.5	4.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 8.7. Demonstration Results for a 15% Damage of the Left Wing

8.3. Demonstration of the AIS AC Detection Through Flight Testing Using a UAV Research Platform

8.3.1. Flight Test Scenario

An initial performance analysis of the proposed ACDIE solution has been assessed through flight tests. A set of two nominal flights that included figure-8-maneuvers as well as control surface doublets in all three axes were performed in order to capture the dynamics of the test aircraft. The platform used for this experimentation was the Skywalker RC aircraft instrumented with an APM 2.5 micro controller. The RC aircraft was flown manually by a pilot on the ground and the maneuvers were implemented in a sequential order. The nominal flight tests consisted of the following stages:

1. Manual flight until an altitude of 80 meters
2. Trim flight at constant speed
3. Figure 8 maneuver
4. Elevator Doublet
5. Figure 8 maneuver
6. Aileron Doublet
7. Figure 8 maneuver
8. Rudder Doublet
9. Figure 8 maneuver

The flight test sequence was performed twice and the data were saved into a flash memory for later processing. The data was recorded at a rate of 50Hz and included angular rates, linear accelerations, neural network angular rate estimations, reference model commands, Euler angles, altitude, stick inputs and velocity. A total of 21 features were recorded, which generate a total of 210 self/non-self 2-D projections. The reduced flight envelope limits of the Skywalker as well as the length of the flights yield a reduced amount of data points. This should be taken into consideration during the generation of projections in order to obtain proper coverage of the self.

For validation purposes, four different types of failures were injected into the system at later flights in order to capture the dynamic fingerprint of abnormal conditions on the test platform. The failures investigated included low and high magnitude aileron failures. The failures were injected manually by the pilot through a PWM signal sent from an RC transmitter. Once a failure was injected, the sequence of maneuvers presented above was attempted by the pilot. Table 8.5 presents the injected failures in the system.

Table 8.5. Failures Injected in Flight Tests

Failure #1	Left Aileron Locked at Wings Level Trim
Failure #2	Right Aileron Locked at Wings Level Trim
Failure #3	Left Aileron Locked at Trim during Bank Turn
Failure #4	Right Aileron Locked at Trim during Bank Turn

8.3.2. Test Platform

The RC airplane chosen for experimentation was the “New Skywalker 1880”. This platform offers a stable and low-cost system that is able to satisfy the needs of the flight tests. Figure 8.8 shows the actual system used for the flight tests. This platform was used in previous projects for which system identification techniques were performed. The physical characteristics of the system are presented in Table 8.6.

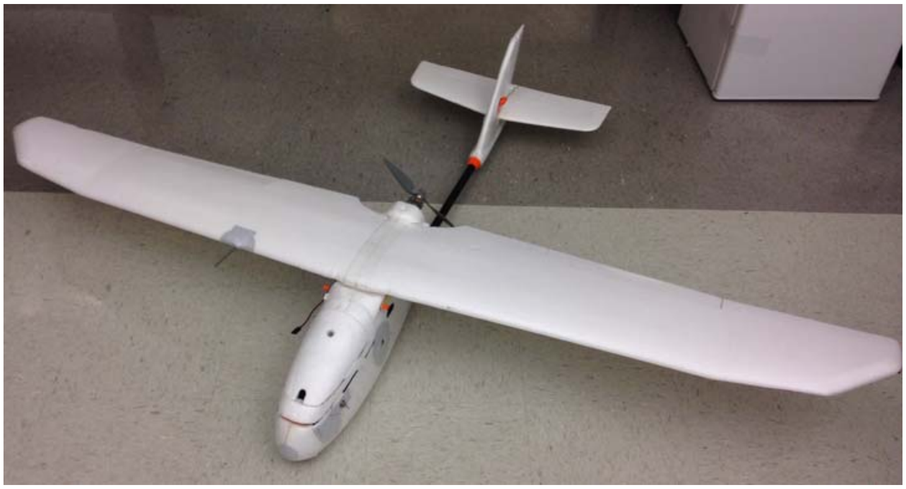


Figure 8.8. Skywalker 1880 RC

Table 8.6. Skywalker Dimensions and Mass Properties

Wing Area (m^2)	0.41143
Wing MAC (m)	0.22647
Wingspan (m)	1.88
Horizontal Tail Span (m)	0.5626
Horizontal Tail MAC (m)	17.1
Vertical Tail Span (m)	0.244
Vertical Tail MAC (m)	19.5
Total Length (m)	1.183
Weight (Kg)	0.9525

The Skywalker 1880 was equipped with a set of analog and digital sensors that provide essential variables for the generation of Selves and Non-selves. Primarily, the onboard microcontroller is an APM 2.5 with an “Atmel ATMEGA 2560” processor. The APM 2.5 board includes embedded sensors such as an IMU, magnetometer and a 4MB data flash chip as well as digital and analog ports for GPS, telemetry and a pitot tube sensor. Figure 8.9 shows the APM and the setup inside the fuselage of the Skywalker.

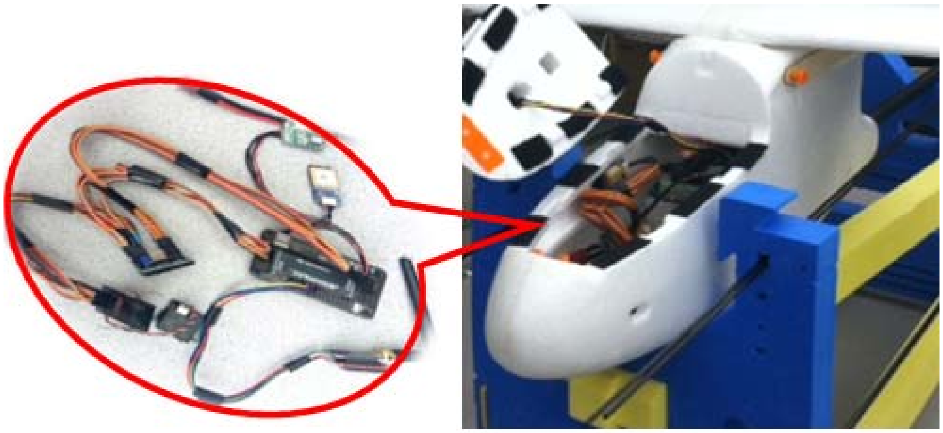


Figure 8.9. Onboard APM 2.5

The Ardupilot Mega (APM) 2.5 is an out-of-the-shelf low-cost autopilot solution produced by 3D Robotics. Its dimensions are approximately 66x40 mm. and with the enclosure it weighs approximately 20 grams. It uses an 8-bit, 16Mhz "Atmel AT Mega 2560" processor which has 54 digital I/O pins for which 14 can be used for PWM signals.

The InvenSense MPU-6000 inertial sensor is a 6-axis motion tracking device. It combines a 3-axis gyroscope and a 3-axis accelerometer in a 4x4x0.9mm QFN footprint and it communicates through a serial interface in an IC2 protocol.

The MEAS Switzerland MS5611 Barometric Pressure Sensor offers a high resolution altimeter sensor with SPI and I2C bus interfaces up to 20MHz. It offers a factory calibrated sensor with a resolution of 10 cm. and its dimensions corresponds to a 5x3x1mm QFN footprint.

The data flash card used is a 4Mb chip embedded in the APM 2.5 board. Previous efforts have shown that recording 20 floating point parameters at 50Hz allows recording approximately 17 minutes of flight**.**

The MediaTek MT3329 is a 66 channel single chip solution with a binary output protocol with an update rate of 10Hz. Its sensitivity can be up to -165dB tracking, a position accuracy of less than 3meters and USB/UART interfaces. Its dimensions are 38x16x7.8mm and it weighs 9.45g.

The Freescale MPXV7002DP Differential Pressure Sensor was connected directly to a miniature pitot tube located on the right wing. Its maximum rating for pressure is 2kPa at 60^{o}C. This sensor provides true air speed measurements that are used for the definitions of selves.

The Spektrum DX7s RC transmitter and receiver is a 7-channel, 2.4 GHz remote control device used for manual control of the aircraft and control of aileron failures in the system. Five channels were used for the control of ailerons, rudder, throttle, and elevator and a sixth channel commanded the aileron to lock at a given position.

The power source chosen for the system was a G6 Pro Lite Thunder Power 5 cell LiPo 18.5V battery.

The electric motor chosen for the Skywalker was a 1000Kv RPM Turnigy brushless motor. Its maximum current is 38A and it has a maximum power of 665W. Its weight is approximately 130g and its size is 35x42mm with a shaft diameter of 5mm.

8.3.3. Simulink Models

The onboard microcontroller has the ability to be targeted through Simulink and the APM 2.0 Block set for Simulink. This feature provides a great advantage for any effort involving low cost autopilots and sensor fusions boards. Several Simulink models including model reference control, artificial neural networks and Kalman filter models, were designed in Simulink and later loaded into the APM 2.5 board for flight test implementation. Figure 8.10 shows the top level of the Simulink model loaded into the APM 2.5 board.

In general, the Simulink model allows the APM board to read several sensors embedded in the board while recording flight test data in real time. Once the data is recorded in the flash memory, it can be downloaded and processed off-line for the generation of projections. Most of the features are obtained from the sensors embedded in the onboard computer. On the other hand, the bank and pitch angles as well as the ANN and model reference control outputs are generated by separate Simulink models. The following sections will briefly describe the Simulink models for the mentioned systems.

It is possible to auto-generate code through the Run-On-Target-Hardware tool in the Simulink environment into the APM 2.5 board. The APM 2.0 Simulink blockset allows users to read data from the sensors embedded in the board, command PWM signals to servos and to run guidance, navigation and control algorithms. This library allows users to read the sensors embedded in the microcontroller. Therefore, a Simulink model that reads the data from such sensors was generated and loaded into the onboard computer. Figure 8.11 shows a sample model of the sensors.

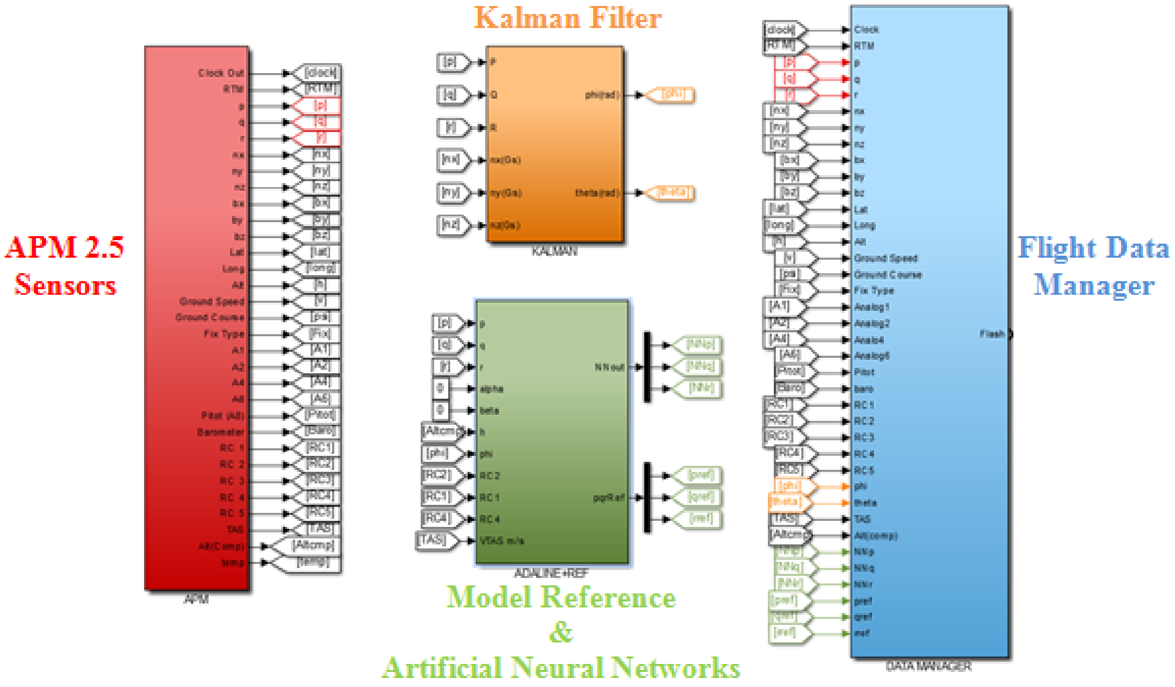


Figure 8.10. Simulink Model Top Level

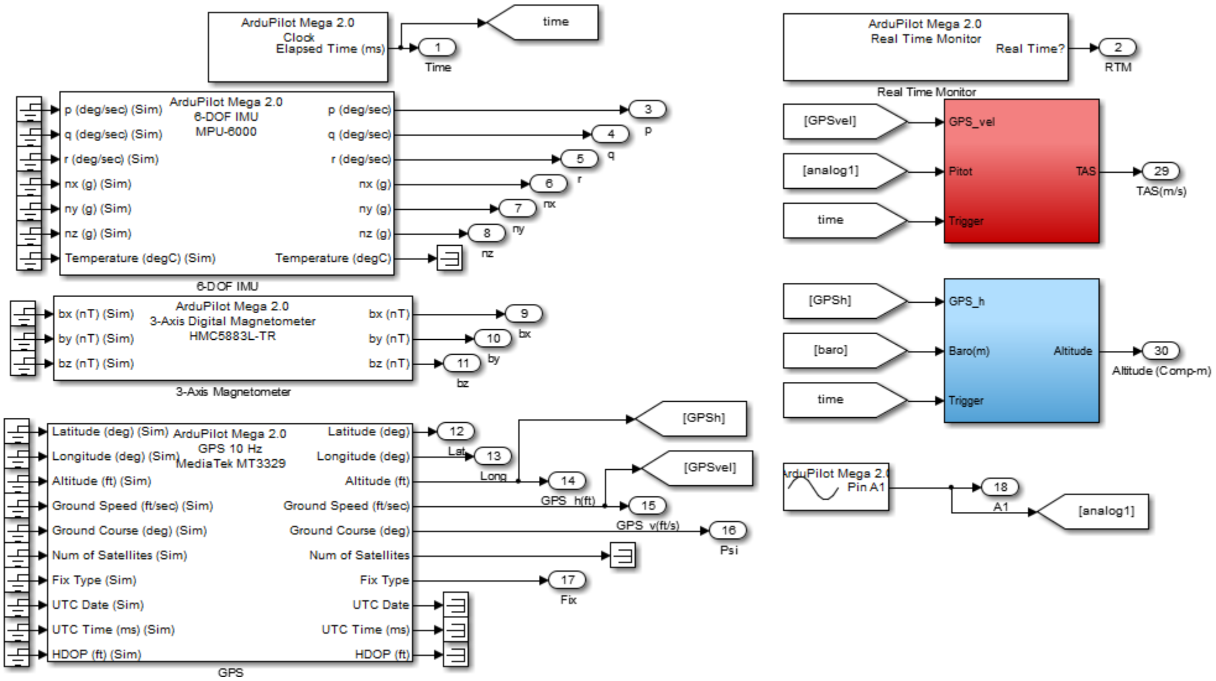


Figure 8.11. Simulink Sensor Model

In previous efforts, the Skywalker 1880 platform was used as an autonomous platform. For this effort, a sensor fusion solution was developed. A discrete Kalman filter was designed in Simulink in order to determine bank and pitch angles of the platform. The Kalman filter received inputs from a complementary filter that integrated data from the gyroscopes and the accelerometer. The Kalman filter demonstrated excellent performance and accuracy in the estimation of the desired Euler angles. Based on these characteristics, the estimator was included in the Simulink model with the objective to generate important variables for the definition of Self/Non-self projections. Figure 8.12 shows a Simulink model with the basic architecture of the Kalman filter implementation.

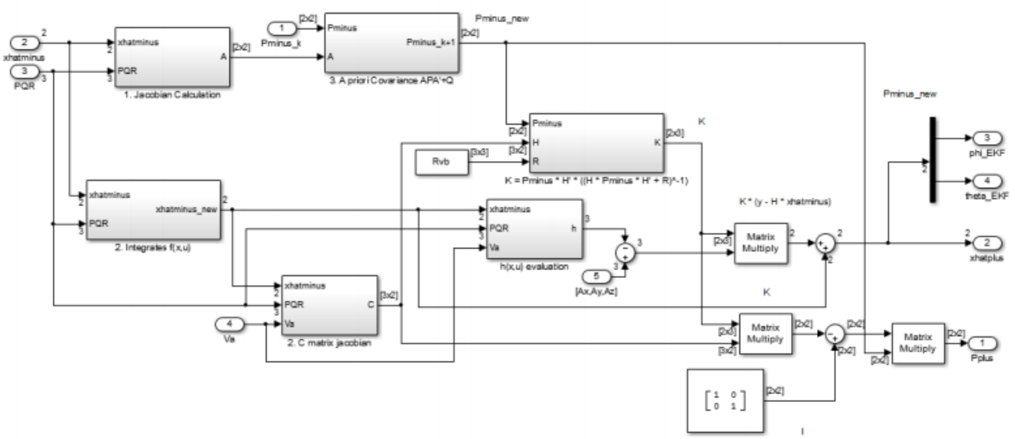


Figure 8.12. Kalman Filter Architecture

A model reference controller and ANNs were incorporated into the system in order to generate features for the definition of self/non-self. The ANN used was an ADALINE network, which requires eight variables as input. The Simulink model presents a relatively complex architecture that will not be shown in detail for simplicity.

8.3.4. Flight Test Results

Two nominal flights and four failure flights were performed. A total of 38 different channels were recorded into the flash memory. Out of those, only 18 were selected for the definition of projections. Table 8.7 presents the features that were recorded and selected for self/non-self definition.

Table 8.7. Self/Non-self Features for Flight Testing

p	NN_p	φ
q	NN_q	θ
r	NN_r	d_e
p_{ref}	a_x	d_a
q_{ref}	a_y	d_r
r_{ref}	a_z	d_T

153 projections were tested against flight test data in order to determine the detection performance of the generated selves. The selves analyzed the data from four different failure flights. The analysis determined that 30 different self projections were able to capture the dynamic fingerprint of abnormal conditions. Table 8.8 presents these projections.

Table 8.8. Skywalker Self Projections

Self#	Features	Self#	Features
1	p_{ref}, q_{ref}	**16**	p_{ref}, θ
2	p_{ref}, r_{ref}	**17**	p_{ref}, φ
3	p_{ref}, p	**18**	r_{ref}, a_x
4	p_{ref}, q	**19**	p, d_a
5	p_{ref}, r	**20**	q, d_a
6	p_{ref}, NN_p	**21**	r, a_x
7	p_{ref}, NN_q	**22**	NN_p, d_a
8	p_{ref}, NN_r	**23**	NN_q, d_a
9	p_{ref}, a_x	**24**	a_x, φ
10	p_{ref}, a_y	**25**	a_x, θ
11	p_{ref}, a_z	**26**	a_x, d_a
12	p_{ref}, d_a	**27**	a_x, d_T
13	p_{ref}, d_e	**28**	θ, d_a
14	p_{ref}, d_T	**29**	φ, d_a
15	p_{ref}, d_r	**30**	d_a, d_e

The selected projections obtained an average 21.11% and 30.68% DR for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver respectively. These detection rates may seems to be low. However, they should be considered in conjunction with the expected 0 FA rate. Table 8.9 presents the DR and FA rate of a sample set of projections.

Table 8.9. Detection Rate and False Alarm Rate for 4 Failures

	Right Aileron Locked at Wings Level		**Right Aileron Locked at Bank Maneuver**		**Left Aileron Locked at Wings Level**		**Left Aileron Locked at Bank Maneuver**	
Self#	**FA**	**DR**	**FA**	**DR**	**FA**	**DR**	**FA**	**DR**
1	0.00	25.6	0.00	28.7	0.00	12.4	0.00	23.8
12	0.00	25.1	0.00	34.8	0.00	19.2	0.00	34.1
13	0.00	32.5	0.00	36.4	0.00	31.6	0.00	39.3
15	0.00	24.6	0.00	44.1	0.00	14.8	0.00	25.5
22	0.00	10.2	0.00	15.6	0.00	11.3	0.00	30.5
30	0.00	20.6	0.00	23.3	0.00	26.2	0.00	32.2
Average	0.00	23.1	0.00	30.5	0.00	19.2	0.00	30.9

The individual selves present a low DR; however, if they are integrated into a single detection mechanism the DR improves greatly. For example, if the 30 selves selected are used for failure detection, then the DR percentage improves to 71.9% and 90.7% for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver, respectively. This configuration is able to obtain a significant improvement in DR but it also increases the FA rates approximately to 5.2%. For this reason, other configurations were tested in order to obtain an acceptable tradeoff between DR and FA. Two more configurations of selves were tested. The second configuration utilizes only the 6 selves shown in Table 8.9. This configuration obtained an average of 43.3% and 58.3% DR for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver respectively, with FA rates of approximately 1% for both cases. The third configuration of selves utilizes only projections that have 0% FA rate for each failure disregarding FA for other types of failures. This configuration guarantees that the FA rate will be low and it also offers the possibility to use this method for identification purposes. The third configuration presents an average of 59.6% and 77.3% DR for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver, respectively and less than 0.5% FA. Table 8.10 presents a summary of these results.

Table 8.10. Detection Rates and False Alarms for 3 Configurations of Selves

	R. Aileron Locked at Wings Level		**R. Aileron Locked at Bank Maneuver**		**Left Aileron Locked at Wings Level**		**Left Aileron Locked at Bank Maneuver**	
Self #	**FA**	**DR**	**FA**	**DR**	**FA**	**DR**	**FA**	**DR**
1	5.2	74.3	1.54	90.0	4.3	69.4	2.33	91.4
2	0.52	47.2	0.12	60.2	1.0	39.4	0.02	56.3
3	0.3	61.2	0.0	80.9	0.2	58.0	0.0	73.6

It should be noted that the four failures investigated in flight test are considered to be low magnitude, which makes them more difficult to detect. Because the locked position of the control surface is close to trim, no significant dynamic failure effects are noticeable until pilot input is provided. The DR is computed over the entire period of time that the failure is active, disregarding the presence of pilot input. This approach produces conservative low DR values as presented in Table 8.10. This effect is increased by the short duration of the flight at failure conditions, which is approximately 20 seconds.

Table 8.9 shows that single projections can only obtain a maximum of 30.9% detection rate. Nevertheless, when the projections are integrated as a single mechanism the DR increases greatly (see Table 8.10). This occurs because each projection at certain time periods of the flight test only captures abnormal dynamics when excited by certain maneuvers and commands. On the other hand, integrating several projections allows the mechanism to capture abnormal dynamic fingerprints at different periods of time during the flight tests. Figures 8.13-8.15 show the detection activity of single projections and Figures 8.16-8.17 shows the detection activity when the projections are integrated into a single mechanism. The value of 1 represents that a detector was been activated and alternatively a value of 0 determines that none of the detectors have been activated. The first 5 seconds are nominal flight test conditions while the remainder of the time corresponds to flight test data under an upset condition.

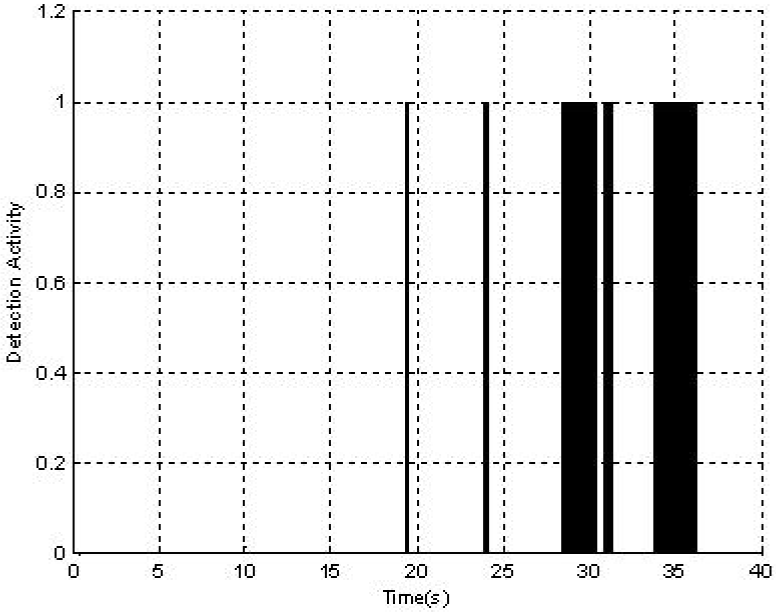


Figure 8.13. Self #1 Detector Activity for Left Aileron Locked at Bank Maneuver

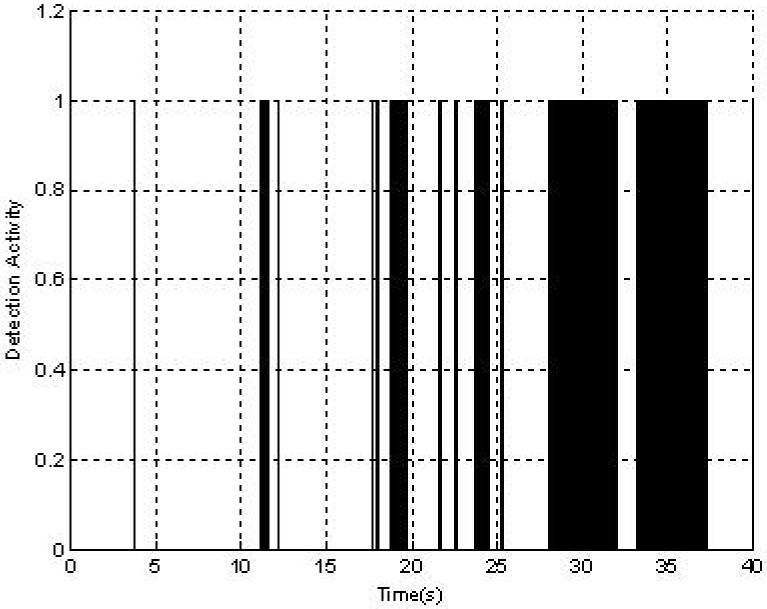


Figure 8.14. Self #17 Detector Activity for Left Aileron Locked at Bank Maneuver

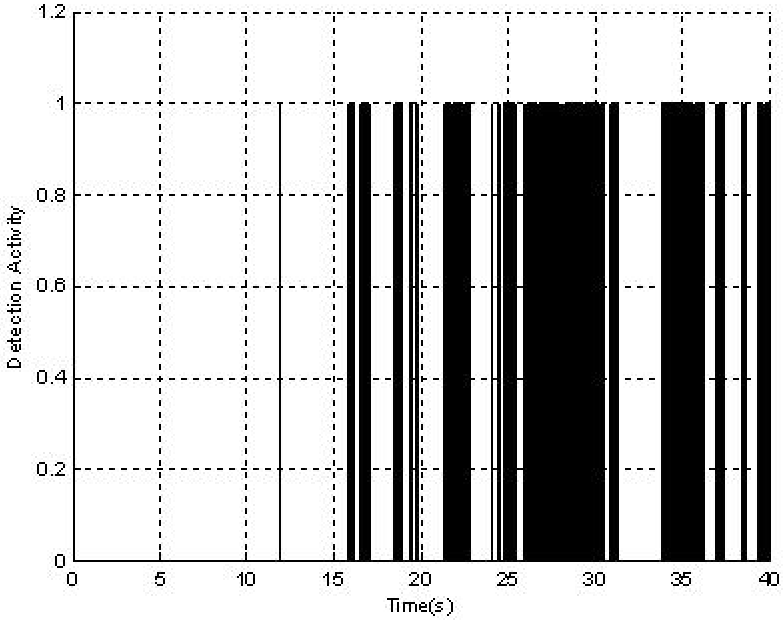


Figure 8.15. Self #18 Detector Activity for Left Aileron Locked at Bank Maneuver

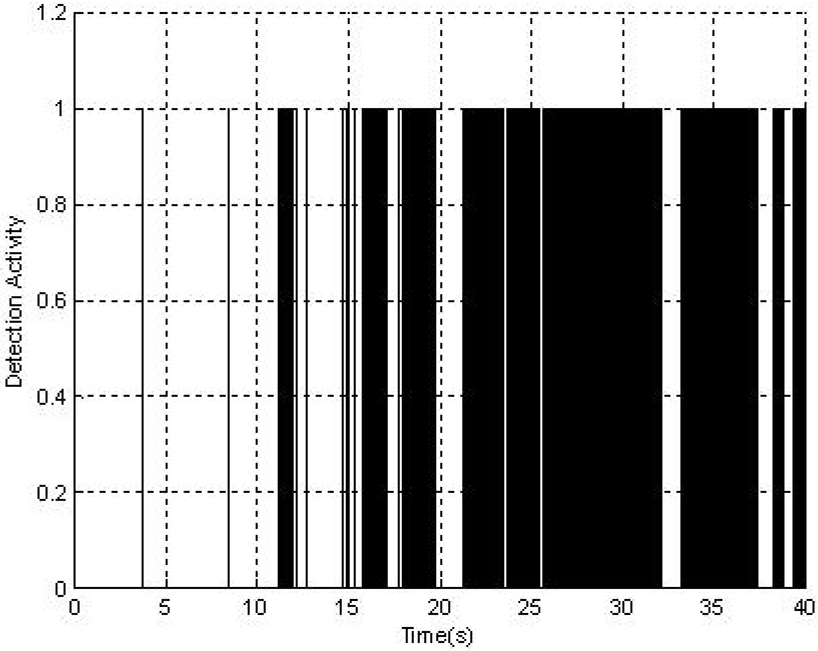


Figure 8.16. Method 2 Detector Activity for Left Aileron Locked at Bank Maneuver

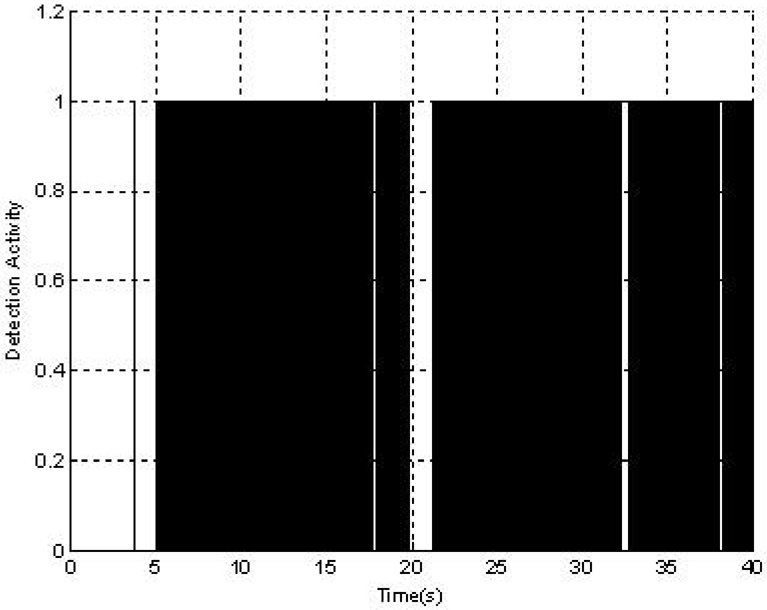


Figure 8.17. Method 1 Detector Activity for Left Aileron Locked at Bank Maneuver

References

[1]Janeway, C. A., Travers, P., Walport, M., and Shlomchik, M. J., *Immunobiology: The Immune System in Health and Disease*, 6th ed., Garland Science, New York, 2005.

[2]Steinman, R. M., "Introduction to Dendritic Cells," Laboratory of Cellular Physiology and Immunology, The Rockefeller University official site available at http://lab.rockefeller.edu/steinman/dendritic_intro/, last accessed Jan.2013.

[3]Banchereau, J., Steinman R. M., "Dendritic Cells and the Control of Immunity," *Nature*, Vol. 392, March 1998, doi:10.1038/32588.

[4]Pletinckx, K., Döhler A., Pavlovic V., and Lutz M. B., "Role of Dendritic Cell Maturity/Co-stimulation for Generation, Homeostasis, and Suppressive Activity of Regulatory T-Cells," *Frontiers in Immunology*, Vol.2, Sept. 2011, doi:10.3389/fimmu.2011.00039.

[5]Dasgupta D., (editor), "Artificial Immune Systems and Their Applications", Springer Verlag, 1998

[6]Dasgupta D., Nino L. F., "Immunological Computation – Theory and Applications", CRC Press, Auerbach Publications, Taylor & Francis Group, 2009

[7]Dasgupta, D., KrishnaKumar, K., Wong, D., and Berry, M., "Negative Selection Algorithm for Aircraft Fault Detection," *ICARIS 2004*, edited by G. Nicosia et al., LNCS 3239,2004, pp. 1–13.

[8]KrishnaKumar, K., "Artificial Immune System Approaches for Aerospace Applications," *Proc. of the 41st Aerospace Sciences Meeting & Exhibit*, AIAA-2003-0457, Reno, Nevada, 2003.

[9]Perhinschi, M. G., Moncayo, H., and Davis, J., "Integrated Framework for Artificial Immunity-Based Aircraft Failure Detection, Identification, and Evaluation," *AIAA Journal of Aircraft*, Vol. 47,No. 6, Nov.-Dec. 2010, pp. 1847-1859.

[10]Moncayo, H., Perhinschi, M. G., and Davis, J., "Aircraft Failure Detection and Identification Using an Immunological Hierarchical Multi-self Strategy," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, Jul.-Aug. 2010, pp. 1105-1114.

[11]Davis, J., Perhinschi, M. G., and Moncayo, H., "Evolutionary Algorithm for Artificial Immune System-Based Failure Detector Generation and Optimization," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, Mar.-Apr. 2010, pp. 302-320.

[12]Perhinschi, M. G., Porter, J., Moncayo, H., Davis, J., and Wayne, W. S., "Artificial Immune System-Based Detection Scheme for Aircraft Engine Failures," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5, Sept.-Oct. 2011, pp 1423-1440.

[13]Moncayo H., Perhinschi M. G., Davis J., "Artificial Immune System – Based Aircraft Failure Detection and Identification Over an Extended Flight Envelope", *The Aeronautical Journal*, Vol. 115, No. 1163, Jan. 2011

[14]Moncayo H., Perhinschi M. G., Davis J., "Artificial Immune System – Based Aircraft Failure Evaluation Over Extended Flight Envelope", *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4 pp 989-1001, Jul.-Aug. 2011

[15]Davis, J., "The Design of an Evolutionary Algorithm for Artificial Immune System Based Failure Detector Generation and Optimization," MS Thesis, West Virginia University, August, 2010.

[16]Moncayo, H., Perhinschi, M. G.,and Davis, J., "Simulation Environment for the Development and Testing of Immunity-Based Aircraft Failure Detection Schemes," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Portland, OR, August 2011.

[17]Hever Moncayo, "Immunity-Based Detection, Identification, and Evaluation of Aircraft Sub-System Failures", PhD Thesis, West Virginia University, December, 2009.

[18] Moncayo, H., and Perhinschi, M. G., *Aircraft Fault Tolerance: A Biologically Inspired Immune Framework for Sub-System Failures*, VDM Verlag Dr. Muller GmbH & Co. KG, VDM Publishing House ltd., Saarbruecken, Germany, 2011.

[19] Napolitano M. R., Younghawn A., Seanor B., “A Fault Tolerant Flight Control System for Sensor and Actuator Failures Using Neural Networks,” Aircraft Design, Vol. 3 No 2, 2000

[20] Takahashi, K., and Yamada, T., “Application of an Immune Feedback Mechanism to Control Systems,” *The Japan Society of Mechanical Engineers, JSME International Journal*, Series C, Vol. 41, No. 2, 1998.

[21] Moncayo H., Perhinschi M. G., Wilburn B., Wilburn J., Karas O., “UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity-based Mechanism”, *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, August 2012

[22] Karr, C., Nishita, K., Graham, K., “Adaptive Aircraft Flight Control Simulation Based on an Artificial Immune System,” *Applied Intelligence*, Vol. 23, No 3, 2005, pp. 295-308.

[23] Perhinschi M. G., Napolitano M.R., “Integration of Computer Simulation for Flight Dynamics and Control Education”, *Computers in Education Journal*, vol. XVIII, no. 1, pp 13-22, Jan.-Mar. 2008

APPENDIX A

List of Publications Resulting from the DARPA Sponsored Research Effort

Peer-reviewed Journal Papers

11). Perhinschi M. G., Al Azzawi D., Moncayo H., “Immunity-based Aircraft Sensor Failure Evaluation”, in preparation.

10). Al Azzawi D., Perhinschi M. G., Moncayo H., Perez A. E., “Dendritic Cell Mechanism for Detection, Identification, and Evaluation of Aircraft Failures”, submitted to *Journal of Control Engineering Practice*, Dec. 2014

9). Togayev A., Perhinschi M. G., Al Azzawi D., Moncayo H., Perez A., “Immunity-based Accommodation of Aircraft Subsystem Failures”, submitted to *Aircraft Engineering and Aerospace Technology*, August 2014

8). Perhinschi M. G., Al Azzawi D., Moncayo H., Perez A., Togayev A., “Immunity-based Aircraft Actuator Failure Evaluation”, submitted to *Aircraft Engineering and Aerospace Technology*, July 2014

7). Al Azzawi D., Moguel I., Perhinschi M. G., Moncayo H., Perez A., Togayev A., “Comparison of Immunity-Based Schemes for Aircraft Failure Detection and Identification”, submitted to *IEEE Transactions on Evolutionary Computation*, Sept. 2014

6). Perez A., Moncayo H., Perhinschi M. G., Al Azzawi D., Togayev A., “A Bio-Inspired Adaptive Control Compensation System for an Aircraft Outside Bounds of Nominal Design”, submitted to *ASME Journal of Dynamic Systems, Measurement, and Control*, Dec. 2014.

5). Perhinschi M. G., Al Azzawi D., Moncayo H., Togayev A., Perez A., Moguel I., “Immunity-based Flight Envelope Prediction at Post-failure Conditions”, submitted to *Aerospace Science and Technology*, Sept 2014.

4). Moguel I., Moncayo H., Perez A., Perhinschi M. G., Al Azzawi D., Togayev A., “Structured Non-Self Approach for Aircraft Failure Identification within an Immunity-based Fault Tolerance Architecture”, submitted to *IEEE Trans. on Aerospace and Electronic Systems*, Jan. 2014.

3). Perhinschi M. G., Moncayo H., Al Azzawi D., “Integrated Immunity-Based Framework for Aircraft Abnormal Conditions Management”, *AIAA Journal of Aircraft*; Vol. 51, Iss. 6, pp 1726-1739, doi: 10.2514/1.C032381, Nov.-Dec., 2014

2). Al Azzawi D., Perhinschi M. G., Moncayo H., “Artificial Dendritic Cell Mechanism for Aircraft Immunity-Based Failure Detection and Identification”, *AIAA Journal of Aerospace Information Systems*, DOI: 10.2514/1.I010214, Vol. 11, No. 7, July, 2014.

1). Perhinschi M. G., Moncayo H., Al Azzawi D., Moguel, I. “Generation of Artificial Immune System Antibodies Using Raw Data and Cluster Set Union”, *IC: International Journal of Immune Computation*, Vol. 2, No. 1, pp 1-15, March, 2014

Peer-reviewed Conference Papers

9). Moguel I., Moncayo H., Perez A. E., Perhinschi M. G., “In-Flight Testing of a Bio-Inspired Approach for Assessment of an UAV Outside Bounds of Nominal Design”, accepted for presentation at the *AIAA Science and Technology Forum*, Orlando, FL., 2015

8). Perez A. E., Moncayo H., Moguel I., Togayev A., Perhinschi M. G., Al Azzawi D., “Pilot-in-the-Loop Assessment of Neurally Augmented Immunity Based Fault Tolerant Control Laws in a Motion-Based Flight Simulator”, accepted for presentation at the *AIAA Guidance, Navigation and Control Conference*, Kissimmee, Florida, Jan. 2015.

7). Togayev A., Perhinschi M. G., Al Azzawi D., Moncayo H., Moguel I., Perez A. E., “Immunity-Based Abnormal Condition Accommodation of Aircraft Sub-system Failures”, *Proc. of the ASME Annual Dynamic Systems and Control Conference*, San Antonio, Texas, Oct. 2014

6). Moguel I., Moncayo H., Perhinschi M. G., Perez A. E., Al Azzawi D., Togayev A., "Bio-Inspired Approach for Aircraft Health Assessment and Flight Envelope Estimation", *Proc. of the ASME Annual Dynamic Systems and Control Conference*, San Antonio, Texas, Oct. 2014

5). Perez A. E., Moncayo H., Moguel I., Perhinschi M. G., Al Azzawi D., Togayev A., "Development of Immunity-based Adaptive Control Laws for Aircraft Fault Tolerance", *Proc. of the ASME Annual Dynamic Systems and Control Conference*, San Antonio, Texas, Oct. 2014

4). Al Azzawi D., Perhinschi M. G., Togayev A., Moncayo H., Moguel I., Perez A. E., "Evaluating Aircraft Abnormal Conditions Using an Artificial Dendritic Cell Mechanism", Proc. of the *International Conference and Exhibition on Mechanical & Aerospace Engineering*, Philadelphia, Pennsylvania, Sept. 2014

3). Moguel I., Moncayo H., Perhinschi M. G., Al Azzawi D., Perez A. E., Togayev A., "Biologically-Inspired Approach for Aircraft Management Under Upset Conditions", Proc. of the *International Conference and Exhibition on Mechanical & Aerospace Engineering*, Philadelphia, Pennsylvania, Sept. 2014

2). Perhinschi M. G., Moncayo H., Al Azzawi D., "Development of Immunity-Based Framework for Aircraft Abnormal Conditions Detection, Identification, Evaluation, and Accommodation", *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Boston MA, Aug. 2013

1). Al Azzawi D., Perhinschi M. G., Moncayo H., "Dendritic Cell Mechanism for Aircraft Immunity-Based Failure Detection and Identification", *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Boston MA, Aug. 2013

Graduate Student Theses

1). Moguel I., "Bio-Inspired Mechanism for Aircraft Assessment under Upset Conditions", MS Thesis, Embry Riddle Aeronautical University, graduated May 2014.

2). Togayev A., "Immunity-Based Accommodation of Aircraft Subsystem Failures", MS Thesis, West Virginia University, defended June 2014, graduated December 2014.

3). Al Azzawi D., "Aircraft Abnormal Conditions Detection, Identification, and Evaluation Using Innate and Adaptive Immune Systems Interaction", PhD Thesis, West Virginia University, graduated August 2014.

4). Perez A., "Bio-Inspired Adaptive Control Laws for Aircraft Accommodation under Upset Conditions", PhD Thesis, Embry Riddle Aeronautical University, in preparation.

APPENDIX B

Aircraft Subsystem ACDIE Scheme – Simulink Implementation

(on DVD)

APPENDIX C

Installation and User Manual for the Aircraft Subsystem ACDIE Scheme

WestVirginiaUniversity®

AIS Aircraft FDIE

USER'S MANUAL

Contents

1. System Requirements 3
2. Installing "AIS Aircraft FDIE" 4
3. Running the Demos with SimDemo 11
4. Running the Demos Inside MATLAB 20

1

System Requirements

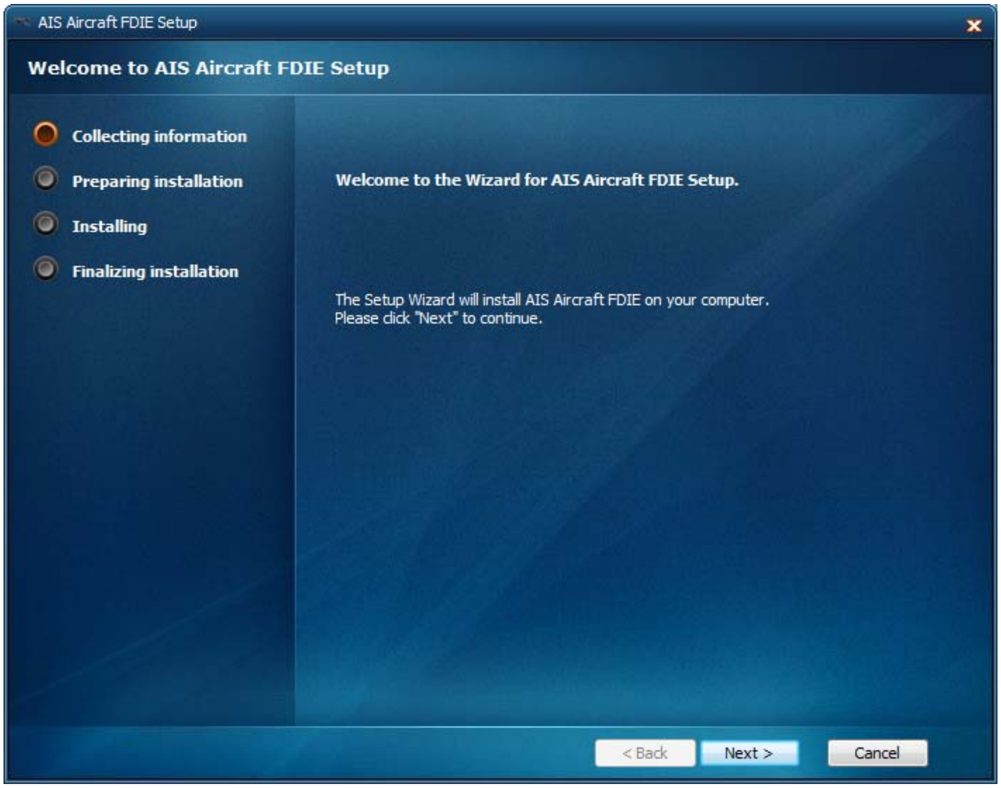
Operating System:	Windows XP SP3, Windows Vista SP2, Windows 7, Windows 8
OS Architecture:	x64 (64-bit)
Processor:	Intel® Core™ @ 1.7GHz or higher
Memory:	8 GB RAM or more
Hard Drive:	4 GB free hard disk space or more
Prerequisites:	• Microsoft® .NET™ Framework 4 or later (included with the installer) • MATLAB®/SIMULINK® R2010a or later (32-bit only)

2

Installing "AIS Aircraft FDIE"

Installing the "AIS Aircraft FDIE" is straightforward. Run the `AISAircraftFDIEInstaller.exe` file on the DVD and follow the instructions that appear on the installer interface.

<u>NOTE:</u> Enter the provided serial number when the installer asks you for one.

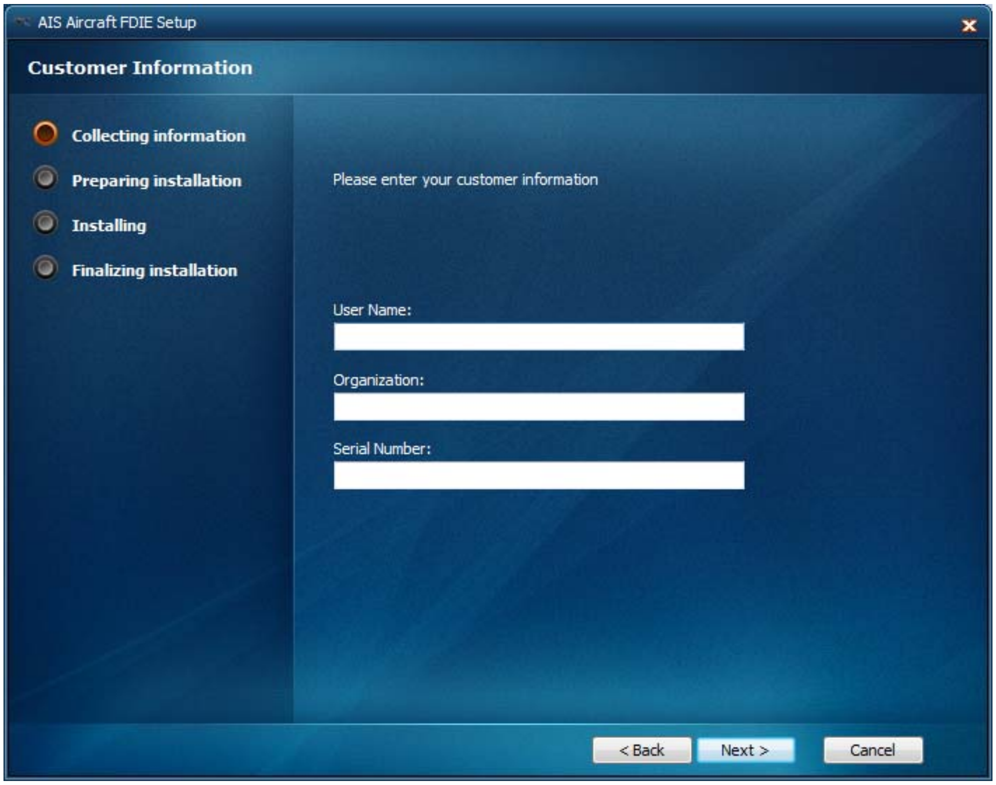


AIS Aircraft FDIE Setup

Welcome to AIS Aircraft FDIE Setup

Welcome to the Wizard for AIS Aircraft FDIE Setup.

The Setup Wizard will install AIS Aircraft FDIE on your computer.
Please click "Next" to continue.

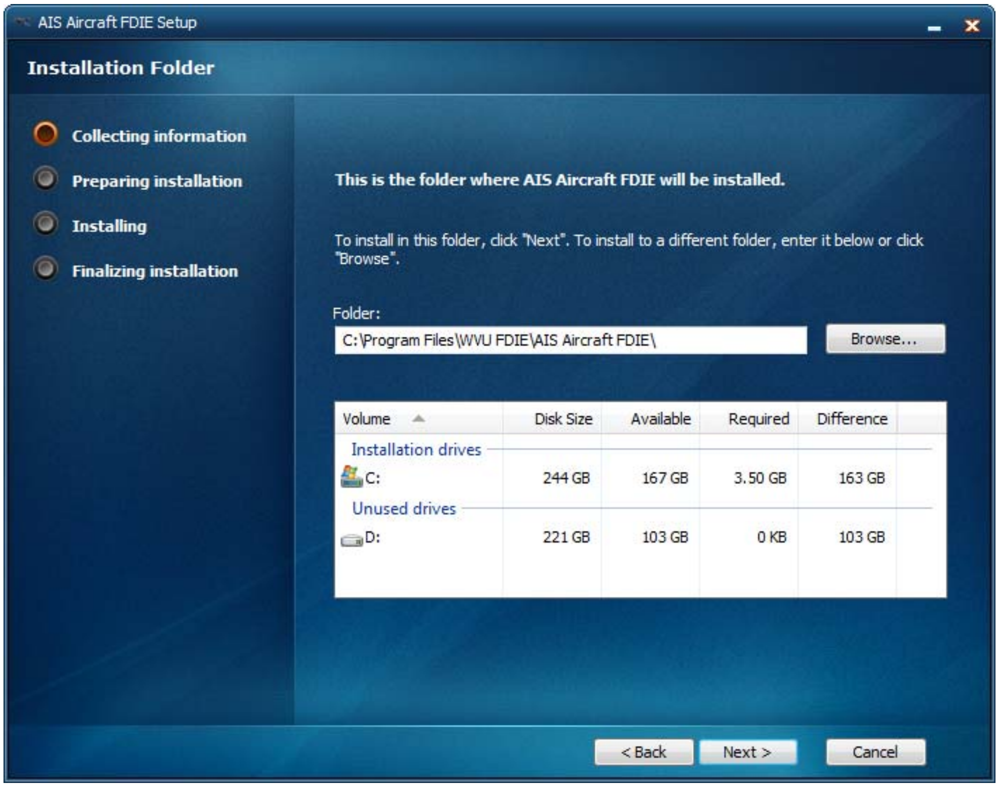


Please enter your customer information

User Name:

Organization:

Serial Number:

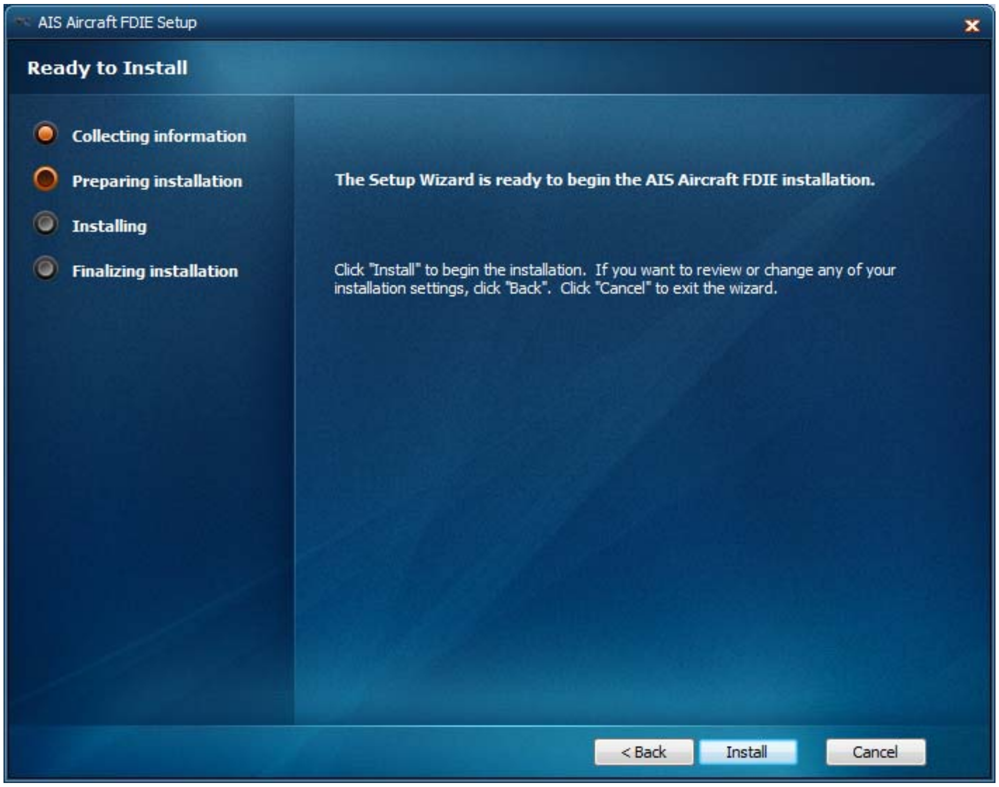


This is the folder where AIS Aircraft FDIE will be installed.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files\WVU FDIE\AIS Aircraft FDIE\

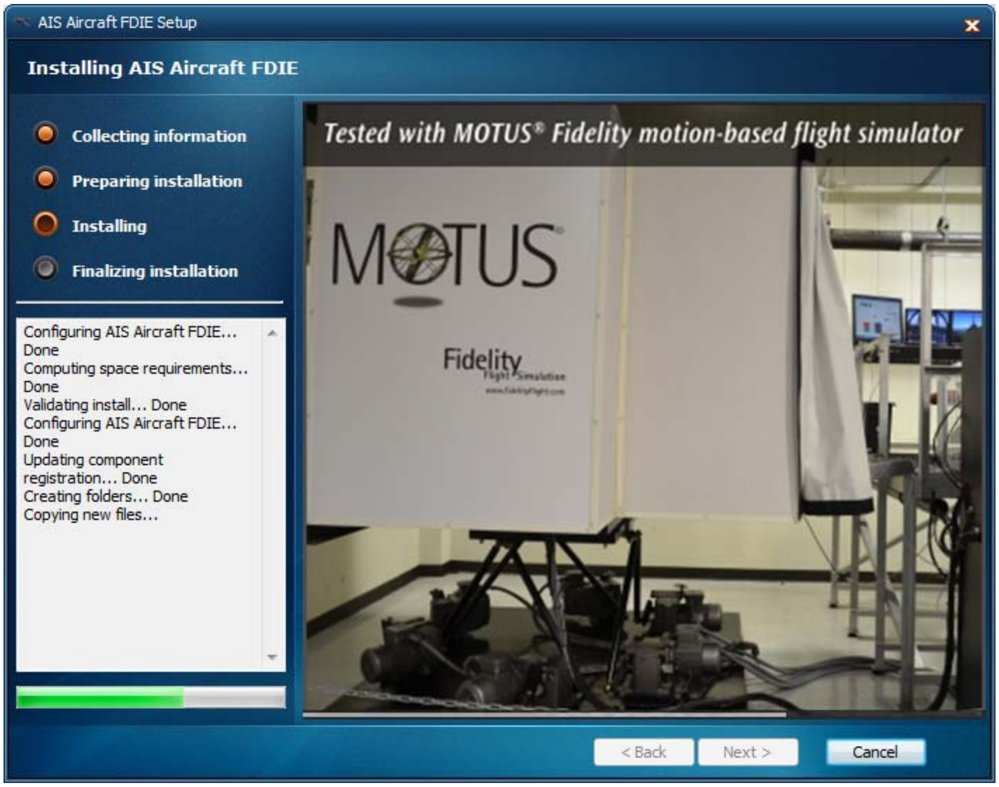
Volume	Disk Size	Available	Required	Difference
Installation drives				
C:	244 GB	167 GB	3.50 GB	163 GB
Unused drives				
D:	221 GB	103 GB	0 KB	103 GB



Ready to Install

The Setup Wizard is ready to begin the AIS Aircraft FDIE installation.

Click "Install" to begin the installation. If you want to review or change any of your installation settings, click "Back". Click "Cancel" to exit the wizard.

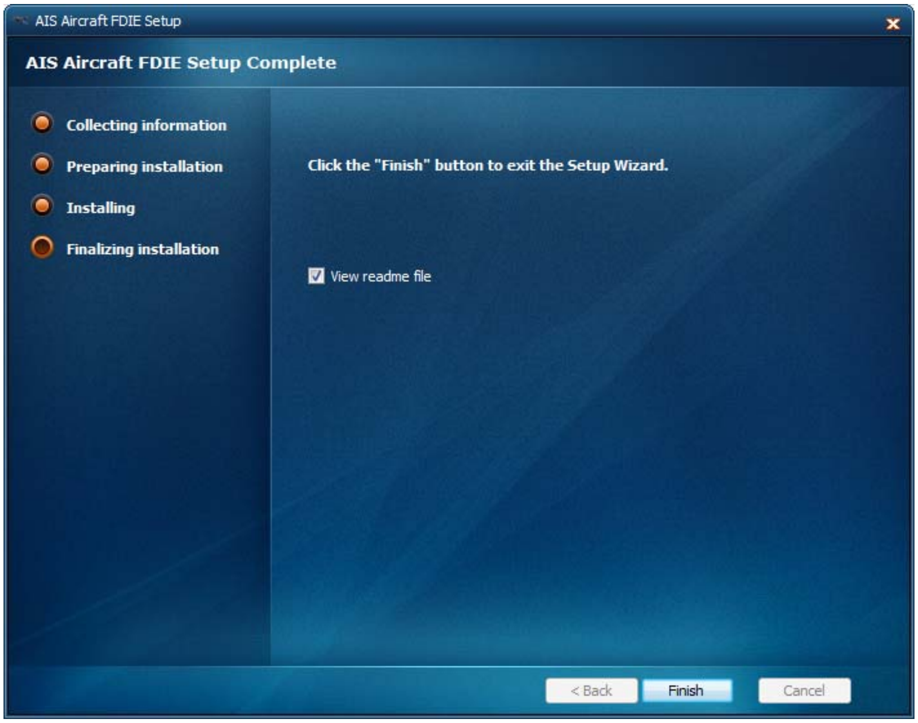


Installing AIS Aircraft FDIE

```
Configuring AIS Aircraft FDIE...
Done
Computing space requirements...
Done
Validating install... Done
Configuring AIS Aircraft FDIE...
Done
Updating component
registration... Done
Creating folders... Done
Copying new files...
```

Tested with MOTUS® Fidelity motion-based flight simulator

After installing the "AIS Aircraft FDIE," the installation folder should appear as shown below.

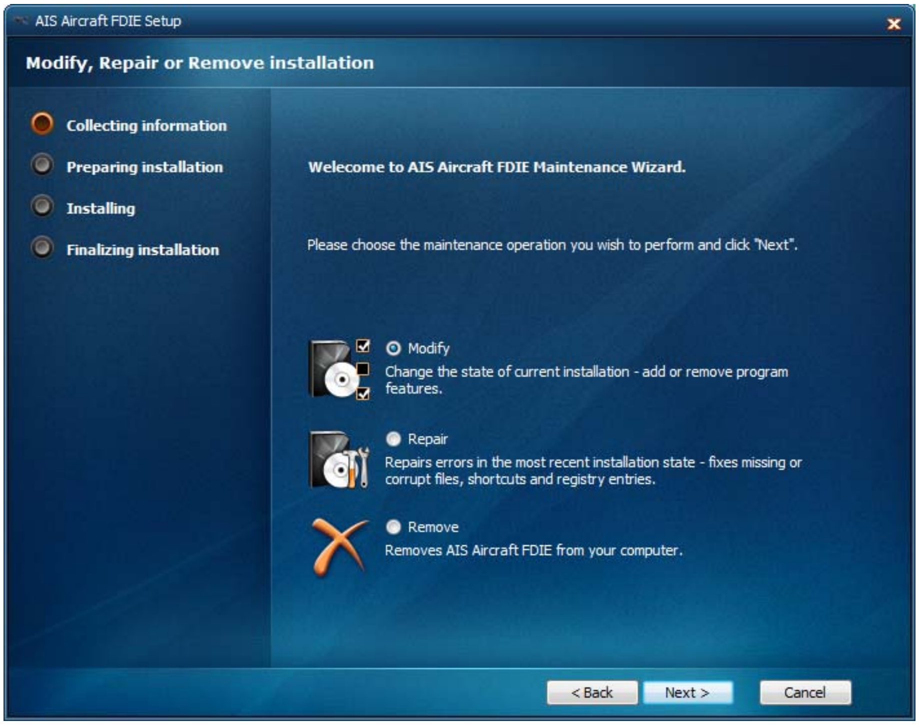


AIS Aircraft FDIE Setup Complete

Click the "Finish" button to exit the Setup Wizard.

☑ View readme file

To uninstall or repair the installation, run the `AISAircraftFDIEInstaller.exe` file again and choose the desired operation.



Modify, Repair or Remove installation

Welecome to AIS Aircraft FDIE Maintenance Wizard.

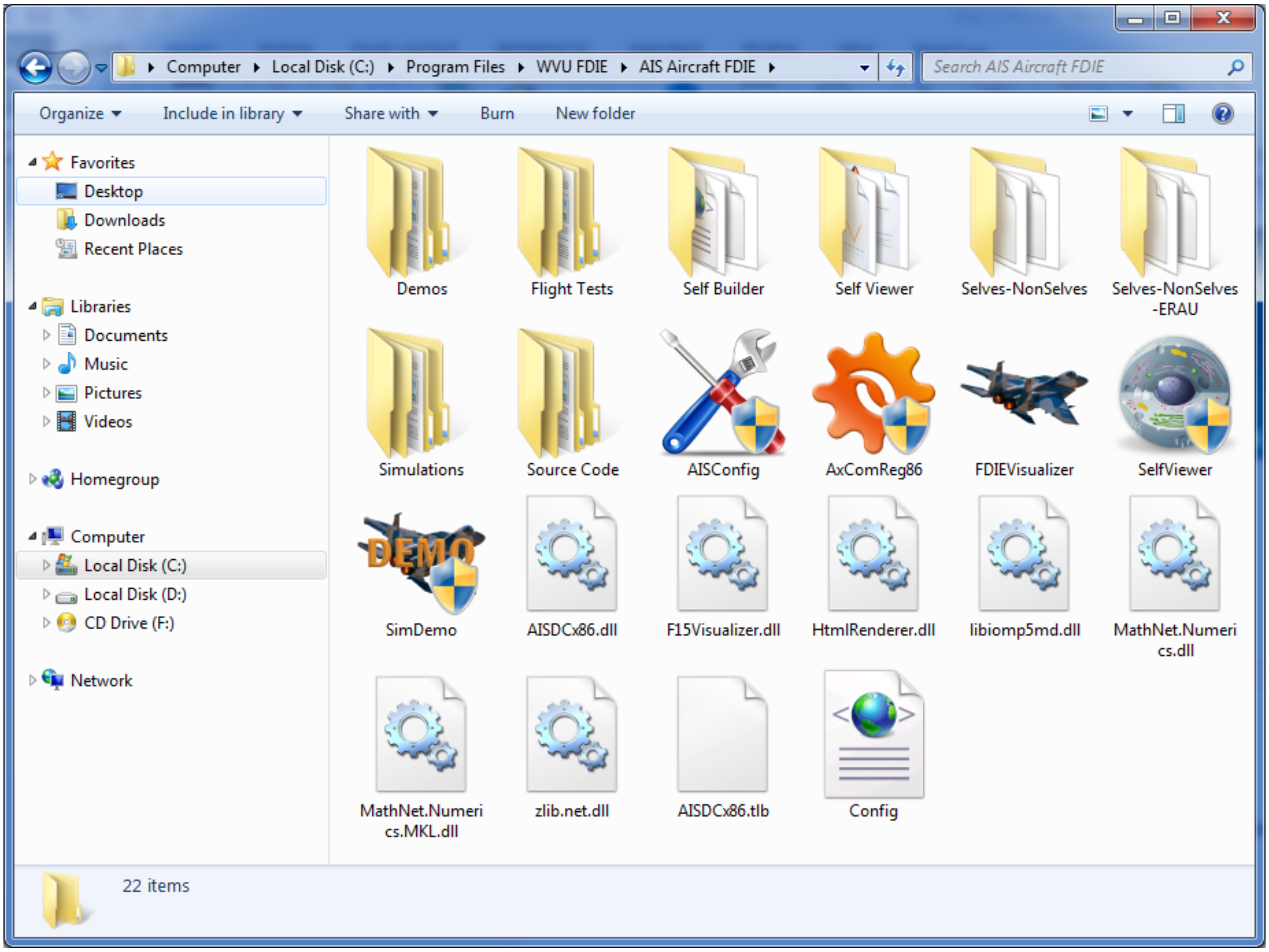
Please choose the maintenance operation you wish to perform and click "Next".

● Modify
Change the state of current installation - add or remove program features.

○ Repair
Repairs errors in the most recent installation state - fixes missing or corrupt files, shortcuts and registry entries.

○ Remove
Removes AIS Aircraft FDIE from your computer.

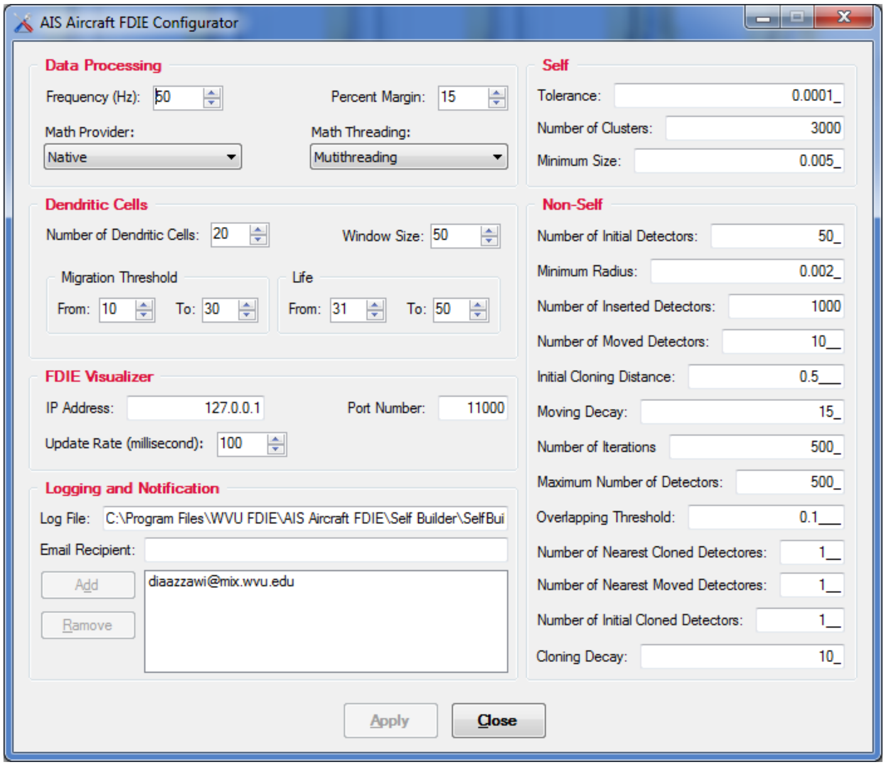
After installing the "AIS Aircraft FDIE," the installation folder should appear as shown below.



The installation folder contains the following components:

1. **Demos** folder containing a set of demo flight tests, each in its own folder. The name of each folder has the pattern `Subsystem_FailureType_FailureMagnitude`. For example, the folder `LeftStabilator_1_6` contains `.mat` data files for a flight test under the left stabilator failure, moving and locked (type 1 failure) at 6°.
2. **Flight Tests** folder containing a set of flight tests used in the development of the different AIS components. The name of each folder has the pattern `Subsystem_FailureType_FailureMagnitude_FlightPath`. For example, the folder `LeftAileron_1_2.5_123` contains `.mat` data files for a flight test under the left aileron failure, moving and locked (Type 1 failure) at 2.5°. Flight tests are categorized in folders representing the different categories of the subsystems (actuators, sensors, aileron, wing, etc.).
3. **Self Builder** folder containing a 64-bit `SelfBuilder.exe` tool for building the selves and nonselves, a 64-bit `AISConfig64.exe` tool for configuring the `SelfBuilder` tool, and a 64-bit `AxComReg64.exe` tool for registering the `AISDCx64.dll COM` Interop library.
4. **Self Viewer** folder containing MATLAB script files for the `SelfViewer` analysis tool.
5. **Selves-NonSelves** folder containing `.mat` data files of the selves and nonselves generated at West Virginia University using the Raw Data Set Union Method.
6. **Selves-NonSelves-ERAU** folder containing `.mat` data files of the selves and nonselves generated at Embry-Riddle Aeronautical University using the Cluster Set Union Method.

7. **Simulations** folder containing implementations of the DC mechanism approach and the Structured Nonself approach with MATLAB/Simulink 2010a (32-bit).
8. **Source Code** folder containing C# code projects for the DC mechanism library and other helper tools.
9. **AISConfig.exe**: A tool for configuring the different parameters needed for the building the selves and nonselves, the DC mechanism, and the FDIE Visualization tool.



Data Processing

Frequency (Hz): 50

Percent Margin: 15

Math Provider: Native

Math Threading: Mutithreading

Dendritic Cells

Number of Dendritic Cells: 20

Window Size: 50

Migration Threshold From: 10 To: 30

Life From: 31 To: 50

FDIE Visualizer

IP Address: 127.0.0.1

Port Number: 11000

Update Rate (millisecond): 100

Logging and Notification

Log File: C:\Program Files\WVU FDIE\AIS Aircraft FDIE\Self Builder\SelfBui

Email Recipient:

diaazzawi@mix.wvu.edu

Self

Tolerance: 0.0001_

Number of Clusters: 3000

Minimum Size: 0.005_

Non-Self

Number of Initial Detectors: 50_

Minimum Radius: 0.002_

Number of Inserted Detectors: 1000

Number of Moved Detectors: 10__

Initial Cloning Distance: 0.5___

Moving Decay: 15_

Number of Iterations 500_

Maximum Number of Detectors: 500_

Overlapping Threshold: 0.1___

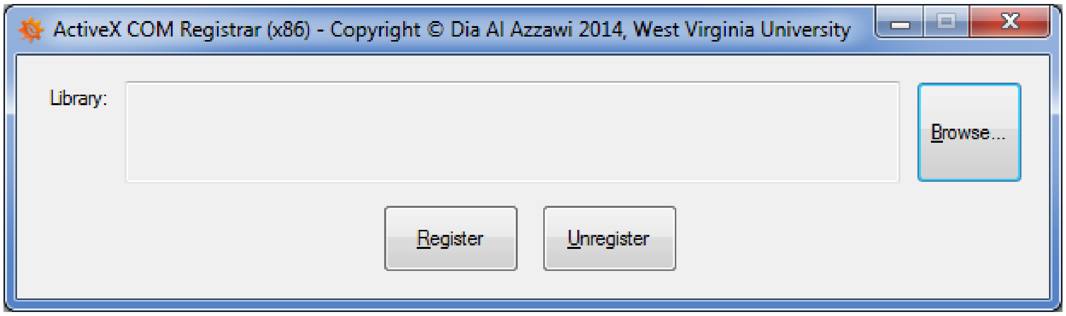
Number of Nearest Cloned Detectores: 1__

Number of Nearest Moved Detectores: 1__

Number of Initial Cloned Detectors: 1__

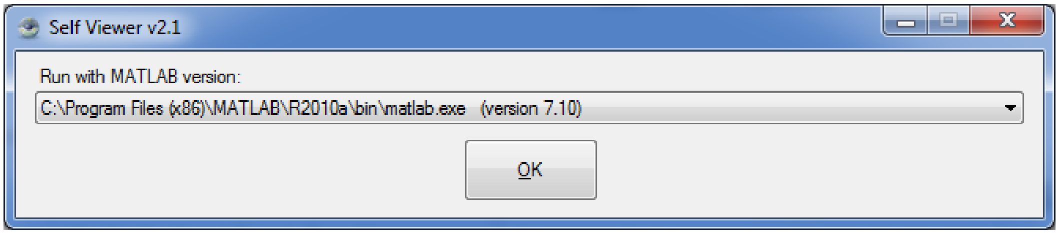
Cloning Decay: 10_

10. **AxComReg64.exe**: A tool for registering/unregistering the `AISDCx86.dll` library. By default, the installer usually registers this library automatically. Check the main installation folder to see if it contains the file `AISDCx86.tlb` or not. If this file exists, then the library registration was successful. If not, then run `AxComReg64.exe`, click the `Browse...` button to select the `AISDCx86.dll` file, then click the `Register` button.


ActiveX COM Registrar (x86) - Copyright © Dia Al Azzawi 2014, West Virginia University
Library:
Browse...
Register
Unregister

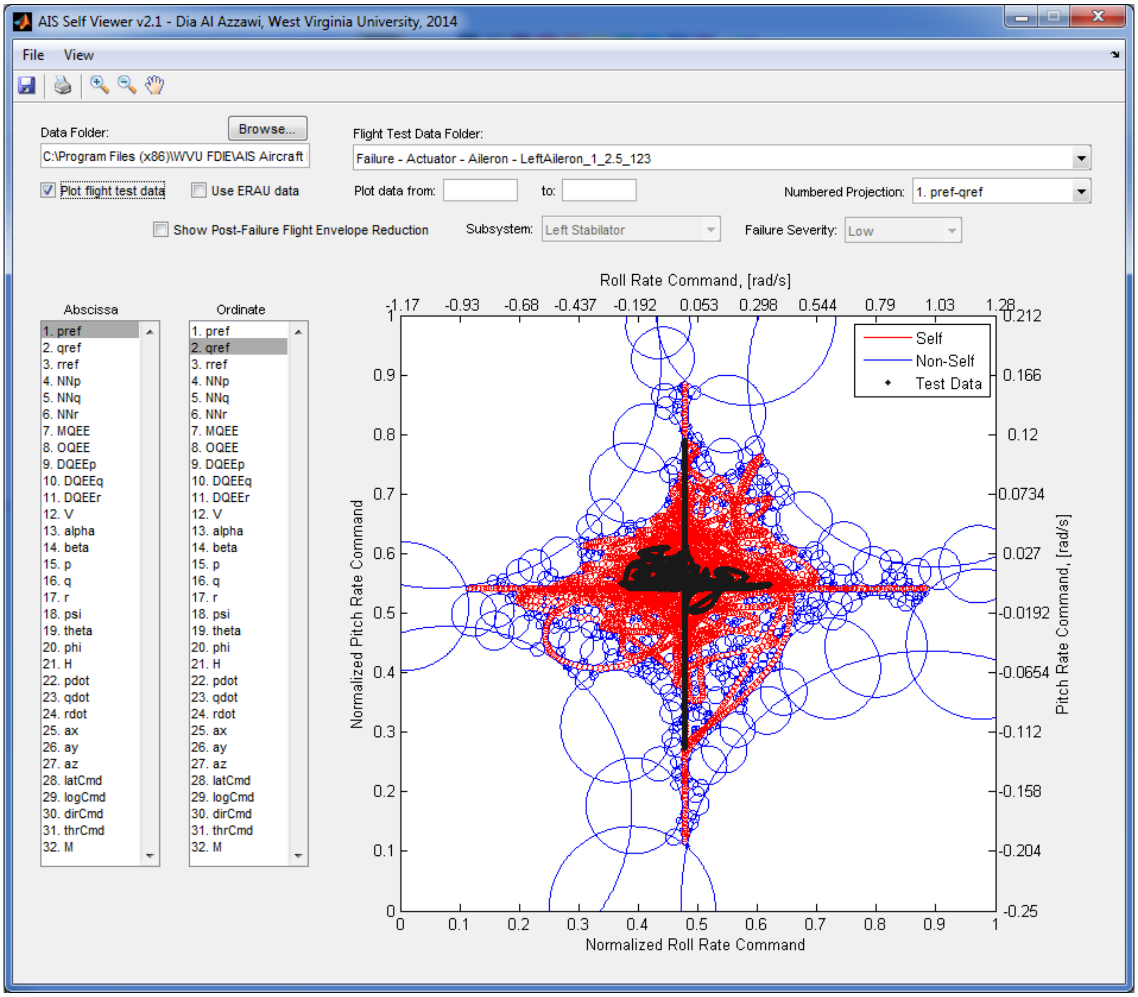
11. **FDIEVisualizer.exe**: A tool for visualizing the FDIE outcome from the simulation models. This tool will run automatically by the simulation models.

12. **SelfViewer.exe**: A tool for viewing and analyzing the 2D selves/nonselves. Be patient when running this tool for the first time as it takes few minutes to respond after doing extensive check and validation of the installed MATLAB versions that are compatible with the tool. After it responds, select the MATLAB version you want to run the SelfViewer with and click OK. This will launch the selected MATLAB command window which, in turn, runs the Self Viewer v2.1 tool.



Self Viewer v2.1

Run with MATLAB version: C:\Program Files (x86)\MATLAB\R2010a\bin\matlab.exe (version 7.10)



AIS Self Viewer v2.1 - Dia Al Azzawi, West Virginia University, 2014

Data Folder: C:\Program Files (x86)\WVU FDIE\AIS Aircraft

Flight Test Data Folder: Failure - Actuator - Aileron - LeftAileron_1_2.5_123

☑ Plot flight test data

□ Use ERAU data

Plot data from: to:

Numbered Projection: 1. pref-qref

□ Show Post-Failure Flight Envelope Reduction

Subsystem: Left Stabilator

Failure Severity: Low

Abscissa: 1. pref

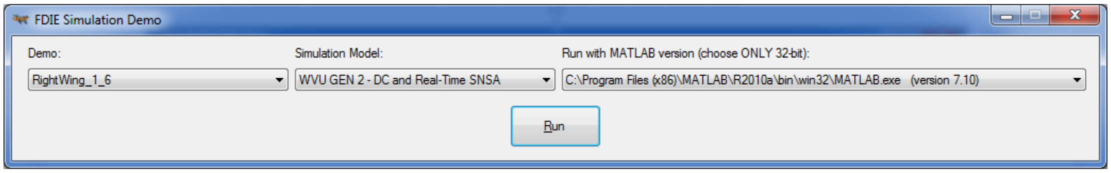
Ordinate: 2. qref

13. **SimDemo.exe**: A tool for the running the demo flight tests automatically. See “Running the Demo Flights Using Automatically” below.
14. **A set of Dynamic Link Libraries (DLLs)** and a **Config.xml** configuration file.

3

Running the Demos with SimDemo

1. To run the demo flights automatically, run `SimDemo.exe` tool, which takes few minutes to load the `FDIE Simulation Demo` window for the first time run.



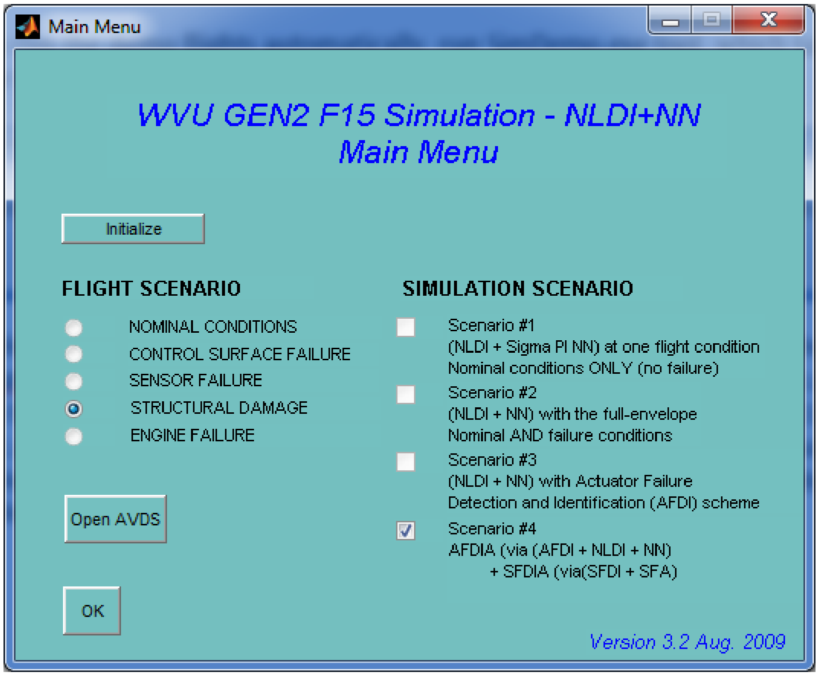
FDIE Simulation Demo

Demo: RightWing_1_6

Simulation Model: WVU GEN 2 - DC and Real-Time SNSA

Run with MATLAB version (choose ONLY 32-bit): C:\Program Files (x86)\MATLAB\R2010a\bin\win32\MATLAB.exe (version 7.10)

2. Select the demo flight you want to run, the simulation model to run the demo with, and the MATLAB version (32-bit only) to run the selected simulation. Click the `Run` button.
3. In the `Main Window`, click the `Initialize` button and wait until the `FDIE Outcome Visualizer v2.0` is loaded. Then select `Flight Scenario` that corresponds to the selected demo in the `FDIE Simulation Demo` window (see the above figure). Finally, select Scenario #4 in the `Simulation Scenario` options then click the `OK` button.



Main Menu

WVU GEN2 F15 Simulation - NLDI+NN
Main Menu

FLIGHT SCENARIO

○ NOMINAL CONDITIONS
○ CONTROL SURFACE FAILURE
○ SENSOR FAILURE
● STRUCTURAL DAMAGE
○ ENGINE FAILURE

SIMULATION SCENARIO

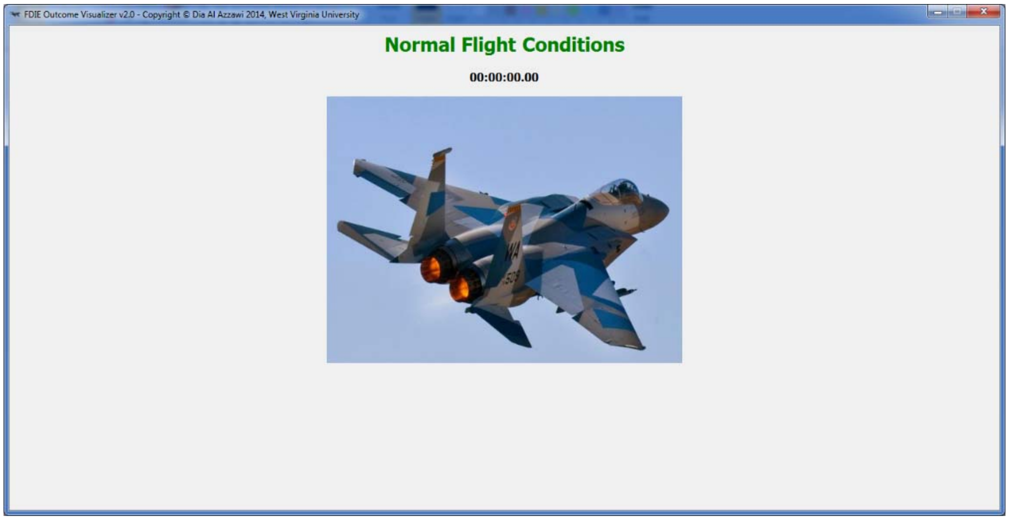
□ Scenario #1
(NLDI + Sigma PI NN) at one flight condition
Nominal conditions ONLY (no failure)

□ Scenario #2
(NLDI + NN) with the full-envelope
Nominal AND failure conditions

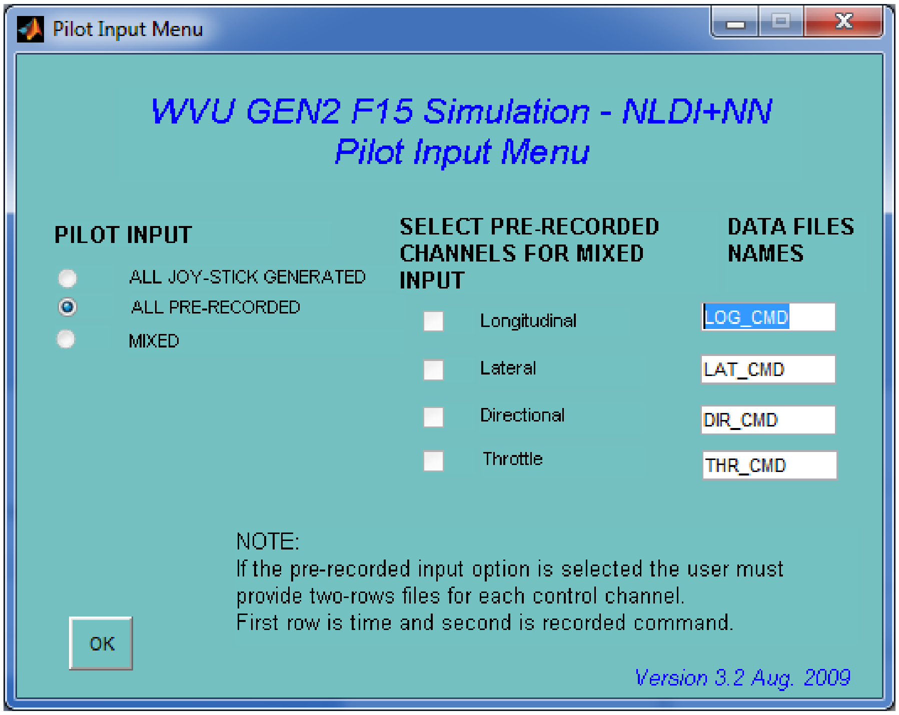
□ Scenario #3
(NLDI + NN) with Actuator Failure
Detection and Identification (AFDI) scheme

☑ Scenario #4
AFDIA (via (AFDI + NLDI + NN)
+ SFDIA (via(SFDI + SFA)

Version 3.2 Aug. 2009


FDIE Outcome Visualizer v2.0 - Copyright © Dia Al Azzawi 2014, West Virginia University
Normal Flight Conditions
00:00:00.00

4. In the `Pilot Input Window`, select `All Pre-Recorded` option in the `Pilot Input` group, and enter `LOG_CMD`, `LAT_CMD`, `DIR_CMD`, and `THR_CMD`, respectively, in the textboxes of the `Data Files Names` group then click the `OK` button.



WVU GEN2 F15 Simulation - NLDI+NN
Pilot Input Menu

PILOT INPUT

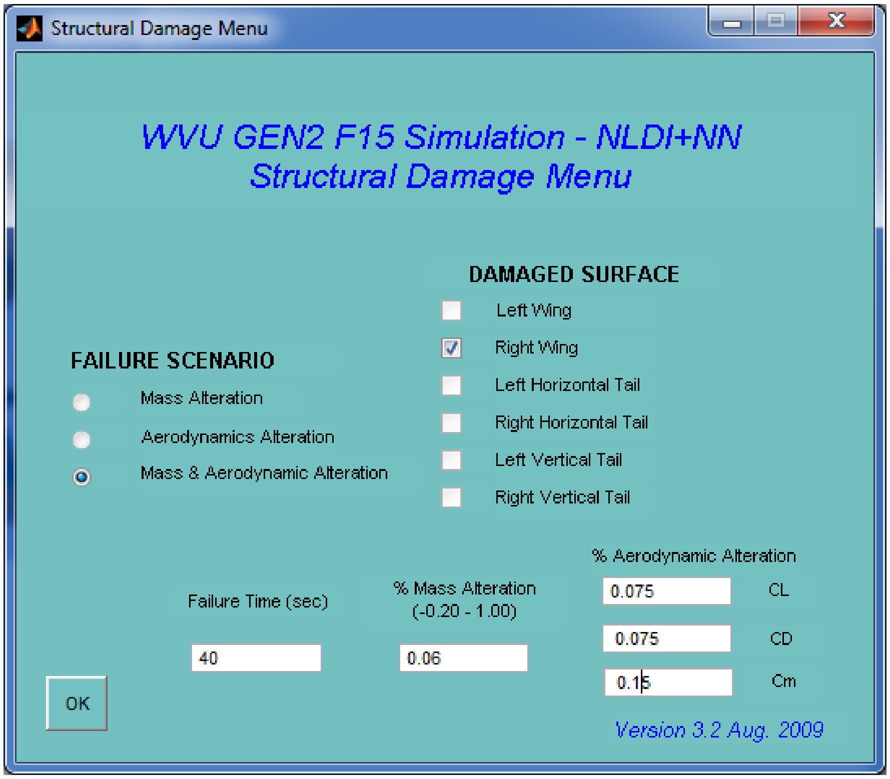
○ ALL JOY-STICK GENERATED
● ALL PRE-RECORDED
○ MIXED

SELECT PRE-RECORDED CHANNELS FOR MIXED INPUT

SELECT PRE-RECORDED CHANNELS FOR MIXED INPUT		DATA FILES NAMES
□	Longitudinal	LOG_CMD
□	Lateral	LAT_CMD
□	Directional	DIR_CMD
□	Throttle	THR_CMD

NOTE:
If the pre-recorded input option is selected the user must provide two-rows files for each control channel.
First row is time and second is recorded command.

5. The next window is specific to the category of the selected flight demo. In this example, the `Structural Damage Menu` window is loaded since the selected demo was a structural damage one. Select the `Mass & Aerodynamic Alteration` option from the `Failure Scenario` group then select the `Right Wing` option from the `Damaged Surface` group and enter 40 in the `Failure Time` textbox, 0.06 (i.e., 6%) in the `Mass Alteration` textbox, 0.075 in the `CL` textbox, 0.075 in the `CD` textbox, and 0.15 in the `Cm` textbox, then click the `OK` button. **NOTE:** Other windows specific to the category of the selected demo will be described later in this chapter.



Structural Damage Menu

WVU GEN2 F15 Simulation - NLDI+NN
Structural Damage Menu

FAILURE SCENARIO

○ Mass Alteration
○ Aerodynamics Alteration
● Mass & Aerodynamic Alteration

DAMAGED SURFACE

□ Left Wing
☑ Right Wing
□ Left Horizontal Tail
□ Right Horizontal Tail
□ Left Vertical Tail
□ Right Vertical Tail

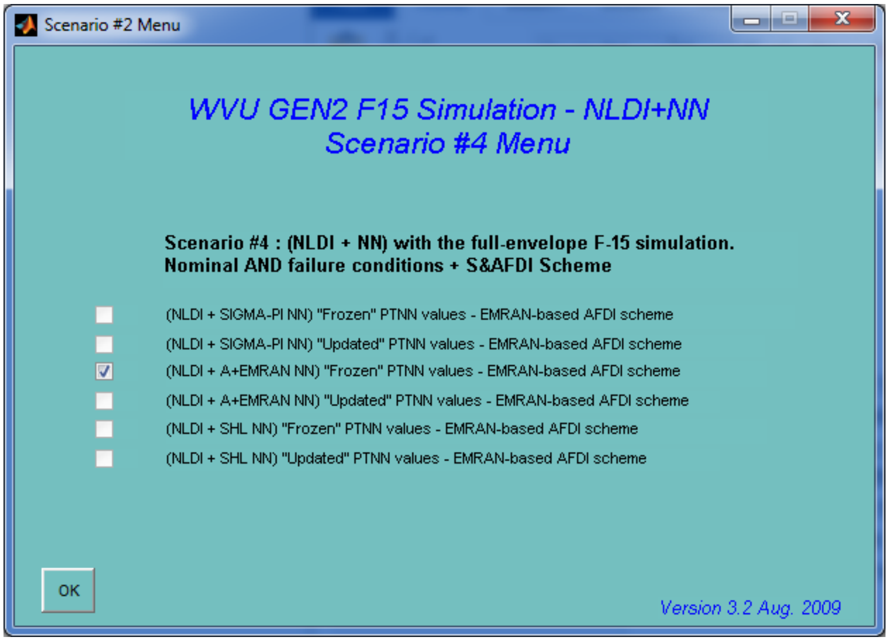
Failure Time (sec): 40

% Mass Alteration (-0.20 - 1.00): 0.06

% Aerodynamic Alteration
CL: 0.075
CD: 0.075
Cm: 0.15

Version 3.2 Aug. 2009

6. In the Scenario #2 window, select the `(NLDI + A+EMRAN NN) “Frozen” PTNN values – EMRAN-based AFDI scheme` then click the `OK` button.



Scenario #2 Menu

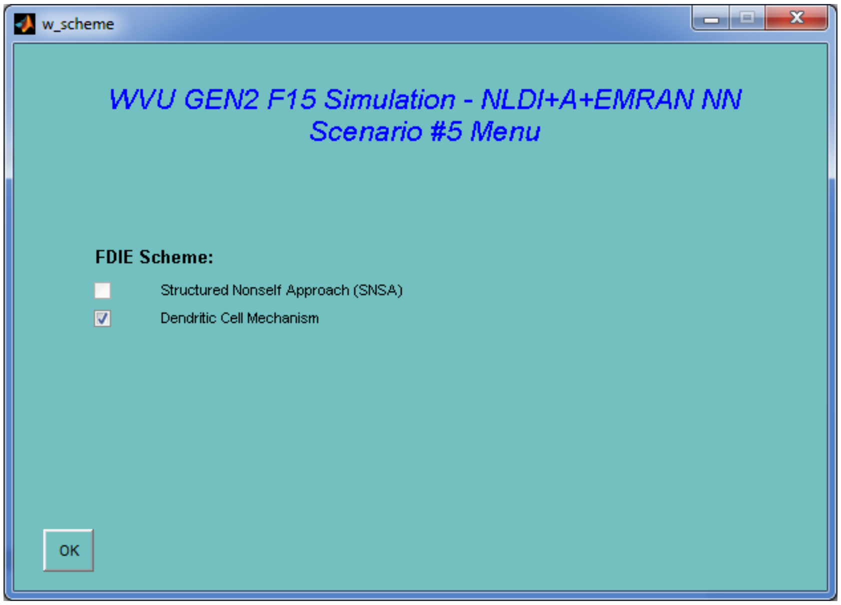
WVU GEN2 F15 Simulation - NLDI+NN
Scenario #4 Menu

Scenario #4 : (NLDI + NN) with the full-envelope F-15 simulation.
Nominal AND failure conditions + S&AFDI Scheme

□ (NLDI + SIGMA-PI NN) "Frozen" PTNN values - EMRAN-based AFDI scheme
□ (NLDI + SIGMA-PI NN) "Updated" PTNN values - EMRAN-based AFDI scheme
☑ (NLDI + A+EMRAN NN) "Frozen" PTNN values - EMRAN-based AFDI scheme
□ (NLDI + A+EMRAN NN) "Updated" PTNN values - EMRAN-based AFDI scheme
□ (NLDI + SHL NN) "Frozen" PTNN values - EMRAN-based AFDI scheme
□ (NLDI + SHL NN) "Updated" PTNN values - EMRAN-based AFDI scheme

Version 3.2 Aug. 2009

7. In the Scenario #2 window, select the `(NLDI + A+EMRAN NN) “Frozen” PTNN values – EMRAN-based AFDI scheme` then click the `OK` button.
8. In the next window, select one the FDIE schemes and click the `OK` button.



w_scheme

WVU GEN2 F15 Simulation - NLDI+A+EMRAN NN
Scenario #5 Menu

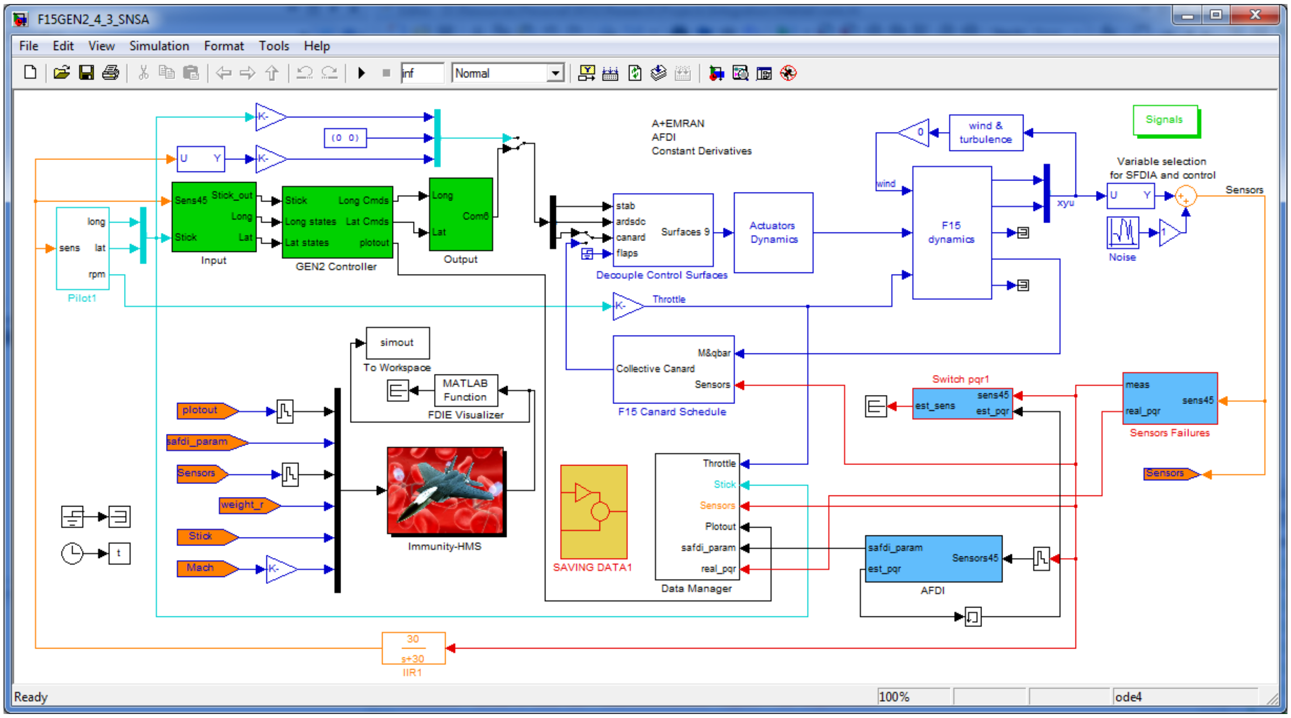
FDIE Scheme:

☐ Structured Nonself Approach (SNSA)

☑ Dendritic Cell Mechanism

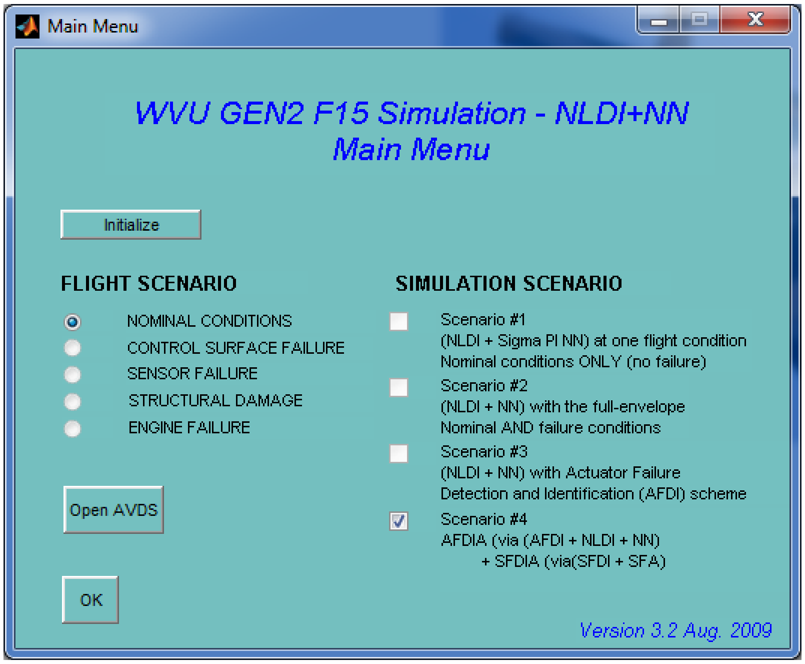
OK

9. Click the `Start Simulation` button on the toolbar of the simulation model to start the demo and monitor the `FDIE Outcome Visualizer` tool as the simulation is running.



Windows and Parameters Specific to Selected Flight Scenarios:

1. **Nominal Flights:** There is no specific window for this group.



WVU GEN2 F15 Simulation - NLDI+NN
Main Menu

FLIGHT SCENARIO

● NOMINAL CONDITIONS
○ CONTROL SURFACE FAILURE
○ SENSOR FAILURE
○ STRUCTURAL DAMAGE
○ ENGINE FAILURE

SIMULATION SCENARIO

□ Scenario #1
(NLDI + Sigma PI NN) at one flight condition
Nominal conditions ONLY (no failure)

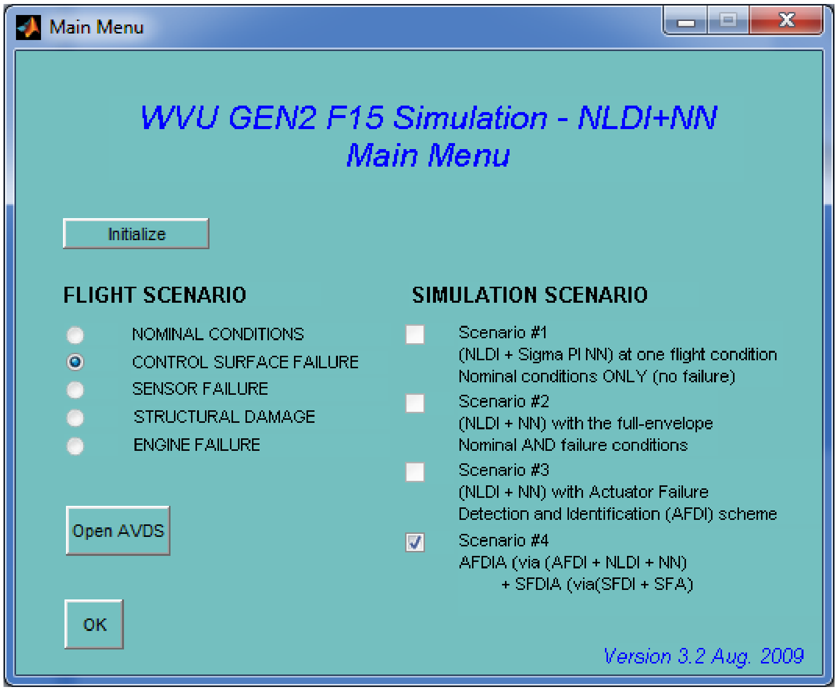
□ Scenario #2
(NLDI + NN) with the full-envelope
Nominal AND failure conditions

□ Scenario #3
(NLDI + NN) with Actuator Failure
Detection and Identification (AFDI) scheme

☑ Scenario #4
AFDIA (via (AFDI + NLDI + NN)
+ SFDIA (via(SFDI + SFA)

Version 3.2 Aug. 2009

2. **Control Surface (Actuator) Failure:**



WVU GEN2 F15 Simulation - NLDI+NN
Main Menu

FLIGHT SCENARIO

○ NOMINAL CONDITIONS
● CONTROL SURFACE FAILURE
○ SENSOR FAILURE
○ STRUCTURAL DAMAGE
○ ENGINE FAILURE

SIMULATION SCENARIO

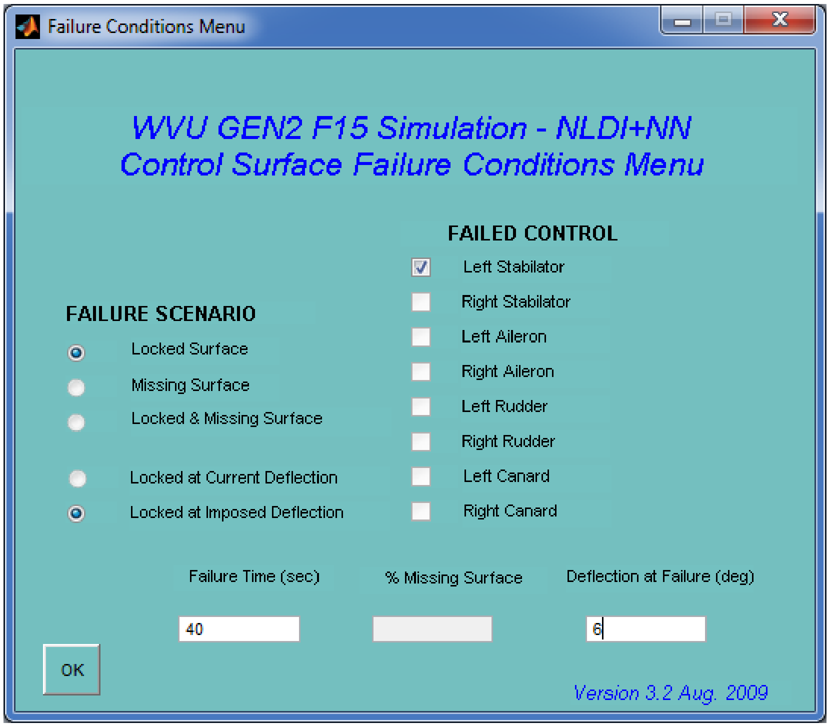
□ Scenario #1
(NLDI + Sigma PI NN) at one flight condition
Nominal conditions ONLY (no failure)

□ Scenario #2
(NLDI + NN) with the full-envelope
Nominal AND failure conditions

□ Scenario #3
(NLDI + NN) with Actuator Failure
Detection and Identification (AFDI) scheme

☑ Scenario #4
AFDIA (via (AFDI + NLDI + NN)
+ SFDIA (via(SFDI + SFA)

Version 3.2 Aug. 2009



WVU GEN2 F15 Simulation - NLDI+NN
Control Surface Failure Conditions Menu

FAILURE SCENARIO

- ● Locked Surface
- ○ Missing Surface
- ○ Locked & Missing Surface
- ○ Locked at Current Deflection
- ● Locked at Imposed Deflection

FAILED CONTROL

- ☑ Left Stabilator
- □ Right Stabilator
- □ Left Aileron
- □ Right Aileron
- □ Left Rudder
- □ Right Rudder
- □ Left Canard
- □ Right Canard

Failure Time (sec): 40

% Missing Surface:

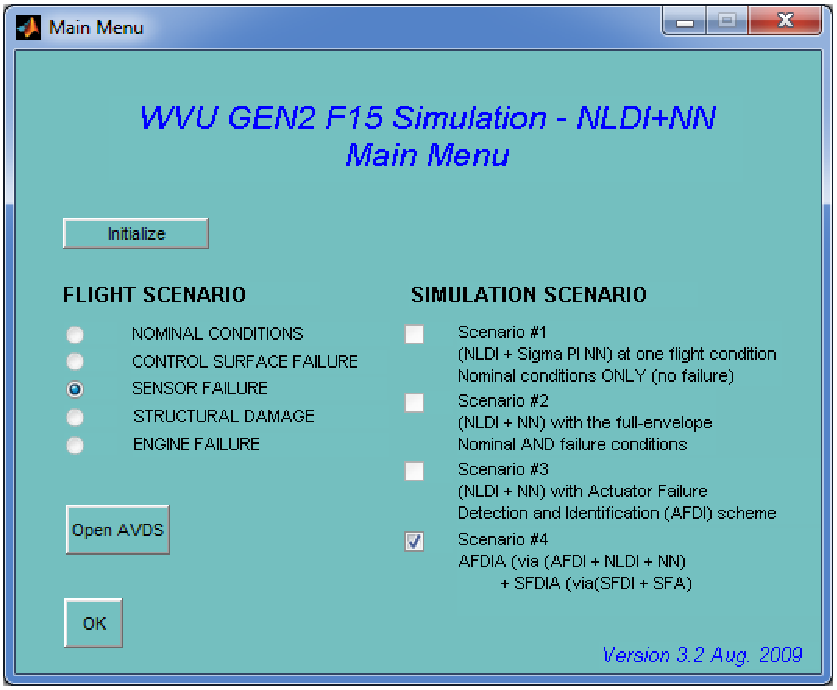
Deflection at Failure (deg): 6

Version 3.2 Aug. 2009

The following table lists the options and values for each of the installed demos that the user must specify:

Demo	Selections/Values
LeftStabilator_1_6	Choose Locked Surface, Locked at Imposed Deflection, and Left Stabilator. Enter 40 in the Failure Time textbox, 6 in the Deflection at Failure textbox.
LeftStabilator_1_8	Choose Locked Surface, Locked at Imposed Deflection, and Left Stabilator. Enter 40 in the Failure Time textbox, 8 in the Deflection at Failure textbox.
RightAileron_1_8	Choose Locked Surface, Locked at Imposed Deflection, and Right Aileron. Enter 40 in the Failure Time textbox, 8 in the Deflection at Failure textbox.

3. **Sensor Failure:**



Main Menu

WVU GEN2 F15 Simulation - NLDI+NN
Main Menu

Initialize

FLIGHT SCENARIO

○ NOMINAL CONDITIONS
○ CONTROL SURFACE FAILURE
● SENSOR FAILURE
○ STRUCTURAL DAMAGE
○ ENGINE FAILURE

Open AVDS

OK

SIMULATION SCENARIO

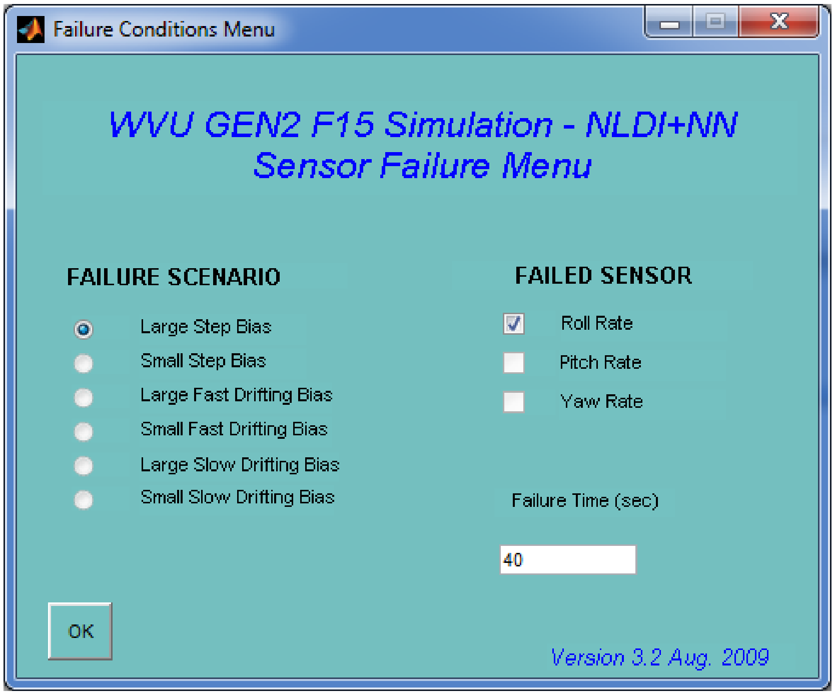
☐ Scenario #1
(NLDI + Sigma PI NN) at one flight condition
Nominal conditions ONLY (no failure)

☐ Scenario #2
(NLDI + NN) with the full-envelope
Nominal AND failure conditions

☐ Scenario #3
(NLDI + NN) with Actuator Failure
Detection and Identification (AFDI) scheme

☑ Scenario #4
AFDIA (via (AFDI + NLDI + NN)
+ SFDIA (via(SFDI + SFA)

Version 3.2 Aug. 2009



Failure Conditions Menu

WVU GEN2 F15 Simulation - NLDI+NN
Sensor Failure Menu

FAILURE SCENARIO

● Large Step Bias
○ Small Step Bias
○ Large Fast Drifting Bias
○ Small Fast Drifting Bias
○ Large Slow Drifting Bias
○ Small Slow Drifting Bias

FAILED SENSOR

☑ Roll Rate
☐ Pitch Rate
☐ Yaw Rate

Failure Time (sec)

40

OK

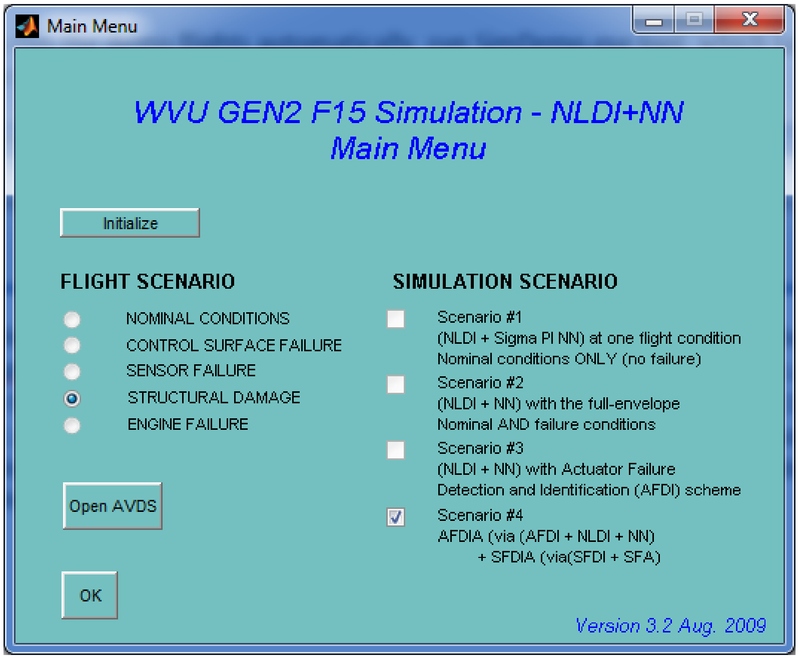
Version 3.2 Aug. 2009

The following table lists the options and values for each of the installed sensor failure demos that the user must specify:

Demo	Selections/Values
Roll_1_LSB10	Choose `Large Step Bias` and `Roll Rate`. Enter 40 in the `Failure Time` textbox.
Roll_LSB_1B	Choose `Large Step Bias` and `Roll Rate`.

	Enter 40 in the `Failure Time` textbox.
Pitch_1_LSB10	Choose `Large Step Bias` and `Pitch Rate`. Enter 40 in the `Failure Time` textbox.
Yaw_1_LSB3	Choose `Large Step Bias` and `Yaw Rate`. Enter 40 in the `Failure Time` textbox.

4. **Structural Damage:**



Main Menu

WVU GEN2 F15 Simulation - NLDI+NN
Main Menu

Initialize

FLIGHT SCENARIO

○ NOMINAL CONDITIONS
○ CONTROL SURFACE FAILURE
○ SENSOR FAILURE
● STRUCTURAL DAMAGE
○ ENGINE FAILURE

SIMULATION SCENARIO

□ Scenario #1
(NLDI + Sigma PI NN) at one flight condition
Nominal conditions ONLY (no failure)

□ Scenario #2
(NLDI + NN) with the full-envelope
Nominal AND failure conditions

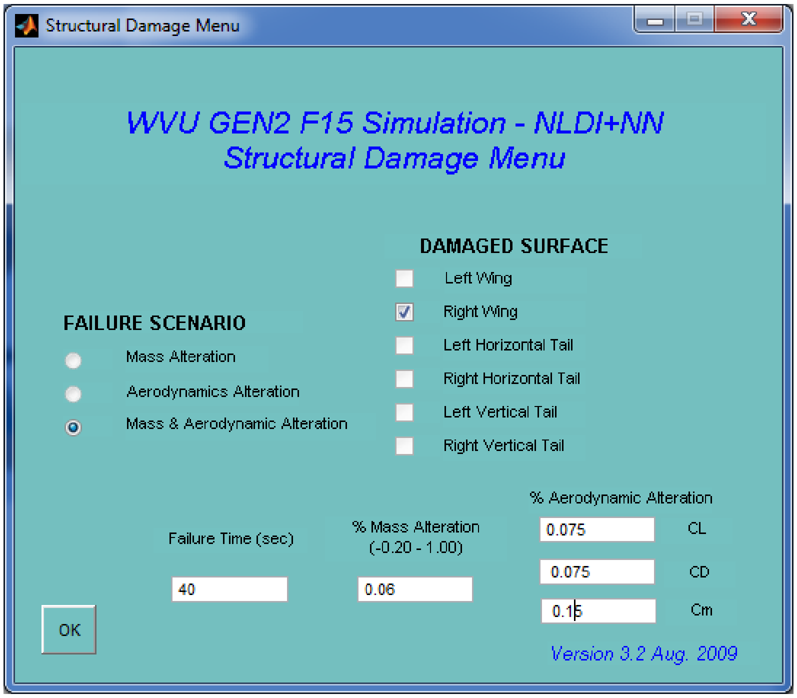
□ Scenario #3
(NLDI + NN) with Actuator Failure
Detection and Identification (AFDI) scheme

☑ Scenario #4
AFDIA (via (AFDI + NLDI + NN)
+ SFDIA (via(SFDI + SFA)

Open AVDS

OK

Version 3.2 Aug. 2009



Structural Damage Menu

WVU GEN2 F15 Simulation - NLDI+NN
Structural Damage Menu

DAMAGED SURFACE

□ Left Wing
☑ Right Wing
□ Left Horizontal Tail
□ Right Horizontal Tail
□ Left Vertical Tail
□ Right Vertical Tail

FAILURE SCENARIO

○ Mass Alteration
○ Aerodynamics Alteration
● Mass & Aerodynamic Alteration

Failure Time (sec): 40

% Mass Alteration (-0.20 - 1.00): 0.06

% Aerodynamic Alteration
CL: 0.075
CD: 0.075
Cm: 0.15

OK

Version 3.2 Aug. 2009

The following table lists the options and values for each of the installed structural damage demos that the user must specify:

Demo	Selections/Values
RightWing_1_6	Choose Mass & Aerodynamic Alteration and Right Wing. Enter 40 in the Failure Time textbox, 0.06 in the Mass Alteration textbox, 0.075 for CL, 0.075 for CD, and 0.15 for Cm.
LeftWing_1_10	Choose Mass & Aerodynamic Alteration and Left Wing. Enter 40 in the Failure Time textbox, 0.1 in the Mass Alteration textbox, 0.1 for CL, 0.1 for CD, and 0.1 for Cm.
LeftWing_1_15	Choose Mass & Aerodynamic Alteration and Left Wing. Enter 40 in the Failure Time textbox, 0.15 in the Mass Alteration textbox, 0.18 for CL, 0.18 for CD, and 0.2 for Cm.

4

Running the Demos Inside MATLAB

Running the demos without the SimDemo tool and directly from MATLAB requires configuring MATLAB only for the first time before running any demo. Follow these steps to configure MATLAB to run the installed simulations smoothly:

1. Run MATLAB and check if it has already been configured to use a compiler for compiling C/C++ files. If not, type `mex –setup` in the MATLAB's `Command Window` and follow the setup a C/C++ compiler.
2. Change the working directory to the "AIS Aircraft FDIE" installation directory
   `[installation root]\WVU FDIE\AIS Aircraft FDIE\Simulations\WVU GEN 2 - DC and Real-Time SNSA\fdm\ann8\source`
3. In the command window, run the command `mex dcsgl2.c` to compile this file. The compiler will generate a binary file named `dcsgl2.mexw32` in the same directory.
4. Repeat Step 3 for to compile the files `emran8.c` and `vrmult.c`.
5. Move all the compiled .mexw32 files to the parent (ann8) directory so that they appear next to their `.dll` files.
6. Redo Steps 2 to 5 for the directory
   `[installation root]\WVU FDIE\AIS Aircraft FDIE\Simulations\WVU GEN 2 - DC and Real-Time SNSA\fdm\smxl\source`
   for the files `VRMULT.C`, `VRPINV.C`, `VRSVD.C`, and `VRTRSP.C`.
   Click the `Set Path`... menu item from the `File` menu.
7. Click the `Add with Subfolders`... button and browse to the folder
   `[installation root]\WVU FDIE\AIS Aircraft FDIE\Simulations\WVU GEN 2 - DC and Real-Time SNSA\fdm`
   and click `OK`.
8. Click `Save` then `Close` to exit the `Set Path` window.

Once MATLAB is configured as described above, you can run the simulations from now on by following these two simple steps:

1. Change the working directory to the folder corresponding to the flight demo you want to run. For example, if you want to run the LeftStabilator_1_6 demo, you should change MATLAB's working directory to
   `[installation root]\WVU FDIE\AIS Aircraft FDIE\Demos\LeftStabilator_1_6`
2. In the command window, type WVU_GEN2 and click enter.
3. Follow the same instructions described previously to interact with simulation interfaces.