WestVirginiaUniversity
THE RESEARCH REPOSITORY @ WVU

Graduate Theses, Dissertations, and Problem Reports

2014

Aircraft Abnormal Conditions Detection, Identification, and Evaluation Using Innate and Adaptive Immune Systems Interaction

Dia Al Azzawi
West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Aircraft Abnormal Conditions Detection, Identification, and Evaluation Using Innate and Adaptive Immune Systems Interaction

Dia Al Azzawi

Dissertation submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Mario Perhinschi, Ph.D., Chair
Larry Banta, Ph.D.
Powsiri Klinkhachorn, Ph.D.
Victor Mucino, Ph.D.
Jacky Prucz, Ph.D.

Department of Mechanical and Aerospace Engineering
Morgantown, West Virginia
2014

Keywords: Aircraft Safety, Intelligent Systems, Artificial Immune Systems,
Failure Detection, Diagnosis, Dendritic Cell Mechanism

ABSTRACT

Aircraft Abnormal Conditions Detection, Identification, and Evaluation Using Innate and Adaptive Immune Systems Interaction

Dia Al Azzawi

Abnormal flight conditions play a major role in aircraft accidents frequently causing loss of control. To ensure aircraft operation safety in all situations, intelligent system monitoring and adaptation must rely on accurately detecting the presence of abnormal conditions as soon as they take place, identifying their root cause(s), estimating their nature and severity, and predicting their impact on the flight envelope.

Due to the complexity and multidimensionality of the aircraft system under abnormal conditions, these requirements are extremely difficult to satisfy using existing analytical and/or statistical approaches. Moreover, current methodologies have addressed only isolated classes of abnormal conditions and a reduced number of aircraft dynamic parameters within a limited region of the flight envelope.

This research effort aims at developing an integrated and comprehensive framework for the aircraft abnormal conditions detection, identification, and evaluation based on the artificial immune systems paradigm, which has the capability to address the complexity and multidimensionality issues related to aircraft systems.

Within the proposed framework, a novel algorithm was developed for the abnormal conditions detection problem and extended to the abnormal conditions identification and evaluation. The algorithm and its extensions were inspired from the functionality of the biological dendritic cells (an important part of the innate immune system) and their interaction with the different components of the adaptive immune system. Immunity-based methodologies for re-assessing the flight envelope at post-failure and predicting the impact of the abnormal conditions on the performance and handling qualities are also proposed and investigated in this study.

The generality of the approach makes it applicable to any system. Data for artificial immune system development were collected from flight tests of a supersonic research aircraft within a motion-based flight simulator. The abnormal conditions considered in this work include locked actuators (stabilator, aileron, rudder, and throttle), structural damage of the wing, horizontal tail, and vertical tail, malfunctioning sensors, and reduced engine effectiveness. The results of applying the proposed approach to this wide range of abnormal conditions show its high capability in detecting the abnormal conditions with zero false alarms and very high detection rates, correctly identifying the failed subsystem and evaluating the type and severity of the failure. The results also reveal that the post-failure flight envelope can be reasonably predicted within this framework.

To my parents, Zeki and Anwar,

my wife, Rawaa,

and my children, Ahmed and Yisra.

It is a great pleasure to take this opportunity to thank my advisor, Dr. Mario G. Perhinschi. Without his true mentoring, indefinite support, and tireless patience, this work would not have been possible.

I would also like to express my sincere thanks to the members of my committee, Dr. Larry Banta, Dr. Powsiri Klinkhachorn, Dr. Victor Mucino, and Dr. Jacky Prucz, for their encouragement and insightful comments I received during the preparation of this dissertation.

Special thanks are reserved to the Department of Mechanical and Aerospace Engineering, particularly Dr. Jacky Prucz and Dr. Victor Mucino, for offering me a teaching assistantship position twice during the early stages of my Ph.D. program. My gratitude must also go to Dr. Larry Banta who enriched my teaching experience.

I gratefully acknowledge the financial support provided by the Tactical Technology Office in the Defense Advanced Research Projects Administration (DARPA) who sponsored this project under contract number HR0011-13-C-0024.

Thanks are also due to Dr. Hever Moncayo at Embry-Riddle Aeronautical University (Daytona Beach, Florida), his graduate students, Israel Moguel and Andres Perez, and Adil Togayev at West Virginia University for the unique collaborative work we have done as a close-knit research team.

Finally, I am truly grateful to my wife for her understanding and support and to my kids for their patience to the little time I spent with them during the four years of my study.

CONTENTS

LIST OF FIGURES

a_x	Longitudinal acceleration [g]	$Ev2$	Direct quantitative evaluation outcome
a_y	Lateral acceleration [g]	$Ev3$	Indirect quantitative evaluation outcome
a_z	Vertical acceleration [g]		
AC_j	Abnormal condition of subsystem j	FA	False alarm rate
b	Wing span	$\mathcal{F}$	Features set
$\mathbb{C}$	Set of dendritic cells	$\mathbf{F}_0$	Nontriggered features matrix
C_D	Drag coefficient	$\mathbf{F}_1$	Triggered features matrix
C_ℓ	Dendritic cell	F_{1M}	Current matrix-pattern matrix
$C_{\ell\delta_a}$	Non-dimensional derivative of the rolling moment coefficient with respect to aileron deflection	F_{1P}	Current projection-pattern vector
		$F_{1\varphi}$	Current feature-pattern vector
		FP	Feature-pattern vector
$C_{\ell\delta_e}$	Non-dimensional derivative of the rolling moment coefficient with respect to elevator deflection	H	Altitude [m]
		h_{VT}	Distance from the aerodynamic center of the vertical tail to the aircraft horizontal plane [m]
C_m	Pitching moment coefficient		
$C_{n\delta_r}$	Non-dimensional derivative of the yawing moment coefficient with respect to rudder deflection	Idt	Identification outcome
		$IL10$	Interleukin-10
		$IL12$	Interleukin-12
$C_{n\delta_T}$	Non-dimensional derivative of the yawing moment coefficient with respect to throttle deflection	k	Subsystem index
		K	Number of cytotoxic T-cells
		$\widetilde{K}$	Number of residual cytotoxic T-cells
C_Y	Lateral force coefficient	$\mathbb{K}$	Set of cytotoxic T-cells
CSM	Costimulatory molecules	$\widetilde{\mathbb{K}}$	Set of residual cytotoxic T-cells
d	Damage factor	ℓ_{AL}	Distance from the aerodynamic center of the left wing to the aircraft plane of symmetry [m]
d_a	Lateral stick displacement [m]		
d_e	Longitudinal stick displacement [m]		
d_r	Pedals displacement [m]	ℓ_{AR}	Distance from the aerodynamic center of the right wing to the aircraft plane of symmetry [m]
d_T	Throttle lever displacement [m]		
D	Drag force [N]		
DR	Detection rate	ℓ_{eL}	Distance from aircraft center of gravity to the axis of the left engine [m]
$\mathbf{D}$	Self/nonself discrimination matrix		
Det	Detection outcome	ℓ_{eR}	Distance from aircraft center of gravity to the axis of the right engine [m]
$DQEE$	Decentralized quadratic estimation error		
		$\mathcal{L}$	Life of a dendritic cell
$Ev1$	Qualitative evaluation outcome	L	Lift force [N]

M	Mach number		s	Wing surface area
$\mathcal{M}$	Migration threshold of a dendritic cell		$\mathbb{S}$	Self
MI	Margin index		$\hat{\mathbb{S}}$	Nonself
M_{WF}	Pitching moment due to wing and fuselage		$\mathcal{S}$	Subself
			t	Current sampling time [s]
M_{HT}	Pitching moment due to horizontal tail		T	Thrust force [N]
$MQEE$	Main quadratic estimation error		$\mathcal{T}$	Size of the moving-time window
MP	Matrix-pattern matrix		$\mathcal{U}$	Features coordinate space
N	Number of features		$\mathbb{V}_E$	Set of flight envelope variables
$\mathbb{N}$	Set of positive natural numbers		$\mathbb{V}_\delta$	Set of directly involved variables
$\mathbb{N}^0$	Set of natural numbers		$\mathbb{V}_\varepsilon$	Set of indirectly involved variables
N_{AC_j}	Number of training patterns with abnormal condition AC_j		V	Aircraft ground velocity [m/s]
			W_{0j}	Nontriggered confidence factor of subself j
N_{fs}	Number of training patterns with failure severity fs		W_{1j}	Triggered confidence factor of subself j
N_F	Number of failures		$[x]$	Nominal range of variable x
$N_{\max}$	Maximum order of failure		$[x]_F$	Post-failure range of variable x
N_{MDC}	Number of migrated dendritic cells		$\mathbf{x}$	Attributes vector
N_{RDC}	Number of regulatory dendritic cells		y	Class variable
N_s	Number of subsystems			
N_{ss}	Number of subselves			
N_{SDC}	Number of stimulatory dendritic cells			
NN_p	Neural network output on roll channel		**Greek**	
NN_q	Neural network output on pitch channel		α	Angle of attack [rad]
NN_r	Neural network output on yaw channel		β	Sideslip angle [rad]
Nft_k	Number of failure types in subsystem k		Γ	Set of feature constraints
			δ_a	Aileron deflection [rad]
$OQEE$	Output quadratic estimation error		δ_e	Stabilator deflection [rad]
p	Roll rate [rad/s]		δ_r	Rudder deflection [rad]
$\dot{p}$	Roll acceleration [rad/s^2]		δ_T	Throttle
p_{ref}	Reference roll rate [rad/s]		ε_a	Aileron alteration factor
P	Feature point		θ	Pitch attitude angle [rad]
PP	Projection-pattern vector		μ	Sample mean
PR	Prediction rate		$\boldsymbol{\mu}$	Sample mean vector
q	Pitch rate [rad/s]		ρ	Air density [kg/m^3]
$\dot{q}$	Pitch acceleration [rad/s^2]		σ	Sample standard deviation
q_{ref}	Reference pitch rate [rad/s]		$\boldsymbol{\Sigma}$	Sample covariance matrix
r	Yaw rate [rad/s]		τ	Sample index in a moving-time window
$\dot{r}$	Yaw acceleration [rad/s^2]			
r_{ref}	Reference yaw rate [rad/s]		ϕ	Roll attitude angle [rad]
R	Number of suppressor T-cells		φ	Feature
REI	Range-exceedance index		ψ	Yaw attitude angle [rad]
$\mathbb{R}$	Set of suppressor T-cells			

Acronyms

AC	Abnormal Conditions
ACDI	Abnormal Conditions Detection and Identification
ACDIE	Abnormal Conditions Detection, Identification, and Evaluation
ACDIEA	Abnormal Conditions Detection, Identification, Evaluation, and Accommodation
ACDQE	Abnormal Conditions Direct Quantitative Evaluation
ACIQE	Abnormal Conditions Indirect Quantitative Evaluation
ACM	Abnormal Conditions Management
ACQE	Abnormal Conditions Qualitative Evaluation
AIS	Artificial Immune System
ANN	Artificial Neural Networks
APC	Antigen-Presenting Cell
CSUM	Clusters Set Union Method
DC	Dendritic Cell
DCA	Dendritic Cell Algorithm
DIV	Directly Involved Variable
EDIV	Equivalent Directly Involved Variable
ERV	Envelope Relevant Variables
FDIE	Failure Detection, Identification, and Evaluation
FDIEA	Failure Detection, Identification, Evaluation, and Accommodation
HMS	Hierarchical Multiself
LSB	Large Step Bias
LFDB	Large Fast Drifting Bias
MHC	Major Histocompatibility Complex
NS	Negative Selection
PS	Positive Selection
RDSUM	Raw Data Set Union Method
SSB	Small Step Bias
SFDB	Small Fast Drifting Bias
WVU	West Virginia University

Introduction

Aircraft safety has been, and will continue to be, an important issue in both the commercial and military aviation industries. Aircraft subsystem failures that result from various sources (such as severe weather conditions, jammed control surfaces, malfunctioning sensors, structural damage during air combat, etc.) may cause catastrophic accidents. Even with the most severe failures, aircraft accident investigations showed that, in many cases, it would have been possible to avoid the accident if the pilot would have taken proper actions at the appropriate time. Although some experienced and highly-skilled pilots can compensate for some failures, they often experience stress and confusion and, therefore, may not take proper actions within few seconds.

Several aircraft safety programs have been launched by research and professional agencies aiming at improving post-failure flight safety. For instance, in the second half of the past decade, NASA commenced the Integrated Resilient Aircraft Control (IRAC) project [1] in its Aviation Safety Program (AvSP) to investigate and develop advanced flight controls that can be implemented to enable safe flight in the presence of adverse conditions such as structural damage, control surface failures, icing, or aerodynamic upsets. The Flight Mechanics Action Group FM-AG(16), part of the Group for Aeronautical Research and Technology in EURope (GARTEUR), has also launched a similar program [2].

Various technologies have been investigated to avoid unrecoverable flight conditions and maintaining control of the aircraft in the presence of failures [3–6]. Fault-tolerant control strategies have been an extensive research topic in failure accommodation [7–9]. However, such strategies often require vital triggering tools that are intelligent enough in gathering the information about the failed subsystem, the nature of the failure, and the severity of the failure as soon as it takes place such that an accommodation strategy knows which of the remaining resources must be used to accommodate the resulting changes in the system. In fact, this information is important to the pilots too since it represents an alarm tool for their continuous situation awareness.

The terms "fault" and "failure" are widely used in the literature to indicate that something is wrong or abnormal in the system. To preserve the general meaning of the abnormal situation or event, the term "abnormal conditions (AC)" will be adopted throughout this dissertation to refer to faults and failures of the 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.

The process of detecting the presence of the AC, identifying the failed subsystem(s), and evaluating the nature, severity, and repercussions of the AC is called *abnormal conditions detection, identification, and evaluation* (ACDIE).

An ideal aircraft ACDIE process must be fast enough in detecting the failure, identifying its origin from other potential anomaly sources, and estimating the severity and nature of the failure with minimum false alarms under normal operating conditions and high performance (detection rate, identification rate, etc.) under abnormal flight conditions. Unreasonable delay in detecting failures when they take place usually lead to undesired consequences such as stall events, loss of control, and severe vibrations. The ideal ACDIE process must also be capable of detecting novel failures and not misclassifying them as one of the known anomalies or as normal operation. It should also be adaptive to system changes, robust to system disturbances and uncertainties, and scalable to dimensionality changes of the system. In addition, an ideal ACDIE should be able to detect and correctly identify multiple failures when they coexist in the system.

Most of the research efforts in the area of aircraft AC detection and identification (ACDI) have considered only individual failures within limited regions of the flight envelope [10–12]. Evaluation of aircraft failures (in particular, flight envelope estimation and protection) have also been conducted in the past several years [13–19]. Although they provided promising solutions to some specific problems, they considered only few system parameters and addressed the evaluation in a limited manner mostly outside the general context, which also includes abnormal conditions detection, identification, and accommodation.

Generally speaking, ACDIE methodologies that are found in the literature can be divided into two main categories: statistical and artificial-intelligence methodologies. Statistical methods are either model-based or non-model-based. Model-based statistical methods require understanding the physics of the system and rely on deriving functional relationships between the inputs and outputs of the system. Diagnostic observers [20, 21], parity relations [22, 23], Kalman filters [20, 24], and parameter estimation [25] are the mostly used model-based statistical methods for the purpose of ACDIE. On the other side, non-model-based methods do not require physical understanding of the system and rely on the availability of large amounts of data. Examples of these methods are trend analysis [26, 27], statistical classifiers [28, 29], and partial least squares [30]. Artificial intelligence methods used for the purpose of ACDIE include digraphs [31], which are based on the cause-and-effect principle, logical fault trees [32], search techniques (such as lookup tables and hypothesis-and-test search) [33], expert systems [34–36], and artificial neural networks (ANN) [34, 37]. However, depending on the technique used, these methods partially satisfy the ACDIE requirements that are of high importance in the aircraft system, as mentioned previously.

Recent research studies [38–41] have acknowledged the need for an integrated and comprehensive solution to the problem of aircraft AC detection, identification, evaluation, and accommodation (ACDIEA) which takes into account the complexity and multidimensionality of aircraft system.

For many years, the biological systems have provided a rich source of inspiration for engineers and scientists in developing new algorithms and computing systems to solve real-life problems that cannot be solved with conventional methods. These inspirations have led to the development of artificial neural networks, evolutionary algorithms, swarm algorithms, and, most recently, artificial immune systems (AISs).

The artificial immune systems, pioneered by Farmer, Packard, and Perelson's work [42] in the mid 80's as a new computational paradigm in artificial intelligence, are "*adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving*" [43]. The immunity-based fault detection operates in a

similar manner as does the biological immune system—according to the principle of self/nonself discrimination—when it detects exogenous antigens while not reacting to the self cells [44, 45].

Plenty of AIS algorithms have been developed in the past two decades and successfully applied to various real-world problems including anomaly detection [46, 47], data mining [48, 49], computer security [50], adaptive control [51, 52], and pattern recognition [53, 54]. In fact, the complexity, adaptability, scalability, and robustness of the AIS make it an appropriate choice for developing adaptive fault-tolerant system [55].

The AIS field is a relatively young field. New AIS algorithms and methods will be developed in the future as new theories and discoveries in immunology appear since many of the observations in this science are not yet fully understood and need solid explanation. For instance, most the existing AIS algorithms have been inspired from the functionality of the components of the adaptive immune system while the innate immune system was totally obsolete until after the emergence of the "Danger Theory" [56], where antigen-presenting cells (APCs), particularly, dendritic cells (DCs), were emphasized to play a vital role in triggering the adaptive immune response. Later, researchers proposed several algorithms based on the danger theory [57–61].

1.1. Research Objectives

The aircraft *abnormal conditions management* (ACM) consists of four major parts: detection, identification, evaluation, and accommodation. Abnormal conditions detection is the process of detecting the failure in the aircraft as soon as it takes place. Abnormal conditions identification is the diagnostic process of isolating the failed subsystem (i.e., actuators, structures, sensors, and propulsion). Abnormal conditions evaluation can be of qualitative or quantitative nature. The qualitative evaluation is the assessment of the failure type. The quantitative evaluation assesses the AC either directly or indirectly. The indirect quantitative evaluation estimates the severity of the AC whereas the indirect quantitative evaluation determines the impact of the AC on the flight envelope of the aircraft. Finally, the abnormal conditions accommodation is the adaptive control of the aircraft system at post-failure.

This research presents an integrated immunity-based framework for aircraft ACDIEA with novel ACDIE algorithms and approaches as indispensable prerequisites for a successful adaptive failure accommodation.

This research aims at developing an integrated and comprehensive framework for aircraft ACM based on the AIS paradigm to tackle the complexity and multidimensionality of the aircraft system. Within this framework, the research aims at designing and implementing reliable and fast schemes for detecting the occurrence of the AC, identifying the aircraft subsystem affected by the AC, and evaluating the type and severity of that AC as indispensable prerequisites for a successful adaptive AC accommodation. Finally, this research effort investigates the capability of AIS paradigm to predict the effect of the AC on the flight envelope.

1.2.   Contributions

The main contributions of this research effort are

- developing an integrated, immunity-based framework for aircraft abnormal conditions detection, identification, evaluation, and accommodation;

- design and implementation of the artificial DC mechanism for AC detection by mimicking the behavior of the biological DCs and using the outcomes of the self/nonself discrimination within the hierarchical multiself strategy;

- extending the artificial DC mechanism to the AC identification, qualitative evaluation, and direct quantitative evaluation by exploiting the information gathered by the DCs about the AC and converting the problem into a pattern recognition one; and

- developing AIS-based methods for post-failure flight envelope prediction under various failure scenarios using analytical approaches with the AIS 2-D projections of the self/nonself.

The work related to the research effort presented in this study has resulted in the following publications and submissions:

16. Perez, Andres, Hever Moncayo, Mario G. Perhinschi, Adil Togayev, and **Dia Al Azzawi**, "Pilot-in-the-Loop Evaluation of a Bio-Inspired Adaptive Fault Tolerant Control System in a Motion Based Flight Simulator," submitted to AIAA Science and Technology Forum, Kissimmee, Florida, Jan. 2015.

15. Perhinschi, Mario G., **Dia Al Azzawi**, Hever Moncayo, Andres Perez, and Adil Togayev, "Immunity-Based Aircraft Actuator Failure Evaluation," *Journal of Aircraft Engineering and Aerospace Technology*, July 2014, submitted, Manuscript ID: AEAT-07-2014-0117.

14. **Al Azzawi, Dia**, Hever Moncayo, Mario G. Perhinschi, Andres Perez, and Adil Togayev, "Comparison of Immunity-Based Schemes for Aircraft Failure Detection and Identification," submitted to *IEEE Computational Intelligence Magazine*, June 2014.

13. Perhinschi, Mario G., **Dia Al Azzawi**, Hever Moncayo, Adil Togayev, and Andres Perez, "Immunity-Based Flight Envelope Prediction at Post-Failure Conditions," submitted to *AIAA Journal of Aerospace Information Systems*, June 2014.

12. Togayev, Adil, Mario G. Perhinschi, **Dia Al Azzawi**, Hever Moncayo, Israel Moguel, and Andres Perez, "Immunity-Based Abnormal Condition Accommodation of Aircraft Subsystem Failures," submitted to ASME 2014 Dynamic Systems & Control Conference, San Antonio, TX, Oct. 2014.

11. Perez, Andres, Hever Moncayo, Israel Moguel, Mario G. Perhinschi, **Dia Al Azzawi**, and Adil Togayev, "Immunity-Based Adaptive Control Laws for Aircraft Fault Tolerance," submitted to ASME 2014 Dynamic Systems & Control Conference, San Antonio, TX, Oct. 2014.

10. Moguel, Israel, Hever Moncayo, Mario G. Perhinschi, Andres Perez, **Dia Al Azzawi**, and Adil Togayev, "Bio-Inspired Approach for Aircraft Health Assessment and Flight Envelope Estimation," submitted to ASME 2014 Dynamic Systems & Control Conference, San Antonio, TX, Oct. 2014.

9. **Al Azzawi, Dia**, Mario G. Perhinschi, Hever Moncayo, Adil Togayev, Andres Perez, and Israel Moguel, "Evaluating Aircraft Abnormal Conditions Using an Artificial Dendritic Cell Mechanism," submitted to the 2nd International Conference and Exhibition on Mechanical & Aerospace Engineering, Philadelphia, PA, Sept. 2014.

8. Moncayo, Hever, Israel Moguel, Mario G. Perhinschi, **Dia Al Azzawi**, Andres Perez, and Adil Togayev, "Biologically-Inspired Approach for Aircraft Management Under Upset Conditions," submitted to the 2nd International Conference and Exhibition on Mechanical & Aerospace Engineering, Philadelphia, PA, Sept. 2014.

7. **Al Azzawi, Dia**, Mario G. Perhinschi, and Hever Moncayo, "Artificial Dendritic Cell Mechanism for Aircraft Immunity-Based Failure Detection and Identification," *AIAA Journal of Aerospace Information Systems*, Vol. 11, No. 7, July 2014, pp. 536-550.

6. Perhinschi, Mario G., Hever Moncayo, **Dia Al Azzawi**, and Israel Moguel, "Generation of Artificial Immune System Antibodies Using Raw Data and Cluster," *IC: International Journal of Immune Computation*, Vol. 2, No. 1, March 2014, pp. 1–15.

5. Perhinschi, Mario G., and **Dia Al Azzawi**, "Undergraduate Experiential Learning Lab for Aircraft Parameter Identification," *Computers in Education (CoED) Journal*, Vol. 5, No. 2, Apr.-Jun. 2014, pp. 79–92.

4. Moguel, Israel, Hever Moncayo, Mario G. Perhinschi, Andres Perez, **Dia Al Azzawi**, and Adil Togayev, "Structured Nonself Approach for Aircraft Failure Identification Within an Immunity-Based Fault Tolerance Architecture," *IEEE Transactions on Aerospace and Electronic Systems*, Jan. 2014, submitted, Manuscript ID: TAES-201400058.

3. Perhinschi, Mario G., Hever Moncayo, and **Dia Al Azzawi**, "Development of Immunity-Based Framework for Aircraft Abnormal Conditions Detection, Identification, Evaluation, and Accommodation," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, Aug. 2013.

2. **Al Azzawi, Dia**, Mario G. Perhinschi, and Hever Moncayo, "Dendritic Cell Mechanism for Aircraft Immunity-Based Failure Detection and Identification," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, Aug. 2013.

1. Perhinschi, Mario G., Hever Moncayo, and **Dia Al Azzawi**, "Integrated Immunity-Based Framework for Aircraft Abnormal Conditions Management," *AIAA Journal of Aircraft*, April 2013, accepted for publication, Manuscript ID: 2013-04-C032381.R1.

Finally, the following papers are under preparation:

1. **Al Azzawi, Dia**, Mario G. Perhinschi, and Hever Moncayo, "Immunity-Based Approach for Flight Envelope Prediction of a Supersonic Aircraft Under Faulty Sensors."

2. **Al Azzawi, Dia**, Mario G. Perhinschi, Hever Moncayo, and Andres Perez, "Flight Envelope Prediction Assessment of an Aircraft Under Various Subsystem Failure."

1.3. Organization of this Dissertation

This dissertation is organized as follows. A concise background on immunology is introduced in Chapter 2 with a survey of research efforts in the area of AIS. The survey is not comprehensive, but it reviews the literature that is directly related to this work. Chapter 3 describes the immunity-based framework for aircraft failure detection, identification, evaluation, and accommodation. Clear definitions of the AIS self and nonself are also presented in Chapter 3. Described in Chapter 4 are the experimental and simulation environment used in this work, the aircraft model, the design of the flight simulation scenarios, and the simulated subsystem failures. The selection of the aircraft AIS features and methods of generating the selves and nonselves are discussed in Chapter 5. The immunity-based schemes and algorithms for abnormal conditions detection, identification, and evaluation proposed in this work are presented in Chapters 6 to 8, respectively. The results and their discussion are presented in Chapter 9. Finally, future research work and recommendations are proposed in Chapter 10.

Literature Review

Artificial intelligence strategies were developed aiming at finding solutions to a broad class of complex problems, which could not be solved by traditional methods, such as linear, nonlinear, multi-objective and dynamic programming, search trees, graph theory etc. The AIS is a relatively new paradigm in the field of artificial intelligence inspired from the biological immune system. Over the past decade, several AIS algorithms have been developed and applied to a wide range of real-world problems including robotics and control, fraud detection, fault detection and identification (diagnosis), scheduling, data mining, anomaly detection, classification, optimization, machine learning, software testing, network security, computer virus and intrusion detection, pattern recognition, and image processing [62].

AIS algorithms were inspired from the functionality of the different components in the biological immune system and its response to the invading pathogens. The biological immune system is divided into two equally important layers: *innate* and *adaptive* [63]. The former is the natural system which we inherit from our ancestors and functions as the first line of defense against an invading organism whereas the latter is the acquired system built through previous exposure to infectious and other foreign 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 looking for antigens (Fig. 2.1). Proteins (or any other markers) of an antigen are recognized by the surface of the phagocyte as the antigen touches the phagocyte. Once recognized as a foreign material, the phagocyte engulfs the pathogen and breaks it up into its constituent molecules. Special proteins in the phagocyte bind to subsets of these molecules (specifically, *peptides*) and transport to the outer surface of the phagocyte along with the bound antigen peptides (Fig. 2.2).
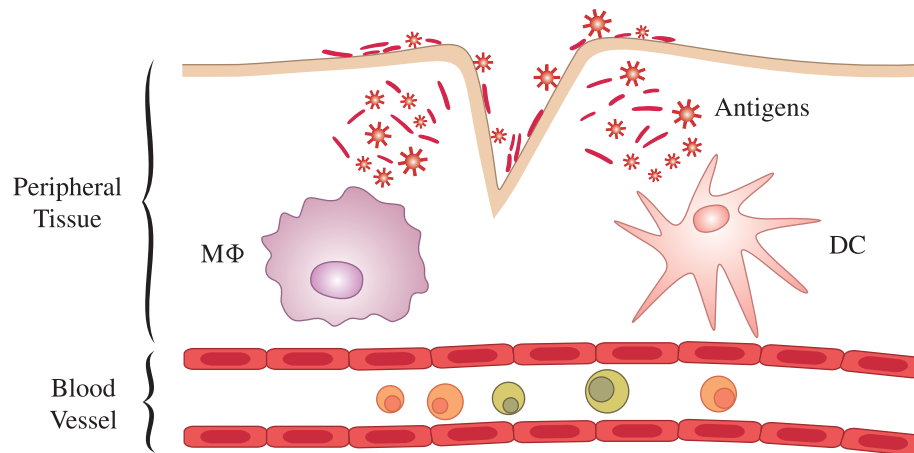
Figure 2.1: Phagocytes in the Peripheral Tissue
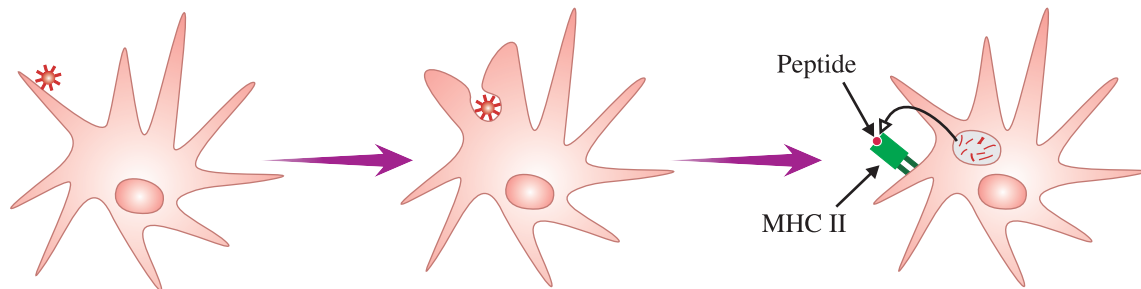
Figure 2.2: A Dendritic Cell Processing an Antigen

These special proteins are called *major histocompatibility complexes* II (MHC II) and 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. It is worth mentioning here that 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.

It is well known in biology that DCs are more efficient APCs than MΦs [64]. 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) [65]. 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* [66, 67].

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 Fig. 2.3, 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 and MΦs, and B-cells to recognize epitopes of exogenous antigens and discriminate self from nonself). 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 that kill the infected cells. 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 called *cell-mediated response*.
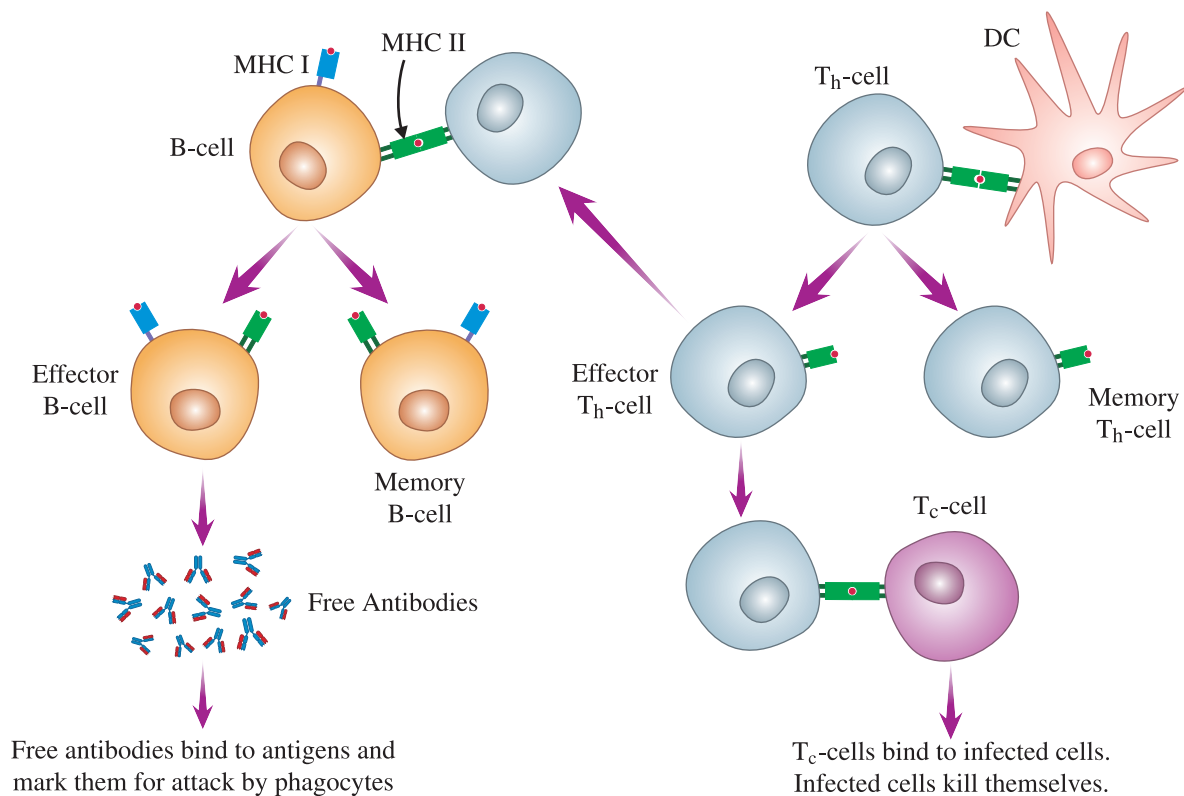
Figure 2.3: Activation of T_h-, B-, and T_c-Cells by the APC

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*.

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 nonself) 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 nonself. Therefore, they are only compatible with the nonself and will be capable to detect its presence. The concept is illustrated in Fig. 2.4, where the term "antibodies" is used in a most generic way.
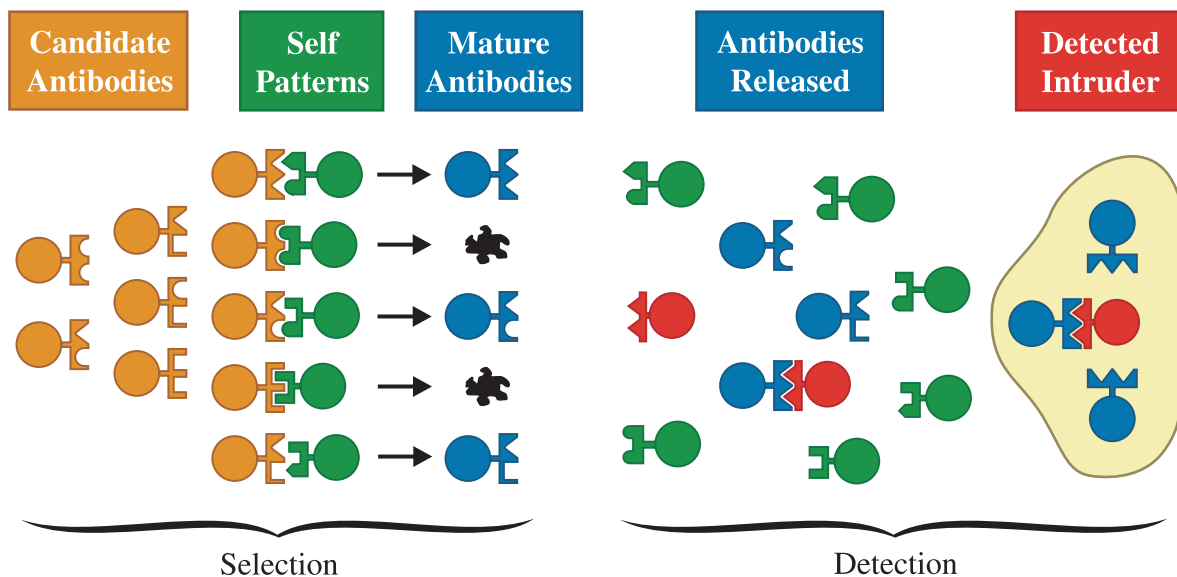
Figure 2.4: The Negative Selection Process

Segel and Cohen [68] examined how to use ideas from the biological immune system to solve engineering problems and, inversely, how the AIS can inspire new conjectures to unrecognized methods by which the immune system is organized.

Ko et al. proposed a general suppression framework based on the AIS discrimination theory [69]. They designed a distributed control system based on their proposed framework to control a modular self-reconfigurable hyper-redundant planar robotic manipulator.

An artificial immune regulation (AIR) scheme was proposed and integrated into an immune model-based fault detection approach for fault diagnosis [70]. This system generated residuals that contained information about the faults. However, various disturbances and errors caused residuals to become nonzero, thus interfering with detection of faults. The AIR scheme produced a set of memory B-cells whose amount depended on several chemical rate constants. This helped in classifying the residuals into distinct patterns, denoting different faulty situations.

Guzella et al. presented an immune-inspired approach for fault detection called dynamic effector regulatory algorithm (DERA) [71]. The proposed approach integrates the role of regulatory T-cells in control and signaling between cells. In DERA, new components of the immune system such as cytokines and regulatory cells are incorporated in the model. DERA uses a population of regulatory and effector cells and combines both positive and negative detection; it also keeps track of the concentration of two cytokines in the environment. It is based on the assumption that there must be an interaction between cells in the population before determining whether an antigen belongs to self or nonself. The system possesses a memory that is represented by cytokine concentrations such that the classification of an antigen depends on the responses against recently classified instances. The system does not include clonal selection, thus the memory is not antigen specific. Unlike NS algorithms, which look for a total coverage of the nonself space, DERA searches for an appropriate distribution of effector and regulatory cells throughout the space. By combining both regulatory and effector cells, to recognize normal and abnormal operation, respectively, DERA's dynamic behavior mediated by cytokines is able to indicate the severity of a fault. The proposed approach was tested on the DAMADICS fault-detection benchmark problem, and it was able to attain considerably lower false-positives than other approaches, because regulatory cells suppress the activation of effector cells.

An immunized computational system (ICS) was proposed by KrishnaKumar and other researchers [72–75]. They successfully tested ICS on an autonomous aircraft control problem.

Ootsuki and Sekiguchi suggested a method for determining control sequences of a sequential control plant based on the immune system [76]. They used Petri nets with control sequences equal to the firing sequences of a Petri net.

An aircraft fault detection system, called multilevel immune learning detection (MILD), was developed to detect a broad spectrum of known as well as unforeseen faults [44]. Empirical study was conducted with datasets collected through simulated failure conditions using National Aeronautics and Space Administration (NASA) Ames C-17 flight simulator. Three sets of in-flight sensory information—namely, body-axes roll rate, pitch rate, and yaw rate were considered to detect five different simulated faults: one for engine, two for the tails, and two for the wings. The MILD implemented a real-valued negative selection (RNS) algorithm, where a small number of specialized detectors (as signatures of known failure conditions) and a set of generalized detectors (for unknown or possible faults) are generated. Once the fault is detected and identified, an adaptive control system would use this detection information to stabilize the aircraft by utilizing available resources (control surfaces). Experiments were performed with datasets collected under normal and various simulated failure conditions using a piloted motion-based NASA simulation facility.

Kaneshige and Krishnakumar demonstrated the potential of using immunized maneuver selection in air combat maneuvers of a UAV [77]. They used a combination of genetic and evolutionary algorithms in emulating the adaptive capabilities of the biological immune system to construct the maneuvers that are necessary for responding to different air combat situations.

Significant research efforts at West Virginia University (WVU) have been focused on AIS-based aircraft ACM. Perhinschi et al. proposed an integrated immunity-based framework for the detection, identification, and evaluation of a wide variety of failures of aircraft subsystems [78]. Moncayo et al. developed an immunity-based aircraft failure detection and identification scheme [79, 80]. They proposed a hierarchical multiself (HMS) strategy where different self configurations are selected for detection and identification of specific abnormal conditions. They have also proposed failure evaluation over extended flight envelope based on the AIS paradigm [81]. The potential of the artificial immune system to provide adaptive control of a UAV has been recently investigated by augmenting an immunity-based mechanism to the nonlinear dynamic inversion (NLDI) of the UAV in an attempt to provide adaptive control laws [82]. An evolutionary algorithm has been developed for the generation and optimization of artificial immune system-based failure detectors using the negative selection strategy [83, 84].

Based on Matzinger's "danger theory" [56] in immunology, the dendritic cell algorithm (DCA) has been recently developed [58], mimicking the functionality of the dendritic cells in the tissue of the biological immune system. Since then, variants of the original algorithm were suggested [59, 61]. These algorithms were applied to error detection in a mobile robotic system [60], robotic classifier [85], anomaly detection in computer security [57], GPS multipath detection [86], pattern recognition [87], and "healthy" model-based fault detection of analog circuits [88].

The danger theory has also inspired Pinto and other researchers to propose a fault detection algorithm in telephone systems [89]. Each call in this fault-detection system is represented by an antigen composed of linear attributes: origin, destination, duration of calls, and a nominal attribute. Two signal levels were identified: signal 1 for perceiving the presence of the antigen and signal 2 for costimulation by using the noncompleted call rate. Signal 2 was responsible for alarming a danger situation. Detector death, detector deactivation, detector population renewal, and a voting routine were significantly employed in this work.

Immunity-Based Framework for Aircraft ACDIEA

This chapter presents an immunity-based framework for the aircraft ACDIEA problem. The chapter starts with clear definitions of the various terms and components used in designing the framework, then introduces the general architecture of the aircraft ACM process. It should be noted that the targeted system in this work is the *aircraft*, including its subsystems (such as the actuators, sensors, propulsion, etc.), the human pilot, and the environment. However, the framework is generally formulated and can be applied to any other system.

3.1. Definitions and Basic Concepts

Definition 3.1: *Abnormal Conditions Detection.*
  The AC detection is the recognition of the presence of an abnormal condition in at least one of the aircraft subsystems. The detection outcome, *Det*, can be expressed as

$$Det = \begin{cases} 1 & \text{for AC,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Definition 3.2: *Abnormal Conditions Identification.*
  The AC identification is the diagnostic process which pinpoints to the root cause of the AC once they are detected. In other words, the AC identification isolates the failed subsystem(s) after the declaration of an AC. The identification outcome, *Idt*, can be expressed as

$$Idt = \begin{bmatrix} id_1 & id_2 & \cdots & id_{N_s} \end{bmatrix}, \quad id_j = \begin{cases} 1 & \text{if subsystem } j \text{ is under AC,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

where N_s is the number of subsystems.
  Alternatively, the identification outcome can be formulated as a set of integers labeling the failed subsystem:

$$Idt = \begin{bmatrix} id_1 & id_2 & \cdots & id_{N_{sf}} \end{bmatrix}, \tag{3.3}$$

where N_{sf} is the number of failed subsystems and id_j represents the label associated to each subsystem. Typically,

$$id_j \in \{id_1, id_2, \ldots, id_{N_{sf}}\}. \tag{3.4}$$

Depending on the complexity of the targeted system, the AC identification may be performed in several phases. For example, in the case of elevator's failure of an aircraft, the first identification phase isolates the actuators as the failed subsystem whereas the second phase determines the elevator as the failed actuator and, finally, the third phase pinpoints to whether the failed elevator is the left or right one.

Definition 3.3:   *Abnormal Conditions Qualitative Evaluation.*
The AC qualitative evaluation (ACQE) is the determination of the failure type. The outcome of the qualitative evaluation, $Ev1$, is an integer labeling the type of failure out of a list of Nft_k failure types associated to each subsystem $k = 1, 2, \ldots, N_s$, such that

$$Ev1 \in \{1, 2, \ldots, Nft_k\} . \tag{3.5}$$

Note that the set of all AC for all subsystems is

$$F = \{f_i \mid i = 1, 2, \ldots, N_F\} , \tag{3.6}$$

where N_F is the total number of AC given by

$$N_F = \sum_{k=1}^{N_s} Nft_k. \tag{3.7}$$

As an example of the ACQE, consider two different types of the elevator's failure: locked at a trim position (type 1) and moving and locked at non-trim position (type 2). If the second type is the actual failure of the elevator, then the outcome of the qualitative evaluation must be $Ev1 = 2$.

Definition 3.4:   *Abnormal Conditions Direct Quantitative Evaluation.*
The AC direct quantitative evaluation (ACDQE) is the determination of the failure magnitude (or severity). The outcome of the direct quantitative evaluation, $Ev2$, can take a numerical value representing the magnitude of the AC. For example, an outcome of

$$Ev2 = 20\% \tag{3.8}$$

in the case of wing failure represents the percentage area of the wing affected by structural damage. Generally speaking, the numerical representation is difficult to achieve, therefore, the ACDQE may, alternatively, provide a severity estimate mapped on a discrete set. For example,

$$Ev2 \in \{\text{low severity, medium severity, high severity}\} , \tag{3.9}$$

where each severity scale maps to a range of actual failure magnitudes. Another alternative is to formulate the ACDQE outcome as a fuzzy set expressed as

$$Ev2 = \{y_i \mid i = 1, 2, \ldots, m\} , \tag{3.10}$$

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

$$Ev2 = \mathcal{D}\{y_i \mid i = 1, 2, \ldots, m\}, \tag{3.11}$$

where $\mathcal{D}$ is the defuzzification operator.

Definition 3.5: *Abnormal Conditions Indirect Quantitative Evaluation.*
The AC indirect quantitative evaluation (ACIQE) is 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 AC. The outcome of the indirect quantitative evaluation, *Ev3*, typically represents the set of new ranges of the flight envelope variables v_i at post-failure conditions, i.e.,

$$Ev3 = \left\{ \left[v_{i\min}, \ v_{i\max} \right] \mid i = 1, 2, \ldots, N_E \right\}, \tag{3.12}$$

where N_E is the number of the flight envelope variables.

Definition 3.6: *Abnormal Conditions Accommodation.*
The AC accommodation is the adaptation of the system to the AC to keep it under control and complete the mission. It can be passive or active. *Passive accommodation* reprocesses the ACDIE outcomes to deliver warnings and information displayed in the cockpit in a form that is easy for the pilot to realize. *Active accommodation* involves triggering of direct compensating modules as integral part of the control laws and/or actual computation of commands at post-failure conditions.

Definition 3.7: *Features.*
The features are the variables of the system that are expected to capture the fingerprints of all AC considered, in terms of their occurrence, nature, and severity. They can be (sub)system states, inputs, control system variables, estimated values, etc. The set of all features

$$\mathcal{F} = \{\varphi_i \mid i = 1, 2, \ldots, N\} \tag{3.13}$$

defines an *N*-dimensional hyper-space referred to as the *universe*, $\mathcal{U}$. An orthogonal coordinate system, equally denoted $\mathcal{U}$, with its origin at $O = \mathbf{0}$ is associated to this hyper-space. A *feature point*, *P*, is the set of simultaneous values of all features φ_i that can be obtained through measurements or simulation, at normal or abnormal conditions. The position vector of *P* relative to *O* is

$$\left[\mathbf{r}^{OP} \right]_{\mathcal{U}} = \left[\begin{array}{cccc} \varphi_{1P} & \varphi_{2P} & \cdots & \varphi_{NP} \end{array} \right]_{\mathcal{U}}^{T}. \tag{3.14}$$

Definition 3.8: *Self and Nonself.*
The self, $\mathbb{S}$, is defined as the hyper-subspace of all possible feature points at normal conditions. All other points in $\mathcal{U}$ form the nonself, $\hat{\mathbb{S}}$. Therefore,

$$\mathbb{S} \cup \hat{\mathbb{S}} = \mathcal{U} \qquad \text{and} \qquad \mathbb{S} \cap \hat{\mathbb{S}} = \varnothing. \tag{3.15}$$

The threshold between self and nonself (i.e., between normal and abnormal conditions) is represented by an *N*-dimensional surface

$$\mathcal{S}\left(\varphi_1, \varphi_2, \ldots, \varphi_N\right) = 0. \tag{3.16}$$

Definition 3.9: *Subselves.*

Subselves are lower-dimensional projections of the self. Consider a 2-dimensional projection (or subself) defined by features φ_1 and φ_2. For clarification, and without loss of generality, let the projection be represented by straight line segments as shown in Fig. 3.1. Note that feature values are typically normalized and the resulting hyper-cube represents a working sub-region in $\mathcal{U}$. The set of all possible feature points collected at normal conditions defines Area 0. Any point outside Area 0 belongs to the nonself $\hat{\mathbb{S}}$ and a failure may be declared. However, a feature point that falls inside Area 0 does not necessarily signify normal conditions. AC for which the dynamic fingerprints depend on one or more other features may produce projections inside Area 0.
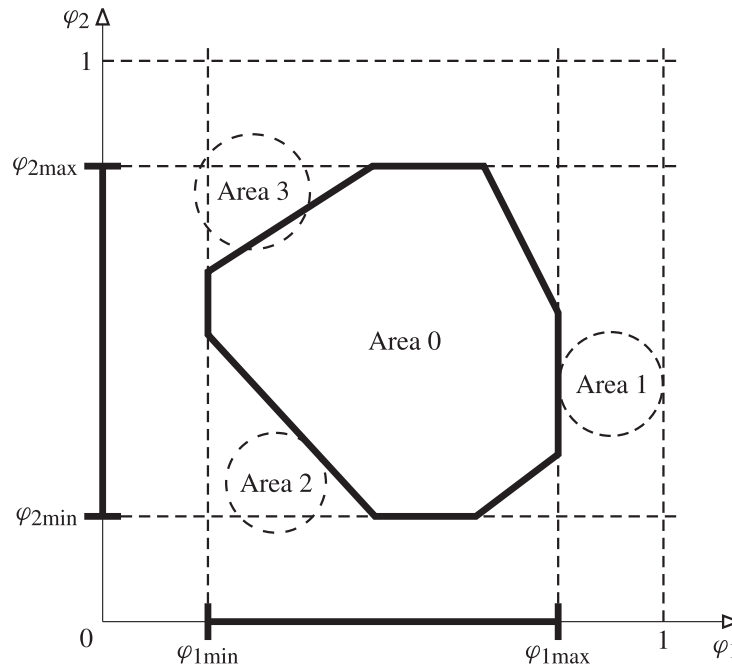
Figure 3.1: Self/Nonself 2-Dimensional Projection

A *complete detection* is said to be possible if *all* feature points under AC fall inside the nonself. The *order* of an AC is defined as the minimum number of features necessary to completely detect that AC. For example, a first order failure would require only one feature, φ^*, for detection. Referring to Fig. 3.1, if Area 1 is the projection of all points under a certain AC, then that AC is of first order with $\varphi^* = \varphi_1$. Similarly, Area 2 represents a second order AC and Area 3 represents a third or higher order one. Furthermore, a first order AC requires *at least* one feature for detection. For instance, the AC represented by Area 1 in Fig. 3.1 can be completely detected using the 2-dimensional self defined by φ_1 and φ_2, as well as only the 1-dimensional subself defined by φ_1, whereas the 2-dimensional self is *sufficient* to completely detect the AC represented by Area 2. Using the subself defined by φ_1 will never succeed in

detecting the AC represented by Area 2. On the other hand, using the subself defined by φ_2 allows 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 may provide very high detection rates for such cases too, if the feature points that are projected inside the self can be reached with very low probability.

Definition 3.10: *Directly Involved Variable.*
A directly involved variable (DIV) in the AC is that variable whose abnormal variation is directly and significantly the result of the AC. In other words, a DIV is a variable that captures the dynamic fingerprint of the AC. It may belong to the features set $\mathscr{F}$ that define the self and nonself. For example, in the case of the left aileron being locked, the DIV can be defined as the left aileron deflection, δ_{aL}, which obviously characterizes the aileron failure. However, this variable might not belong to the features set defining the self/nonself.

Definition 3.11: *Equivalent Directly Involved Variable.*
An equivalent directly involved variable (EDIV) is a variable that belongs to the features set $\mathscr{F}$ and reflects the abnormal variation of the DIV, which does not belong to the features set. Back to the locked left aileron example, the EDIV which captures the abnormal variation of δ_{aL} could be the lateral stick displacement, d_a, which belongs to the features set defining the self/nonself. The mapping $d_a = f(\delta_{aL})$ defines the relationship between δ_{aL} and d_a. The importance of defining EDIVs in a particular AC is to enable the use of certain 2-dimensional projections for predicting the post-failure flight envelope, as will be discussed Sec. 8.3.

3.2. Architecture of the AIS-Based Aircraft ACM

The AIS-based aircraft ACM includes three main components functionally connected in a closed loop as shown in Fig. 3.2:

- offline ACM system design and implementation,
- online ACDIEA, and
- post-processing of flight data and ACDIEA outcomes.
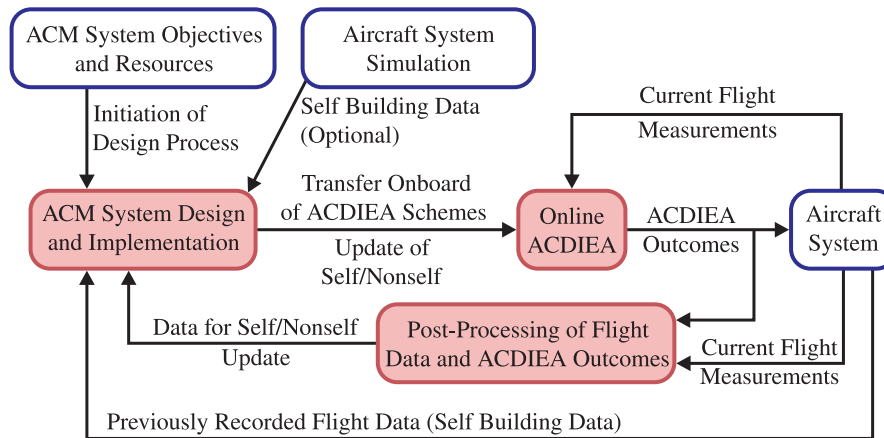
Figure 3.2: The AIS-Based Aircraft ACM

The outcome of the *offline ACM system design and implementation* is the development of an integrated and comprehensive ACDIEA scheme. The design depends on a clear definition of the aircraft subsystems that are targeted, the types of the AC (including known and unknown failures), the severity scales of the AC, the flight envelope variables, and the nature and level of the 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/nonself generation. The block diagram of the ACM design component is presented in Fig. 3.3.
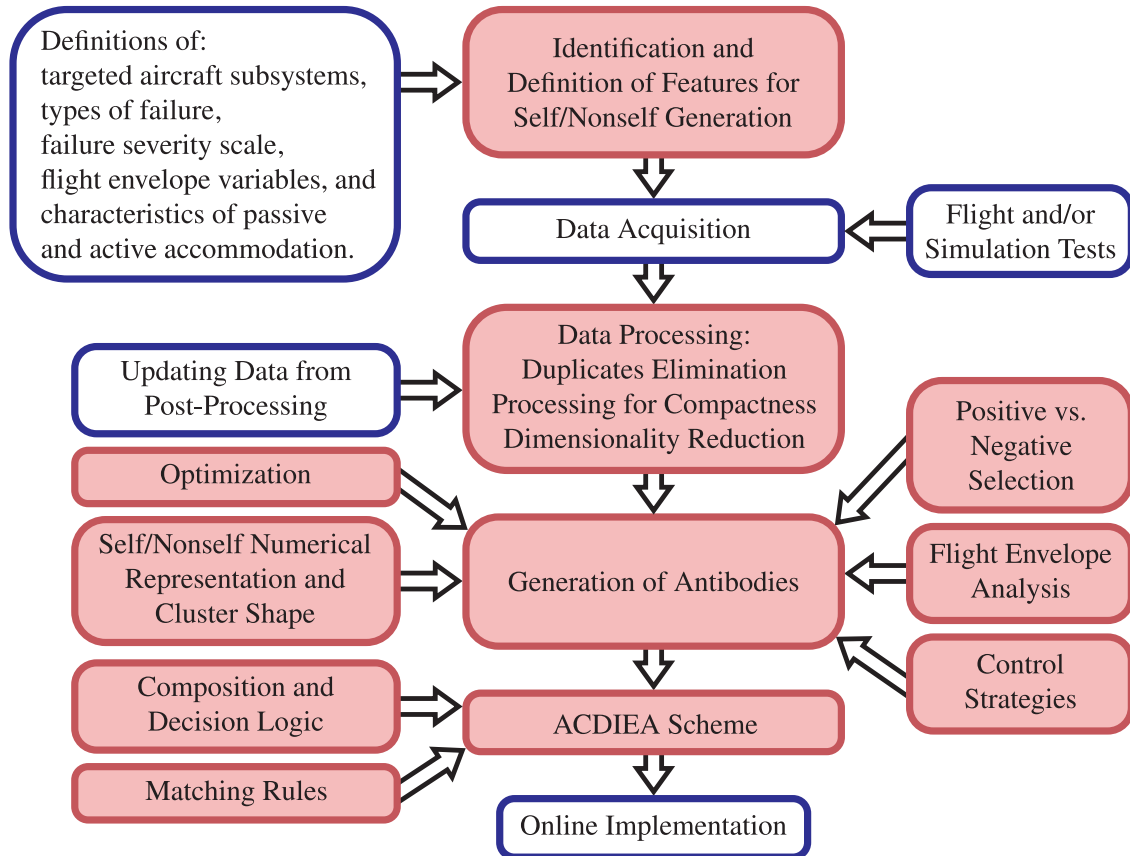
Figure 3.3: Offline ACM System Design and Implementation

One of the most critical steps in the design of the ACM system is the selection of features. The features selected for generating the self/nonself must be relevant to all four components of the ACDIEA. Their number and nature depend on the aircraft subsystems that are targeted, the types of the AC, the severity scales of the AC, the flight envelope variables, and the nature and level of the 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-th order AC requires N relevant features and, therefore, an N-dimensional self. If the number of subsystems considered is N_s, then the states X_i, inputs U_i, and other relevant parameters P_i corresponding to subsystem i, can be expressed as

$$X_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iN_{si}} \end{bmatrix}, \tag{3.17}$$

$$U_i = \begin{bmatrix} u_{i1} & u_{i2} & \cdots & u_{iN_{ui}} \end{bmatrix}, \tag{3.18}$$

$$P_i = \begin{bmatrix} p_{i1} & p_{i2} & \cdots & p_{iN_{pi}} \end{bmatrix}, \tag{3.19}$$

respectively. For any subsystem i, the maximum order an AC can have is

$$N_i = N_{si} + N_{ui} + N_{pi}. \tag{3.20}$$

Therefore, for complete detection of all AC from an exhaustive set of AC, the number of necessary features to build the self/nonself is

$$N = \sum_{i=1}^{N_s} \left(N_{si}^* + N_{ui}^* + N_{pi}^* \right), \tag{3.21}$$

where the *-variables exclude duplication among the subsystems. If all states, inputs, and other parameters are distinct, then

$$N_{si}^* = N_{si}, \qquad N_{ui}^* = N_{ui}, \qquad N_{pi}^* = N_{pi}. \tag{3.22}$$

Otherwise, the *-variables can be given by, for example,

$$N_{zi}^* = \begin{cases} N_{z1} & \text{for } i = 1, \\ N_{zi} - n & \text{for } i > 1, \end{cases} \tag{3.23}$$

where $z = s$, u, or p and n is the number of duplicate states for each subsystem, i.e., 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/nonself can become excessively large and create significant computational problems [78]. These problems can be avoided under certain conditions by using lower-dimensional subselves within a HMS strategy [79].

Let the maximum order of failures in the set of AC, F, be

$$N_{\max} = \max_j \left[\text{ord} \left(f_j \right) \right], \qquad j = 1, 2, \ldots, N_F, \tag{3.24}$$

and

$$\text{ord} \left(f_m \right) = N_{\max}. \tag{3.25}$$

If $N_{\max} = N$, then a complete detection of failure f_m can only be obtained, in general, with an N-dimensional self. In some cases, not all of the lower-dimensional projections can capture the fingerprint of the AC. For example, consider the hypothetical 3-D self shown in Fig. 3.4. An AC feature point AC_1 is projected outside all the three 2-D projections. Feature point AC_2 is projected outside one of the $(N–1)$-dimensional subselves. For specific shapes of the self/nonself, it is possible that some AC feature points are projected inside all subselves as illustrated by feature point AC_3.
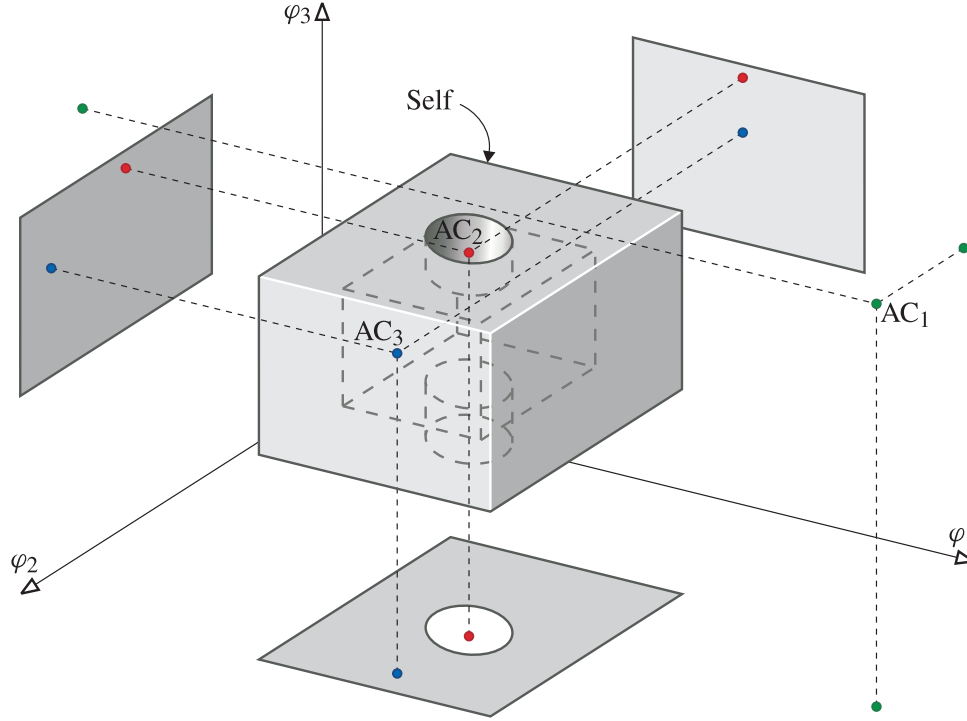
Figure 3.4: Potential of Subselves to Capture the Abnormal Conditions

If $N_{\max} < N$, then a complete detection of f_m can be obtained by using an $N_{\max}$-dimensional subself with proper features. In this case, 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 subselves. The approach is conservative and some of the subselves may be ignored or replaced by lower-dimensional subselves without loss of performance.

The order of an AC is often difficult (or even impossible) to determine; therefore, $N_{\max}$ may be unknown. In this situation, a low value for $N_{\max}$ may be assumed and the self/nonself can be generated as the set of all possible $N_{\max}$-dimensional subselves, 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 subselves for N features is given by

$$N_{ss} = C_N^{N_{\max}} = \frac{N!}{N_{\max}!\left(N - N_{\max}\right)!}. \tag{3.26}$$

Feature points that are collected under normal conditions and have common predefined properties are usually grouped to form *self clusters*. These clusters can be geometrically shaped as hyper-rectangles, hyper-spheres, or hyper-ellipsoids. Feature points that do not fall on any of the self clusters are also clustered with the same shape to form the antibodies. These antibodies may be directly used in the AC detection, where they are referred to as *detectors*. The *structured nonself approach* [90] can be used to restructure subsets of the antibodies by adding information to them such that they can be used in the AC identification, evaluation, and accommodation. In this case, the antibodies are referred to as *identifiers*, *evaluators*, and *compensators*, respectively.

The *online ACDIEA* process implies the real time operation of the ACDIEA schemes. 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 online ACDIEA component is presented in Fig. 3.5.
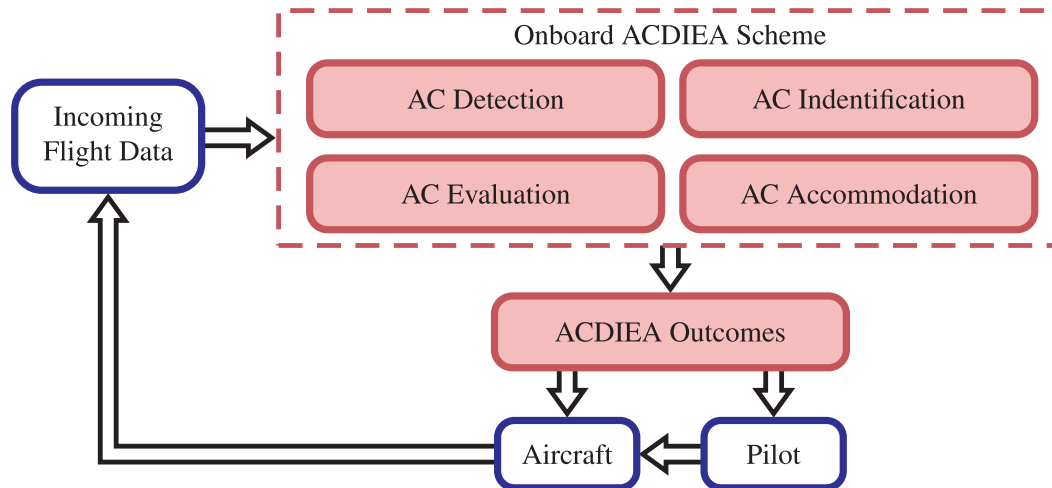
Figure 3.5: Online ACDIEA Scheme

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. The block diagram of the post-processing component is presented in Fig. 3.6.
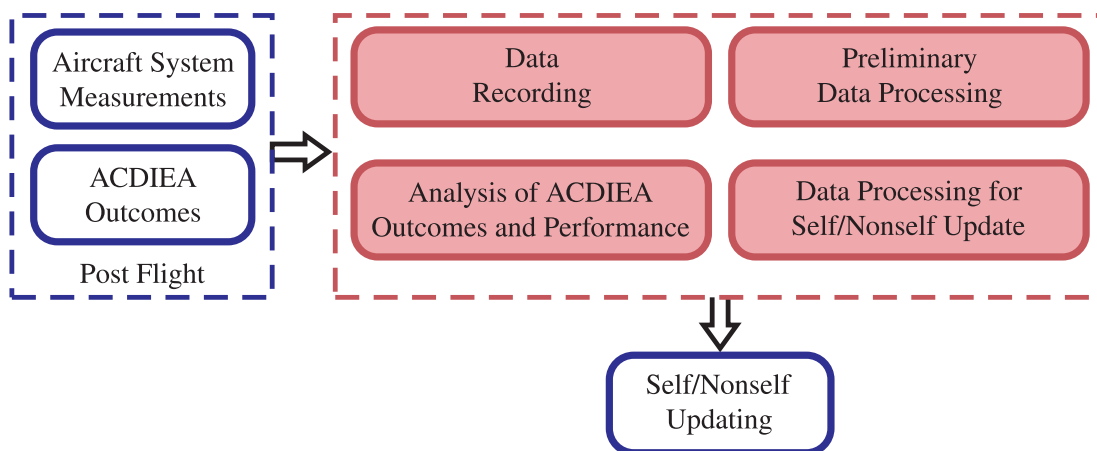
Figure 3.6: Post-Processing of Flight Data and ACDIEA Outcomes

Experimental and Simulation Environment

4.1. WVU Motion-Based Flight Simulator

The WVU motion-based flight simulator system used in this research is shown Fig. 4.1. The simulator includes a 6-DOF motion platform driven by electrical induction motors (shown in Fig. 4.2), laminar research X-Plane flight simulation software, LCD mosaic wall four-monitor external visual display, instructors operating station, and computer and control cabinet.
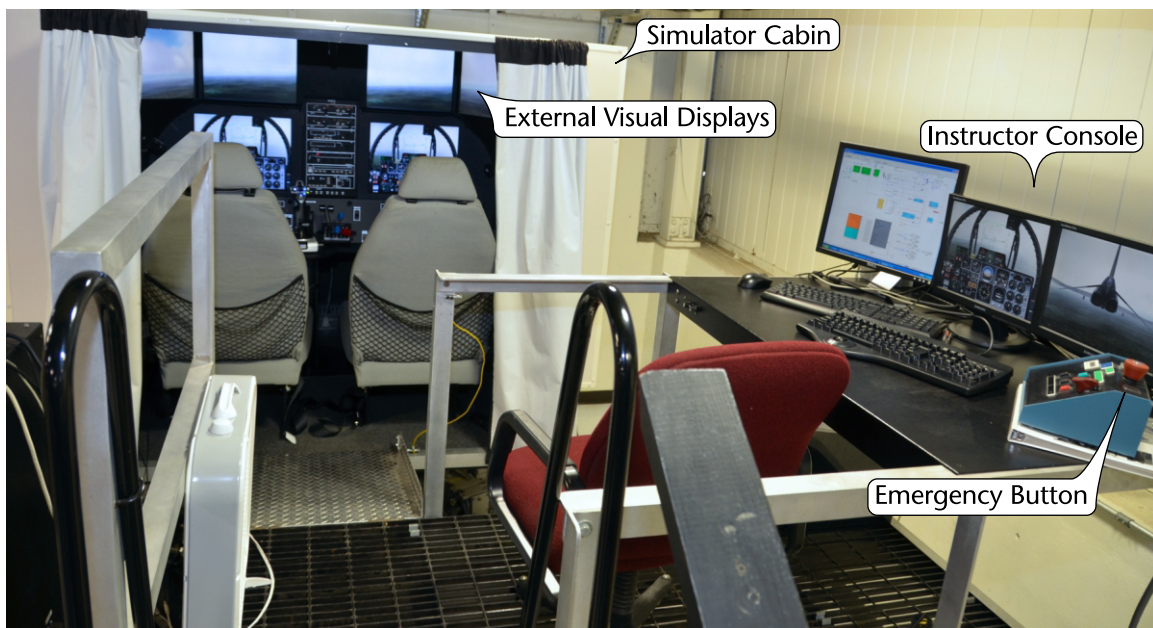
Figure 4.1: The WVU 6-DOF Motion-Based Flight Simulator

The motion platform provides adequate 6-DOF translational and rotational motion cues. Electrical motors are used to drive the motion base, which represents a very versatile and inexpensive solution to this type of application. Motion drive algorithms convert the motion of the aircraft as resulting from the dynamic model into motion of the platform such that the perception of the pilot is optimized within the physical limitations of the ground based simulator.
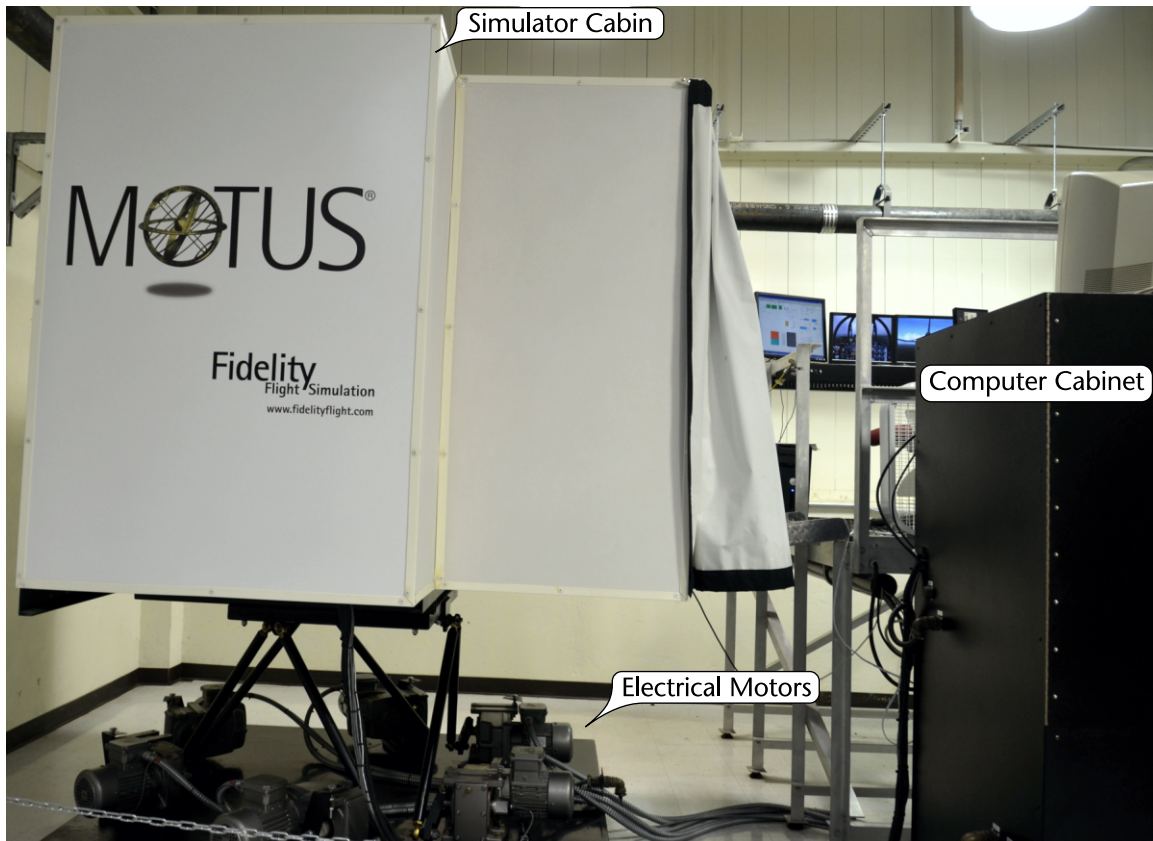
Figure 4.2: The Actuators of the WVU Motion-Based Flight Simulator

The simulator has been interfaced [91] with an external computer (see Fig. 4.3) on which the aircraft model can be run within the Matlab®/Simulink® environment to drive the entire simulator system. Pilot input signals are transferred from the simulator cockpit into the Matlab®/Simulink® model. The outputs of this model are sent to X-Plane [92], for the control of all the simulator subsystems including the generation of visual cues. However, the connection of X-Plane to the motion computer is deactivated and the signals from the 20 external computer are sent directly to the motion computer, which drives the motion base. This setup allows the use of any Simulink aircraft model including customized failures to drive the simulator.

Figure 4.4 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 (as described in Sec. 4.2) and models failure/damages of actuator, sensors, wing, and engine. 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.

Figure 4.3: Interface of the WVU Flight Simulator with External Models [93]

Figure 4.4: Top Level Simulink Model Interfaced with WVU 6-DOF Flight Simulator [91]

4.2. The Aircraft Model

The aircraft model used in this work is a supersonic fighter NASA IFCS F-15 research aircraft. The aircraft aerodynamic model is derived from a nonlinear model of a high performance military aircraft distributed by NASA to academic institutions in 1990 within a student design competition [94]. This generic model was customized through the addition of the aerodynamics modeling of canard surfaces for the purpose of simulating the NASA IFCS F-15 research aircraft [95]. The aerodynamic and thrust characteristics are provided through 42 lookup tables: 16 tables for the longitudinal dynamics as functions of Mach number, angle of attack and stabilator deflection, 20 tables for the lateral-directional dynamics as functions of Mach number, angle of attack, sideslip angle and rudder, and 2 tables for engine thrust and fuel flow as functions of Mach number and altitude. Additional lookup tables have been added for the modeling the canards. The lookup tables have been subdivided to isolate the contribution of individual aerodynamic surfaces and control surfaces in order to be able to simulate structural damage and control surface failure.

The aircraft includes model following adaptive control laws based on nonlinear dynamic inversion and ANN augmentation which also produces estimates of aircraft angular rates and angular acceleration errors that are used as features [96].

4.3. Modeling of Aircraft Subsystem Abnormal Conditions

The aircraft subsystems and the failure types considered in this effort are summarized in Table 4.1. Four major subsystems were modeled to support the development and testing of the AIS-based FDIE scheme: actuators, sensors, structure, and propulsion.

A total of 30 failures were simulated for the 19 subsystems, as shown in the table. The failure modeling of each of the main subsystems is briefly described next.

Table 4.1: The Aircraft Subsystems and Their Failure Types

Main Subsystem	k	Subsystem	AC_j	Failure Type
Actuators	1	Left Stabilator	1	Locked at current trim position
			2	Moving and locking at non-trim position
	2	Right Stabilator	1	Locked at current trim position
			2	Moving and locking at non-trim position
	3	Left Aileron	1	Locked at current trim position
			2	Moving and locking at non-trim position
	4	Right Aileron	1	Locked at current trim position
			2	Moving and locking at non-trim position
	5	Left Rudder	1	Locked at current trim position
			2	Moving and locking at non-trim position
	6	Right Rudder	1	Locked at current trim position
			2	Moving and locking at non-trim position

Table 4.1 – Cont'd.

Main Subsystem	k	Subsystem	AC_j	Failure Type
Actuators	7	Left Throttle	1	Locked at current trim position
			2	Moving and locking at non-trim position
	8	Right Throttle	1	Locked at current trim position
			2	Moving and locking at non-trim position
Sensors	9	Roll Rate Sensor	1	Sensor bias
			2	Constant sensor output
	10	Pitch Rate Sensor	1	Sensor bias
			2	Constant sensor output
	11	Yaw Rate Sensor	1	Sensor bias
			2	Constant sensor output
Structure	12	Left Wing	1	Missing part
	13	Right Wing	1	Missing part
	14	Left Horizontal Tail	1	Missing part
	15	Right Horizontal Tail	1	Missing part
	16	Left Vertical Tail	1	Missing part
	17	Right Vertical Tail	1	Missing part
Propulsion	18	Left Engine	1	Reduced effectiveness
	19	Right Engine	1	Reduced effectiveness
	$N_s = 19$		$N_F = \sum\limits_{k=1}^{N_s} Nft_k = 2 \times 8 + 2 \times 3 + 1 \times 6 + 1 \times 2 = 30$	

1. **Actuators:** The actuators considered are the aerodynamic control surfaces (stabilator, aileron, rudder, and throttle). A jammed control surface for an airplane is a control surface that is locked at a certain deflection and can no longer be moved. The deflection at post-failure condition can be the one existing at the moment of the failure occurrence or the surface can move to a failed position within the deflection range of the surface and remain there. This failure does not alter the aerodynamic characteristics of the control surface and can be simulated by simply disconnecting the corresponding input to the aircraft dynamic system and keeping it constant. However, each surface in a pair (left and right) will have different deflections and the resulting moments and forces must be computed individually. The aerodynamic lookup tables have been divided such that the contribution of each individual control surface is isolated [97].

2. **Sensors:** The malfunction of the sensors for which the outputs are used in the control laws feedback may have a significant impact on the overall performance of the system, potentially leading to catastrophic system failure. The sensors considered are the roll, pitch, and yaw rate

gyros. Several types of failures have been modeled and implemented for these sensors [98], including large and small step bias, large and small fast drifting bias, and large and small slow drifting bias.

A step bias occurs when a sensor continuously reports a constant offset from the correct value. For example, the pitch rate sensor is indicating the aircraft is pitching upward at a certain constant rate when the aircraft is actually in steady-state level flight. A large step bias or a small step bias indicates the relative magnitude of the sensor failure.

Drifting bias arises when the sensor values are increasing or decreasing at a constant rate from the actual values until a bias saturation level is reached. Large drifting bias and small drifting bias are relative indications to the severity of the failure. Furthermore, fast drifting bias and slow drifting bias quantify how fast the saturation level is reached.

3. **Structure:** A simple model of wing damage is developed considering both aerodynamic and gravimetric effects. The failure type corresponds to a total or partial physical destruction and/or deformation of the wing and different percent values along the wing can be selected as damage affected area. In addition, the effect of this type of failure on the ailerons control (physically damaged aerodynamic control surface) has been modeled for certain wing damage percentages. At post failure conditions, the asymmetry of aerodynamic forces produced by the left and right aerodynamic surfaces requires that the aircraft forces and moment computation be adjusted. An additional rolling moment is introduced due to the asymmetric lift produced by the damaged semi-wing, and an additional yawing moment due to the asymmetric drag produced by the damaged semi-wing.

4. **Propulsion:** Reduced control efficiency corresponds to the situation when less additional thrust is produced by the same displacement of the throttle control. This failure is modeled by scaling down the throttle input by a constant factor depending on the severity of the failure (percentage loss of efficiency) [94].

4.4.   Flight Simulation Scenarios

To define the AIS self/nonself 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 Fig. 4.5 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/nonself of the aircraft. Additional intermediate points (A, B, C, and D in Fig. 4.5) 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.
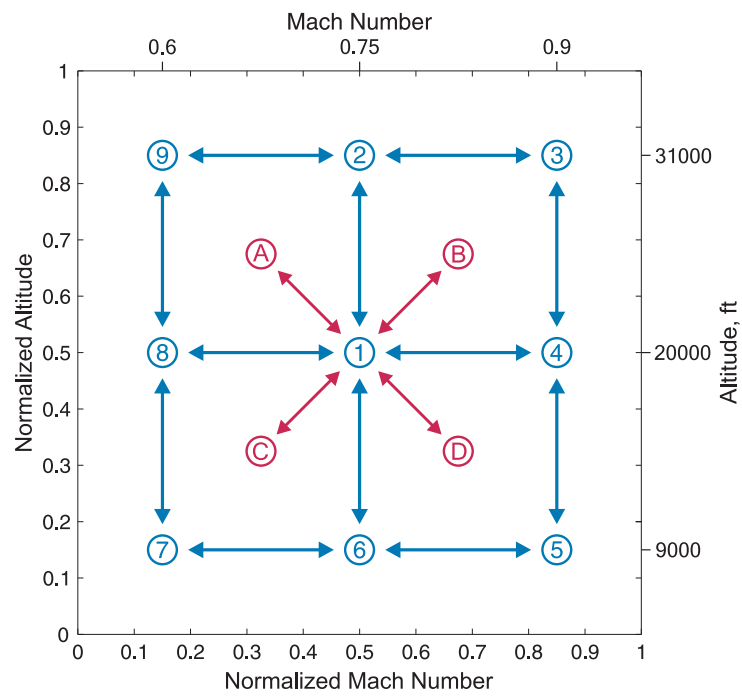
Figure 4.5: Testing Flight Envelope

Generation of the Self and Nonself

Generating the self, $\mathbb{S}$, requires collecting large amounts of measured feature values at normal conditions, ideally covering the entire flight envelope. In practice, these N-dimensional feature points are usually clustered using a clustering algorithm. *Self clusters* are geometrical hyper-bodies that can typically take one the following shapes:

- 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.

Similar hyper-bodies are used to represent the nonself, $\hat{\mathbb{S}}$, and will be referred to as *antibodies*. These antibodies primarily serve as *detectors* of the AC. The geometrical shape of these hyper-bodies can potentially affect the efficiency of the detector generation and the performance of AC detection [99]. They determine how well the nonself is covered, how many detectors are necessary, and how intensive the computational process is.

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 set of self clusters can be expressed as

$$S = \{c_1, c_2, \ldots, c_{Nc}\}, \qquad c_i = \begin{bmatrix} C_i & Rc_i \end{bmatrix} = \begin{bmatrix} \varphi_{1i} & \varphi_{2i} & \cdots & \varphi_{Ni} & Rc_i \end{bmatrix}, \qquad (5.1)$$

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

$$\hat{S} = \{d_1, d_2, \ldots, d_{Nd}\}, \qquad d_j = \begin{bmatrix} D_j & Rd_j \end{bmatrix} = \begin{bmatrix} \varphi_{1j} & \varphi_{2j} & \cdots & \varphi_{Nj} & Rd_j \end{bmatrix}, \qquad (5.2)$$

where D_j is the center and Rd_j is the radius of detector j.

When generating the self and the nonself, the following optimization criteria should be considered [78, 83, 84]:

- no overlapping among detectors and self clusters;
- minimum empty space in the self clusters;
- minimum uncovered areas in the nonself;
- minimum overlapping among self clusters;

- minimum overlapping among detectors;
- minimum number of detectors (note that this criterion may be constrained by imposing a maximum size for the detectors in order to achieve a desirable resolution).

   Two alternative methods can be used for generating the self and nonself [100]: the *raw data set union method* (RDSUM) and the *cluster set union method* (CSUM). The block diagram summarizing both methods is shown in Fig. 5.1. In the first method, 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 the data sets and then combining the clusters in a single set for antibodies generation. Using the two approaches, data for each combination of features corresponding to a particular projection are processed separately to produce a set of antibodies by covering the respective nonself. These methods are presented in Sections 5.2 and 5.3.
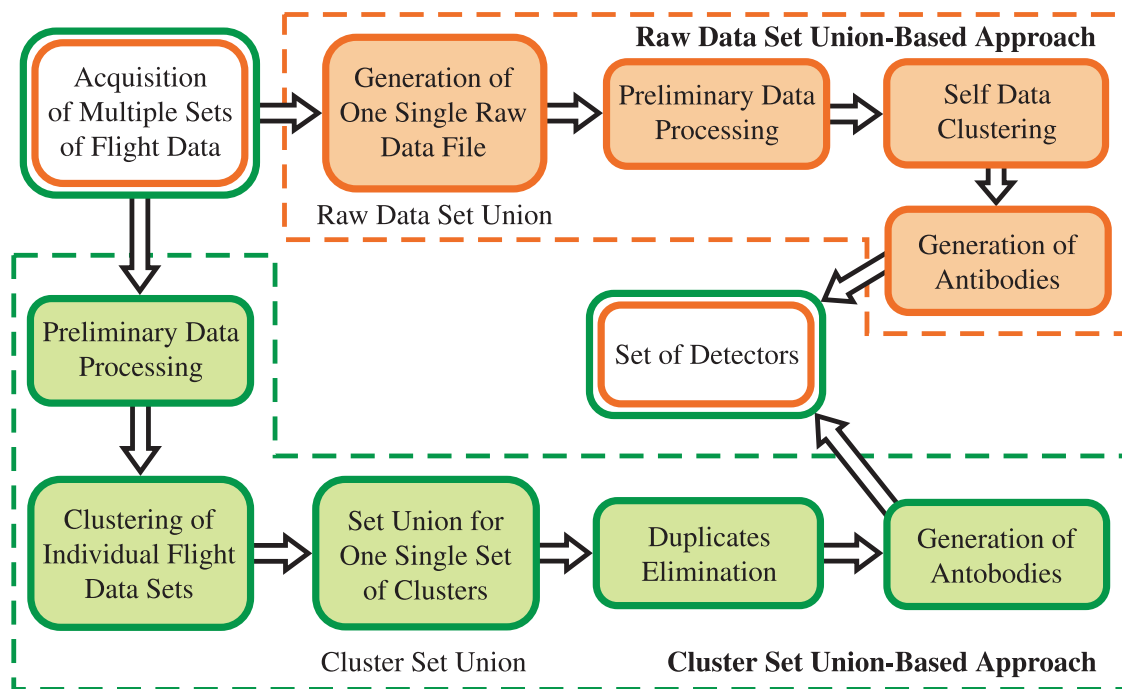
Figure 5.1: Two Methods for AIS Antibodies Generation

5.1.   Features Selection

The candidate parameters for self/nonself generation of the supersonic aircraft model described in Sec. 4.2 can be grouped in the following five categories:

- Aircraft state variables.
- Pilot input variables.
- Stability and control derivatives.
- Variables generated within the control laws.
- Derived variables.

The aircraft state variables are a natural choice since measurements of aircraft angular rates have been used for self/nonself definition and failure detection with promising results [78].

Intentional pilot input may reproduce the dynamic fingerprint of some failures on lower-dimensional projections. The stability and control derivatives may also provide useful information regarding subsystem failures [101]. It should be noted that determining them online is not trivial and they have not been considered in this work.

The ANN control augmentation implemented within the model provides useful signals with significant ACDIE capabilities; in particular, the NN outputs and their derivatives have ACDIE potential [95]. Since the adaptation activity increases after the occurrence of a failure, these signals capture the increased adaptation activity and, thus, can detect the failure.

Additional features have been defined based on the neural network estimates of the aircraft angular rates [95]:

$$MQEE\left(t\right) = \frac{1}{2}\left\{\left[p\left(t\right) - \hat{p}_{MNN}\left(t\right)\right]^2 + \left[q\left(t\right) - \hat{q}_{MNN}\left(t\right)\right]^2 + \left[r\left(t\right) - \hat{r}_{MNN}\left(t\right)\right]^2\right\}, \qquad (5.3)$$

$$OQEE\left(t\right) = \frac{1}{2}\left\{\left[\hat{p}_{DNN}\left(t\right) - \hat{p}_{MNN}\left(t\right)\right]^2 + \left[\hat{q}_{DNN}\left(t\right) - \hat{q}_{MNN}\left(t\right)\right]^2 + \left[\hat{r}_{DNN}\left(t\right) - \hat{r}_{MNN}\left(t\right)\right]^2\right\}, \quad (5.4)$$

$$DQEE_x\left(t\right) = \frac{1}{2}\left[\hat{x}_{DNN}\left(t\right) - x\left(t\right)\right]^2, \qquad x = p, q, r, \qquad (5.5)$$

where $p\left(t\right)$, $q\left(t\right)$, and $r\left(t\right)$ are measurements of angular rates at sample t; $\hat{p}_{MNN}\left(t\right)$, $\hat{q}_{MNN}\left(t\right)$, and $\hat{r}_{MNN}\left(t\right)$ are neural estimates of the angular rates based on sensor measurements *including* the respective gyro, over a specified time window; and $\hat{p}_{DNN}\left(t\right)$, $\hat{q}_{DNN}\left(t\right)$, and $\hat{r}_{DNN}\left(t\right)$ are neural estimates of the angular rates based on sensor measurements that do *not* include the respective gyro, over a specified time window.

Table 5.1 lists all the F-15 aircraft features used for the AIS development.

Table 5.1: The AIS Development Features of the Supersonic Aircraft

ID	Feature	Description
1	p_{ref}	Reference roll rate
2	q_{ref}	Reference pitch rate
3	r_{ref}	Reference yaw rate
4	NN_p	Neural network output on the roll channel
5	NN_q	Neural network output on the pitch channel
6	NN_r	Neural network output on the yaw channel
7	$MQEE$	Main quadratic estimation error
8	$OQEE$	Output quadratic estimation error
9	$DQEE_p$	Decentralized quadratic roll rate estimation error
10	$DQEE_q$	Decentralized quadratic pitch rate estimation error
11	$DQEE_r$	Decentralized quadratic yaw rate estimation error
12	V	Ground velocity

Table 5.1 – Cont'd.

ID	Feature	Description
13	α	Angle of attack
14	β	Sideslip angle
15	p	Roll rate
16	q	Pitch rate
17	r	Yaw rate
18	ϕ	Roll attitude angle
19	θ	Pitch attitude angle
20	ψ	Yaw attitude angle
21	H	Altitude
22	$\dot{p}$	Roll acceleration
23	$\dot{q}$	Pitch acceleration
24	$\dot{r}$	Yaw acceleration
25	a_x	Longitudinal acceleration
26	a_y	Lateral acceleration
27	a_z	Vertical acceleration
28	d_a	Lateral stick displacement
29	d_e	Longitudinal stick displacement
30	d_r	Pedal displacement
31	d_T	Throttle displacement
32	M	Mach number

5.2.   Raw Data Set Union Method

The RDSUM processes experimental data at normal conditions in four main modules, as shown in Fig. 5.1:

1. **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.

2. **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.

3. **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 [96] was used to represent the clusters as hyper-spheres, but they could also be represented as hyper-rectangles or hyper-ellipsoids.

4. **Generation of Antibodies:** Self clusters are used to generate antibodies by covering the nonself hyper-space with hyper-bodies similar to the clusters. An enhanced negative selection algorithm for real-valued representation with variable nonself radius (ENSA-RV) [96] was used for the purpose of this research. The algorithm ensures that there is no overlapping with the self and that the nonself 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 nonself is achieved.

5.3. Cluster Set Union Method

The mechanism to generate antibodies using the CSUM is based on a 5-phase/modules process that uses an optimized algorithm to fuse different sets of clusters generated from single sets of experimental flight or simulation data. The main components of this methodology are described next.

1. **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 subsets 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.

2. **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.

3. **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.

4. **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.

5. **Generation of Antibodies:** The same enhanced negative selection algorithm for real-valued representation with variable detector radius (ENSA-RV) [96] 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 uncovered areas in the nonself.

5.4.   The Hierarchical Multiself Strategy

The HMS strategy, proposed in [79], attempts to mitigate the dimensionality issues associated to the high-dimensional self by using lower-dimensional projections for ACDIE. Considering the different level of capabilities of the lower-dimensional projections in capturing the fingerprint of an abnormal condition, the projections are assigned confidence factors (weights) and used in hierarchical configurations for the three ACDIE processes. With the HMS strategy, the ACDIE scheme is capable of detecting and identifying a large variety of AC of the main aircraft subsystems over a wide range of severity. Figure 5.2 illustrates the block diagram of the HMS strategy for the online ACDIE.
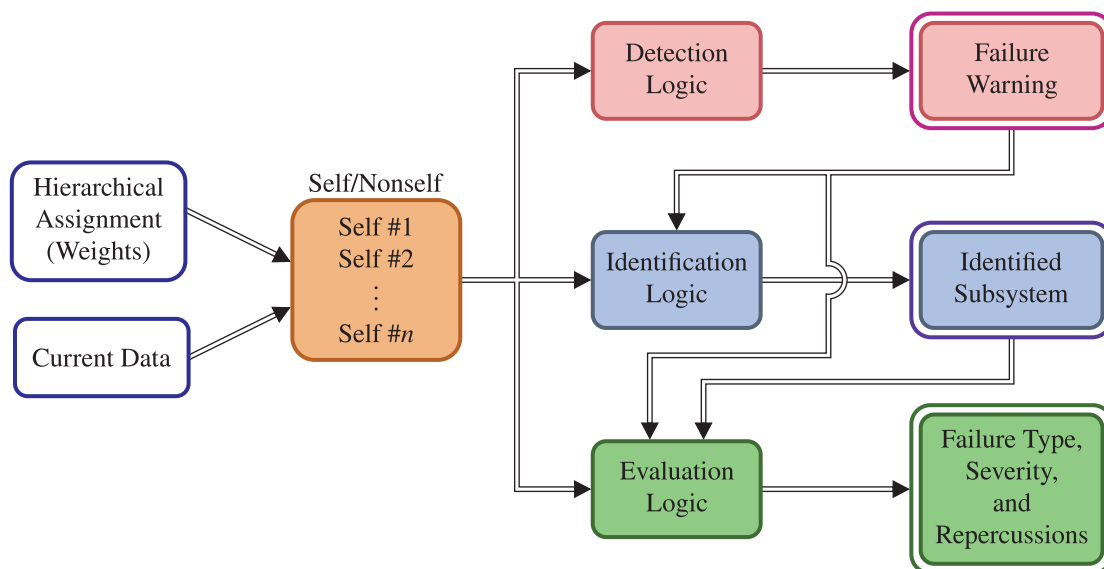
Figure 5.2: Hierarchical Multiself Strategy for Online ACDIE

5.5. Self/Nonself Analysis Tool

Visualizing the 2-dimensional subselves 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 Fig. 5.3) was developed to analyze all the generated 2-D subselves and project a given flight test data on these subselves in order to validate them before adopting them as a basis to the ACDIE schemes.
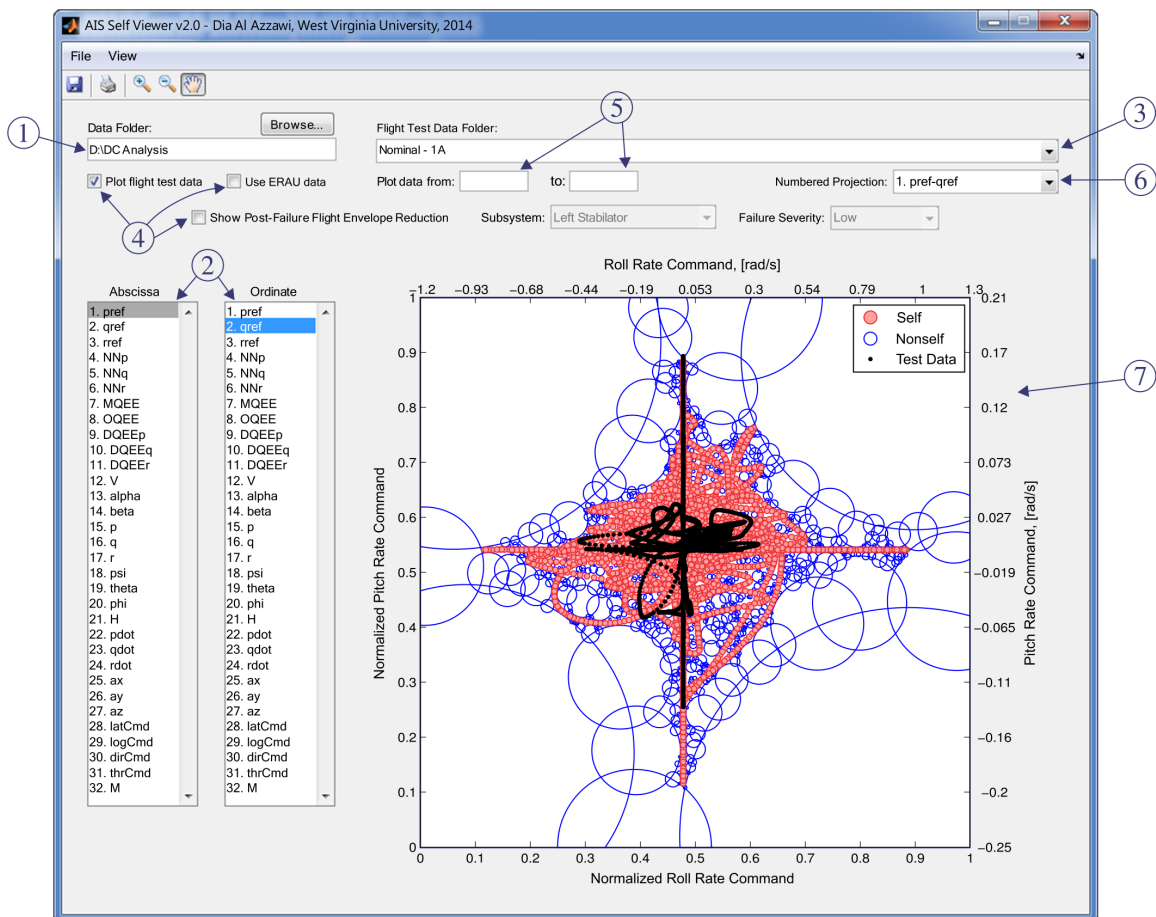
Figure 5.3: The Interactive Tool for Self/Nonself Analysis

The tool is user friendly and has the following main components referenced in Fig. 5.3:

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 listboxes of features containing the features listed in Table 5.1. Any feature selected from the left listbox represents the abscissa of the 2-D plot and any feature selected from the right listbox represents the ordinate.
3. "Flight test" dropdown menu containing all available flight tests.
4. Three checkboxes providing options for plotting flight test data points on the figure, plotting self/nonself 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 will be plotted.

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

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

The zooming feature of the analysis tool facilitates the validation of the self/nonself generation process such as how well the detectors cover the nonself 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. For instance, one might conclude from Figs. 5.3 and 5.4 that there is an issue with the self generating process. However, checking the range (minimum and maximum values) of p_{ref} in the original flight data used in generating the subselves and the corresponding range in the selected flight test reveals that the maximum value used in the flight test was higher than that used in the original flight data. This explains why there are few test points outside the self clusters in Fig. 5.4.
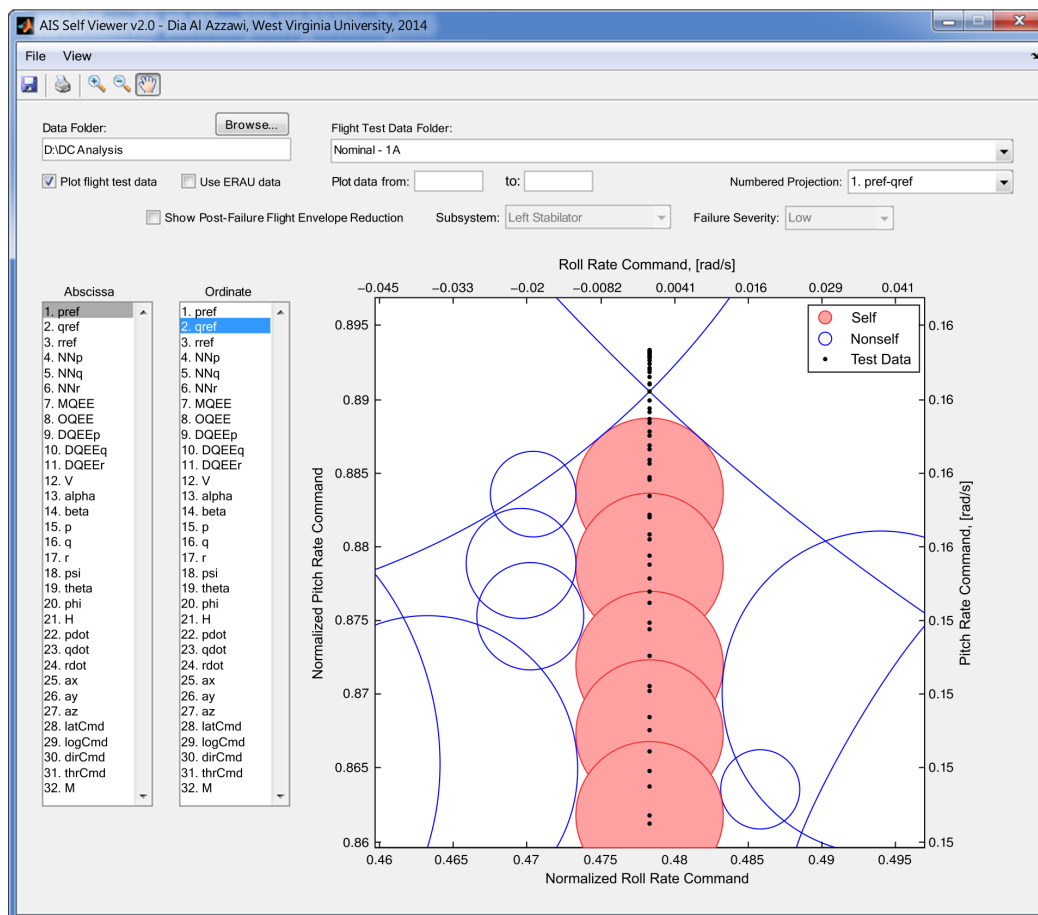
Figure 5.4: The Zooming Action of the Interactive Tool for Self/Nonself Analysis

Immunity-Based Abnormal Conditions Detection

Detecting the presence of an AC is the first and critical step in the ACM process. Without a reliable and fast detection scheme, the AC identification and evaluation schemes will not provide timely and reliable outcomes as expected and, eventually, the AC accommodation will not properly compensate for the abnormal variations in the system parameter(s) that are affected by the AC. This chapter introduces the basic concept of self/nonself discrimination and discusses its general applicability to the AC detection. From the immune system point of view, the self/nonself discrimination relies on the functionality of the antibodies generated in the adaptive immune system to detect the existence of the antigens. Due to the computational issues involved in generating the self and nonself, a perfect definition of the self clusters and the antibodies is very difficult to achieve in practice. Recalling the additional issues that may arise from the lower-dimensional projections of the self, as discussed in Chapter 3, the self/nonself discrimination may lead to false alarms and/or missed detections. However, investigations within this research effort have shown that false alarms and missed detections can be reduced by properly processing the outcomes of the self/nonself discrimination. This has led to the development of a novel computational algorithm for AC detection inspired from recent findings in immunology that the immune response is governed by the interaction of the various components of *both* the innate and adaptive immune systems [102, 103]. In particular, dendritic cells in the innate immune system play a major role in activating special populations of T-cells in the adaptive immune system as well as interacting with the antibodies in attacking the invading antigens. Section 6.2 presents the artificial DC mechanism which processes the outcomes of the self/nonself discrimination, within the HMS strategy, to minimize the false alarms and increase the detection rate.

6.1. Self/Nonself Discrimination

The discrimination between the self and nonself is a fundamental concept in the AIS field. The outcome of the AC detection can be obtained based on two mechanisms that distinguish between the self and nonself, namely, the *positive selection* (PS) and *negative selection* (NS).

In the PS mechanism, the measured features point, P, is compared against the self clusters. Assuming hyper-spherical self clusters as defined by Eq. (5.1), the detection outcome at each sample time, t, is given by

$$Det_t = \begin{cases} 0 & \text{if } \left| \mathbf{r}^{OP_t} - \mathbf{r}^{OC_i} \right| \le Rc_i, \\ 1 & \text{otherwise,} \end{cases} \quad \forall\, i = 1, 2, \ldots, Nc, \tag{6.1}$$

where $\mathbf{r}^{OP_t}$ is the position vector of the measured feature point as given by Eq. (3.14) relative to the origin O of the coordinate system $\mathcal{U}$.

On the contrary, the NS mechanism compares the measured features point, P, with the detectors. If the detectors are defined as in Eq. (5.2), then the NS mechanism provides the detection outcome at each sample time, t, as

$$Det_t = \begin{cases} 0 & \text{if } \left| \mathbf{r}^{OP_t} - \mathbf{r}^{OD_j} \right| > Rd_j, \\ 1 & \text{otherwise,} \end{cases} \qquad \forall \, j = 1, 2, \ldots, Nd. \qquad (6.2)$$

If the HMS strategy with N_{ss} subselves is used, then either Eq. (6.1) or (6.2) has to be applied for each subself to obtain sets of values Det_i, $i = 1, 2, \ldots, N_{ss}$. In this case, the detection outcome is expressed as

$$Det_t = \max_{i=1, N_{ss}} \left(Det_{ti} \right). \qquad (6.3)$$

If the self and nonself are perfectly defined, the discrimination between them would definitely provide the desired outcome of the AC detection. If, on the other hand, a perfect definition of the self/nonself is unattainable, false alarms and/or missed detections are inevitable. However, excessive false alarms and missed detections can be minimized if the outcomes of the self/nonself discrimination are processed in a certain way before providing the detection outcome. This can be achieved by incorporating an additional detection logic to provide the detection outcome as a function of current and past discrimination outcomes within the HMS strategy. Within this research effort [104, 105], it was revealed that such a detection logic can be inspired from the functionality of the biological dendritic cells in the innate immune system and their interaction with the different components of the adaptive immune system. The next section presents an AC detection algorithm based on the DC mechanism.

6.2. The Artificial Dendritic Cell Mechanism for AC Detection

The various significant components of the biological immune system serve as sources of inspiration for mathematical and computational elements pertinent to aircraft abnormal condition detection and identification within the AIS paradigm. These correspondences are summarized in Table 6.1 and will be further explained throughout this section.

Table 6.1: The Biological Terms and Their AIS Paradigm Representations

Biological Term	AIS Paradigm Representation
Self or Organism	Data clusters of feature values acquired from development tests under normal operating conditions
Nonself or Antibodies	Complementary clusters of self clusters that function as detectors in the self/nonself discrimination process
Antigen	Set of feature values in a validation test
DC	Computational unit that processes the outcomes of the self/nonself discrimination process
Regulatory DC	Computational unit that votes a normal condition

Table 6.1 – Cont'd.

Biological Term	AIS Paradigm Representation
Stimulatory DC	Computational unit that votes an abnormal condition
Costimulatory Molecules	Number of times a computational unit has been selected for processing the inputs
DC Life	Number of times allowed for the computational unit to be selected for processing the inputs
Interleukin-10	Counter of number of zeros in a self/non-discrimination outcome processed by the computational unit
Interleukin-12	Counter of number of ones in a self/non-discrimination outcome processed by the computational unit
Regulatory T-Cells	Number of computational units voting a normal condition over a moving time window
Stimulatory T-Cells	Number of computational units voting an abnormal condition over a moving time window

The artificial DC mechanism proposed here is a novel computational algorithm inspired from the functionality of the biological DCs in the tissue and their interaction with with T-cells and antibodies in the adaptive immune system [104, 105]. Cells from a pool of DCs are randomly selected to process the outcomes of the self/nonself discrimination process. 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, whereas 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. The block diagram of the algorithm is illustrated in Fig. 6.1.

The inputs to the algorithm are the outcomes of the self/nonself discrimination over a moving time window of size T. Let these outcomes be defined as $Det_{\tau j}$, where $\tau = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, N_{ss}$, and let $\mathbf{D}_t$ and $\overline{\mathbf{D}}_t$ be the discrimination matrices defined by

$$
\mathbf{D}_t =
\begin{matrix}
\mathcal{S}_1 & \mathcal{S}_2 & \cdots & \mathcal{S}_{N_{ss}} \\
\begin{bmatrix}
Det_{11} & Det_{12} & \cdots & Det_{1N_{ss}} \\
Det_{21} & Det_{22} & \cdots & Det_{2N_{ss}} \\
\vdots & \vdots & \ddots & \vdots \\
Det_{T1} & Det_{T2} & \cdots & Det_{TN_{ss}}
\end{bmatrix}
\end{matrix},
\qquad
Det_{\tau j} =
\begin{cases}
0 & \text{if } P_{\tau j} \text{ is inside } \mathcal{S}_j, \\
1 & \text{if } P_{\tau j} \text{ is outside } \mathcal{S}_j,
\end{cases}
\tag{6.4}
$$

$$
\overline{\mathbf{D}}_t = \begin{matrix} \mathcal{S}_1 & \mathcal{S}_2 & \cdots & \mathcal{S}_{N_{ss}} \end{matrix} \\
\begin{bmatrix} \overline{Det}_{11} & \overline{Det}_{12} & \cdots & \overline{Det}_{1N_{ss}} \\ \overline{Det}_{21} & \overline{Det}_{22} & \cdots & \overline{Det}_{2N_{ss}} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{Det}_{T1} & \overline{Det}_{T2} & \cdots & \overline{Det}_{TN_{ss}} \end{bmatrix}, \qquad \overline{Det}_{\tau j} = \begin{cases} 0 & \text{if } P_{\tau j} \text{ is outside } \mathcal{S}_j, \\ 1 & \text{if } P_{\tau j} \text{ is inside } \mathcal{S}_j. \end{cases} \tag{6.5}
$$
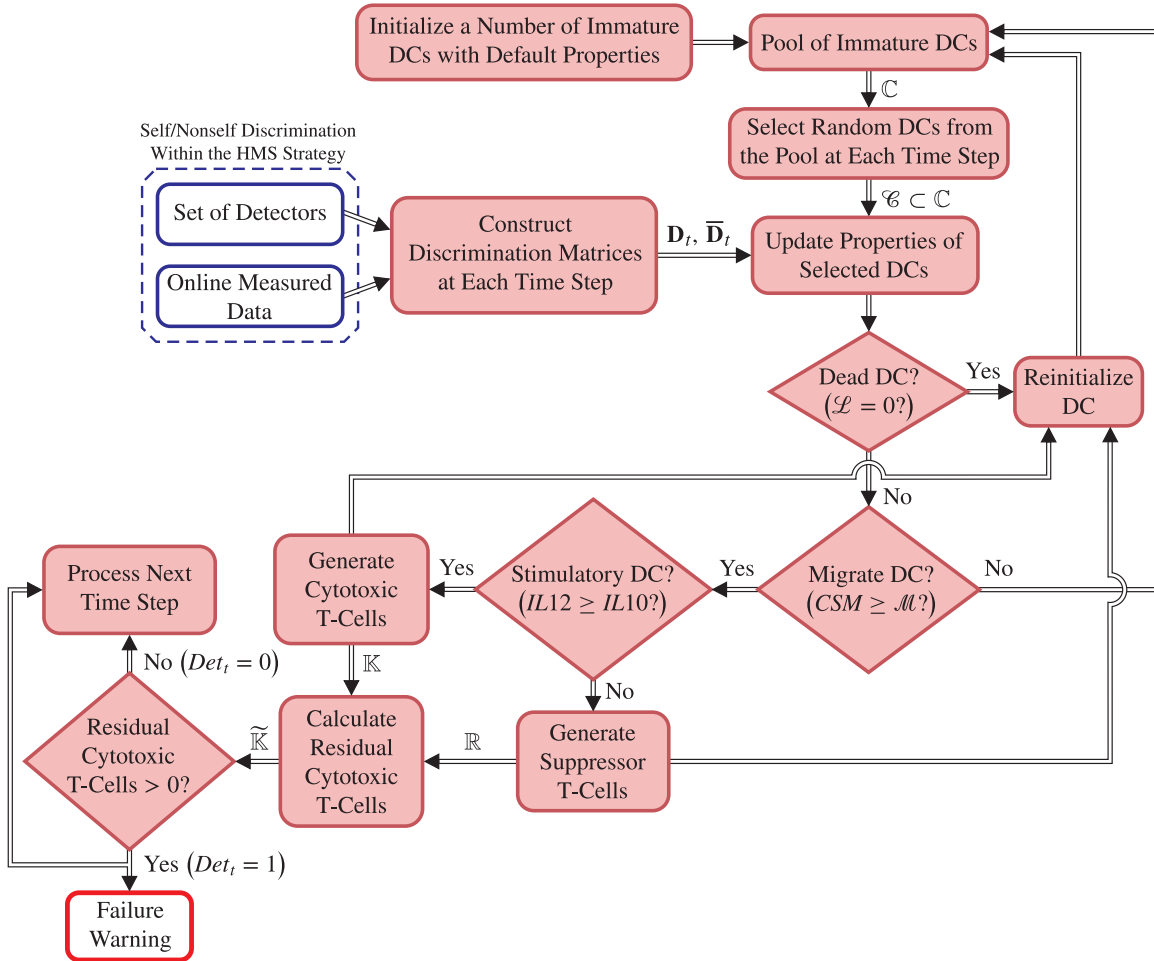
Figure 6.1: Block Diagram of the Artificial DC Mechanism for AC Detection

Note that $\overline{\mathbf{D}}_t$ is the complement of $\mathbf{D}_t$. Each column of matrix $\mathbf{D}_t$ is a set of detection outcomes of a particular subself over the entire time window. Ideally, all elements of this matrix are equal to zero under normal conditions and one under abnormal conditions. In practice, $\mathbf{D}_t$ usually contains both zeros and ones whether the time window is under normal or abnormal conditions due to imperfections in designing and building the subselves and/or due to the shape of the self that may project abnormal condition feature points onto the self area [106]. This situation may lead to false

alarms and missed detections, if not properly handled.

A pool containing a set $\mathbb{C}$ of N_{DC} immature artificial DCs is first initialized with each cell $C_\ell \in \mathbb{C}$ ($\ell = 1, 2, \ldots, N_{DC}$) having the data structure shown in Fig. 6.2.
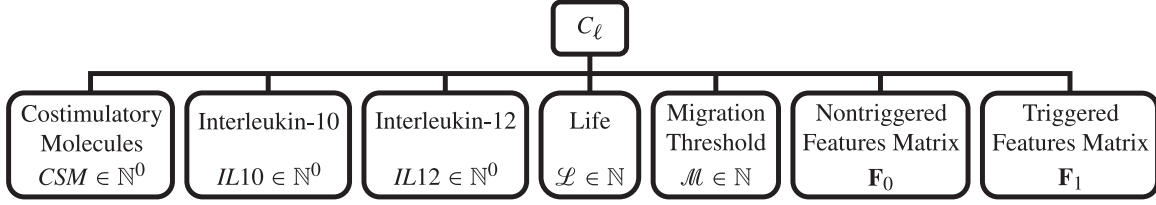
Figure 6.2: Data Structure of an Artificial Dendritic Cell

At each sample time, a *random* number of DCs is selected from the pool and each selected DC processes the input discrimination matrices, $\mathbf{D}_t$ and $\overline{\mathbf{D}}_t$, and updates its data structure as follows:

1. **Costimulatory Molecules,** *CSM***:** These are secreted proteins that appear on the surface of the biological DC. The concentration of these proteins increases whenever the DC processes a substance, regardless whether it is a harmful antigen or not. Similarly, when an artificial DC receives the input $Det_{\tau j}$, the new *CSM* value becomes

$$CSM_\tau = CSM_{\tau-1} + 1, \tag{6.6}$$

regardless of the value of the input $Det_{\tau j}$. All DCs in the pool are initialized with $CSM = 0$.

2. **Interleukin-10,** *IL*10**:** These are special cytokines that are produced by the biological DC when the substance is *suspected* to be either part of the self or a foreign, but harmless, one. Similarly, when an artificial DC receives an input $Det_{\tau j} = 0$, the new *IL*12 value becomes

$$IL10_\tau = IL10_{\tau-1} + \sum_{j=1}^{N_{ss}} \overline{Det}_{\tau j} W_{0j}, \tag{6.7}$$

where $\mathbf{W}_0 = \begin{bmatrix} W_{01} & W_{02} & \cdots & W_{0N_{ss}} \end{bmatrix}^T$ is a vector of confidence factors such that $W_{0j} \in [0, 1]$ to take into consideration the different level of capabilities of the subselves to capture the fingerprint of the normal condition. All DCs in the pool are initialized with $IL10 = 0$.

3. **Interleukin-12,** *IL*12**:** These are special cytokines that are produced by the biological DC when the substance is *suspected* to be either an antigen or a foreign harmful one. Similarly, when an artificial DC receives an input $Det_{\tau j} = 1$, the new *IL*12 value becomes

$$IL12_\tau = IL12_{\tau-1} + \sum_{j=1}^{N_{ss}} Det_{\tau j} W_{1j}, \tag{6.8}$$

where $\mathbf{W}_1 = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1N_{ss}} \end{bmatrix}^T$ is a vector of confidence factors such that $W_{1j} \in [0, 1]$ to take into consideration the different level of capabilities of the subselves to capture the fingerprint of the abnormal condition. All DCs in the pool are initialized with $IL12 = 0$.

4. **Life, $\mathcal{L}$:** Like any other cell in the body, the biological DC experiences a healthy programmed cell death [63]. In the artificial DC mechanism, this can be done by assigning a *random* positive integer to each DC in the pool representing the life remaining for that DC to keep processing the input of the next time step. Each DC then updates its life property according to

$$\mathcal{L}_\tau = \mathcal{L}_{\tau-1} - 1. \tag{6.9}$$

5. **Migration Threshold, $\mathcal{M}$:** When the concentration of the *CSM* reaches a certain level, the biological DC becomes mature and migrates from the tissue to the lymph node in order to present the information they have collected about the processed substance to the adaptive immune system. Similarly, an artificial DC is said to be mature when its *CSM* reaches a predefined migration threshold, $\mathcal{M}$. Each DC in the pool is initialized with a *random* migration threshold.

6. **Triggered and Nontriggered Features Matrices:** When a biological DC engulfs the antigen, it breaks it up into its constituent molecules and moves them to the outer surface of the DC which helps the adaptive immune system to recognize the type of the antigen presented by the DC. Similarly, an artificial DC presents information about the processed input by constructing the triggered features matrix, $\mathbf{F}_1$, to record the number of times subself $\mathcal{S}_j$ is triggered and the nontriggered features matrix, $\mathbf{F}_0$, to record the number of times subself $\mathcal{S}_j$ is not triggered. These matrices are defined as

$$
\mathbf{F}_1 =
\begin{array}{cccc}
\mathcal{S}_1 & \mathcal{S}_2 & \cdots & \mathcal{S}_{N_{ss}}
\end{array}
\begin{bmatrix}
F_{11} & F_{12} & \cdots & F_{1N_{ss}} \\
F_{21} & F_{22} & \cdots & F_{2N_{ss}} \\
\vdots & \vdots & \ddots & \vdots \\
F_{N1} & F_{N2} & \cdots & F_{NN_{ss}}
\end{bmatrix}
\begin{array}{c}
\varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N
\end{array},
\qquad
\mathbf{F}_0 =
\begin{bmatrix}
\overline{F}_{11} & \overline{F}_{12} & \cdots & \overline{F}_{1N_{ss}} \\
\overline{F}_{21} & \overline{F}_{22} & \cdots & \overline{F}_{2N_{ss}} \\
\vdots & \vdots & \ddots & \vdots \\
\overline{F}_{N1} & \overline{F}_{N2} & \cdots & \overline{F}_{NN_{ss}}
\end{bmatrix}
\begin{array}{c}
\varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_N
\end{array},
\tag{6.10}
$$

where $F_{ij} = F_{ij}+1$ if φ_i is one of the feature coordinates of the *triggered* subself $\mathcal{S}_j$ and $\overline{F}_{ij} = \overline{F}_{ij}+1$ if φ_i is one of the feature coordinates of the *nontriggered* subself $\mathcal{S}_j$.

Any DC which migrates to the lymph node (i.e., the adaptive immune system) with $IL12 \geq IL10$ is called *stimulatory DC* since it activates the production of cytotoxic T-cells in the adaptive immune system. The set of activated cytotoxic T-cells can be expressed as

$$\mathbb{K} = \left\{ K_i \in \mathbb{N}^0 \mid i = 1, 2, \ldots, N \right\}, \qquad K_i = \sum_{m=1}^{N_{SDC}} \sum_{j=1}^{N_{ss}} F_{ij}^{(m)}, \tag{6.11}$$

where N_{SDC} is the number of stimulatory DCs and K_i is the number of cytotoxic T-cells corresponding to feature φ_i.

On the other hand, any migrated DC with $IL12 < IL10$ is called *regulatory DC* since it activates the production of suppressor (regulatory) T-cells in the adaptive immune system. The set of activated suppressor T-cells can be expressed as

$$\mathbb{R} = \left\{ R_i \in \mathbb{N}^0 \mid i = 1, 2, \ldots, N \right\}, \qquad R_i = \sum_{m=1}^{N_{RDC}} \sum_{j=1}^{N_{ss}} \overline{F}_{ij}^{(m)}, \tag{6.12}$$

where N_{RDC} be the number of regulatory DCs and R_i is the number of suppressor T-cells corresponding to feature φ_i.

The role of the suppressor T-cells is to regulate the adaptive immune response by suppressing an equal number of activated cytotoxic T-cells which results in a set of residual cytotoxic T-cells given by

$$\widetilde{\mathbb{K}} = \left\{ \widetilde{K}_i = K_i - R_i \mid i = 1, 2, \ldots, N \right\}. \tag{6.13}$$

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.

Once all the elements of the discrimination matrices, $\mathbf{D}_t$ and $\overline{\mathbf{D}}_t$, are processed as described above, the detection outcome for the t-th time window can be calculated from

$$Det_t = \begin{cases} 0 & \text{if } \sum_{i=1}^{N} \widetilde{K}_i = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{6.14}$$

Immunity-Based Abnormal Conditions Identification

The AC identification is the diagnostic process which isolates the subsystem that is mostly affected by the detected AC. In addition to its robust capability of detecting the intruders, the biological immune system is also capable of correctly identifying both the intruder and the infected area in the body. In particular, the peptides that appear on the surface of a mature DC are *specific* to the invading antigen [63]. Similarly, the migrated DCs in the artificial DC mechanism presented in Chapter 6 carry information that is useful in identifying the failed subsystem [104, 105]. This information is summarized by the triggered-features matrix, $\mathbf{F}_1$, defined in Eq. (6.8). Moreover, three patterns can be extracted from the $\mathbf{F}_1$ matrices of the migrated DCs, depending on how $\mathbf{F}_1$ is viewed: *features pattern*, *projections pattern*, and *matrix pattern*. These patterns can be generated both offline and online. Patterns that are generated offline using *training tests* are called *reference patterns*. It should be pointed out here that the AC identification requires establishing N_s *different* reference patterns, one associated to each subsystem. Patterns that are generated online either during a simulation test or from recorded *validation tests* are called *current patterns*. Thus, using the artificial DC mechanism converts the AC identification into a pattern recognition problem in which the failed subsystem is identified as the one for which the reference pattern best matches the current pattern.

7.1. The Features-Pattern Approach

In this approach, the features are used to define the pattern. For the *j*-th AC ($j = 1, 2, \ldots, Nft_k$), a reference pattern may be established in terms of how strongly each feature is affected by the AC. In other words, a vector of membership values of each feature to the set of EDIVs (see Definition 3.11) is defined as

$$FP_{AC_j} = \begin{bmatrix} m_1 & m_2 & \cdots & m_N \end{bmatrix}^T, \qquad m_i \in [0, 1]. \tag{7.1}$$

Considering that each subsystem may be affected by more than one AC and assuming superposition, the features-pattern vector can be defined for each subsystem k as

$$FP_k = \frac{\sum_{j=1}^{Nft_k} FP_{AC_j}}{\left\| \sum_{j=1}^{Nft_k} FP_{AC_j} \right\|} = \begin{bmatrix} \overline{m}_1 & \overline{m}_2 & \cdots & \overline{m}_N \end{bmatrix}, \tag{7.2}$$

where the norm in the denominator is the Euclidean norm and $\overline{m}_i \in [0, 1]$ are membership values. All these membership values can be determined from tests and/or heuristics. Note that binary logic

can be used instead of fuzzy logic. In this case, $m_i \in \{0, 1\}$ and $\overline{m}_i \in \{0, 1\}$.

At each sample time, after an AC is detected, the current features pattern can be obtained from the triggered-feature matrices of all migrated DCs as

$$F_{1\varphi} = \frac{\left(\sum_{m=1}^{N_{MDC}} \mathbf{F}_1^{(m)} \right) I_{N_{ss}}}{\left\| \left(\sum_{m=1}^{N_{MDC}} \mathbf{F}_1^{(m)} \right) I_{N_{ss}} \right\|}, \qquad I_{N_{ss}} \in \{1\}^{N_{ss} \times 1}, \tag{7.3}$$

where the norm in the denominator is the Euclidean norm. Note that Eq. (7.3) can be used to define the reference features pattern, as it will be shown in Sec. 7.4. Also note that both Eqs. (7.2) and (7.3) define normalized vectors so that they can be compared without bias to the actual magnitudes of their corresponding elements.

At this point, a pattern recognition algorithm is required in order to determine which of the FP_k vectors of Eq. (7.2) is the closest to $F_{1\varphi}$ and ultimately constitute the outcome of the AC identification process as

$$Idt_t = \begin{bmatrix} id_1 & id_2 & \cdots & id_{N_s} \end{bmatrix}, \tag{7.4}$$

where

$$id_k = \begin{cases} 1 & \text{if } FP_k \text{ best matches } F_{1\varphi}, \\ 0 & \text{otherwise.} \end{cases} \tag{7.5}$$

7.2.   The Projections-Pattern Approach

The projections-pattern approach is analogous to the features-pattern approach with the exception that the reference and current patterns are defined based on the projections (i.e., subselves) that are triggered during the detection phase. For the j-th AC ($j = 1, 2, \ldots, Nft_k$), 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) is defined as

$$PP_{AC_j} = \begin{bmatrix} n_1 & n_2 & \cdots & n_{N_{ss}} \end{bmatrix}^T, \qquad n_i \in \begin{bmatrix} 0, 1 \end{bmatrix}. \tag{7.6}$$

Considering again that each subsystem may be affected by more than one AC and assuming superposition, the projections-pattern vector can be defined for each subsystem k as

$$PP_k = \frac{\sum_{j=1}^{Nft_k} PP_{AC_j}}{\left\| \sum_{j=1}^{Nft_k} PP_{AC_j} \right\|} = \begin{bmatrix} \overline{n}_1 & \overline{n}_2 & \cdots & \overline{n}_{N_{ss}} \end{bmatrix}, \tag{7.7}$$

where the norm in the denominator is the Euclidean norm and $\overline{n}_i \in \begin{bmatrix} 0, 1 \end{bmatrix}$ are membership values. All these membership values can be determined from tests, analysis, and/or heuristics. Binary logic can be used here as well, for simplicity.

At each sample time, after an AC is detected, the current projections pattern can be obtained from the triggered-feature matrices of all migrated DCs as

$$F_{1P} = \frac{I_N \left(\sum_{m=1}^{N_{MDC}} \mathbf{F}_1^{(m)} \right)}{\left\| I_N \left(\sum_{m=1}^{N_{MDC}} \mathbf{F}_1^{(m)} \right) \right\|}, \qquad I_N \in \{1\}^{1 \times N}, \tag{7.8}$$

where the norm in the denominator is the Euclidean norm.

Similar to the features-pattern approach, Eq. (7.8) can be used to define the reference projections pattern. At this point, a pattern recognition algorithm is required in order to determine which of the PP_k vectors of Eq. (7.7) best matches F_{1P}. Once the pattern is recognized, the outcome of the AC identification is obtained as

$$Idt_t = \begin{bmatrix} id_1 & id_2 & \cdots & id_{N_s} \end{bmatrix}, \tag{7.9}$$

where

$$id_k = \begin{cases} 1 & \text{if } PP_k \text{ best matches } F_{1P}, \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

7.3. The Matrix-Pattern Approach

The matrix-pattern approach combines both the features- and projections-pattern approaches. Here, the patterns are established based on the entire matrix $\mathbf{F}_1$. For each subsystem k, both the FP_k and PP_k vectors must be defined using the algorithms presented in the previous sections to eventually construct an $N \times N_{ss}$ matrix given by

$$MP_k = \begin{bmatrix} m_{11}^* & m_{12}^* & m_{1N_{ss}}^* \\ m_{21}^* & m_{22}^* & m_{2N_{ss}}^* \\ & & \\ m_{N1}^* & m_{N2}^* & m_{NN_{ss}}^* \end{bmatrix}, \tag{7.11}$$

where $m_{ij}^* \in [0, 1]$ are membership values determined from

$$m_{ij}^* = \begin{cases} \max(m_i, n_j) & \text{if } \varphi_j \text{ is one of the feature coordinates} \\ & \qquad \text{of the triggered subself } S_j, \\ 0 & \qquad \text{otherwise.} \end{cases} \tag{7.12}$$

At each sample time, after an AC is detected, the sum of $\mathbf{F}_1$ matrices for all migrated DCs can be considered as an overall metric to establish the matrix pattern associated to the affected subsystem:

$$F_{1M} = \sum_{m=1}^{N_{SDC}} \mathbf{F}_1^{(m)}. \tag{7.13}$$

As was the case in the first two approaches, using a pattern recognition algorithm to determine which of the MP_k matrices of Eq. (7.11) best matches F_{1M} provides the outcome of the AC identification as

$$Idt_t = \begin{bmatrix} id_1 & id_2 & \cdots & id_{N_s} \end{bmatrix}, \tag{7.14}$$

where

$$id_k = \begin{cases} 1 & \text{if } MP_k \text{ best matches } F_{1M}, \\ 0 & \text{otherwise.} \end{cases} \tag{7.15}$$

7.4. The AC Identification Scheme Using the DC Mechanism

The three approaches described in the previous sections need a pattern recognition algorithm to recognize the current patterns produced by the migrated DCs. In machine learning, there are many pattern recognition algorithms among which the naïve Bayes classifier is the most popular one. Naïve Bayes classifier is a *probabilistic supervised learning* algorithm which provides very high classification rate with very fast *training* and *validation* phases in most practical applications. Besides, the classifier provides an automated way of establishing the reference patterns. These advantages of the classifier make it the most suitable algorithm for the purpose of pattern recognition of the AC identification problem. The reader is referred to Appendix A for detailed derivation of the classifier probabilistic models.

This section describes the AC identification scheme using the features-pattern approach. However, the scheme is not limited to the features-pattern approach; any of the other approaches (i.e., the projections-pattern or the matrix-pattern) can be used instead.

For normally distributed continuous attributes, the quadratic discriminant function, $\Delta_k(\mathbf{x})$, is given by (see Appendix A for a detailed derivation of the function),

$$\Delta_k(\mathbf{x}) = \ln N_k - \frac{1}{2} \ln |\mathbf{\Sigma}_k| - \frac{1}{2} (\mathbf{x} - \mathbf{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x} - \mathbf{\mu}_k), \tag{7.16}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T$ is the vector of continuous values of N attributes to be classified into subsystem k, N_k is the number of samples in subsystem k, $\mathbf{\mu}_k$ is the sample mean vector, and $\mathbf{\Sigma}_k$ is the sample covariance matrix.

Once the mean vectors and covariance matrices are known, the classifier model of Eq. (7.16) can be used to calculate the discriminant functions of all subsystems for the current $\mathbf{x}$. The best-matching subsystem can then be identified from

$$k^* = \underset{k=1, 2, ..., N_s}{\operatorname{argmax}} \big[\Delta_k(\mathbf{x}) \big]. \tag{7.17}$$

The mean vectors and covariance matrices are obtained by *training* the classifier offline against samples from a set of training tests. Referring to Fig. 7.1, the training phase is summarized by the following steps:

1. For each subsystem k, prepare a set of training flight tests.
2. For each training test, run the DC algorithm for detection. When a failure is detected, compute $\mathbf{x}_k = F_{1\varphi}$ using Eq. (7.3).

3. Denote the number of $\mathbf{x}_k$ vectors for all training tests by N_k and compute the mean vector and covariance matrix from

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{m=1}^{N_k} \mathbf{x}_k^{(m)}, \tag{7.18}$$

and

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k - 1} \sum_{m=1}^{N_k} \left(\mathbf{x}_k^{(m)} - \boldsymbol{\mu}_k \right) \left(\mathbf{x}_k^{(m)} - \boldsymbol{\mu}_k \right)^T, \tag{7.19}$$

respectively.
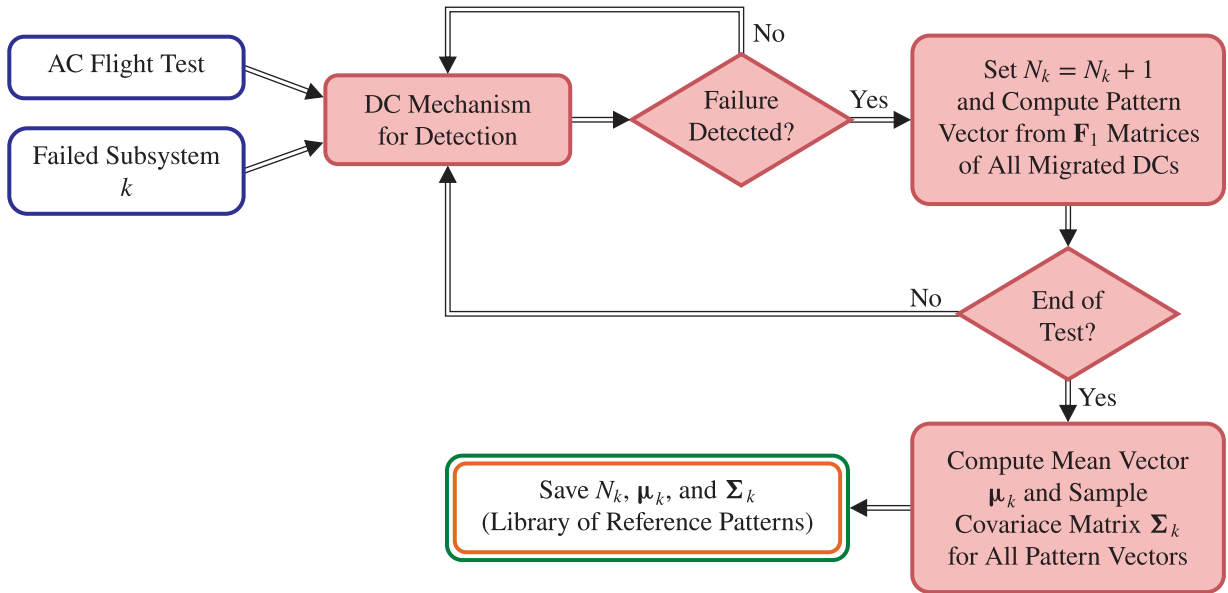
Figure 7.1: Training the Naïve Bayes Classifier for AC Identification

Obviously, the mean vectors and covariance matrices *implicitly* define the reference features pattern, FP_k, as a function of the training current features pattern. After constructing all the reference patterns, the classifier can be used online to recognize the failed subsystem from the current feature patterns in a given AC test as follows (see the Fig. 7.2):

1. Whenever the DC algorithm for detection declares a failure, compute $\mathbf{x}_k = F_{1\varphi}$ using Eq. (7.3).
2. For each subsystem k, compute $\Delta_k (\mathbf{x})$ from Eq. (7.16) using the mean vectors and covariance matrices calculated in the training phase.
3. Use Eq. (7.17) to determine the failed subsystem k^*.
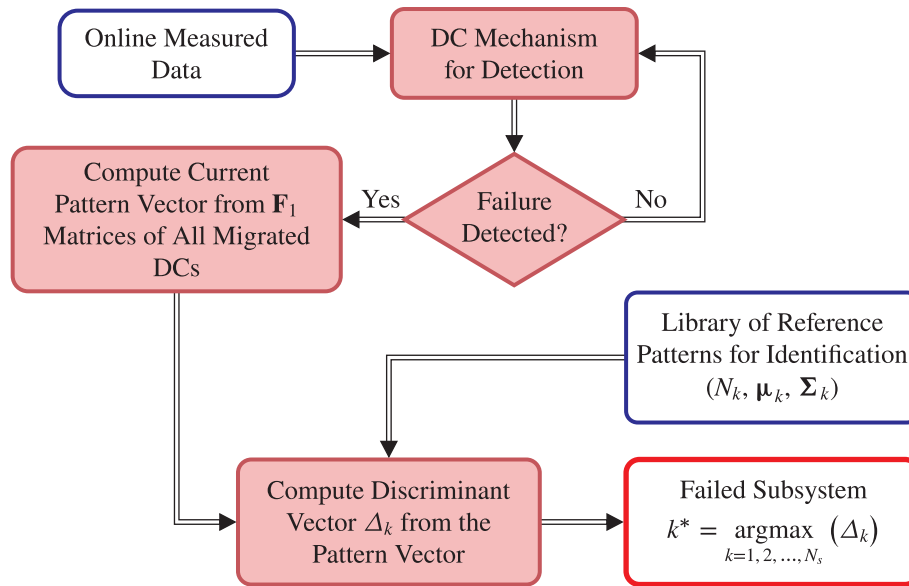
Figure 7.2: AC Identification Scheme Using the Dendritic Cell Mechanism

Immunity-Based Abnormal Conditions Evaluation

Aircraft AC evaluation is a vital prerequisite to a successful post-failure accommodation. AC are evaluated both *qualitatively* and *quantitatively*. The qualitative evaluation determines the type of the failure, whereas the quantitative evaluation estimates the severity of the failure (direct quantitative evaluation) and the limitations of the flight envelope at post-failure (indirect quantitative evaluation). It should be noted that correct flight envelope estimation requires correct evaluation of both the type and severity of the AC once the failed subsystem is correctly identified.

Extensions of the DC mechanism, introduced in Chapter 6, to the ACQE and ACDQE problems are presented in this chapter. Similar to the DC mechanism for identification presented in Chapter 7, the patterns carried by the migrated DCs are used to define the reference patterns by retraining the pattern recognition algorithm for each problem separately [107]. The three pattern approaches discussed in Chapter 7 can be applied to both ACQE and ACDQE. However, only the features-pattern approach will be discussed here, since using the projections-pattern or the matrix-pattern follow the same logical steps. Section 8.3 presents an immunity-based approach for the post-failure flight envelope prediction under various failure scenarios of the aircraft subsystems.

8.1. Abnormal Conditions Qualitative Evaluation

The ACQE starts once an AC is detected, assuming correct detection. Within the HMS strategy, the ACQE can be formulated as a pattern recognition problem, where the pattern represents the level of impact of the AC on each feature. Therefore, patterns for AC qualitative evaluation purposes corresponding to each type of failure of the affected subsystem can be defined based on the detection outcome of all subselves, which is summarized by the $\mathbf{F}_1$ matrix provided by the migrated DCs.

If the number of AC of subsystem k is Nft_k, then Nft_k *different* reference patterns should be established: one associated to each type of failure. The reference features-pattern vector for the failure type can be expressed as

$$FP_{AC_j} = \begin{bmatrix} m_1 & m_2 & \cdots & m_N \end{bmatrix}^T, \tag{8.1}$$

where $m_i \in \begin{bmatrix} 0, & 1 \end{bmatrix}$ are fuzzy membership values of each feature with respect to the set of EDIVs (see Definition 3.11). They can be determined, in general, from analysis, tests, and/or heuristics. Each membership value m_i can be associated to the level of triggering of all projections corresponding to the respective feature, of which $\mathbf{F}_1$ matrices provide a measure. This allows the reference features-pattern vector to be determined by training the naïve Bayes classifier offline in a similar fashion to that described in Sec. 7.4. The training samples from a set of training tests are represented by the

current features-pattern vectors provided by the triggered-features matrices of all migrated DCs as in Eq. (7.3). Figure 8.1 illustrates the block diagram of the training phase of the classifier.
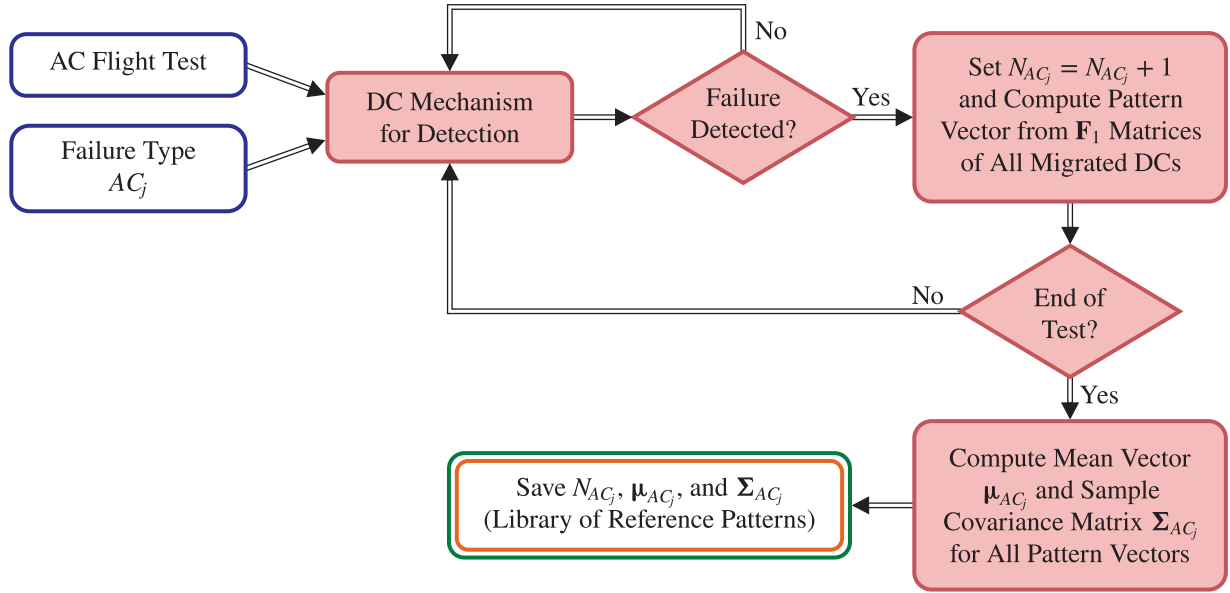
Figure 8.1: Training the Naïve Bayes Classifier for ACQE

The online ACQE starts immediately after a failure is declared by the DC mechanism. The current features-pattern $F_{1\varphi}$ of Eq. (7.3) is computed and compared to the library of reference patterns to determine the discriminant Δ_{AC_j} for each failure type from

$$\Delta_{AC_j}\left(\mathbf{x}\right) = \ln N_{AC_j} - \frac{1}{2}\ln\left|\boldsymbol{\Sigma}_{AC_j}\right| - \frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_{AC_j}\right)^T \boldsymbol{\Sigma}_{AC_j}^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_{AC_j}\right), \tag{8.2}$$

where N_{AC_j} is the number of training patterns, $\mathbf{x} = F_{1\varphi}$, and $\boldsymbol{\mu}_{AC_j}$ and $\boldsymbol{\Sigma}_{AC_j}$ are the mean vector and sample covariance matrix given by

$$\boldsymbol{\mu}_{AC_j} = \frac{1}{N_{AC_j}}\sum_{m=1}^{N_{AC_j}} \mathbf{x}_{AC_j}^{(m)}, \tag{8.3}$$

and

$$\boldsymbol{\Sigma}_{AC_j} = \frac{1}{N_{AC_j}-1}\sum_{m=1}^{N_{AC_j}}\left(\mathbf{x}_{AC_j}^{(m)} - \boldsymbol{\mu}_{AC_j}\right)\left(\mathbf{x}_{AC_j}^{(m)} - \boldsymbol{\mu}_{AC_j}\right)^T, \tag{8.4}$$

respectively.

The block diagram of ACQE scheme described in this section is shown in Fig. 8.2.
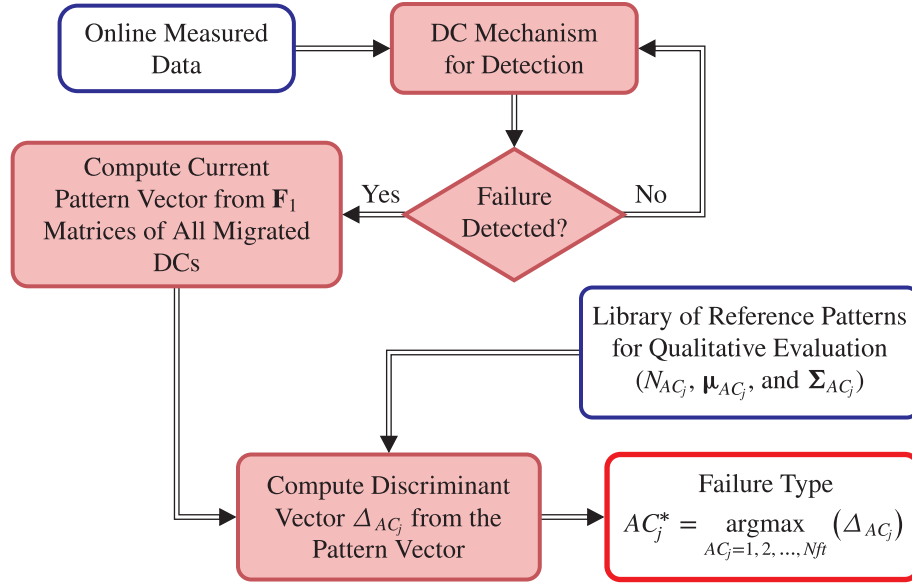
Figure 8.2: ACQE Scheme Using the Dendritic Cell Mechanism

8.2. Abnormal Conditions Direct Quantitative Evaluation

Similar to the ACQE, patterns for each failure severity can be defined based on the detection outcome of all subselves based on the $\mathbf{F}_1$ matrices of the migrated DCs. If the number of severity scales of failure AC_j is *Nfs*, then *Nfs different* reference patterns must be established: one associated to each severity scale. The reference features-pattern vector corresponding to the failure severity $AC_{j\varsigma}$ can be expressed as

$$FP_{AC_{j\varsigma}} = \begin{bmatrix} \widetilde{m}_1 & \widetilde{m}_2 & \cdots & \widetilde{m}_N \end{bmatrix}^T, \tag{8.5}$$

where $\widetilde{m}_i \in [0, 1]$ are fuzzy membership values of each feature with respect to the set of EDIVs (see Definition 3.11).

The reference features-pattern of Eq. (8.5) 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 ACDQE is very similar to training it for ACQE as shown in the schematic diagram of Fig. 8.3.

The online ACDQE starts immediately after a failure is declared by the DC mechanism. The current features-pattern $F_{1\varphi}$ of Eq. (7.3) is computed and compared to the library of reference patterns to determine the discriminant Δ_{fs} for each severity scale from

$$\Delta_{fs}(\mathbf{x}) = \ln N_{fs} - \frac{1}{2}\ln|\mathbf{\Sigma}_{fs}| - \frac{1}{2}(\mathbf{x} - \mathbf{\mu}_{fs})^T \mathbf{\Sigma}_{fs}^{-1}(\mathbf{x} - \mathbf{\mu}_{fs}), \tag{8.6}$$

where N_{fs} is the number of samples in tests with failure severity *fs*, $\mathbf{x} = F_{1\varphi}$, and $\mathbf{\mu}_{fs}$ and $\mathbf{\Sigma}_{fs}$ are the mean vector and sample covariance matrix given by
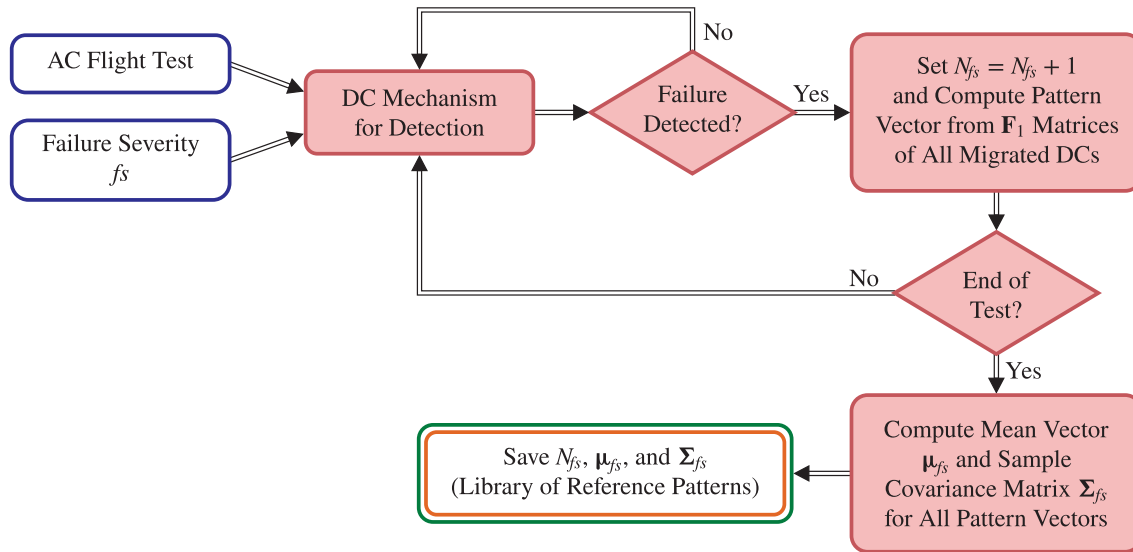
Figure 8.3: Training the Naïve Bayes Classifier for ACDQE

$$\boldsymbol{\mu}_{fs} = \frac{1}{N_{fs}} \sum_{m=1}^{N_{fs}} \mathbf{x}_{fs}^{(m)}, \tag{8.7}$$

and

$$\boldsymbol{\Sigma}_{fs} = \frac{1}{N_{fs}-1} \sum_{m=1}^{N_{fs}} \left(\mathbf{x}_{fs}^{(m)} - \boldsymbol{\mu}_{fs} \right) \left(\mathbf{x}_{fs}^{(m)} - \boldsymbol{\mu}_{fs} \right)^{T}, \tag{8.8}$$

respectively.

The block diagram of ACDQE scheme is presented in Fig. 8.4.
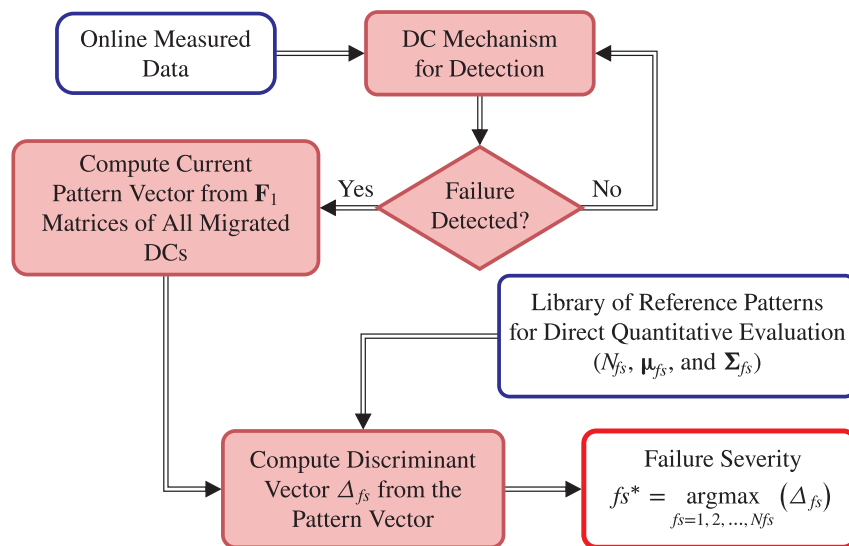
Figure 8.4: ACDQE Scheme Using the Dendritic Cell Mechanism

8.3. Abnormal Conditions Indirect Quantitative Evaluation

The presence of AC in one or more of the subsystems of an aircraft may result in detrimental effects on the flight envelope and, consequently, the overall safety of the aircraft and capability to complete the mission. To support the AC accommodation, it is highly desirable to estimate the achievable flight envelope at post-failure. Within the immuntiy-based ACDIE framework proposed in this work, this section presents an AIS-based approach for flight envelope estimation under various AC of the aircraft subsystems.

The approach requires defining a set of DIVs (see Definition 3.10) for each failed subsystem along with a set of *envelope related variables* (ERVs) that could be affected by the AC and establishing analytical relationships between the DIVs and EDIVs (see Definition 3.11) so that the post-failure ranges can be extracted from the 2-D subselves in which an EDIV is one of its coordinates and an ERV is the other coordinate.

Let $\mathbb{V}_\delta = \{v_{\delta i} \mid i = 1, 2, \ldots, N_\delta\}$ be the set of DIVs that characterizes the AC and let $\mathbb{V}_E = \{v_{Ei} \mid i = 1, 2, \ldots, N_E\}$ be the set of ERVs that could be affected by that AC such that $\mathbb{V}_E \subset \mathscr{F}_E \subset \mathscr{F}$, where $\mathscr{F}_E$ is the set of ERVs for all subsystems. Since, by definition, the DIVs do not necessarily belong to the features set $\mathscr{F}$ that define the self/nonself, it is necessary to find a function that maps $v_{\delta i} \notin \mathscr{F}$ to an EDIV which belongs to $\mathscr{F}$ and, subsequently, extracting the post-failure range of the ERV from the corresponding 2-D subself if there is a mutual constraint between the two features involved in the projection. Note that special attention should be paid to the case when such a constraint does not exist, since values of one of the features can cover the entire nominal range regardless of the value of the other feature and, therefore, the post-failure range of the ERV cannot be obtained from the corresponding 2-D subself in this case.

To illustrate the approach, suppose $v_{\delta 1} \notin \mathscr{F}$ is the only DIV that characterizes a hypothetical failure which takes place when $v_{\delta 1} = c$, where c is some constant. In order to estimate the post-failure range of the envelope variable v_{E1} using one of the 2-D projections of the self/nonself, then v_{E1} should be one of the coordinates of that projection and the other coordinate should be an EDIV, $v_{\varepsilon 1} \in \mathscr{F}$, evaluated by the mapping $v_{\varepsilon 1} = f(v_{\delta 1})$. In fact, the constraint $v_{\delta 1} = c$ and the post-failure range $\left[v_{E1}\right]_F$ define a "new" self $\overline{\overline{\mathbb{S}}}$ representing reduced envelope at post-failure conditions such that

$$\overline{\overline{\mathbb{S}}} = \{c_i \mid c_i \text{ satisfies } \Gamma_i\}, \tag{8.9}$$

where $\Gamma_i = \{\gamma_{i1} \mid i = 1, 2, \ldots, N_{\Gamma_i}\}$ is the set of constraints produced by the AC on the set of EDIVs defined by $\mathbb{V}_\varepsilon = \{v_{\varepsilon i} \mid i = 1, 2, \ldots, N_\varepsilon\}$. This concept is illustrated in Fig. 8.5.

The immunity-based ACIQE scheme described here is generally formulated and can be applied to any system. However, the relationships between the DIVs to EDIVs are application specific and obtaining them often requires analysis of the system under consideration.

For the F-15 aircraft model, these relationships can be obtained from analyzing the aircraft dynamics for each failed subsystem individually. Table 8.1 presents typical lists of the DIVs, EDIVs, and ERVs for each failed subsystem. Note that in this context the flight envelope of this aircraft is defined by the sixteen-features set

$$\mathscr{F}_E = \{H, M, p, q, r, \dot{p}, \dot{q}, \dot{r}, a_x, a_z, \beta\} = \bigcup_{i=1}^{N_F} v_{E_i}. \tag{8.10}$$

The rest of this section is devoted to the derivation of the DIV-EDIV relationships for each of the subsystems listed in Table 8.1.

Figure 8.5: Post-Failure Flight Envelope Estimation Using the AIS Paradigm

Table 8.1: The Indirect Quantitative Evaluation Variables for the Supersonic Aircraft

k_F	k	**Failure Type**	$\mathbb{V}_\delta$	$\mathbb{V}_\varepsilon$	$\mathbb{V}_E$
1	1	Left Stabilator Locked	δ_{eL}	d_e	$H, M, p, q, \dot{q}$
2	1	Left Stabilator Locked and Moving	δ_{eL}	d_e	$H, M, p, q, \dot{q}$
3	2	Right Stabilator Locked	δ_{eR}	d_e	$H, M, p, q, \dot{q}$
4	2	Right Stabilator Locked and Moving	δ_{eR}	d_e	$H, M, p, q, \dot{q}$
5	3	Left Aileron Locked	δ_{aL}	d_a	$p, \dot{p}$
6	3	Left Aileron Locked and Moving	δ_{aL}	d_a	$p, \dot{p}$
7	4	Right Aileron Locked	δ_{aR}	d_a	$p, \dot{p}$
8	4	Right Aileron Locked and Moving	δ_{aR}	d_a	$p, \dot{p}$

<div align="center">**Table 8.1** – Cont'd.</div>

k_F	k	**Failure Type**	$\mathbb{V}_\delta$	$\mathbb{V}_\varepsilon$	$\mathbb{V}_E$
9	5	Left Rudder Locked	δ_{rL}	d_r	$r, \dot{r}$
10	5	Left Rudder Locked and Moving	δ_{rL}	d_r	$r, \dot{r}$
11	6	Right Rudder Locked	δ_{rR}	d_r	$r, \dot{r}$
12	6	Right Rudder Locked and Moving	δ_{rR}	d_r	$r, \dot{r}$
13	7	Left Throttle Locked	δ_{TL}	d_T	a_x, a_z, r, H, M
14	7	Left Throttle Locked and Moving	δ_{TL}	d_T	a_x, a_z, r, H, M
15	8	Right Throttle Locked	δ_{TR}	d_T	a_x, a_z, r, H, M
16	8	Right Throttle Locked and Moving	δ_{TR}	d_T	a_x, a_z, r, H, M
17	9	Roll Rate Sensor Bias	p_{meas}	d_a	$p, \dot{p}$
18	9	Roll Rate Sensor Constant Output	p_{meas}	d_a	$p, \dot{p}$
19	10	Pitch Rate Sensor Bias	q_{meas}	d_e	$q, \dot{q}$
20	10	Pitch Rate Sensor Constant Output	q_{meas}	d_e	$q, \dot{q}$
21	11	Yaw Rate Sensor Bias	r_{meas}	d_r	$r, \dot{r}, \beta$
22	11	Yaw Rate Sensor Constant Output	r_{meas}	d_r	$r, \dot{r}, \beta$
23	12	Left Wing Partially Missing	$\Delta C_{LWL}, \Delta C_{mWL}$	d_e, d_a, d_r, d_T	$H, M, p, q, \dot{p}, \dot{q}$
24	13	Right Wing Partially Missing	$\Delta C_{LWR}, \Delta C_{mWR}$	d_e, d_a, d_r, d_T	$H, M, p, q, \dot{p}, \dot{q}$
25	14	Left Horizontal Tail Partially Missing	ΔC_{LHL}	d_e, d_a	$H, M, p, q, \dot{p}, \dot{q}$
26	15	Right Horizontal Tail Partially Missing	ΔC_{LHR}	d_e, d_a	$H, M, p, q, \dot{p}, \dot{q}$
27	16	Left Vertical Tail Partially Missing	ΔC_{LVL}	d_a, d_r, d_T	$p, r, \dot{p}, \dot{r}$
28	17	Right Vertical Tail Partially Missing	ΔC_{LVR}	d_a, d_r, d_T	$p, r, \dot{p}, \dot{r}$
29	18	Left Engine Reduced Efficiency	T_L	d_T	a_x, a_z, r, H, M
30	19	Right Engine Reduced Efficiency	T_R	d_T	a_x, a_z, r, H, M

8.3.1. Locked Stabilator

For a locked stabilator failure, there is only one DIV:

$$\mathbb{V}_\delta = \begin{cases} \{\delta_{eL}\} & \text{for left stabilator failure,} \\ \{\delta_{eR}\} & \text{for right stabilator failure.} \end{cases} \tag{8.11}$$

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_e\} \quad \text{and} \quad \mathbb{V}_E = \{H, M, p, q, \dot{q}\}, \tag{8.12}$$

respectively.

In the following analysis, the elevator deflection is considered positive when it is deflected downward such that it produces negative pitching moment, as shown in Fig. 8.6.
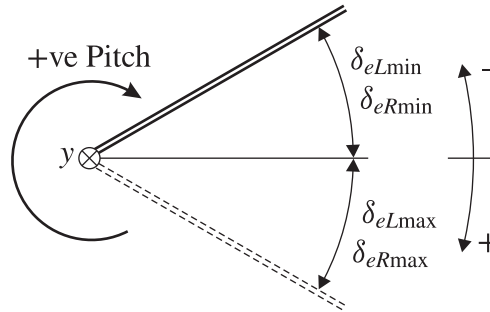
Figure 8.6: Sign Convention for Elevator Deflections (Left View)

The longitudinal stick displacement is assumed to be given by the relationship

$$d_e = k_e \delta_e, \tag{8.13}$$

where k_e is some constant and δ_e is the collective elevator deflection given by

$$\delta_e = \frac{\delta_{eL} + \delta_{eR}}{2}. \tag{8.14}$$

At normal flight conditions, the range of the longitudinal stick displacement is

$$\left[d_e\right] = \left[k_e \delta_{e\,\min}, \ k_e \delta_{e\,\max}\right]. \tag{8.15}$$

The post-failure range of d_e for a locked stabilator can be calculated from

$$\left[d_e\right]_F = \begin{cases} \left[k_e \dfrac{\delta_{eLF} + \delta_{eR\,\min}}{2}, \ k_e \dfrac{\delta_{eLF} + \delta_{eR\,\max}}{2}\right] & \text{for left stabilator failure,} \\[3mm] \left[k_e \dfrac{\delta_{eL\,\min} + \delta_{eRF}}{2}, \ k_e \dfrac{\delta_{eL\,\max} + \delta_{eRF}}{2}\right] & \text{for right stabilator failure.} \end{cases} \tag{8.16}$$

where δ_{eLF} is the value at which the left elevator is locked and δ_{eRF} is the value at which the right elevator is locked. These values can be obtained from the outcome of the ACDQE.

The post-failure range of d_e in Eq. (8.16) is virtual since the pilot is still capable of moving the stick to its extremities. If the left and right stabilators have identical δ_e ranges under normal conditions, then the post-failure range of the longitudinal stick displacement for the left stabilator would be the same as that for the right one. Note that $[d_e]_F \subset [d_e]$.

After determining $[d_e]_F$, the post-failure ranges for H, M, q, and $\dot{q}$ may be found from the corresponding 2-D projections of the self.

The asymmetry induced by a locked stabilator produces an adverse rolling. Ignoring the slight coupling between the lateral and longitudinal channels, the longitudinal stick displacement does not produce roll rate and, therefore, the d_e-p projection cannot be used to determine the post-failure range of the roll rate. The adverse rolling can be compensated by a non-zero aileron trim deflection δ_{aPF}. Considering that the aileron deflection is positive when the left aileron is deflected downward and the right one upward, such that it produces positive rolling moment as shown in Fig. 8.7, the compensating aileron deflection can be approximated by using the balance of the rolling moment

$$C_{\ell\delta_{eL}}\delta_{eLF} - C_{\ell\delta_{eR}}\delta_{eR} = -C_{\ell\delta_a}\delta_{aPF} \qquad \Rightarrow \qquad \delta_{aPF} = -\frac{C_{\ell\delta_{eL}}}{C_{\ell\delta_a}}\left(\delta_{eLF} - \delta_{eR}\right), \qquad (8.17)$$

if the locked stabilator is the left one, or

$$C_{\ell\delta_{eR}}\delta_{eRF} - C_{\ell\delta_{eL}}\delta_{eL} = -C_{\ell\delta_a}\delta_{aPF} \qquad \Rightarrow \qquad \delta_{aPF} = -\frac{C_{\ell\delta_{eR}}}{C_{\ell\delta_a}}\left(\delta_{eRF} - \delta_{eL}\right), \qquad (8.18)$$

if the locked stabilator is the right one.
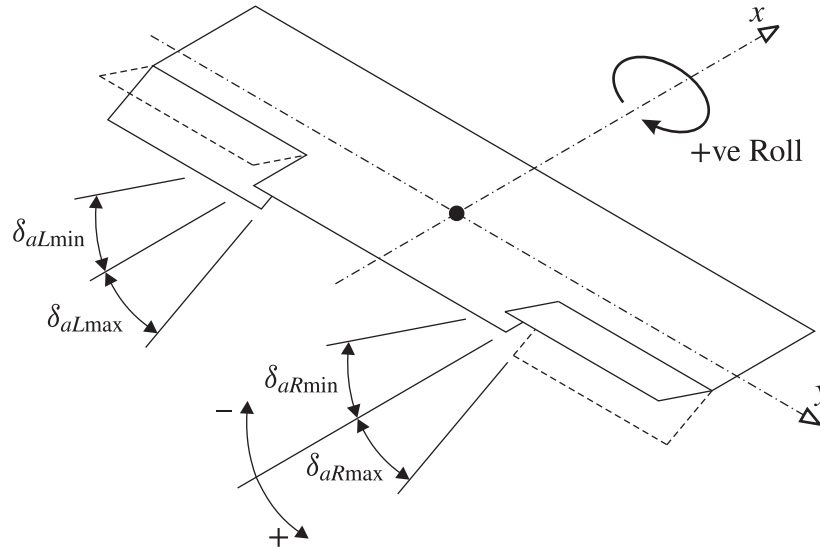
Figure 8.7: Sign Convention for Aileron Deflections

The relationship between the aileron deflection and the lateral stick displacement is given by

$$d_a = k_a\delta_a, \qquad (8.19)$$

where k_a is a constant and

$$\delta_a = \delta_{aL} - \delta_{aR}. \tag{8.20}$$

Under stabilator failure, the post-failure range of the lateral stick displacement can be determined from

$$\left[d_a\right]_F = \left[k_a\left(\delta_{a\min} - \delta_{aPF}\right),\ k_a\left(\delta_{a\max} - \delta_{aPF}\right)\right], \tag{8.21}$$

where

$$\delta_{a\min} = \delta_{aL\min} - \delta_{aR\max} \quad \text{and} \quad \delta_{a\max} = \delta_{aL\max} - \delta_{aR\min}. \tag{8.22}$$

Note that this is equivalent to shifting the nominal range of d_a in one direction (i.e., $\left[d_a\right]_F \not\subset \left[d_a\right]$), depending on the sign of the stabilator failure, while maintaining the same span (i.e., $d_{a\max F} - d_{a\min F} = d_{a\max} - d_{a\min}$).

Once the post-failure range of d_a is known, the d_a-p projection can be used to determine post-failure range of p. Note that since $C_{\ell\delta_{eL}} = -C_{\ell\delta_{eR}}$, the aileron pseudo-failure range for the left stabilator failure is the same as that of the right stabilator failure if the right stabilator is locked in the opposite direction relative to the trim position.

8.3.2. Locked Aileron

In the case of a locked aileron failure, there is only one DIV:

$$\mathbb{V}_\delta = \begin{cases} \{\delta_{aL}\} & \text{for left aileron failure,} \\ \{\delta_{aR}\} & \text{for right aileron failure.} \end{cases} \tag{8.23}$$

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_a\} \quad \text{and} \quad \mathbb{V}_E = \{p, \dot{p}\}, \tag{8.24}$$

respectively.

The lateral stick displacement is given by the relationship

$$d_a = k_a\delta_a, \tag{8.25}$$

where k_a is some constant and δ_a is the differential aileron deflection given by

$$\delta_a = \delta_{aL} - \delta_{aR}. \tag{8.26}$$

At normal flight conditions, the range of the lateral stick displacement is

$$\left[d_a\right] = \left[k_a\delta_{a\min},\ k_a\delta_{a\max}\right]. \tag{8.27}$$

The post-failure range of d_a for a locked aileron can be calculated from

$$\left[d_a\right]_F = \begin{cases} \left[k_a\left(\delta_{aLF} - \delta_{aR\max}\right),\ k_a\left(\delta_{aLF} - \delta_{aR\min}\right)\right] & \text{for left aileron failure,} \\ \left[k_a\left(\delta_{aL\min} - \delta_{aLF}\right),\ k_a\left(\delta_{aL\max} - \delta_{aLF}\right)\right] & \text{for right aileron failure,} \end{cases} \tag{8.28}$$

where δ_{aLF} is the value at which the left aileron is locked and δ_{aRF} is the value at which the right aileron is locked. These values can be obtained from the outcome of the ACDQE.

Similar to the stabilator failure, the post-failure range of d_a in Eq. (8.28) is virtual since the pilot is still capable of moving the stick to its extremities. Note that this range is the same if the opposite control surface is locked in the opposite direction. Also note that $[d_a]_F \subset [d_a]$ (i.e., the range of d_a is reduced after the failure).

It should be mentioned here that a locked aileron produces adverse yawing. However, simulations have shown that this undesirable yawing is negligibly small and, therefore, it was excluded from this analysis.

Once the post-failure range of d_a is known, the post-failure ranges for the ERVs may be found from the corresponding 2-D projections of the self.

8.3.3. Locked Rudder

For a locked rudder, there is only one DIV:

$$\mathbb{V}_\delta = \begin{cases} \{\delta_{rL}\} & \text{for left rudder failure,} \\ \{\delta_{rR}\} & \text{for right rudder failure.} \end{cases} \tag{8.29}$$

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_r\} \quad \text{and} \quad \mathbb{V}_E = \{r, \dot{r}\}, \tag{8.30}$$

respectively.

In the following analysis, the rudder deflection is considered positive when it is deflected to the right of the pilot such that it produces positive yawing moment, as shown in Fig. 8.8.
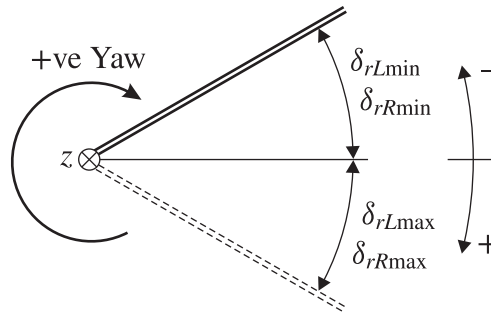
Figure 8.8: Sign Convention for Rudder Deflections (Top View)

The pedal displacement is given by the relationship

$$d_r = k_r \delta_r, \tag{8.31}$$

where k_r is some constant and δ_r is the collective rudder deflection given by

$$\delta_r = \frac{\delta_{rL} + \delta_{rR}}{2}. \tag{8.32}$$

At normal flight conditions, the range of the pedal displacement is

$$\left[d_r\right] = \left[k_r \delta_{r\min}, \ k_r \delta_{r\max}\right].$$ (8.33)

The post-failure range of d_r for a locked rudder can be calculated from

$$\left[d_r\right]_F = \begin{cases} \left[k_r \dfrac{\delta_{rLF} + \delta_{aR\min}}{2}, \ k_r \dfrac{\delta_{rLF} + \delta_{aR\max}}{2}\right] & \text{for left rudder failure,} \\[3ex] \left[k_r \dfrac{\delta_{aL\min} + \delta_{rRF}}{2}, \ k_r \dfrac{\delta_{aL\max} + \delta_{rRF}}{2}\right] & \text{for right rudder failure,} \end{cases}$$ (8.34)

where δ_{rLF} is the value at which the left rudder is locked and δ_{rRF} is the value at which the right rudder is locked. These values can be obtained from the outcome of the ACDQE.

The post-failure range of d_r in Eq. (8.34) is virtual since the pilot is still capable of moving the pedals to their extremities. Note that $\left[d_r\right]_F \subset \left[d_r\right]$. Also note that, because of its collective deflection, the locked rudder failure is similar to locked stabilator failure in many respects, if the aircraft is equipped with dual rudder, as is the case of the F-15 aircraft.

It should be mentioned here that a locked rudder produces adverse rolling. However, simulations have shown that this undesirable rolling is negligibly small and, therefore, it was excluded from this analysis.

Once the post-failure range of d_r is known, the post-failure ranges for the ERVs may be found from the corresponding 2-D projections of the self.

8.3.4. Locked Throttle

For a locked throttle failure, there is only one DIV:

$$\mathbb{V}_\delta = \begin{cases} \{\delta_{TL}\} & \text{for left throttle failure,} \\ \{\delta_{TR}\} & \text{for right throttle failure.} \end{cases}$$ (8.35)

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_T\} \quad \text{and} \quad \mathbb{V}_E = \{a_x, \ a_z, \ r, \ H, \ M\},$$ (8.36)

respectively.

The throttle lever displacement is given by the relationship

$$d_T = k_T \delta_T,$$ (8.37)

where k_T is some constant and δ_T is the collective throttle given by

$$\delta_T = \frac{\delta_{TL} + \delta_{TR}}{2}.$$ (8.38)

At normal flight conditions, the range of the throttle lever displacement is

$$\left[d_T\right] = \left[k_T \delta_{T\min}, \ k_T \delta_{T\max}\right] = \left[0\%, \ 100\%\right].$$ (8.39)

After failure, the range of d_T becomes

$$[d_T]_F = \begin{cases} \left[k_T \dfrac{\delta_{TLF} + \delta_{TR\min}}{2}, \ k_T \dfrac{\delta_{TLF} + \delta_{TR\max}}{2} \right] & \text{for left throttle failure,} \\[4mm] \left[k_T \dfrac{\delta_{TL\min} + \delta_{TRF}}{2}, \ k_T \dfrac{\delta_{TL\max} + \delta_{TRF}}{2} \right] & \text{for right throttle failure,} \end{cases} \tag{8.40}$$

where δ_{TLF} is the value at which the left throttle is locked and δ_{TRF} is the value at which the right throttle is locked. These values can be obtained from the outcome of the ACDQE.

The post-failure range of d_T in Eq. (8.40) is virtual since the pilot is still capable of moving the throttle lever to its extremities. If the left and right throttle have identical δ_T ranges under normal conditions, then the post-failure range of the throttle lever displacement for the left throttle would be the same as that for the right one. Note that $[d_T]_F \subset [d_T]$.

After determining $[d_T]_F$, the post-failure ranges for a_x, a_z, H, and M can be found from the corresponding 2-D projections of the self.

The asymmetry induced by a locked throttle produces an adverse yawing (depending on the distance between the engine and the centerline of the aircraft). Since the throttle lever displacement does not produce yaw rate, therefore, the d_T-r projection cannot be used to determine the post-failure range of the yaw rate. The adverse yawing can be compensated by a shift of the trim position of the rudder at δ_{rPF}. Considering the sign convention shown in Fig. 8.8, the compensating rudder deflection can be approximated by using the balance of the yawing moment

$$C_{n\delta_{TL}} \left(\delta_{TLF} - \delta_{TR} \right) = -C_{n\delta_r} \delta_{rPF} \qquad \Rightarrow \qquad \delta_{rPF} = -\frac{C_{n\delta_{TL}}}{C_{n\delta_r}} \left(\delta_{TLF} - \delta_{TR} \right), \tag{8.41}$$

if the locked throttle is the left one or

$$C_{n\delta_{TR}} \left(\delta_{TL} - \delta_{TRF} \right) = C_{n\delta_r} \delta_{rPF} \qquad \Rightarrow \qquad \delta_{rPF} = \frac{C_{n\delta_{TR}}}{C_{n\delta_r}} \left(\delta_{TL} - \delta_{TRF} \right), \tag{8.42}$$

if the locked throttle is the right one. Note that $C_{n\delta_{TL}} > 0$, $C_{n\delta_{TR}} < 0$, and $C_{n\delta_r} < 0$.

Considering that $\delta_{TLF} \in [0, 100]$, the compensating rudder deflection for the left throttle failure can be expressed as, by Eq. (8.41),

$$\delta_{rPF} \in \left[\frac{C_{n\delta_{TL}}}{C_{n\delta_r}} \delta_{TR}, \ -\frac{C_{n\delta_{TL}}}{C_{n\delta_r}} \left(100 - \delta_{TR} \right) \right]. \tag{8.43}$$

For full range of δ_{TR}, the compensating rudder deflection for left throttle failure becomes

$$\delta_{rPF} \in \left[\frac{C_{n\delta_{TL}}}{C_{n\delta_r}} 100, \ -\frac{C_{n\delta_{TL}}}{C_{n\delta_r}} 100 \right]. \tag{8.44}$$

Similarly, considering that $\delta_{TRF} \in [0, 100]$, the compensating rudder deflection in the case of right throttle failure can be expressed as, by Eq. (8.42),

$$\delta_{rPF} \in \left[\frac{C_{n\delta_{TR}}}{C_{n\delta_r}} \delta_{TL}, \ \frac{C_{n\delta_{TR}}}{C_{n\delta_r}} \left(\delta_{TL} - 100 \right) \right]. \tag{8.45}$$

For full range of δ_{TL}, the compensating rudder deflection for right throttle failure becomes

$$\delta_{rPF} \in \left[-\frac{C_{n\delta_{TR}}}{C_{n\delta_r}} 100, \ \frac{C_{n\delta_{TR}}}{C_{n\delta_r}} 100 \right].$$

(8.46)

The relationship between the rudder deflection and the pedal displacement is given by

$$d_r = k_r \delta_r,$$

(8.47)

where k_r is a constant and

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

(8.48)

Under throttle failure, the post-failure range of the pedal displacement can be determined from

$$\left[d_r \right]_F = \left[k_r \left(\delta_{r\min} - \delta_{rPF} \right), \ k_r \left(\delta_{r\max} - \delta_{rPF} \right) \right],$$

(8.49)

where

$$\delta_{r\min} = \frac{\delta_{rL\min} + \delta_{rR\min}}{2} \qquad \text{and} \qquad \delta_{r\max} = \frac{\delta_{rL\max} + \delta_{rR\max}}{2}.$$

(8.50)

Once the post-failure range of d_r is known, the d_r-r projection can be used to determine the post-failure range of r. Note that this is equivalent to shifting the nominal range of d_r in one direction (i.e., $\left[d_r \right]_F \not\subset \left[d_r \right]$), depending on the sign of the locked throttle, while maintaining the same span (i.e., $d_{r\max F} - d_{r\min F} = d_{r\max} - d_{r\min}$). To determine the direction in which the nominal range of d_r is shifted at post-failure, let's consider the left throttle failure first. Substituting Eq. (8.41) into Eq. (8.49) gives the following minimum and maximum values of $\left[d_r \right]_F$, respectively:

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

(8.51)

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

(8.52)

Considering the full range of δ_{TR}, the minimum value of $d_{r\min F}$ occurs at $\delta_{TR} = 0$ whereas $d_{r\max F}$ reaches its maximum value at $\delta_{TR} = 100$, that is,

$$\min \left(d_{r\min F} \right) = k_r \left[\delta_{r\min} + \frac{C_{n\delta_{TL}}}{C_{n\delta_r}} \delta_{TLF} \right],$$

(8.53a)

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

(8.53b)

For the right throttle failure, substituting Eq. (8.42) into Eq. (8.49) gives the following minimum and maximum values of $\left[d_r \right]_F$, respectively:

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

(8.54)

$$d_{r\max F} = k_r \left[\delta_{r\max} - \frac{C_{n\delta_{TR}}}{C_{n\delta_r}} \left(\delta_{TL} - \delta_{TRF} \right) \right].$$

(8.55)

Considering the full range of δ_{TL}, the minimum value of $d_{r\min F}$ in this case occurs at $\delta_{TL} = 100$ and the maximum value of $d_{r\max F}$ occurs at $\delta_{TL} = 0$, that is,

$$\min\left(d_{r\min F}\right) = k_r \left[\delta_{r\min} + \frac{C_{n\delta_{TR}}}{C_{n\delta_r}}\left(\delta_{TRF} - 100\right)\right], \tag{8.56a}$$

$$\max\left(d_{r\max F}\right) = k_r \left[\delta_{r\max} + \frac{C_{n\delta_{TR}}}{C_{n\delta_r}}\delta_{TRF}\right]. \tag{8.56b}$$

Note that $C_{n\delta_{TL}} = -C_{n\delta_{TR}}$. Therefore, $\left[d_r\right]_F$ shifts towards positive deflections for a left throttle failure and towards negative deflections for a right throttle failure.

8.3.5. Roll Rate Sensor Faults

For the roll rate sensor, the sets of DIVs, EDIVs, and ERVs are

$$\mathbb{V}_\delta = \{p_{meas}\}, \quad \mathbb{V}_\varepsilon = \{d_a\}, \quad \text{and} \quad \mathbb{V}_E = \{p, \dot{p}\}, \tag{8.57}$$

respectively.

At normal flight conditions, the relationship between the reference roll rate and the lateral stick displacement can be expressed as

$$p_{ref} = f_1\left(d_a\right). \tag{8.58}$$

Assuming error-free measurements, the measured roll rate is equal to the actual value, i.e.,

$$p_{meas} = p_{act}, \tag{8.59}$$

where p_{meas} is the measured roll rate and p_{act} is the actual roll rate.

The commanded roll rate is given by

$$p_{cmd} = f_2\left(p_{ref} - p_{act}\right). \tag{8.60}$$

For a faulty roll rate sensor, the measured and commanded roll rates become

$$p_{meas, F} = \begin{cases} p_{act} + p_{bias} & \text{for } p_{bias} \text{ in the roll rate output,} \\ p_{const} & \text{for } p_{const} \text{ roll rate output,} \end{cases} \tag{8.61}$$

and

$$p_{cmd, F} = \begin{cases} f_2\left(p_{ref} - p_{act} - p_{bias}\right) & \text{for } p_{bias} \text{ in the roll rate output,} \\ f_2\left(p_{ref} - p_{const}\right) & \text{for } p_{const} \text{ roll rate output,} \end{cases} \tag{8.62}$$

respectively.

Consequently, the faulty sensor produces an additional roll rate command given by

$$\Delta p_{cmd} = p_{cmd, F} - p_{cmd} = \begin{cases} -p_{bias} & \text{for } p_{bias} \text{ in the roll rate output,} \\ p_{ref} - p_{const} & \text{for } p_{const} \text{ roll rate output.} \end{cases} \tag{8.63}$$

Note that the sign of the resulting roll of the aircraft would be opposite to that of the bias or constant output of the sensor. This undesirable effect can be eliminated by deflecting the aileron by a certain amount which will, eventually, restrict the excursion of the lateral stick displacement by an amount equal to

$$
d_{aF} = \begin{cases} f_1^{-1}\left(p_{bias}\right) & \text{for } p_{bias} \text{ in the roll rate output,} \\ f_1^{-1}\left(p_{const}\right) & \text{for } p_{const} \text{ roll rate output.} \end{cases} \tag{8.64}
$$

Therefore, the post-failure range of the lateral stick displacement can be found from

$$
\left[d_a\right]_F = \left[d_{a\min} - d_{aF}, \; d_{a\max} - d_{aF}\right]. \tag{8.65}
$$

Using this post-failure range of d_a, the post-failure ranges for all the ERVs can be obtained from the corresponding 2-D projections of the self.

8.3.6. Pitch Rate Sensor Faults

For the pitch rate sensor, the sets of DIVs, EDIVs, and ERVs are

$$
\mathbb{V}_\delta = \{q_{meas}\}, \quad \mathbb{V}_\varepsilon = \{d_e\}, \quad \text{and} \quad \mathbb{V}_E = \{q, \dot{q}\}, \tag{8.66}
$$

respectively.

At normal flight conditions, the relationship between the reference pitch rate and the longitudinal stick displacement can be expressed as

$$
q_{ref} = f_3\left(d_e\right). \tag{8.67}
$$

Assuming error-free measurements, the measured pitch rate is equal to the actual value, i.e.,

$$
q_{meas} = q_{act}, \tag{8.68}
$$

where q_{meas} is the measured pitch rate and q_{act} is the actual pitch rate.

The commanded pitch rate is given by

$$
q_{cmd} = f_4\left(q_{ref} - q_{act}\right). \tag{8.69}
$$

For a faulty pitch rate sensor, the measured and commanded pitch rates become

$$
q_{meas,\,F} = \begin{cases} q_{act} + q_{bias} & \text{for } q_{bias} \text{ in the pitch rate output,} \\ q_{const} & \text{for } q_{const} \text{ pitch rate output,} \end{cases} \tag{8.70}
$$

and

$$
q_{cmd,\,F} = \begin{cases} f_4\left(q_{ref} - q_{act} - q_{bias}\right) & \text{for } q_{bias} \text{ in the pitch rate output,} \\ f_4\left(q_{ref} - q_{const}\right) & \text{for } q_{const} \text{ pitch rate output,} \end{cases} \tag{8.71}
$$

respectively.

Consequently, the faulty sensor produces an additional pitch rate command given by

$$\Delta q_{cmd} = q_{cmd,F} - q_{cmd} = \begin{cases} -q_{bias} & \text{for } q_{bias} \text{ in the pitch rate output,} \\ q_{ref} - q_{const} & \text{for } q_{const} \text{ pitch rate output.} \end{cases} \tag{8.72}$$

Note that the sign of the resulting pitch of the aircraft is opposite to that of the bias or constant output of the sensor. This undesirable effect can be eliminated by deflecting the elevator by a certain amount which will, eventually, restrict the excursion of the longitudinal stick displacement by an amount equal to

$$d_{eF} = \begin{cases} f_3^{-1}\left(q_{bias}\right) & \text{for } q_{bias} \text{ in the pitch rate output,} \\ f_3^{-1}\left(q_{const}\right) & \text{for } q_{const} \text{ pitch rate output.} \end{cases} \tag{8.73}$$

Therefore, the post-failure range of the longitudinal stick displacement can be found from

$$\left[d_e\right]_F = \left[d_{e\min} - d_{eF}, \ d_{e\max} - d_{eF}\right]. \tag{8.74}$$

Using this post-failure range of d_e, the post-failure ranges for all the ERVs can be obtained from the corresponding 2-D projections of the self.

8.3.7. Yaw Rate Sensor Faults

For the yaw rate sensor, the sets of DIVs, EDIVs, and ERVs are

$$\mathbb{V}_\delta = \{r_{meas}\}, \quad \mathbb{V}_\varepsilon = \{d_r\}, \quad \text{and} \quad \mathbb{V}_E = \{r, \dot{r}, \beta\}, \tag{8.75}$$

respectively.

At normal flight conditions, the relationship between the reference yaw rate and the pedal displacement can be expressed as

$$r_{ref} = f_5\left(d_r\right). \tag{8.76}$$

Assuming error-free measurements, the measured yaw rate is equal to the actual value, i.e.,

$$r_{meas} = r_{act}, \tag{8.77}$$

where r_{meas} is the measured yaw rate and r_{act} is the actual yaw rate.

The commanded yaw rate is given by

$$r_{cmd} = f_6\left(r_{ref} - r_{act}\right). \tag{8.78}$$

For a faulty yaw rate sensor, the measured and commanded yaw rates become

$$r_{meas,F} = \begin{cases} r_{act} + r_{bias} & \text{for } r_{bias} \text{ in the yaw rate output,} \\ r_{const} & \text{for } r_{const} \text{ yaw rate output,} \end{cases} \tag{8.79}$$

and

$$r_{cmd,F} = \begin{cases} f_6\left(r_{ref} - r_{act} - r_{bias}\right) & \text{for } r_{bias} \text{ in the yaw rate output,} \\ f_6\left(r_{ref} - r_{const}\right) & \text{for } r_{const} \text{ yaw rate output,} \end{cases} \tag{8.80}$$

respectively.

Consequently, the faulty sensor produces an additional yaw rate command given by

$$\Delta r_{cmd} = r_{cmd, F} - r_{cmd} = \begin{cases} -r_{bias} & \text{for bias in the yaw rate of } r_{bias}, \\ r_{ref} - r_{const} & \text{for } r_{const} \text{ yaw rate output.} \end{cases} \tag{8.81}$$

Note that the sign of the resulting yaw of the aircraft is opposite to that of the bias or constant output of the sensor. This undesirable effect can be eliminated by deflecting the rudder by a certain amount which will, eventually, restrict the excursion of the pedal displacement by an amount equal to

$$d_{rF} = \begin{cases} f_5^{-1}\left(r_{bias}\right) & \text{for } r_{bias} \text{ in the yaw rate output,} \\ f_5^{-1}\left(r_{const}\right) & \text{for } r_{const} \text{ yaw rate output.} \end{cases} \tag{8.82}$$

Therefore, the post-failure range of the pedal displacement can be found from

$$\left[d_r\right]_F = \left[d_{r\min} - d_{rF},\ d_{r\max} - d_{rF}\right]. \tag{8.83}$$

Using this post-failure range of d_r, the post-failure ranges for all the ERVs can be obtained from the corresponding 2-D projections of the self.

8.3.8. Wing Damage

For a damaged wing, the set of DIVs consists of two variables:

$$\mathbb{V}_\delta = \{\Delta C_{LWL},\ \Delta C_{mWL}\}, \tag{8.84}$$

where ΔC_{LWL} is the change in the lift moment and ΔC_{mWL} is the change in the pitching moment.

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_a,\ d_e,\ d_r,\ d_T\}, \tag{8.85}$$

and

$$\mathbb{V}_E = \{H,\ M,\ p,\ q,\ \dot{p},\ \dot{q}\}, \tag{8.86}$$

respectively.

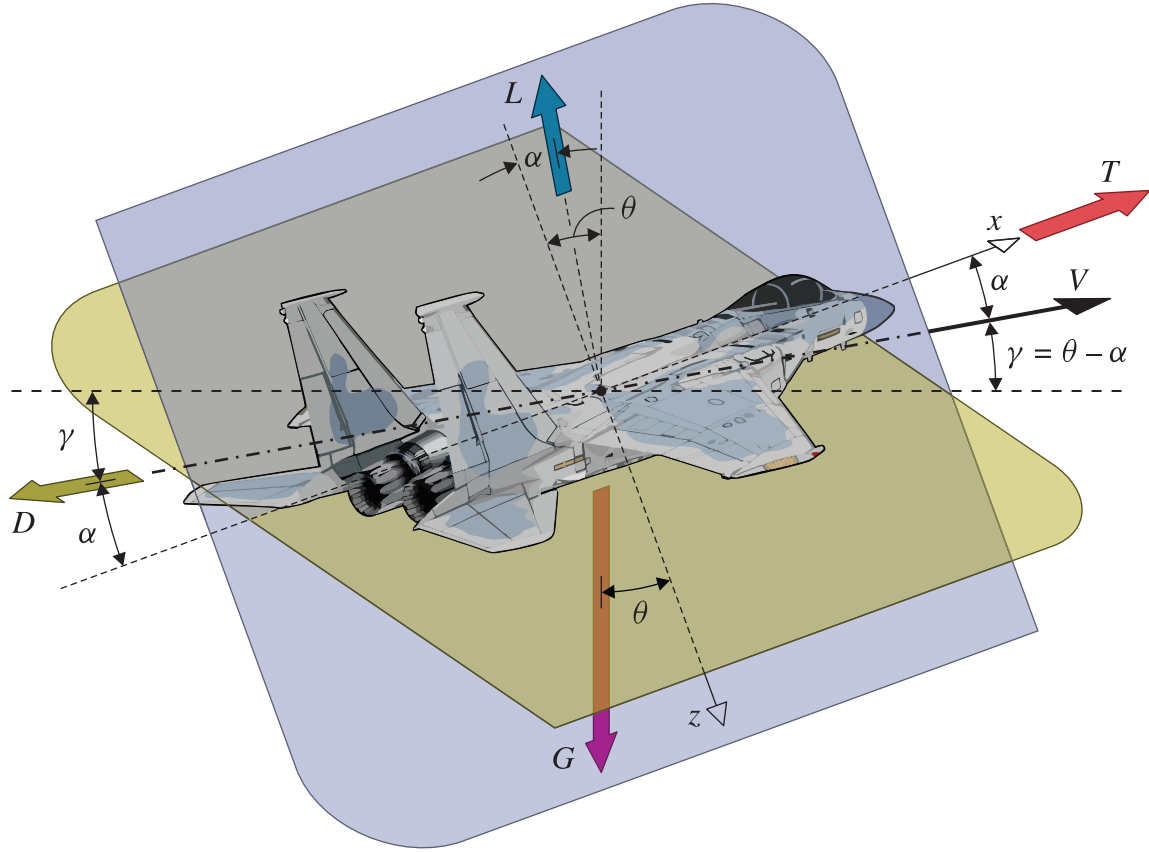Referring to Fig. 8.9, the aircraft is under equilibrium when it is flying at normal conditions, i.e.,

$$\sum F_x = 0: \quad T + L\sin\alpha - D\cos\alpha - G\sin\theta = 0, \tag{8.87a}$$

$$\sum F_z = 0: \quad -L\cos\alpha - D\sin\alpha + G\cos\theta = 0, \tag{8.87b}$$

$$\sum M_y = 0: \quad M_{WF} + M_{HT} = 0, \tag{8.87c}$$

$$C_L\left(M,\alpha\right) = C_L\left(C_D\left(M,\alpha\right)\right), \tag{8.87d}$$

$$C_{\ell WF} = 0. \tag{8.87e}$$

Figure 8.9: Aircraft Aerodynamic Forces

A damaged wing produces changes on the variables involved in the above equations. To maintain M, H, and γ when part of the wing is damaged, the following equations hold:

$$T + \Delta T + (L + \Delta L)\sin(\alpha + \Delta\alpha) - (D + \Delta D)\cos(\alpha + \Delta\alpha) - G\sin(\theta + \Delta\theta) = 0, \tag{8.88a}$$

$$-(L + \Delta L)\cos(\alpha + \Delta\alpha) - (D + \Delta D)\sin(\alpha + \Delta\alpha) + G\cos(\theta + \Delta\theta) = 0, \tag{8.88b}$$

$$M_{WF} + \Delta M_{WF} = -M_{HT} - \Delta M_{HT}, \tag{8.88c}$$

$$C_{LF} = C_{LF}(C_D), \tag{8.88d}$$

$$\Delta C_{\ell WF} = -\Delta C_{LWL}\ell_{AL} = -\Delta C_{LWR}\ell_{AR} \neq 0, \tag{8.88e}$$

where ℓ_{AL} is the distance to the plane of symmetry of the aerodynamic center of the left damaged wing and ℓ_{AR} is the distance to the plane of symmetry of the aerodynamic center of the right damaged wing. Note that $\ell_{AL} < 0$, while $\ell_{AR} > 0$.

A rolling moment is needed to compensate the non-zero rolling moment in Eq. (8.88e). This can be achieved from an aileron deflection, which will apparently reduces the authority of the aileron command. Similar analysis to the aileron pseudo-failure of the stabilator failure can be carried out here. The compensating aileron deflection can be approximated using the balance of the rolling moment

$$- \Delta C_{LWL} \ell_{AL} = -\varepsilon_a b C_{\ell \delta_a} \delta_{aPF} \qquad \Rightarrow \qquad \delta_{aPF} = \frac{\Delta C_{LWL} \ell_{AL}}{\varepsilon_a b C_{\ell \delta_a}}, \qquad (8.89)$$

where b is the wingspan and ε_a is an alteration factor which counts for how the left aileron is affected by the wing failure such that

$$\varepsilon_a = \begin{cases} 0.5 & \text{if the aileron is completely out of work,} \\ 1 & \text{if the aileron is not affected by the wing damage.} \end{cases} \qquad (8.90)$$

The post-failure range of the lateral stick displacement is given by Eq. (8.21). Note that, in this case, the direction of the shift the post-failure range 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 the wing damage is expected to decrease the lift. At this point, $[d_a]_F$ can be used to determine the post-failure ranges of p and $\dot{p}$ from the corresponding 2-D projections of the self.

Wing damage causes a reduction in the longitudinal control authority as well. Considering the pitching moment balance

$$\Delta M_{WF} = -\Delta M_{HT}, \qquad (8.91)$$

and

$$\Delta C_{mWL} = -C_{m\delta_e} \Delta \delta_{ePF} \qquad \Rightarrow \qquad \Delta \delta_{ePF} = -\frac{\Delta C_{mWL}}{C_{m\delta_e}}, \qquad (8.92)$$

the reduction in the longitudinal stick displacement at post-failure can be expressed as

$$[d_e]_F = \left[d_{e\min} - k_e \left(\Delta \delta_{ePF} + \delta_{etrim} \right), \ d_{e\max} - k_e \left(\Delta \delta_{ePF} + \delta_{etrim} \right) \right], \qquad (8.93)$$

with which the post-failure ranges for q and $\dot{q}$ can be obtained from the corresponding 2-D projections of the self.

To determine post failure-ranges for H and M, consider the following equations for the longitudinal force, the vertical force, and the polar:

$$T + \Delta T + (L + \Delta L) \sin \alpha \cos \Delta \alpha + (L + \Delta L) \cos \alpha \sin \Delta \alpha - (D + \Delta D) \cos \alpha \cos \Delta \alpha +$$
$$(D + \Delta D) \sin \alpha \sin \Delta \alpha - G \sin \theta \cos \Delta \theta - G \cos \theta \sin \Delta \theta = 0, \qquad (8.94a)$$

$$-(L + \Delta L) \cos \alpha \cos \Delta \alpha + (L + \Delta L) \sin \alpha \sin \Delta \alpha - (D + \Delta D) \sin \alpha \cos \Delta \alpha -$$
$$(D + \Delta D) \cos \alpha \sin \Delta \alpha + G \cos \theta \cos \Delta \theta - G \sin \theta \sin \Delta \theta = 0, \qquad (8.94b)$$

$$\Delta D = C_{LF}^{-1} (\Delta L). \qquad (8.94c)$$

In order to keep the same trajectory slope γ, we have to ensure that $\Delta\alpha = \Delta\theta$. Recalling that ΔL is known, if the polar $C_{LF}(C_D)$ is known, then we end up with three equations and three unknowns, namely, ΔT, ΔD, and $\Delta\alpha$. Solving for ΔT allows us to determine the new "virtual" ranges for $d_T = f(T)$ at post-failure and, hence, the post-failure ranges for M and H from the corresponding 2-D projections of the self.

8.3.9. Horizontal Tail Damage

In this case, the DIV is

$$
\mathbb{V}_\delta = \begin{cases} \{\Delta C_{L_{HL}}\} & \text{for left horizontal tail failure,} \\ \{\Delta C_{L_{HR}}\} & \text{for right horizontal tail failure.} \end{cases} \tag{8.95}
$$

The sets of EDIVs and ERVs are

$$
\mathbb{V}_\varepsilon = \{d_a, d_e\}, \tag{8.96}
$$

and

$$
\mathbb{V}_E = \{H, M, p, q, \dot{p}, \dot{q}\}, \tag{8.97}
$$

respectively.

In this analysis, it is assumed that the stabilator corresponding to the damaged side of the horizontal tail can be still deflected; however, the respective control derivative may be affected, such that

$$
\Delta C_{L_{HL}} = \Delta C_{L_{HL0}} + \Delta C_{L\delta_e}\delta_e, \tag{8.98}
$$

for left horizontal tail and

$$
\Delta C_{L_{HR}} = \Delta C_{L_{HR0}} + \Delta C_{L\delta_e}\delta_e, \tag{8.99}
$$

for right horizontal tail.

To represent the reduction of the damaged horizontal tail capability in producing lift, the alterations $\Delta C_{L_{HL0}}$, $\Delta C_{L_{HR0}}$, and $\Delta C_{L\delta_e}$ are assumed negative. The additional elevator deflection necessary to compensate for the reduction in lift can be found from the longitudinal balance as

$$
-\Delta C_{L_{HL}} = \left(\frac{C_{L\delta_e}}{2} - \Delta C_{L\delta_e}\right)\Delta\delta_e + \frac{C_{L\delta_e}}{2}\Delta\delta_e \quad \Rightarrow \quad \Delta\delta_e = -\frac{\Delta C_{L_{HL}}}{C_{L\delta_e} - \Delta C_{L\delta_e}}. \tag{8.100}
$$

With d denoting the damage factor ($0 \le d \le 1$) which can be obtained from the outcome of the ACDQE, $\Delta C_{L_{HL}}$ can be estimated as

$$
\begin{aligned}
\Delta C_{L_{HL}} &= \left(\Delta C_{L_{HL}}\right)_{\text{before failure}} - \left(\Delta C_{L_{HL}}\right)_{\text{after failure}} \\
&= \left(\Delta C_{L_{HL}}\right)_{\text{before failure}} - (1-d)\left(\Delta C_{L_{HL}}\right)_{\text{before failure}} \\
&= \frac{d\,C_{L\delta_e}\delta_e}{2},
\end{aligned} \tag{8.101}
$$

where δ_e is given by Eq. (8.14).

Substituting Eq. (8.101) into Eq. (8.100) and simplifying, we have

$$\Delta \delta_e = -\frac{d}{2-d}\,\delta_e. \tag{8.102}$$

The post-failure range for the longitudinal stick displacement can be expressed as

$$\left[d_e\right]_F = \left[d_{e\,\min} - k_e \Delta \delta_e,\ d_{e\,\max} - k_e \Delta \delta_e\right]. \tag{8.103}$$

With this range, the post-failure ranges for H, M, q and $\dot{q}$ can now be found from the corresponding 2-D projections of the self.

Aileron deflection can be used to compensate for the rolling moment produced by the damaged horizontal tail. This can be found from the moments balance in the lateral direction as

$$\Delta C_{L_{HL}} \frac{\ell_{LHT}}{b} = C_{\ell \delta_a} \Delta \delta_a \qquad \Rightarrow \qquad \Delta \delta_a = \frac{\Delta C_{L_{HL}} \ell_{LHT}}{b C_{\ell \delta_a}}, \tag{8.104}$$

where b is the wing span and $\ell_{LHT} < 0$ is the distance from the left horizontal tail aerodynamic center to the plane of symmetry, which can be estimated from the following moment balance:

$$\frac{C_{\ell \delta_a}}{2} \Delta \delta_{eL} \ell_{LHT} = C_{\ell \delta_{eL}} \delta_{eL} \qquad \Rightarrow \qquad \ell_{LHT} = 2\frac{C_{\ell \delta_{eL}}}{C_{L \delta_e}}. \tag{8.105}$$

Substituting Eqs. (8.101) and (8.105) into Eq. (8.104) and simplifying, we get

$$\Delta \delta_a = \left(\frac{d}{b}\right) \left(\frac{C_{\ell \delta_{eL}}}{C_{\ell \delta_a}}\right) \delta_e, \tag{8.106}$$

Recall that $C_{\ell \delta_{eL}} = -C_{\ell \delta_{eR}} = C_{\ell \delta_e}$, therefore Eq. (8.101) becomes

$$\Delta \delta_a = \left(\frac{d}{b}\right) \left(\frac{C_{\ell \delta_e}}{C_{\ell \delta_a}}\right) \delta_e. \tag{8.107}$$

The post-failure range for the lateral stick displacement can now be according to

$$\left[d_a\right]_F = \left[d_{a\,\min} - k_a \Delta \delta_a,\ d_{a\,\max} - k_a \Delta \delta_a\right], \tag{8.108}$$

with which the post-failure ranges for p and $\dot{p}$ can now be found from the corresponding 2-D projections of the self.

8.3.10. Vertical Tail Damage

For a damaged vertical tail, the DIV is

$$\mathbb{V}_\delta = \begin{cases} \{\Delta C_{Y_{VL}}\} & \text{for left vertical tail failure,} \\ \{\Delta C_{Y_{VR}}\} & \text{for right vertical tail failure.} \end{cases} \tag{8.109}$$

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_a,\, d_r\}, \qquad \text{and} \qquad \mathbb{V}_E = \{p,\, r,\, \dot{p},\, \dot{r}\}, \tag{8.110}$$

respectively.

In this analysis, it is assumed that the rudder corresponding to the damaged side of the vertical tail can be still deflected; however, the respective control derivative may be affected, such that

$$\Delta C_{Y_{VL}} = \Delta C_{Y_{VL0}} + \Delta C_{Y\delta_r}\delta_r, \tag{8.111}$$

for left vertical tail and

$$\Delta C_{Y_{VR}} = \Delta C_{Y_{VR0}} + \Delta C_{Y\delta_r}\delta_r, \tag{8.112}$$

for right vertical tail.

The additional rudder deflection necessary to compensate for the reduction in lift can be found from the directional balance as

$$-\Delta C_{Y_{VL}} = \left(\frac{C_{Y\delta_r}}{2} - \Delta C_{Y\delta_r}\right)\Delta\delta_r + \frac{C_{Y\delta_r}}{2}\Delta\delta_r \qquad \Rightarrow \qquad \Delta\delta_r = -\frac{\Delta C_{Y_{VL}}}{C_{Y\delta_r} - \Delta C_{Y\delta_r}}. \tag{8.113}$$

The post-failure range for the pedal displacement can be expressed as

$$\left[d_r\right]_F = \left[d_{r\min} - k_r\Delta\delta_r, \ d_{r\max} - k_r\Delta\delta_r\right]. \tag{8.114}$$

With this range, the post-failure ranges for r and $\dot{r}$ can now be found from the corresponding 2-D projections of the self.

Aileron deflection can be used to compensate for the rolling moment produced by the damaged vertical tail. This can be found from the moments balance in the lateral direction as

$$\Delta C_{Y_{VL}}\frac{h_{VT}}{b} = C_{\ell\delta_a}\Delta\delta_a \qquad \Rightarrow \qquad \Delta\delta_a = \frac{\Delta C_{Y_{VL}}h_{VT}}{bC_{\ell\delta_a}}, \tag{8.115}$$

where b is the wing span and $h_{VT} < 0$ is the distance from the vertical tail aerodynamic center to the body horizontal plane.

The post-failure range for the lateral stick displacement can now be according to

$$\left[d_a\right]_F = \left[d_{a\min} - k_a\Delta\delta_a, \ d_{a\max} - k_a\Delta\delta_a\right], \tag{8.116}$$

with which the post-failure ranges for p and $\dot{p}$ can now be found from the corresponding 2-D projections of the self.

8.3.11. Engine Reduced Efficiency

In the case of engine failure, there is only one DIV:

$$\mathbb{V}_\delta = \begin{cases} \{T_L\} & \text{for left engine failure,} \\ \{T_R\} & \text{for right engine failure.} \end{cases} \tag{8.117}$$

The sets of EDIVs and ERVs are

$$\mathbb{V}_\varepsilon = \{d_T\} \quad \text{and} \quad \mathbb{V}_E = \{a_x, \ a_z, \ r, \ H, \ M\}, \tag{8.118}$$

respectively.

The throttle lever displacement is given by the relationship

$$d_T = f_{\varepsilon\delta 1}\left(T_L + T_R\right). \tag{8.119}$$

At normal flight conditions, the range of the throttle lever displacement is

$$\begin{aligned}
\left[d_T\right] &= \left[f_{\varepsilon\delta 1}\left(T_{L\min} + T_{R\min}\right),\; f_{\varepsilon\delta 1}\left(T_{L\max} + T_{R\max}\right)\right] \\
&= \left[f_{\varepsilon\delta 1}\left(2T_{\min}\right),\; f_{\varepsilon\delta 1}\left(2T_{\max}\right)\right] = \left[0\%,\; 100\%\right].
\end{aligned} \tag{8.120}$$

The post-failure range of d_T for can be calculated from

$$\left[d_T\right]_F = \begin{cases} \left[f_{\varepsilon\delta 1}\left(T_{L\min F} + T_{R\min}\right),\; f_{\varepsilon\delta 1}\left(T_{L\max F} + T_{R\max}\right)\right] & \text{for left engine failure,} \\[2mm] \left[f_{\varepsilon\delta 1}\left(T_{L\min} + T_{R\min F}\right),\; f_{\varepsilon\delta 1}\left(T_{L\max} + T_{R\max F}\right)\right] & \text{for right engine failure,} \end{cases} \tag{8.121}$$

where $T_{L\min F}$, $T_{L\max F}$, $T_{R\min F}$, and $T_{R\max F}$ are values that can be obtained from the outcome of the ACDQE.

The post-failure range of d_T in Eq. (8.121) is virtual since the pilot is still capable of moving the throttle lever to its extremities. If the left and right engine have identical thrust ranges under normal conditions, then the post-failure range of the throttle lever displacement for the left engine would be the same as that for the right one. Note that $\left[d_T\right]_F \subset \left[d_T\right]$.

After determining $\left[d_T\right]_F$, the post-failure ranges for a_x, a_z, H, and M can be found from the corresponding 2-D projections of the self.

The asymmetry induced by an engine failure produces an adverse yawing (depending on the distance between the engine and the centerline of the aircraft). Since the throttle lever displacement does not produce yaw rate, therefore, the d_T-r projection cannot be used to determine the post-failure range of the yaw rate. The adverse yawing can be compensated by a rudder deflection δ_{rPF} in a similar manner to that of the locked throttle case. However, in the case of engine failure, the pseudo-failure of the rudder consists of a locked rudder at variable deflection, depending on d_T. Assuming the total thrust and the throttle deflection vary linearly with the throttle lever displacement, the total yawing moment can be approximated as

$$C_{n\delta_{TL}}\delta_{TL} + C_{n\delta_{TR}}\delta_{TR} = 0 \qquad \Rightarrow \qquad \left(C_{n\delta_{TL}} + C_{n\delta_{TR}}\right)d_T = 0, \tag{8.122}$$

at normal operating conditions.

If the efficiency of the left engine is reduced, then Eq. (8.122) becomes

$$\left(C_{n\delta_{TLF}} + C_{n\delta_{TR}}\right)d_T \neq 0. \tag{8.123}$$

Let ℓ_{eL} be the distance from the aircraft center of gravity to the axis of the left engine and let ℓ_{eR} be the distance from the aircraft center of gravity to the axis of the right engine. For $\ell_{eL} = -\ell_{eR}$, we have

$$\left(C_{n\delta_{TLF}} + C_{n\delta_{TR}}\right)d_T = -\frac{\ell_{eL}\left(T_L - T_R\right)}{\frac{\rho}{2}sbV^2}, \tag{8.124}$$

where ρ is the air density, s is the surface area of the wing, and b is the wing span. Since $C_{n\delta_{TR}} d_T = \frac{\ell_{eL} T_R}{\frac{\rho}{2} s b V^2}$, therefore, Eq. (8.124) reduces to

$$C_{n\delta_{TLF}} d_T = -\frac{\ell_{eL} T_L}{\frac{\rho}{2} s b V^2}. \tag{8.125}$$

Assuming a linear relationship and $T_{L\min F} = 0$, we get

$$C_{n\delta_{TLF}} = -\frac{\ell_{eL} T_{L\max F}}{\frac{\rho}{2} s b V^2 d_{T\max}}. \tag{8.126}$$

Similar derivation can be carried out for the right engine failure. The compensating rudder deflection can be obtained from the balance of the yawing moment

$$\left(C_{n\delta_{TLF}} + C_{n\delta_{TR}} \right) d_T = -C_{n\delta_r} \delta_{rPF} \quad \Rightarrow \quad \delta_{rPF} = -\frac{C_{n\delta_{TLF}} + C_{n\delta_{TR}}}{C_{n\delta_r}} d_T, \tag{8.127}$$

if the left engine experiences a reduction in efficiency or from

$$\left(C_{n\delta_{TL}} + C_{n\delta_{TRF}} \right) d_T = -C_{n\delta_r} \delta_{rPF} \quad \Rightarrow \quad \delta_{rPF} = -\frac{C_{n\delta_{TL}} + C_{n\delta_{TRF}}}{C_{n\delta_r}} d_T, \tag{8.128}$$

if the right engine experiences a reduction in efficiency. In either case, the post-failure range of the pedal displacement can be determined from

$$\left[d_r \right]_F = \left[k_r \left(\delta_{r\min} - \delta_{rPF} \right), \ k_r \left(\delta_{r\max} - \delta_{rPF} \right) \right], \tag{8.129}$$

where

$$\delta_{r\min} = \frac{\delta_{rL\min} + \delta_{rR\min}}{2} \quad \text{and} \quad \delta_{r\max} = \frac{\delta_{rL\max} + \delta_{rR\max}}{2}. \tag{8.130}$$

Note that this is very similar to the locked throttle failure except that $\left[d_r \right]_F$ varies with the d_T in this case. Once the post-failure range of d_r is known, the d_r-r projection can be used to determine post-failure range of r.

Results and Discussion

9.1. Generation of the Self and Nonself

Features data were collected from experimental flight tests of a fighter research aircraft performed on the WVU motion-based flight simulator in two sets: development tests and validation tests. Eleven flight tests under nominal conditions covering a wide range of the flight envelope, as shown in Fig. 4.5, were used as development tests to generate the self and nonself using the RDSUM and CSUM presented in Chapter 5. Both methods were implemented in MATLAB® to output a set of 496 two-dimensional subselves (projections) from unique combinations of the aircraft features listed in Table 5.1. Figure 9.1 depicts sample 2-dimensional subself clusters along with the AIS antibodies (nonself) generated for a subself defined by the non-dimensional roll rate neural network output feature NN_p and the reference roll rate p_{ref} using the RDSUM. An example of the clusters and antibodies generated with the CSUM is presented in Fig. 9.2 for the same subself.
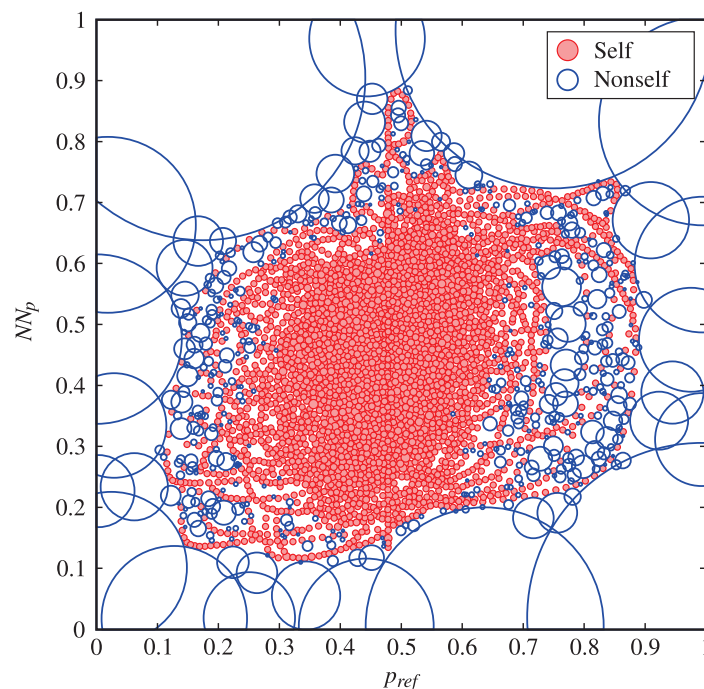
Figure 9.1: Sample 2-D Subself Clusters with AIS Antibodies Generated Using RDSUM
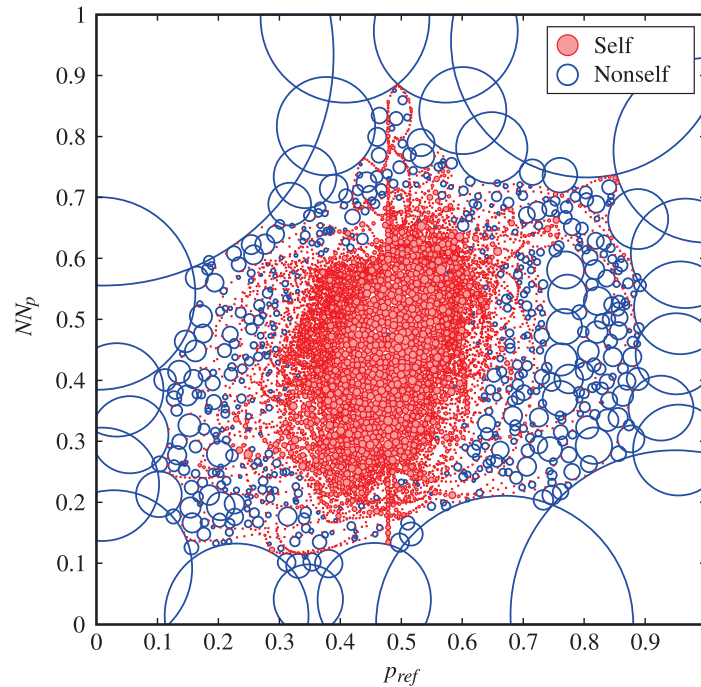
Figure 9.2: Sample 2-D Subself Clusters with AIS Antibodies Generated Using CSUM

Figures 9.3 and 9.4 illustrate the reduction of the subself clusters with the CSUM. Figure 9.3 shows the union of different sets of clusters generated from 11 single simulator test data before applying the fusion process. The union set has a total of 22,000 clusters. Figure 9.4 shows the reduced set of 6,343 clusters after the complete fusion process. This shows capability of the fusion algorithm of reducing 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.

Validation tests were used to analyze and compare the performance of the detectors generated with both RDSUM and CSUM in terms of detection rates (DR) and false alarms (FA). Figure 9.5 shows a typical 2-dimensional subself with validation test points for a nominal flight test. Validation test points for a flight test under abnormal conditions are shown in Fig. 9.6 using the same subself. Assuming typical binary outcomes, the results of the detection based on the self/nonself discrimination for each feature point can be categorized as

- True Positives, *TP*: the number of abnormal data points detected as abnormal.
- True Negatives, *TN*: the number of normal data points *not* detected as abnormal.
- False Positives, *FP*: the number of normal data points detected as abnormal.
- False Negatives, *FN*: the number of abnormal data points *not* detected as abnormal.

The *detection rate*, *DR*, is defined as the ratio of true positives to the total amount of abnormal data points, i.e.,

$$DR = \frac{TP}{TP + FN} \times 100. \tag{9.1}$$
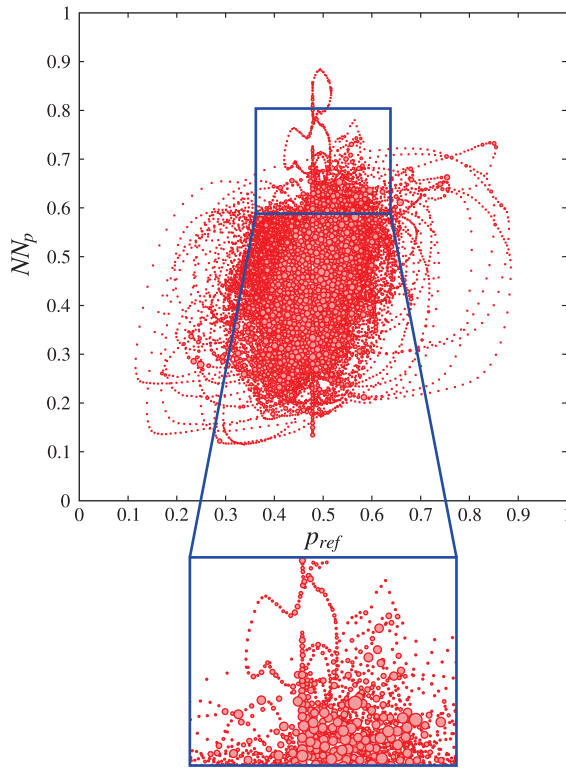
Figure 9.3: A 2-D Subself with 22,000 Clusters
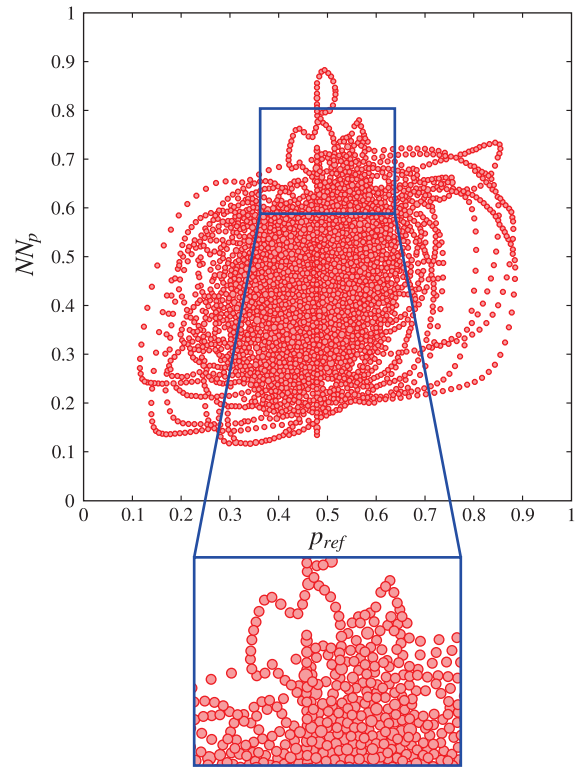
Figure 9.4: A 2-D Subself with 6,343 Clusters
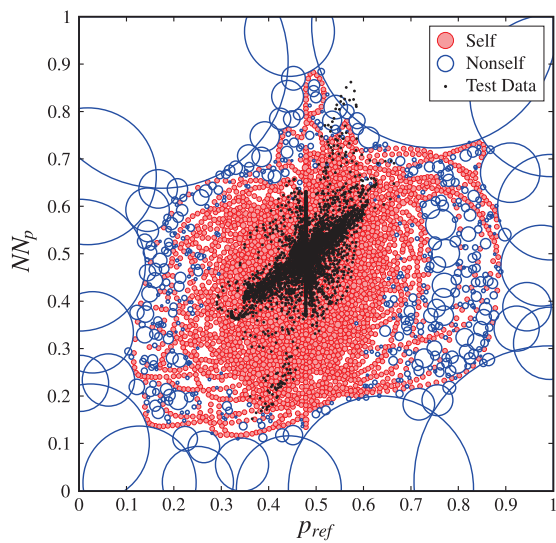
Figure 9.5: p_{ref}-NN_p Projection with Nominal Flight Test Data Points
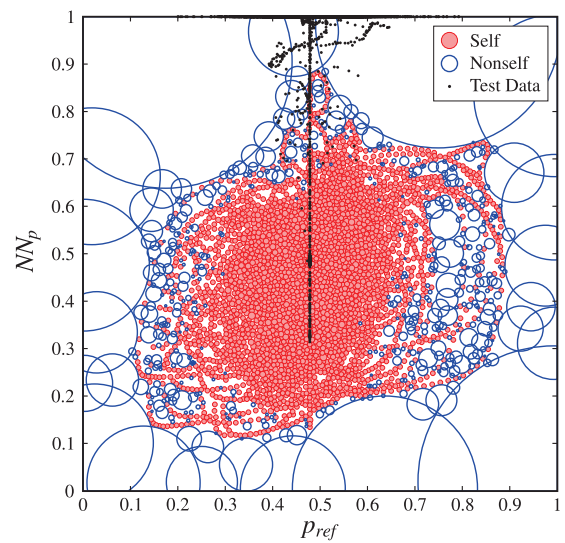
Figure 9.6: p_{ref}-NN_p Projection with Failure Flight Test Data Points

The *false alarm rate*, *FA*, is defined as the ratio of false positives to the total amount of normal data points, i.e.,

$$FA = \frac{FP}{TN + FP} \times 100. \tag{9.2}$$

Listed in Table 9.1 are selected subselves among the 496 two-dimensional subselves along with their corresponding features.

Table 9.1: Features of Selected 2-Dimensional Subselves

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

Table 9.2 provides a comparison between the detection performance of these subselves generated with RDSUM and the detection performance of the same subselves generated with CSUM under four different failures: left aileron locked at 2.5°, left stabilator locked at 2°, 6% loss of the left wing, and 5° bias in the pitch rate sensor output. The number of data points in each of these tests was about 60,000.

Table 9.2: Detection Performance of Selected 2-Dimensional Subselves

Subself #	Left Aileron Locked at 2.5°				Left Stabilator Locked at 2°				6% Loss of Left Wing				5° Bias in Pitch Rate Sensor			
	RDSUM		CSUM		RDSUM		CSUM		RDSUM		CSUM		RDSUM		CSUM	
	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA
3	81.2	0.0	82.0	1.0	99.3	0.0	99.3	1.85	98.8	0.0	99.0	1.1	0.08	0.0	2.27	1.0
30	82.1	0.0	83.5	0.0	99.2	0.0	99.2	0.0	99.4	0.0	99.5	0.0	2.96	0.0	4.16	0.0
42	0.0	0.0	0.0	0.0	5.33	0.0	8.22	0.0	0.01	0.0	0.1	0.0	99.9	0.0	99.9	0.0
53	16.0	0.0	22.9	0.0	23.1	0.0	24.8	0.0	20.8	0.0	25.1	0.0	85.6	0.0	86.7	0.0
82	90.2	0.0	92.3	0.0	99.7	0.0	99.7	0.0	99.1	0.0	99.2	0.0	3.87	0.0	4.63	0.0
83	86.3	0.0	88.1	0.0	99.5	0.0	99.6	0.0	98.8	0.0	99.0	0.0	0.08	0.0	0.22	0.0
84	85.7	0.0	86.2	0.0	99.5	0.0	99.5	0.0	98.9	0.0	99.0	0.0	0.01	0.0	0.05	0.0
85	85.4	0.0	88.7	0.0	99.4	0.0	99.4	0.0	98.9	0.0	98.9	0.0	25.1	0.0	26.5	0.0

<div align="center">Table 9.2 – Cont'd.</div>

Subself #	Left Aileron Locked at 2.5°				Left Stabilator Locked at 2°				6% Loss of Left Wing				5° Bias in Pitch Rate Sensor			
	RDSUM		CSUM		RDSUM		CSUM		RDSUM		CSUM		RDSUM		CSUM	
	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA
86	89.8	0.0	84.2	0.0	99.4	0.0	99.4	0.0	98.9	0.0	98.9	0.0	23.2	0.0	24.2	0.0
87	79.9	0.0	80.4	0.0	99.3	0.0	99.4	0.0	99.0	0.0	99.1	0.0	41.6	0.0	46.8	0.0
89	89.5	0.0	90.0	0.0	99.6	0.0	99.6	0.0	99.1	0.0	99.2	0.0	1.37	0.0	1.46	0.0
90	90.9	0.0	91.1	0.0	99.5	0.0	99.5	0.0	99.5	0.0	99.5	0.0	0.28	0.0	0.68	0.0
92	81.3	0.0	80.8	0.0	99.3	0.0	99.3	0.0	99.0	0.0	99.0	0.0	0.06	0.0	0.09	0.0
93	81.5	0.0	82.9	1.25	99.4	0.0	99.4	1.55	99.3	0.0	99.4	0.0	0.48	0.0	1.3	0.0
94	85.1	0.0	85.8	2.25	99.4	0.0	99.5	2.25	98.9	0.0	99.0	2.4	0.00	0.0	0.51	2.25
95	91.0	0.0	90.9	0.0	99.5	0.0	99.6	0.0	99.1	0.0	99.1	0.0	0.23	0.0	0.87	0.0
96	89.0	0.0	89.5	0.0	99.5	0.0	99.5	0.0	99.2	0.0	99.3	0.0	2.33	0.0	3.18	0.0
97	61.1	0.0	86.3	0.0	50.2	0.0	99.5	0.0	99.0	0.0	99.0	0.0	0.02	0.0	0.08	0.0
98	93.2	0.0	94.6	0.0	99.6	0.0	99.6	0.0	99.1	0.0	99.2	0.0	10.0	0.0	11.2	0.0
99	85.1	0.0	86.0	0.0	99.4	0.0	99.4	0.0	99.1	0.0	99.2	0.0	2.59	0.0	2.98	0.0
100	86.2	0.0	86.9	0.0	99.5	0.0	99.0	0.0	99.0	0.0	97.7	0.0	0.01	0.0	0.16	0.0
105	90.4	0.0	91.1	0.0	99.5	0.0	99.6	0.0	98.9	0.0	99.0	0.0	6.26	0.0	12.2	0.0
106	92.7	0.0	93.1	0.0	99.5	0.0	99.5	0.0	99.0	0.0	99.1	0.0	6.31	0.0	6.76	0.0
224	0.0	0.0	0.36	0.0	1.26	0.0	1.85	0.0	0.66	0.0	0.9	0.0	60.2	0.0	63.7	0.0
233	4.03	0.0	5.51	0.0	6.06	0.0	8.04	0.0	10.3	0.0	12.4	0.0	85.7	0.0	87.8	0.0
410	91.5	0.0	81.4	0.0	99.6	0.0	99.4	0.0	99.3	0.0	99.1	0.0	3.23	0.0	0.06	0.0
441	85.0	0.0	80.9	0.0	99.6	0.0	99.4	0.0	99.4	0.0	99.1	0.0	2.88	0.0	0.64	0.0
471	83.1	0.0	80.5	0.0	99.4	0.0	99.3	0.0	99.0	0.0	98.9	0.0	0.05	0.0	0.05	0.0

These results show that different selves favor the detection of the same particular types of failures for both RDSUM and CSUM methodologies. For instance, selves #82, #96, and #410, among others, show poor detection capabilities for the sensor failure, but an acceptable one for actuator and structural failures with low false alarms. On the other hand, selves #42, #53, and #233 present good performance for sensor failure but low detection rate for the other three abnormal conditions. Note that every self-case presents acceptable performance for at least one of the failures considered. The fact that different selves favor the detection of particular types of failures has been used by [79]. to develop an integrated scheme where different self configurations ensure overall high detection rate and low number of false alarms. The mentioned mechanism uses lower order projections to build subselves instead of using one single higher dimensional hyper-space and makes use of a specific hierarchy of feature relevance with respect to each type of failure.

Figures 9.7 through 9.10 present a direct comparison between the two methodologies, RDSUM and CSUM, for the four aforementioned abnormal conditions for some of the selves outlined in Table 9.1. For left aileron failure, as shown in Fig. 9.7, the CSUM performed slightly better with respect to RDSUM for most of the selves. The same characteristic is present in the other three failures as shown in Figs. 9.8, 9.9, and 9.10. Figure 9.11 shows a comparison between the two methodologies for one single self (self #87) where both RDSUM and CSUM have almost the same performance pattern with acceptable detection rate for all the failures except the
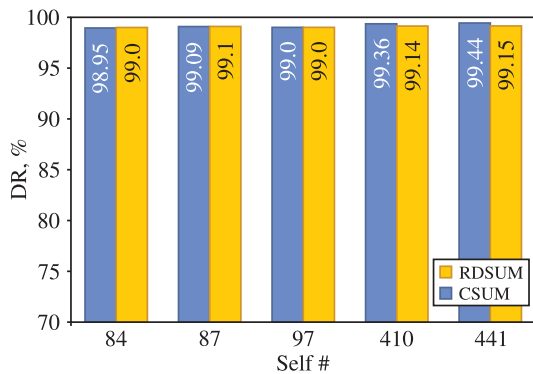
abnormal condition of the pitch rate sensor. This consistent trend has been noticed for all other subselves/projections. In general, the percentage detection rate and false alarms remain around the same values for both methodologies without significant differences.   This fact allows concluding that, in terms of detection performance, both RDSUM and CSUM are equivalent.
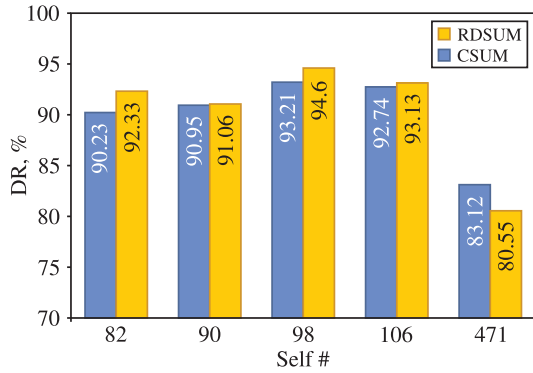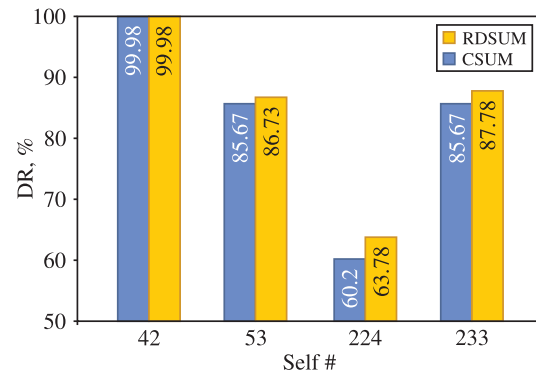

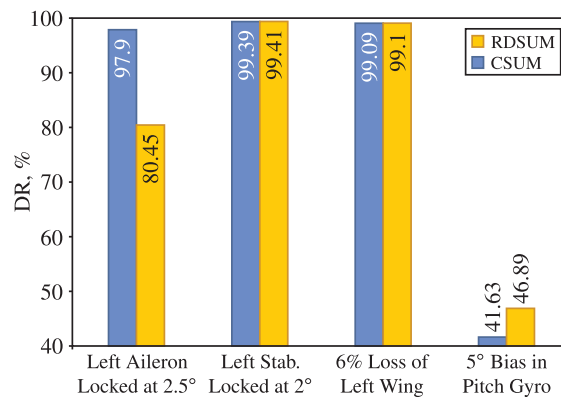**Figure 9.7:** Performance Comparison Between RDSUM and CSUM for Left Aileron Locked at 2.5°
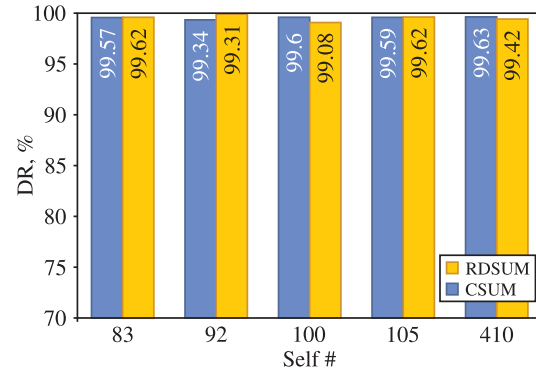
Figure 9.8: Performance Comparison Between RDSUM and CSUM for Left Stabilator Locked at 2°

Figure 9.9: Performance Comparison Between RDSUM and CSUM for 6% Loss of the Left Wing

Figure 9.10: Performance Comparison Between RDSUM and CSUM for 5° Bias in the Pitch Rate Sensor

Figure 9.11: Performance Comparison for Different Abnormal Conditions Using Self #87

The computational time needed by RDSUM is consistently less than the one needed by CSUM. The difference varies quite largely depending on the self. The fewer the number of distinct data, the less the computational time RDSUM needs for clustering. Typically, the size of the raw data file is reduced significantly after eliminating 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 nominal flight tests used) 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 60,000 (which is comparable to the number of records in each of the 11 flight files).

The values of the computational times for the two proposed detector generation methods are presented in Table 9.3 for several different selves. 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 algorithm on computers with less than 8GB RAM.

Table 9.3: Computational Time Comparison Between the Two Detector Generation Methods

Computer Specifications		Intel® Core™ i7 @ 3.33GHz, 12GB RAM	
Operating System		Windows® XP Professional 64-bit	
Development Environment		MATLAB® 2009b	
Generation Method		RDSUM	CSUM
	Self #3	700	6020
	Self #30	1120	5589
	Self #42	801	9032
	Self #53	266	2190
	Self #82	4404	4857
	Self #90	1126	4521
	Self #92	1425	3030
	Self #93	2134	2653
Computational Time [sec]	Self #94	2507	2703
for Detector Generation	Self #95	4217	9527
	Self #96	1903	2277
	Self #97	3797	11315
	Self #98	1900	2342
	Self #99	1276	2443
	Self #105	2056	2245
	Self #106	2283	2285
	Self #224	1562	2623

9.2.   AC Detection Using the Artificial DC Mechanism

The artificial DC mechanism for detection proposed in Chapter 6 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 subselves, 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 $\sum_{i=1}^{N} \widetilde{K}_i$ in Eq. 6.14 versus time is shown in Figs. 9.12 to 9.15 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, as expected, capturing the occurrence of the failure and providing a reliable detection criterion.
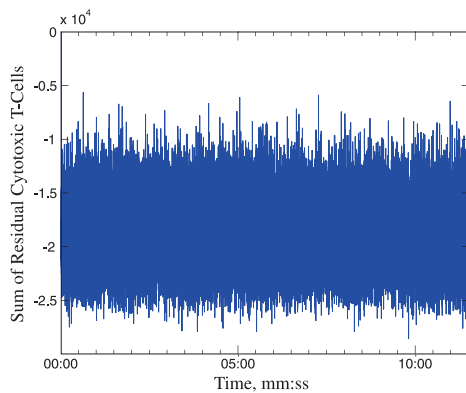
Figure 9.12: Sum of Residual Cytotoxic T-cells vs. Time of a Nominal Flight Test
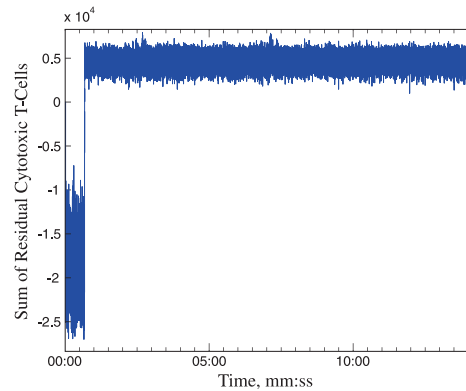
Figure 9.13: Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: Right Aileron Locked at 8°. Actual Failure Time = 40 s; Detection Time = 40.76 s
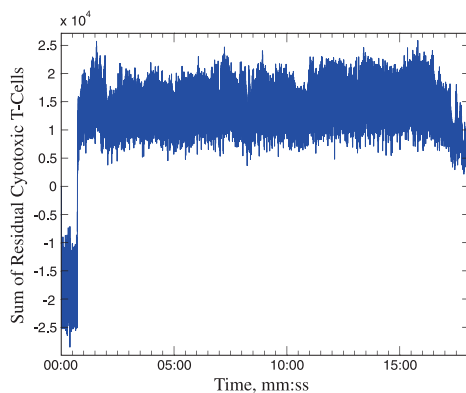
Figure 9.14: Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: 3° LFDB in Yaw Rate Sensor. Actual Failure Time = 40 s; Detection Time = 43.38 s
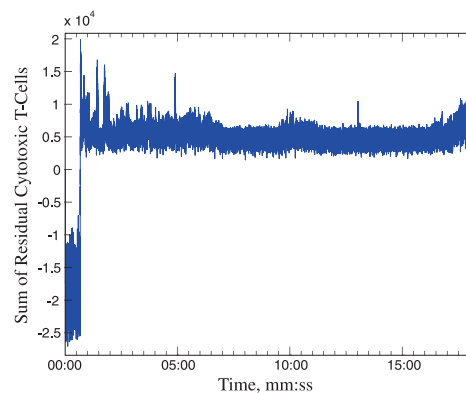
Figure 9.15: Sum of Residual Cytotoxic T-cells vs. Time of an AC Flight Test: 15% Loss of the Left Wing. Actual Failure Time = 40 s; Detection Time = 40.52 s

Table 9.4 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 9.4: Average 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	80.30	2
14	Right Vertical Tail	83.84	2
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

The lower average detection rate for the cases of locked right rudder and missing vertical tail can be attributed to imperfections in the self/nonself generation in regions relevant to directional channel maneuvers and to the specific shape of the high-dimensional self, which prevents some of the projections from capturing the dynamic fingerprint of the failure. It is expected that the inclusion of higher-dimensional projections would eliminate or mitigate this effect.

9.3. AC Identification Using the Artificial DC Mechanism

Among the three pattern approaches proposed in Chapter 7, the features-pattern approach was selected since it requires less computational resources as compared to the projections-pattern and matrix-pattern approaches. The artificial DC mechanism for identification with the features-pattern approach was implemented in C#. The naïve Bayes classifier, discussed in Sec. 7.4, was trained with the current features-pattern 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 9.16 illustrates the variation of the current features-pattern vectors $F_{1\varphi}$ over the entire test time of a failed right wing subsystem ($k = 13$) in one of these tests. Figures 9.17 to 9.35 depict the reference FP_k vectors of the failed subsystems presented in Table 4.1. Once a failure is declared by the DC mechanism in a validation test, 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 Fig. 9.36 for a validation failure test with right wing damage.

Figure 9.16: Variation of the Feature-Pattern Vector with
Time for a Failed Right Wing Subsystem

Figure 9.17: Reference Features-Pattern for a Failed Left Stabilator
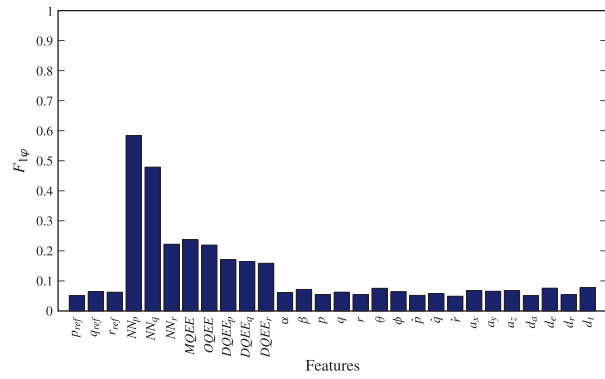
Figure 9.18: Reference Features-Pattern for a Failed Right Stabilator
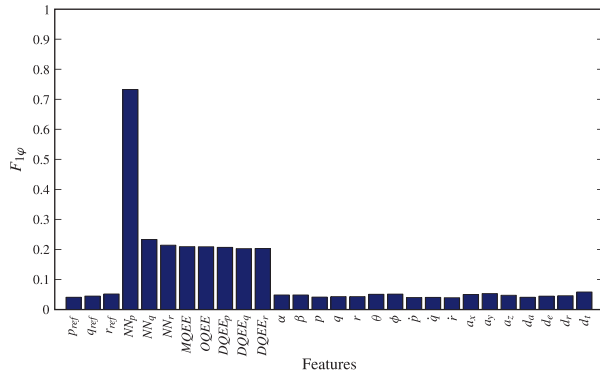
Figure 9.19: Reference Features-Pattern for a Failed Left Aileron
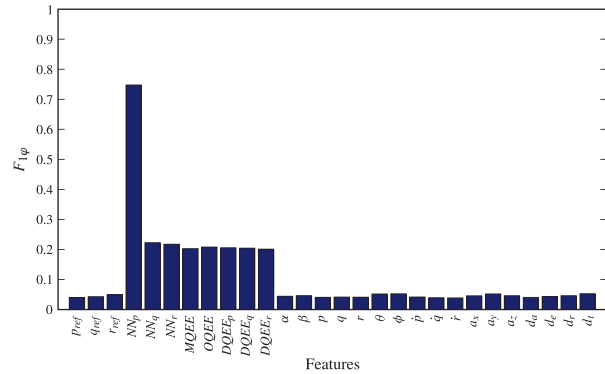
Figure 9.20: Reference Features-Pattern for a Failed Right Aileron
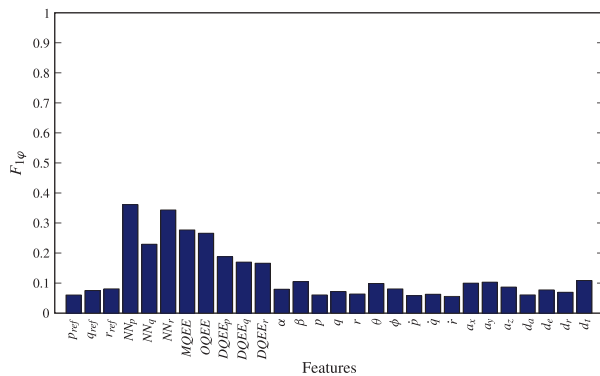
Figure 9.21: Reference Features-Pattern for a Failed Left Rudder
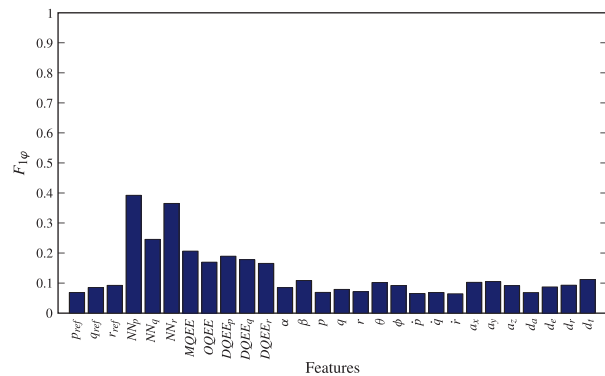
Figure 9.22: Reference Features-Pattern for a Failed Right Rudder

Figure 9.23: Reference Features-Pattern for a
Failed Left Throttle

Figure 9.24: Reference Features-Pattern for a
Failed Right Throttle

Figure 9.25: Reference Features-Pattern for a
Failed Left Wing

Figure 9.26: Reference Features-Pattern for a
Failed Right Wing

Figure 9.27: Reference Features-Pattern for a
Failed Left Horizontal Tail

Figure 9.28: Reference Features-Pattern for a
Failed Right Horizontal Tail

Figure 9.29: Reference Features-Pattern for a Failed Left Vertical Tail

Figure 9.30: Reference Features-Pattern for a Failed Right Vertical Tail

Figure 9.31: Reference Features-Pattern for a Failed Roll Rate Sensor

Figure 9.32: Reference Features-Pattern for a Failed Pitch Rate Sensor

Figure 9.33: Reference Features-Pattern for a Failed Yaw Rate Sensor

Figure 9.34: Reference Features-Pattern for a Failed Left Engine
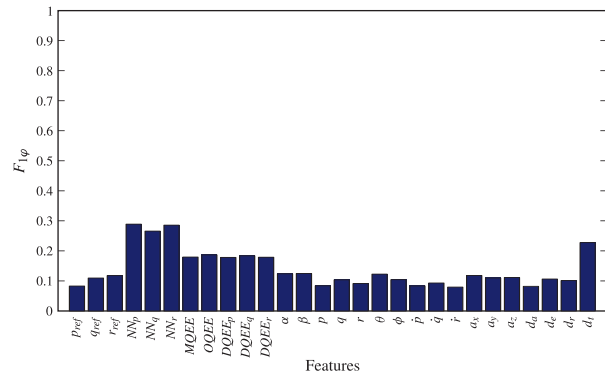
Figure 9.35: Reference Features-Pattern for a Failed Right Engine
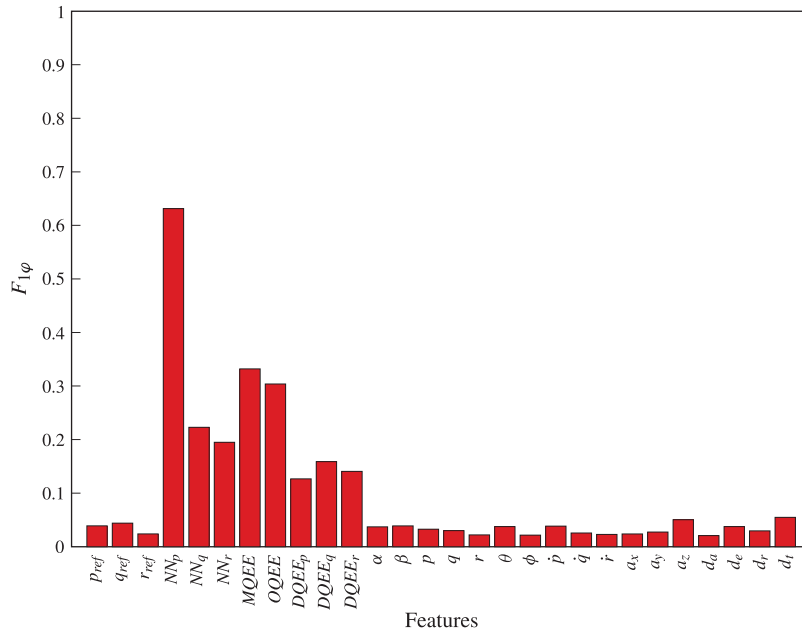
Figure 9.36: Sample Current Feature-Pattern for a Failed Right Wing

Listed in Table 9.5 are the average identification rates for different subsystems. Note that the same number of tests shown in Table 9.4 was also used here. 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 9.5 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.

Table 9.5: Average Identification Rate for Different Subsystems

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**

9.4. AC Qualitative Evaluation

The features-pattern approach was also used to implement the ACQE scheme proposed in Sec. 8.1. The naïve Bayes classifier was trained using a set of training failure tests to define the reference features-pattern FP_{AC_j} for each failure type AC_j. Figures 9.37 to 9.42 illustrate the reference features-patterns for the actuators subsystems. Note that the ACQE scheme was tested for these subsystems only since other subsystems were simulated with only one type of failure (see Table 4.1).

Figure 9.37: Reference Features-Pattern for a Stabilator Under Failure Type 1

Figure 9.38: Reference Features-Pattern for a Stabilator Under Failure Type 2

Figure 9.39: Reference Features-Pattern for an Aileron Under Failure Type 1

Figure 9.40: Reference Features-Pattern for an Aileron Under Failure Type 2

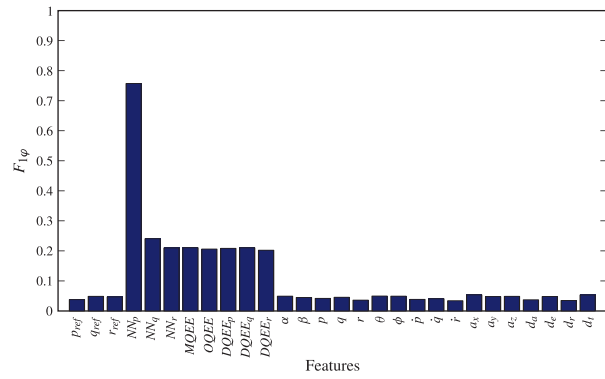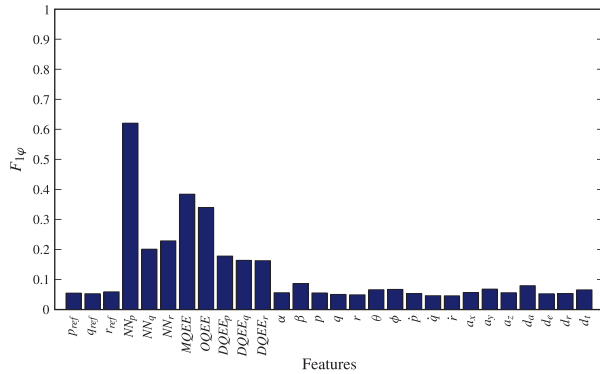**Figure 9.41:** Reference Features-Pattern for a Rudder Under Failure Type 1

Figure 9.42: Reference Features-Pattern for a Rudder Under Failure Type 2

Table 9.6 lists the ACQE rates for the actuator subsystems. The ACQE rate is defined as the number of time steps for which the failure type is correctly evaluated divided by the total number of time steps after failure. The second column in Table 9.6 represents the "known" failure type of

the corresponding subsystem whereas the first row represents the evaluated failure. For example, 99.98% of the failure samples for failure type 1 of the stabilator was correctly evaluated as failure type 1 while for 0.02% of these samples was incorrectly evaluated as failure type 2. These results show the capability of the proposed ACQE scheme in isolating the failure type with very high rates.

Table 9.6: ACQE Rates for the Actuator Subsystems

Subsystem	Failure Type	1	2
Stabilator	1	**99.98**	0.02
	2	0.01	**99.99**
Aileron	1	**99.99**	0.01
	2	0.01	**99.99**
Rudder	1	**100.00**	0.00
	2	0.00	**100.00**

9.5. AC Direct Quantitative Evaluation

The features-pattern approach was also used to implement the ACQE scheme proposed in Sec. 8.2. The naïve Bayes classifier was trained using a set of training failure tests to define the reference features-pattern FP_{fs} for each failure severity fs. Figures 9.43 to 9.68 illustrate the reference features-patterns for subsystems under different failure severity scales.
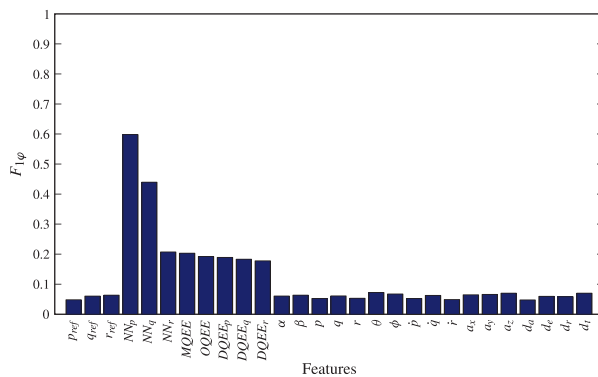
Figure 9.43: Reference Features-Pattern for a Stabilator Under Low Severity Failure
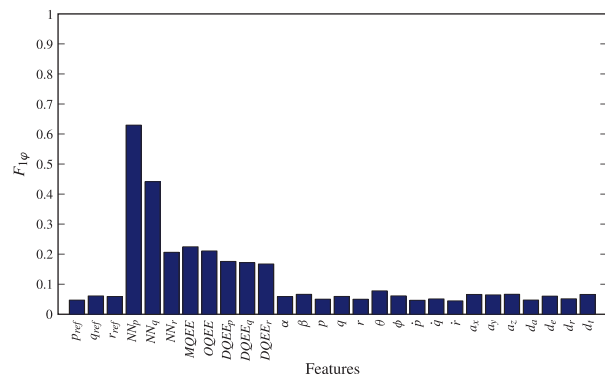
Figure 9.44: Reference Features-Pattern for a Stabilator Under Medium Severity Failure
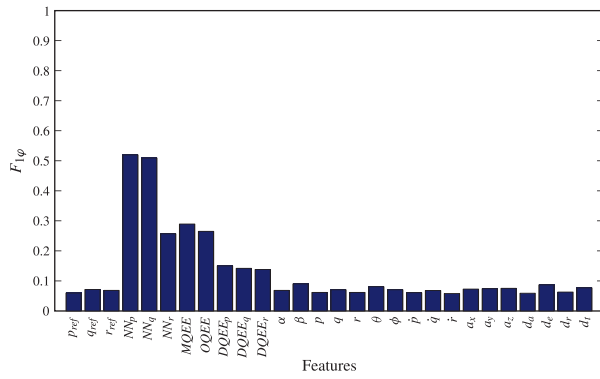
Figure 9.45: Reference Features-Pattern for a Stabilator Under High Severity Failure
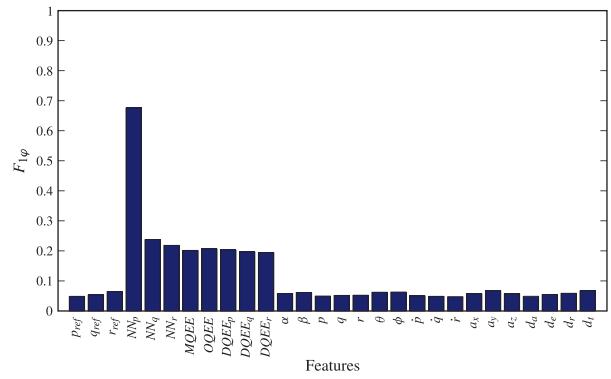
Figure 9.46: Reference Features-Pattern for an Aileron Under Low Severity Failure
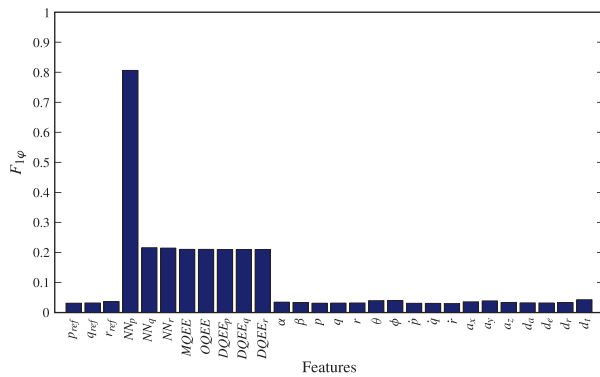
Figure 9.47: Reference Features-Pattern for an Aileron Under Medium Severity Failure
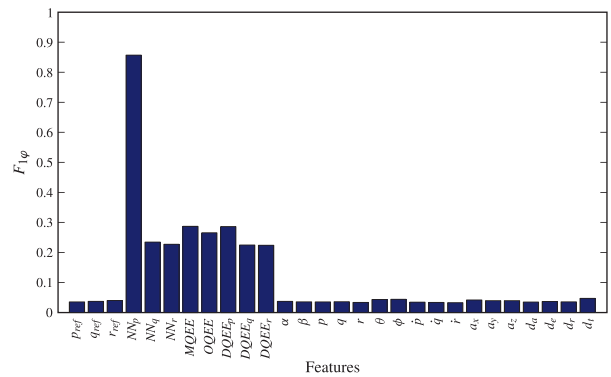
Figure 9.48: Reference Features-Pattern for an Aileron Under High Severity Failure
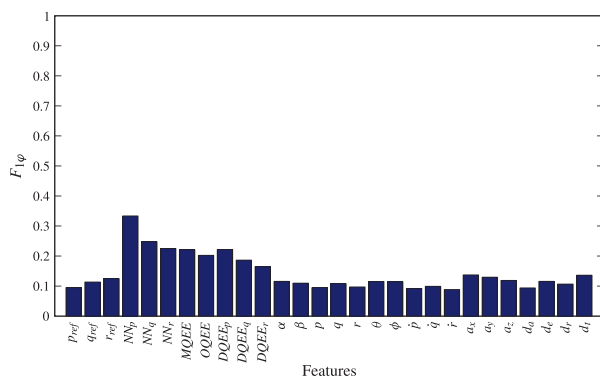
Figure 9.49: Reference Features-Pattern for a Rudder Under Low Severity Failure
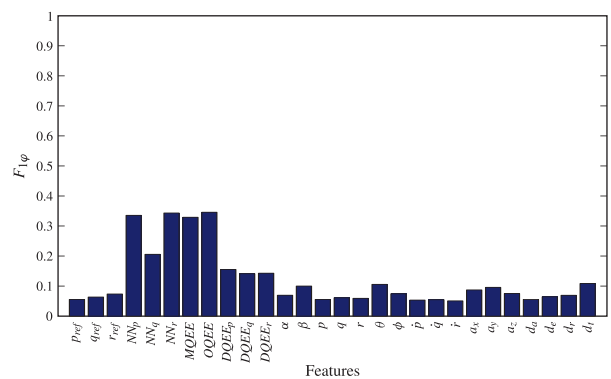
Figure 9.50: Reference Features-Pattern for a Rudder Under Medium Severity Failure
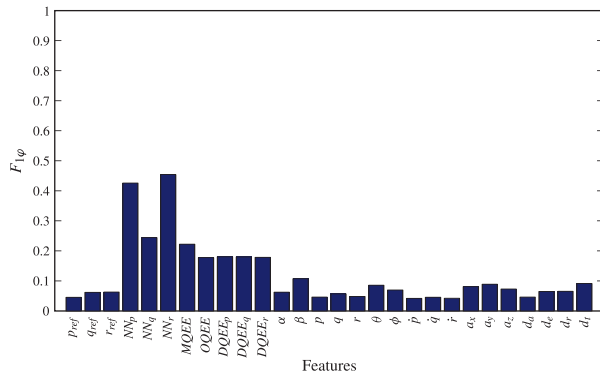
Figure 9.51: Reference Features-Pattern for a Rudder Under High Severity Failure

Figure 9.52: Reference Features-Pattern for a Wing Under Low Severity Failure

Figure 9.53: Reference Features-Pattern for a Wing Under Medium Severity Failure
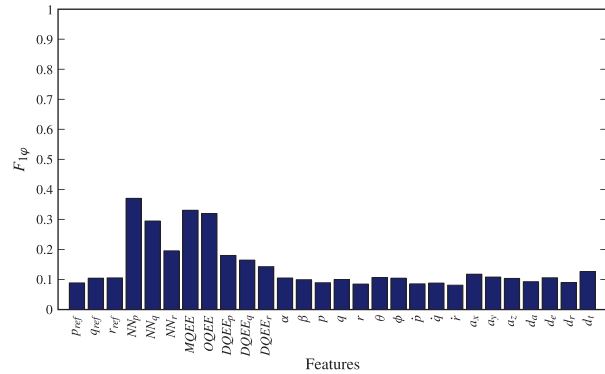
Figure 9.54: Reference Features-Pattern for a Wing Under High Severity Failure
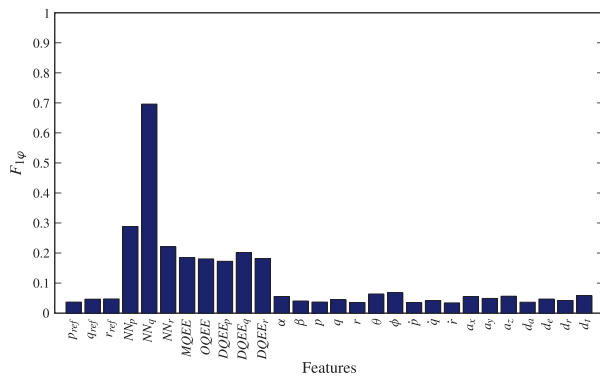
Figure 9.55: Reference Features-Pattern for a Horizontal Tail Under Low Severity Failure
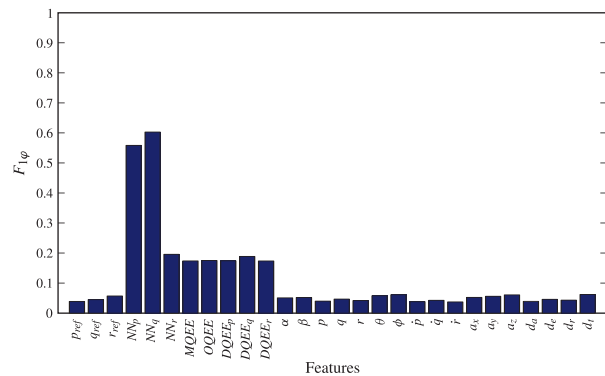
Figure 9.56: Reference Features-Pattern for a Horizontal Tail Under Medium Severity Failure

Figure 9.57: Reference Features-Pattern for a
Horizontal Tail Under High Severity Failure

Figure 9.58: Reference Features-Pattern for a
Vertical Tail Under Low Severity Failure

Figure 9.59: Reference Features-Pattern for a
Vertical Tail Under Medium Severity Failure

Figure 9.60: Reference Features-Pattern for a
Vertical Tail Under High Severity Failure

Figure 9.61: Reference Features-Pattern for a
Roll Rate Sensor Under Low Severity Failure

Figure 9.62: Reference Features-Pattern for a
Roll Rate Sensor Under High Severity Failure

Figure 9.63: Reference Features-Pattern for a Pitch Rate Sensor Under Low Severity Failure
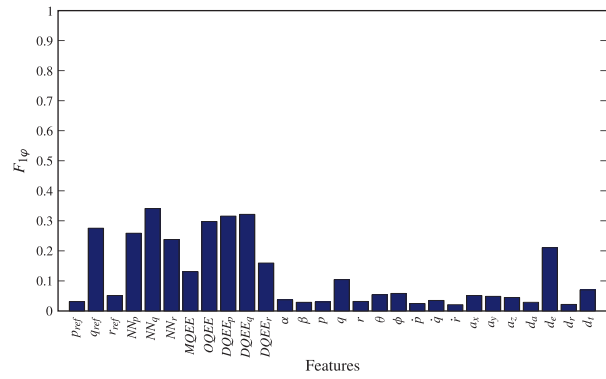
Figure 9.64: Reference Features-Pattern for a Pitch Rate Sensor Under High Severity Failure
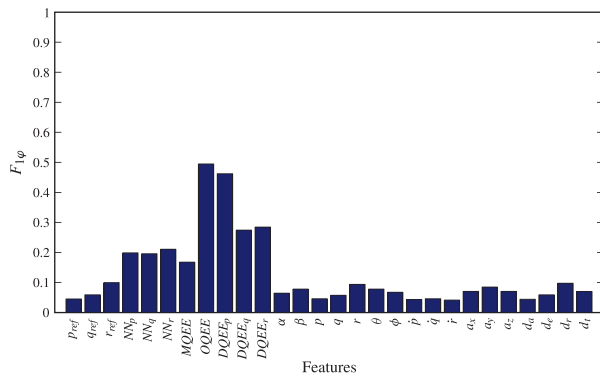
Figure 9.65: Reference Features-Pattern for a Yaw Rate Sensor Under Low Severity Failure
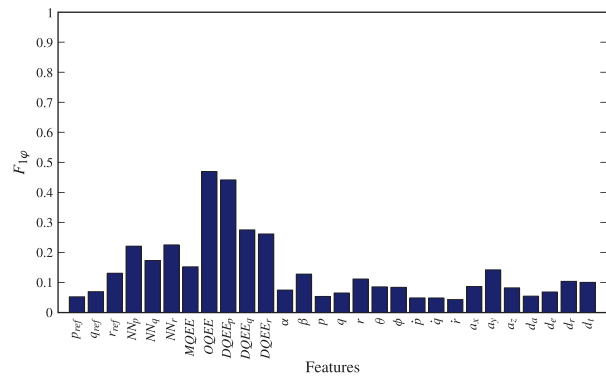
Figure 9.66: Reference Features-Pattern for a Yaw Rate Sensor Under High Severity Failure
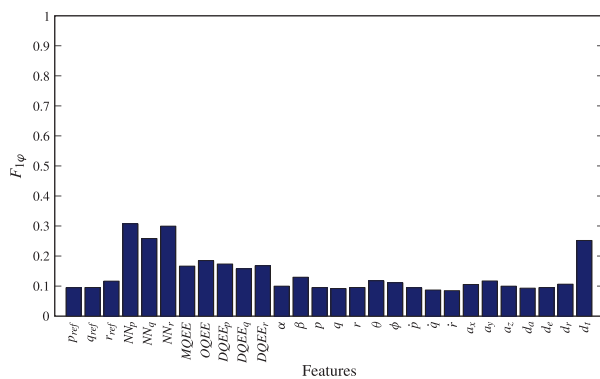
Figure 9.67: Reference Features-Pattern for an Engine Under Low Severity Failure
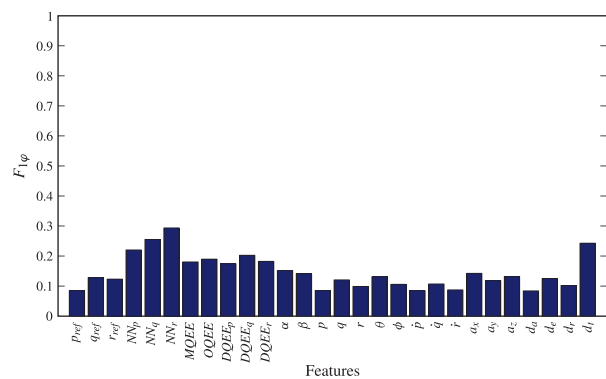
Figure 9.68: Reference Features-Pattern for an Engine Under High Severity Failure

Table 9.7 presents the ACDQE rates for the actuator and structural subsystems. The ACDQE 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 scales were simulated for the sensors and engine failures. These results show the capability of the proposed ACDQE scheme in isolating the failure severity with very high rates.

Table 9.7: ACDQE Rates for Subsystems Under Different Failure Severity Scales

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**
Horizontal Tail	Low	**100.00**	0.00	0.00
	Medium	0.00	**99.99**	0.01
	High	0.01	0.00	**99.99**
Vertical Tail	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**

9.6. AC Indirect Quantitative Evaluation

To achieve desirable performance of the post-failure envelope prediction methodology presented in Sec. 8.3, data points acquired through properly designed validation tests under abnormal conditions must stay within the predicted range and be close to the predicted limits. Therefore, for performance evaluation of these schemes, 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 validation points that fall inside the predicted range. If N_R is the number of the points inside the predicted range and $N_{\overline{R}}$ is the number of points outside the that range, then the total number of validation points is given by

$$N_V = N_R + N_{\overline{R}}. \tag{9.3}$$

The prediction rate can the be calculated from

$$PR = \frac{N_R}{N_V} 100. \tag{9.4}$$

The level of possible range exceedance is assessed as the relative maximum amount by which points in the validation data set exceed the predicted range. The predicted range exceedance index, REI, is defined as

$$REI = \frac{\max\limits_{N_{\overline{R}}} \left[\min \left(\left| v_E - v_{E\min} \right|, \ \left| v_E - v_{E\max} \right| \right) \right]}{\left| v_{E\max} - v_{E\min} \right|} 100. \tag{9.5}$$

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 these two evaluation metrics must be considered in conjunction with a margin index, MI, that is expected to capture the level of predicted range conservativeness. The margin index is defined as

$$MI = \frac{\max\limits_{N_R} \left[\min \left(\left| v_E - v_{E\min} \right|, \ \left| v_E - v_{E\max} \right| \right) \right]}{\left| v_{E\max} - v_{E\min} \right|} 100. \tag{9.6}$$

It should be also noted that, if the validation 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.

Table 9.8 presents the predicted post-failure flight envelope ranges under a stabilator failure along with the corresponding nominal ranges and the performance metrics defined above. Figure 9.69 shows an example of a 2-D projection with the validation test points and the corresponding predicted post-failure ranges represented by the vertical and horizontal lines. The failure considered consists of left stabilator locked at 8°. Two different validation 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 stabilator 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 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 9.8: Predicted Flight Envelope and Performance Metrics 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
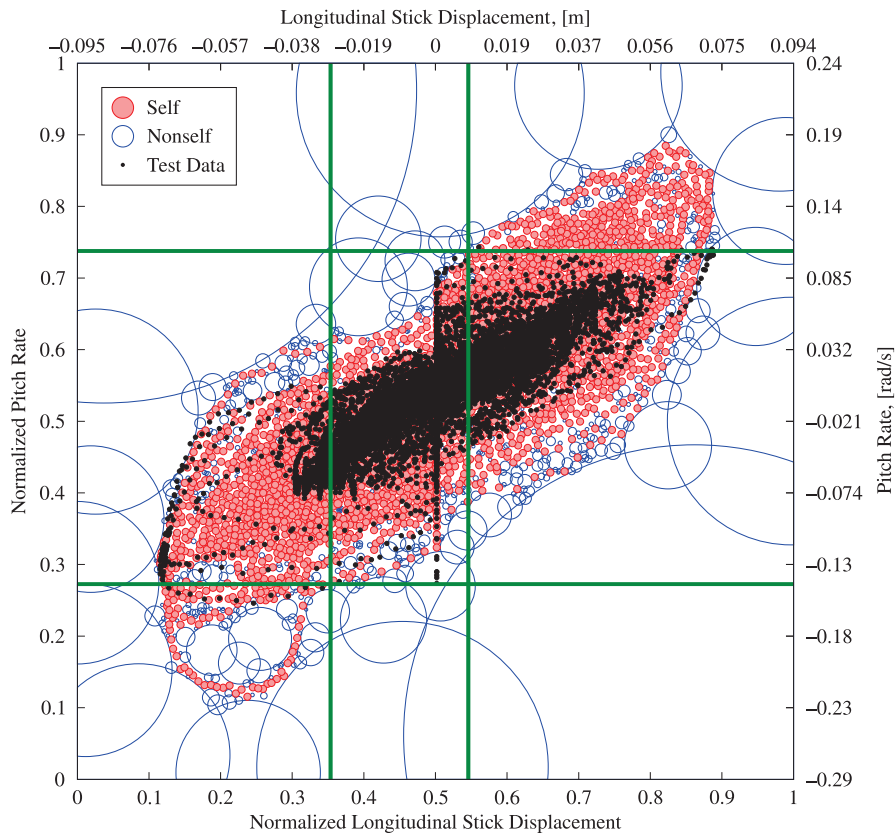
Figure 9.69: Post-Failure Range of the Pitch Rate Under Stabilator Failure

The predicted post-failure flight envelope ranges under the aileron failure are presented in Table 9.9. 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 ranges 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 exceedance as indicated by *REI*. The low values of *REI* and *MI* indicate that the predicted ranges are not unrealistically large.

Table 9.9: Predicted Flight Envelope and Performance Metrics 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

The predicted post-failure flight envelope ranges under the rudder failure are presented in Table 9.10. 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.

Table 9.10: Predicted Flight Envelope and Performance Metrics 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
$\dot{r}$	$[-0.117, 0.126]$	$[-0.114, 0.104]$	100.0	0.0	10.4

The predicted post-failure flight envelope ranges under a wing damage affecting 15% of the left surface are listed in Table 9.11. 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 9.12 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

Table 9.11: Predicted Flight Envelope and Performance Metrics Under Wing Failure

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
$\dot{p}$	$[-2.95, 3.1]$	$[-2.97, 2.5]$	99.9	2.5	0.0
q	$[-0.225, 0.183]$	$[-0.05, 0.064]$	100.0	0.0	1.2
$\dot{q}$	$[-0.383, 0.386]$	$[-0.091, 0.072]$	99.8	9.7	3.7

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 9.12: Predicted Flight Envelope and Performance Metrics Under Horizontal Tail Failure

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
$\dot{q}$	$[-0.383, 0.386]$	$[-0.37, 0.355]$	100.0	0.0	9.1
p	$[-0.94, 1.08]$	$[-0.925, 1.09]$	100.0	0.0	18.1
$\dot{p}$	$[-2.95, 3.1]$	$[-2, 3.13]$	100.0	0.0	24.5

Table 9.13 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 both variables considered, as reflected by the corresponding *REI* values. The relatively low values of *MI* show that the predicted limits of the respective variables are not unrealistically large.

Table 9.13: Predicted Flight Envelope and Performance Metrics 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
$\dot{p}$	$[-2.95, 3.1]$	$[-1.8, 3.13]$	100.0	0.0	15.5

Listed in Table 9.14 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 9.14: Predicted Flight Envelope and Performance Metrics 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
$\dot{q}$	$[-0.383, 0.386]$	$[-0.417, 0.32]$	99.9	30.9	6.0

The predicted post-failure flight envelope ranges under the yaw rate sensor failure are listed in Table 9.15. 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 9.15: Predicted Flight Envelope and Performance Metrics 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
$\dot{r}$	$[-0.117, 0.126]$	$[-0.093, 0.128]$	100.0	4.8	5.8
β	$[-9.6, 9.8]$	$[-9.6, 9.8]$	100.0	0.7	16.9

Table 9.16 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 9.16: Predicted Flight Envelope and Performance Metrics Under Engine Failure

ERV	Nominal Range	Post-Failure Range	*PR*, %	*REI*, %	*MI*, %
a_x	$\left[-0.344, 0.597\right]$	$\left[-0.344, 0.353\right]$	99.9	4.8	20.4
a_z	$\left[-2.46, 3.05\right]$	$\left[-2.46, 2.16\right]$	99.6	10.3	10.4
r	$\left[-0.0637, 0.0621\right]$	$\left[-0.0637, 0.0393\right]$	99.9	5.3	21.5
H	$\left[2012, 9929\right]$	$\left[2012, 9929\right]$	95.0	3.4	50.0
M	$\left[0.526, 0.95\right]$	$\left[0.526, 0.95\right]$	100.0	0.0	54.1

Conclusions and Future Work

The generalized algorithmic framework presented in this research provides tools for an integrated and comprehensive solution to the problem of safe aircraft operation under abnormal conditions. The main concepts and foundations were established for immunity-based methodologies and algorithms that are expected to facilitate the design of onboard augmentation systems to increase aircraft survivability and improve operation safety.

Two methods for generating the AIS antibodies have been presented and compared in terms of the detection performance and computational resources. The detection performance of the 2-dimensional projections under various types of AC showed that both the RDSUM and CSUM are equivalent from this point of view. Generating the antibodies with the RDSUM requires more computer memory as compared to the CSUM. However, the computational time needed by the RDSUM is less than that required by the CSUM, unless parallel computation is used. The parameters related to the self/nonself generation (such as the shape, number, and size of the self clusters and the overlapping between self clusters and detectors) affect both the detection performance of the subselves and the computational resources required by each method. To achieve better detection performance with less computational resources, these parameters need to be tuned through optimization (for example, by using the genetic algorithm suggested in [84]). Additional nominal flight tests covering a wider range of the flight envelope are expected to improve the overall detection performance of the generated subselves but at the expense of the computer memory and computational speed.

Conceptual formulations and computational algorithms have been outlined for aircraft subsystem failure detection, identification, and evaluation based on the novel artificial DC mechanism within the AIS paradigm. Implementation results of the proposed DC mechanism for AC detection show the high capability of the approach in detecting aircraft failures and minimizing the false alarms rate to 0% for most nominal flights.

Despite their success in some application domains, the DCA proposed by [58] and its variants [59, 61] rely on the mapping of their input signals to the appropriate parameters of the particular application. This mapping becomes extremely difficult, if not impossible, in complex and multidimensional systems such as the aircraft. The DCA was based on the behavior of the DCs that are part of the innate immune system, whereas the principle of self/non-self-discrimination was based on the functionality of the antibodies in the adaptive immune system. The synergistic combination of the HMS strategy and the novel artificial DC mechanism presented in this research effort was inspired from both the functionality of the biological DCs in the innate immune system and their interaction with the different constituents of the adaptive immune system (T-cells, B-cells, and antibodies). The artificial DCs in the proposed algorithm take the

outcomes of the self/non-self-discrimination as the only two signals (safe and danger) that influence the resulting failure detection, thus overcoming the challenge of mapping the input signals to the application's parameters, as required by the original DCA, and facilitating the implementation to the complex and multidimensional aircraft system.

The ability of the artificial DCs of storing information about each processed nonself and fusing them in a certain pattern useful for failure identification is another advantage of the proposed algorithm over the original DCA, which was designed solely for failure detection. Table 10.1 presents a summary of the qualitative comparison between the original DCA and the artificial DC mechanism proposed in the research effort.

Table 10.1: Comparison Between the Original DCA and the Artificial DC Mechanism

Original DC Algorithm	**Artificial DC Mechanism**
Inspired from only the functionality of the biological DCs in the innate immune system.	Inspired from both the functionality of the biological DCs in the innate immune system and their interaction with the different constituents of the adaptive immune system.
Relies on the mapping of 4 input signals to the appropriate parameters of the particular application, which is extremely difficult, if not impossible, in complex and multidimensional systems, such as the aircraft.	Needs only 2 input signals (safe and danger) from the self/nonself discrimination without special mapping to the application's parameters. Synergistically combined with the HMS strategy, thus addressing the complexity and multidimensionality of the aircraft system.
Designed solely for AC detection.	Designed for AC detection, identification, and qualitative and direct quantitative evaluation.

While the proposed DC mechanism performs quite similar to the Structured-Nonself Approach [90] for identification and evaluation, it has the advantage of avoiding structuring the nonself, thus making the design of the detection, identification, and evaluation scheme more streamlined, general, and consistent.

Tuning the various parameters of the artificial DC mechanism (such as the number of DCs in the pool and the triggered and nontriggered weights) has to be further investigated in a future research effort. The detection performance for the cases of throttle failure and engine reduced efficiency can be improved by considering additional features (such as temperature and pressure) that capture the dynamic fingerprint of these failures, as investigated in [108].

The results of implementing the features-pattern approach of the DC mechanism for AC identification, qualitative evaluation, and direct quantitative evaluation showed its high capability in correctly identifying the failed subsystem and evaluating the type and severity of the failure. The patterns provided by the migrated DCs in the artificial DC mechanism convert the AC identification and evaluation problems into pattern-recognition ones. The naïve Bayes classifier used as the pattern-recognition algorithm in these schemes is limited to classifying the current pattern as one of the known patterns included in the training phase. In order to identify

novel/unseen failed subsystems, one can use another classifier which is capable of classifying unseen patterns, such as Support Vector Machines and Neural Networks [28, 29], keeping in mind other application-specific requirements that need to be satisfied (e.g., speed and accuracy of the classification).

The alternative pattern approaches (i.e., the projections-pattern and matrix-pattern) proposed in Chapter 7 are recommended for future study. Special care should be taken about selecting the features or projections required for the particular classifier when applying the proposed schemes to the system under consideration in that they must be distinguishing or informative for the classifier and invariant to irrelevant data transformation. These features/projections may be selected via analysis or, more appropriately, by using a selection algorithm such as the wrapper approaches proposed by Kohavi and John [109].

The average computational time required for the artificial DC mechanism to provide the FDIE outcome at each time step is about 40 millisecond on Windows® XP Professional 64-bit running on Intel® Core™ i7 @ 3.33GHz.

Immunity-based algorithms for the flight envelope prediction at post-failure of various subsystems have been developed in this work and successfully demonstrated through numerical examples. The prediction of altered ranges of the relevant flight envelope variables at post-failure conditions requires prior abnormal condition detection, identification, direct evaluation, and failure specific algorithms. The performance metrics defined for the purpose of validating the altered ranges show that the proposed methodology can be used to yield reasonable post-failure ranges. Future work should address the issues related to the resolution of the self clusters to improve the accuracy of the predicted ranges.

The Naïve Bayes Classifier

Naïve Bayes classifier is one of the simplest *probabilistic supervised learning* algorithms in machine learning [28]. It is based on the statistical Bayes' theorem with strong independence assumption (i.e., the presence or absence of a particular attribute of a class is unrelated to the presence or absence of any other attribute). Although this assumption is "naïve," the classifier surprisingly provides excellent performance (high classification rate) in practice, even when this assumption is violated. In many practical applications, parameter estimation for naïve Bayes models uses the method of *maximum likelihood* or *maximum a posteriori* (MAP).

Another advantage of the naïve Bayes classifier is that it is very fast in both the *learning* and *testing* phases, even for large datasets [29]. Besides its high speed and performance, the naïve Bayes classifier works pretty well even when a small amount of training data is available to estimate the parameters (means and variances of the attributes) necessary for classification.

Let $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T$ be an $m \times 1$ vector of continuous values of m attributes to be classified into a class variable $y = \{y_j \mid j = 1, 2, \ldots, n\}$. The naïve Bayes probabilistic model is a probability model $P(y \mid \mathbf{x})$ over the dependent class variable y conditional on attribute variables x_1 through x_m. From Bayes' theorem, we have

$$P(y \mid \mathbf{x}) = \frac{P(y) \, P(\mathbf{x} \mid y)}{P(\mathbf{x})}, \tag{A.1}$$

where $P(y \mid \mathbf{x})$ is the *posterior* class probability, $P(y)$ is the *prior* class probability, $P(\mathbf{x} \mid y)$ is the *likelihood*, and $P(\mathbf{x})$ is the *evidence*.

Using the chain rule for repeated applications of the definition of the conditional probability, the likelihood can be written as

$$
\begin{aligned}
P(\mathbf{x} \mid y) &= P(x_1, x_2, \ldots, x_m \mid y) \\
&= P(x_1 \mid y) \, P(x_2, \ldots, x_m \mid y, x_1) \\
&= P(x_1 \mid y) \, P(x_2 \mid y, x_1) \, P(x_3, \ldots, x_m \mid y, x_1, x_2) \\
&= P(x_1 \mid y) \, P(x_2 \mid y, x_1) \, P(x_3 \mid y, x_1, x_2) \, P(x_4, \ldots, x_m \mid y, x_1, x_2, x_3) \\
&= P(x_1 \mid y) \, P(x_2 \mid y, x_1) \, P(x_3 \mid y, x_1, x_2) \cdots P(x_m \mid y, x_1, x_2, \ldots, x_{m-1}).
\end{aligned} \tag{A.2}
$$

Using the "naïve" assumption of the *conditional independence* of the attributes, we have

$$P(x_i \mid y, x_j) = P(x_i \mid y), \quad \forall \, i \neq j. \tag{A.3}$$

Therefore, Eq. (A.2) can be written as

$$P\left(\mathbf{x}\,\middle|\,y\right) = \prod_{i=1}^{m} P\left(x_i\,\middle|\,y\right). \tag{A.4}$$

Assuming normal distribution of the attributes over class y_j, Eq. (A.4) becomes

$$P\left(\mathbf{x}\,\middle|\,y = y_j\right) \sim \mathcal{N}_m\left(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) = \frac{1}{\sqrt{\left(2\pi\right)^m \left|\boldsymbol{\Sigma}_j\right|}} \exp\left[-\frac{\left(\mathbf{x} - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1} \left(\mathbf{x} - \boldsymbol{\mu}_j\right)}{2}\right]. \tag{A.5}$$

where $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the sample mean vector and the sample covariance matrix in class y_j, respectively. The sample mean vector and the sample covariance matrix can be estimated from a given sample attribute vectors.   If n_j is the number of samples in class y_j and $\mathbf{x}_j^{(k)} = \left[\begin{array}{cccc} x_{1j}^{(k)} & x_{2j}^{(k)} & \cdots & x_{mj}^{(k)} \end{array}\right]^T$ is the k-th sample in that class, then we have

$$\boldsymbol{\mu}_j = \left[\begin{array}{cccc} \mu_{1j} & \mu_{2j} & \cdots & \mu_{mj} \end{array}\right]^T = \frac{1}{n_j} \sum_{k=1}^{n_j} \mathbf{x}_j^{(k)}, \tag{A.6}$$

$$\boldsymbol{\Sigma}_j = \left[\begin{array}{cccc} \sigma_{1j}^2 & \sigma_{1j}\sigma_{2j} & \cdots & \sigma_{1j}\sigma_{mj} \\ \sigma_{1j}\sigma_{2j} & \sigma_{2j}^2 & \cdots & \sigma_{2j}\sigma_{mj} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1j}\sigma_{mj} & \sigma_{2j}\sigma_{mj} & \cdots & \sigma_{mj}^2 \end{array}\right] = \frac{1}{n_j - 1} \sum_{k=1}^{n_j} \left(\mathbf{x}_j^{(k)} - \boldsymbol{\mu}_j\right) \left(\mathbf{x}_j^{(k)} - \boldsymbol{\mu}_j\right)^T. \tag{A.7}$$

The classification *decision* of the naïve Bayes classifier is based on the mode of Eq. (A.1), i.e., the MAP, given by

$$\hat{y} = y_j^* \quad \Leftrightarrow \quad y_j^* = \operatorname*{argmax}_{y} \left[P\left(y\,\middle|\,\mathbf{x}\right)\right] = \operatorname*{argmax}_{y} \left[\frac{P\left(y\right) P\left(\mathbf{x}\,\middle|\,y\right)}{P\left(\mathbf{x}\right)}\right]. \tag{A.8}$$

Since the denominator of the fraction in Eq. (A.8) does not depend on y and the values of the attributes $\mathbf{x}$ are given, therefore, only the numerator affects the classification outcome. Hence, Eq. (A.8) reduces to

$$y_j^* = \operatorname*{argmax}_{y} \left[P\left(y\right) P\left(\mathbf{x}\,\middle|\,y\right)\right]. \tag{A.9}$$

Since both the product $P\left(y\right) P\left(\mathbf{x}\,\middle|\,y\right)$ and its logarithm attain their maximum values at the same location, therefore, Eq. (A.9) is equivalent to

$$y_j^* = \operatorname*{argmax}_{y} \left[\ln P\left(y\right) + \ln P\left(\mathbf{x}\,\middle|\,y\right)\right] = \operatorname*{argmax}_{y} \left[\Delta_j\left(\mathbf{x}\right)\right], \tag{A.10}$$

where $\Delta_j\left(\mathbf{x}\right)$ is the discriminant.

Using Eq. (A.5), we have

$$
\begin{aligned}
\Delta_j(\mathbf{x}) &= \ln P(y) + \ln P(\mathbf{x}\,|\,y) \\
&= \ln P(y) - \frac{m}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j). \quad (\text{A.11})
\end{aligned}
$$

The prior class probabilities $P(y)$ can be easily estimated from the given data either by using the relative frequencies or a more sophisticated estimation method, such as the m-probability estimate in order to smooth the estimation of a small dataset. The priors can be computed from

$$
P(y = y_j) = \frac{n_j}{N}, \quad (\text{A.12})
$$

where $N = \sum_{j=1}^{n} n_j$ is the total number of samples.

Since the second term in Eq. (A.11) is constant, therefore, it can be dropped out without affecting the classification decision of Eq. (A.10). Thus, Eq. (A.11) becomes, after substituting $P(y)$ from Eq. (A.12),

$$
\begin{aligned}
\Delta_j(\mathbf{x}) &= \ln\left(\frac{n_j}{N}\right) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j) \\
&= \ln n_j - \ln N - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j) \\
&= \ln n_j - \frac{1}{2}\ln|\boldsymbol{\Sigma}_j| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j), \quad (\text{A.13})
\end{aligned}
$$

where we have dropped out the constant term $(\ln N)$ since it does not affect the maximum posterior. This analysis is called *quadratic discrimination analysis* since it results in the quadratic discriminant function in Eq. (A.13).

If we assume the covariance matrices are the same over the classes, i.e.,

$$
\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2 & \cdots & \sigma_1\sigma_m \\ \sigma_1\sigma_2 & \sigma_2^2 & \cdots & \sigma_2\sigma_m \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1\sigma_m & \sigma_2\sigma_m & \cdots & \sigma_m^2 \end{bmatrix} = \frac{1}{N-n}\sum_{j=1}^{n}\sum_{k=1}^{n_j}\left(\mathbf{x}_j^{(k)} - \boldsymbol{\mu}_j\right)\left(\mathbf{x}_j^{(k)} - \boldsymbol{\mu}_j\right)^T, \quad \forall j, \quad (\text{A.14})
$$

then, Eq. (A.13) becomes

$$
\begin{aligned}
\Delta_j(\mathbf{x}) &= \ln n_j - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_j) \\
&= \ln n_j - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j \\
&= \ln n_j - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j - \frac{1}{2}\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j. \quad (\text{A.15})
\end{aligned}
$$

Recalling that $\mathbf{\Sigma}$ is symmetric and using the properties of the inverse and transpose of matrices, it can be easily shown that $\frac{1}{2}\boldsymbol{\mu}_j^T\mathbf{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_j = \boldsymbol{\mu}_j^T\mathbf{\Sigma}^{-1}\mathbf{x}$. Hence, Eq. (A.15) reduces to

$$\Delta_j\left(\mathbf{x}\right) = \ln n_j - \frac{1}{2}\ln\left|\mathbf{\Sigma}\right| - \frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_j^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_j^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_j. \tag{A.16}$$

Since both the second and third term in Eq. (A.16) are independent of the class $y = y_j$, therefore, they can be dropped out without affecting the classification decision, upon which we get

$$\Delta_j\left(\mathbf{x}\right) = \ln n_j + \boldsymbol{\mu}_j^T\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_j^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_j. \tag{A.17}$$

This is the so called *linear discrimination analysis* for that it results in the linear discriminant function in Eq. (A.17).

[1] KrishnaKumar, Kalmanje, Nhan T. Nguyen, and John T. Kaneshige, *Integrated Resilient Aircraft Control*, Encyclopedia of Aerospace Engineering, John Wiley & Sons, 2010, doi: 10.1002/9780470686652.eae510.

[2] Edwards, Christopher, Thomas Lombaerts, and Hafid Smaili, *Fault Tolerant Flight Control*, Lecture Notes in Control and Information Sciences, Vol. 399, Springer-Verlag, Berlin, 2010.

[3] White, John, "NASA's Aviation Safety Program," 44th Annual AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 2006.

[4] Kwatny, Harry, Celeste Belcastro, Christine M. Belcastro, and Bor-Chin Chang, "Aircraft Loss-of-Control Accident Prevention, Part I: Safety State Assessment and Preliminary Results," Proceedings of the AIAA Guidance, Navigation and Control Conference, Honolulu, HI, Aug. 2008, doi: 10.2514/6.2008-6506.

[5] KrishnaKumar, Kalmanje, and Karen Gundy-Burlet, "Intelligent Control Approaches for Aircraft Applications," ANAFF Interagency Propulsion Committee Meeting, Destin, FL, 2002.

[6] Jacobson, Steven R., "Aircraft Loss of Control Causal Factors and Mitigation Challenges," Proceedings of the AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Aug. 2010, doi: 10.2514/6.2010-8007.

[7] Zhang, Youmin, and Jin Jiang, "Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems," *Annual Reviews in Control*, Vol. 32, No. 2, 2008, pp. 229–252, doi: 10.1016/j.arcontrol.2008.03.008.

[8] Campbell, Stefan F., John T. Kaneshige, Nhan T. Nguyen, and Kalmanje KrishnaKumar, "Implementation and Evaluation of Multiple Adaptive Control Technologies for a Generic Transport Aircraft Simulation," Proceedings of the AIAA Infotech@Aerospace Conference, Atlanta, GA, Aug. 2010, doi: 10.2514/6.2010-3322.

[9] Nguyen, Nhan T., and Kalmanje KrishnaKumar, "Hybrid Intelligent Flight Control with Adaptive Learning Parameter Estimation," *Journal of Aerospace Computing, Information, and Communication*, Vol. 6, 2009, pp. 171–186, doi: 10.2514/1.35929.

[10] Azam, Mohammad, Krishna Pattipati, Jeffrey Allanach, Scott Poll, and Ann Patterson-Hine, "In-Flight Fault Detection and Isolation in Aircraft Flight Control Systems," IEEE Aerospace Conference, Big Sky, MT, March 2005, pp. 3555–3565, doi: 10.1109/AERO.2005.1559659.

[11] Oonk, Stephen, Francisco J. Maldonado, Fernando Figueroa, and Ching-Fang Lin, "Predictive Fault Diagnosis System for Intelligent and Robust Health Monitoring," *Journal of Aerospace Computing, Information, and Communication*, Vol. 9, No. 4, 2012, pp. 125–143, doi: 10.2514/1.54961.

[12] Boskovic, Jovan D., Joshua Redding, and Nathan Knoebel, "An Adaptive Fault Management (AFM) System for Resilient Flight Control," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, 2009, doi: 10.2514/6.2009-6263.

[13] Unnikrishnan, Suraj, and J. V. R. Prasad, "Carefree Handling using Reactionary Envelope Protection Method," Proceedings of the AIAA Infotech@Aerospace Conference, Keystone, CO, Aug. 2006, doi: 10.2514/6.2006-6219.

115

[14] Yavrucuk, Ilkay, J. V. R. Prasad, and Suraj Unnikrishnan, "Envelope Protection for Autonomous Unmanned Aerial Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 248–261, doi: 10.2514/1.35265.

[15] Tang, Liang, Michael Roemer, Jianhua Ge, Agamemnon Crassidis, J. V. R. Prasad, and Christine M. Belcastro, "Methodologies for Adaptive Flight Envelope Estimation and Protection," Proceedings of the AIAA Infotech@Aerospace Conference, Chicago, IL, Aug. 2009, doi: 10.2514/6.2009-6260.

[16] Keller, Jeffrey D., Robert M. McKillip, and Sungwan Kim, "Aircraft Flight Envelope Determination using Upset Detection and Physical Modeling Methods," Proceedings of the AIAA Infotech@Aerospace Conference, Chicago, IL, Aug. 2009, doi: 10.2514/6.2009-6259.

[17] Urnes, James M., Eric Y. Reichenbach, and Timothy A. Smith, "Dynamic Flight Envelope Assessment and Prediction," Proceedings of the AIAA Guidance, Navigation and Control Conference, Honolulu, HI, Aug. 2008, doi: 10.2514/6.2008-6983.

[18] Pandita, Rohit, Abhijit Chakraborty, Peter Seiler, and Gary Balas, "Reachability and Region of Attraction Analysis Applied to GTM Dynamic Flight Envelope Assessment," Proceedings of the AIAA Infotech@Aerospace Conference, Chicago, IL, Aug. 2009, doi: 10.2514/6.2009-6258.

[19] Gingras, David R., Billy Barnhart, Richard Ranaudo, Borja Martos, Thomas P. Ratvasky, and Eugene Morelli, "Development and Implementation of a Model-Driven Envelope Protection System for In-Flight Ice Contamination," Proceedings of the AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Aug. 2010, doi: 10.2514/6.2010-8141.

[20] Frank, Paul M., and J. Wünnenberg, *Robust Fault Diagnosis Using Unknown Input Observer Schemes*, Ron Patton, Paul M. Frank, and R. Clark, editors, Fault Diagnosis in Dynamic Systems: Theory and Applications, Prentice Hall, NY, 1989, pp. 47–98.

[21] Frank, Paul M., "Online Fault Detection in Uncertain Nonlinear Systems Using Diagnostic Observers: A Survey," *International Journal of Systems Science*, Vol. 25, No. 12, 1994, pp. 2129–2154, doi: 10.1080/00207729408949341.

[22] Chow, Edward Y., and Alan S. Willsky, "Analytical Redundancy and the Design of Robust Failure Detection Systems," *IEEE Transactions on Automatic Control*, Vol. 29, No. 7, July 1984, pp. 603–614, doi: 10.1109/TAC.1984.1103593.

[23] Gertler, Janos J., Xiaowen Fang, and Qiang Luo, *Detection and Diagnosis of Plant Failures: The Orthogonal Parity Equation Approach*, Vol. 37, Control and Dynamic Systems, Academic Press, San Diego, CA, 1990, pp. 159–216.

[24] Basseville, Michèle, and Albert Benveniste, *Detection of Abrupt Changes in Signals and Dynamic Systems*, Lecture Notes in Control and Information Sciences, Vol. 77, Springer-Verlag, Berlin, Germany, 1986.

[25] Isermann, Rolf, *Process Fault Diagnosis Based on Dynamic Models and Parameter Estimation Methods*, Ron Patton, Paul M. Frank, and R. Clark, editors, Fault Diagnosis in Dynamic Systems: Theory and Applications, Prentice Hall, NY, 1989, pp. 253–291.

[26] Cheung, J. T.-Y., and , "Representation of Process Trends, Part I: A formal Representation Framework," *International Journal of Computers and Chemical Engineering*, Vol. 14, No. 4-5, May 1990, pp. 495–510, doi: 10.1016/0098-1354(90)87023-I.

[27] Vedam, Hiranmayee, and Venkat Venkatasubramanian, "A Wavelet Theory-Based Adaptive Trend Analysis System for Process Monitoring and Diagnosis," Proceedings of the American Control Conference, Vol. 1, IEEE, Albuquerque, NM, June 1997, pp. 309–313, doi: 10.1109/ACC.1997.611807.

[28] Bishop, Christopher M., *Pattern Recognition and Machine Learning*, Springer, Singapore, 2006.

[29] Murphy, Kevin P., *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, MA, 2006.

[30] MacGregor, John F., and Theodora Kourti, "Statistical Process Control of Multivariate Processes," *Control Engineering Practice*, Vol. 3, No. 3, March 1995, pp. 403–414, doi: 10.1016/0967-0661(95)00014-L.

[31] Li, R. F., and X. Z. Wang, "Qualitative/Quantitative Simulation of Process Temporal Behavior Using Clustered Fuzzy Digraphs," *American Institute of Chemical Engineers Journal*, Vol. 27, No. 4, April 2001, pp. 906–919, doi: 10.1002/aic.690470413.

[32] Ulerich, N. H., and G. A. Powers, "Online Hazard Aversion and Fault Diagnosis in Chemical Processes: The Digraph + Fault-Tree Method," *IEEE Transactions on Reliability*, Vol. 37, No. 2, June 1988, pp. 171–177, doi: 10.1109/24.3738.

[33] Peng, Y., and J. A. Reggia, "A Probabilistic Causal Model for Diagnostic Problem Solving, Part II: Diagnostic Search," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, No. 3, May 1987, pp. 395–406, doi: 10.1109/TSMC.1987.4309056.

[34] Becraft, W., and P. Lee, "An Integrated Neural Network/Expert System Approach for Fault Diagnosis," *International Journal of Computers and Chemical Engineering*, Vol. 17, No. 10, Oct. 1993, pp. 1001–1014, doi: 10.1016/0098-1354(93)80081-W.

[35] Chen, Lioun Wee, and M. Modarres, "Hierarchical Decision Process for Fault Administration," *International Journal of Computers and Chemical Engineering*, Vol. 16, No. 5, May 1992, pp. 425–448, doi: 10.1016/0098-1354(92)85009-W.

[36] Rich, Steven H., and Venkat Venkatasubramanian, "Causality-Based Failure-Driven Learning in Diagnostic Expert Systems," *American Institute of Chemical Engineers Journal*, Vol. 35, No. 6, June 1989, pp. 943–950, doi: 10.1002/aic.690350607.

[37] Zhao, Jinsong, Bingzhen Chen, and Jingzhu Shen, "A Hybrid ANN-ES System for Dynamic Fault Diagnosis of Hydrocracking Process," *International Journal of Computers and Chemical Engineering*, Vol. 21, No. Supplement, May 1997, pp. S929–S933, doi: 10.1016/S0098-1354(97)87621-1.

[38] Belcastro, Christine M., and Steven R. Jacobson, "Future Integrated Systems Concept for Preventing Aircraft Loss-of-Control Accidents," Proceedings of the AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Aug. 2010, doi: 10.2514/6.2010-8142.

[39] Belcastro, Christine M., "Validation and Verification of Future Integrated Safety-Critical Systems Operating under Off-Nominal Conditions," Proceedings of the AIAA Guidance, Navigation and Control Conference, Toronto, Canada, Aug. 2010, doi: 10.2514/6.2010-8143.

[40] Roemer, Michael, Liang Tang, Sudarshan Bharadwaj, and Celeste Belcastro, "An Integrated Aircraft Health Assessment and Fault Contingency Management System," Proceedings of the AIAA Guidance, Navigation and Control Conference, Honolulu, HI, Aug. 2008, doi: 10.2514/6.2008-6505.

[41] Figueroa, Fernando, John Schmalzel, Mark Walker, Meera Venkatesh, Ravi Kapadia, Jon Morris, Mark Turowski, and Harvey Smith, "Integrated System Health Management: Foundational Concepts, Approach, and Implementation," Proceedings of the AIAA Infotech@Aerospace Conference, Seattle, WA, April 2009, doi: 10.2514/6.2009-1915.

[42] Farmer, J. Doyne, Norman H. Parkard, and Alan S. Perelson, "The Immune System, Adaptation, and Machine Learning," *Physica D: Nonlinear Phenomena*, Vol. 22, No. 1-3, 1986, pp. 187–204, doi: 10.1016/0167-2789(86)90240-X.

[43] Castro, Leandro Nunes, and Jonathan Timmis, *Artificial Immune Systems: A New Computation Intelligence Approach*, Springer, 2002.

[44] Dasgupta, Dipankar, Kalmanje KrishnaKumar, D. Wong, and Misty Berry, "Negative Selection Algorithm for Aircraft Fault Detection," Proceedings of the 3rd International Conference on Artificial Immune Systems, Catania, Sicily, Sept. 2004, pp. 1–13, doi: 10.1007/978-3-540-30220-9_1.

[45] KrishnaKumar, Kalmanje, "Artificial Immune System Approaches for Aerospace Applications," Proceedings of the 41st Aerospace Sciences Meeting & Exhibit, Reno, NV, April 2003, doi: 10.2514/6.2003-457.

[46] González, Fabio, *A Study of Artificial Immune Systems Applied to Anomaly Detection*, Ph.D. thesis, The University of Memphis, Memphis, TN, 2003.

[47] Yeom, Ki-Won, *Immune-Inspired Algorithm for Anomaly Detection*, Computational Intelligence in Information Assurance and Security, Vol. 57, Springer-Verlag, Berlin, Germany, 2007, pp. 129–154, doi: 10.1007/978-3-540-71078-3_5.

[48] Castro, Leandro Nunes, and Fernando J. Von Zuben, *aiNet: An Artificial Immune Network for Data Analysis*, H. A. Abbas, R. A. S. Charles and S. Newton, editors, Data Mining: A Heuristic Approach, Idea Group Publishing, Hershey, PA, 2001, pp. 231–259, doi: 10.4018/978-1-930708-25-9.ch012.

[49] Nasraoui, Olfa, Dipankar Dasgupta, and Fabio González, "A Novel Artificial Immune System Approach to Robust Data Mining," Proceedings of the International Genetic and Evolutionary Computation Conference, New York, NY, July 2002, pp. 356–363.

[50] Forrest, Stepahnie, Steven A. Hofmeyr, Anil Somayaji, and Thomas A. Longstaff, "A Sense of Self for Unix Processes," Proceedings of the IEEE Symposium on Computer Security and Privacy, Oakland, CA, July 1996, pp. 120–128, doi: 10.1109/SECPRI.1996.502675.

[51] Singh, Surya, and Scott M. Thayer, "Immunology Directed Methods for Distributed Robotics: A Novel, Immunity-Based Architecture for Robust Control and Coordination," Proceedings of SPIE: Mobile Robots XVI, Vol. 4573, Boston, MA, Oct. 2001, doi: 10.1117/12.457453.

[52] Lee, Jongan, Mootaek Roh, Jinsung Lee, and Doheon Lee, "Clonal Selection Algorithms for 6-DOF PID Control of Autonomous Underwater Vehicles," Proceedings of the 6th International Conference on Artificial Immune Systems, Santos, Brazil, Aug. 2007, pp. 182–190, doi: 10.1007/978-3-540-73922-7_16.

[53] Carter, Jerome H., "The Immune System as a Model for Pattern Recognition and Classification," *Journal of the American Medical Informatics Association*, Vol. 7, No. 1, Jan.-Feb. 2000, pp. 29–41, doi: 10.1136/jamia.2000.0070028.

[54] Castro, Leandro Nunes, and Jonathan Timmis, *Artificial Immune Systems: A Novel Approach to Pattern Recognition*, J. Corchado, L. Alonso, and C. Fyfe, editors, Artificial Neural Networks in Pattern Recognition, University of Paisley, UK, Jan. 2002, pp. 67–84.

[55] Avizienis, Algirdas, "Toward Systematic Design of Fault-Tolerant System," *Computer*, Vol. 30, No. 4, April 1997, pp. 51–58, doi: 10.1109/2.585154.

[56] Matzinger, Polly, "The Danger Model: A Renewed Sense of Self," *Science*, Vol. 296, No. 5566, April 2002, pp. 301–305, doi: 10.1126/science.1071059.

[57] Greensmith, Julie, Jamie Twycross, and Uwe Aickelin, "Dendritic Cells for Anomaly Detection," Proceedings of the 6th IEEE Congress on Evolutionary Computation, Vancouver, BC, 2006, pp. 664–671, doi: 10.1109/CEC.2006.1688374.

[58] Greensmith, Julie, *The Dendritic Cell Algorithm*, Ph.D. thesis, University of Nottingham, Nottingham, UK, Oct. 2007.

[59] Greensmith, Julie, and Uwe Aickelin, "The Deterministic Dendritic Cell Algorithm," Proceedings of the 7th International Conference on Artificial Immune Systems, Phuket, Thailand, May 2008, pp. 291–302, doi: 10.1007/978-3-540-85072-4_26.

[60] Mokhtar, Maizura, Ran Bi, Jonanthan Timmis, and Andy M. Tyrrell, "A Modified Dendritic Cell Algorithm for Online Error Detection in Robotic Systems," IEEE Congress on Evolutionary Computation, Trondheim, Norway, May 2009, pp. 2055–2062, doi: 10.1109/CEC.2009.4983194.

[61] Chelly, Zeineb, and Zied Elouedi, "FDCM: A Fuzzy Dendritic Cell Method," Proceedings of the 9th International Conference on Artificial Immune Systems, Edinburgh, UK, July 2010, pp. 102–115, doi: 10.1007/978-3-642-14547-6_9.

[62] Dasgupta, Dipankar, editor, *Artificial Immune Systems and Their Applications*, Springer, 1999, doi: 10.1007/978-3-642-59901-9.

[63] Janeway, Charles A., Paul Travers, Mark Walport, and Mark Shlomchik, *Immunobiology: The Immune System in Health and Disease*, Garland Science, New York, 6th ed., 2005.

[64] Steinman, Ralph M., and Zanvil A. Cohn, "Identification of a Novel Cell Type in Peripheral Lymphoid Organs of Mice," *Journal of Experimental Medicine*, Vol. 137, No. 5, May 1973, pp. 1142–1162, doi: 10.1084/jem.137.5.1142.

[65] Steinman, Ralph 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.

[66] Banchereau, Jacques, and Ralph M. Steinman, "Dendritic Cells and the Control of Immunity," *Nature*, Vol. 392, March 1998, doi: 10.1038/32588.

[67] Pletinckx, Katrien, Anja Döhler, Vladimir Pavlovic, and Manfred B. Lutz, "Role of Dendritic Cell Maturity/Costimulation for Generation, Homeostasis, and Suppressive Activity of Regulatory T-Cells," *Frontiers in Immunology*, Vol. 2, Sept. 2011, doi: 10.3389/fimmu.2011.00039.

[68] Segel, Lee A., and Irun R. Cohen, *Design Principles for the Immune System and Other Distributed Autonomous Systems*, Oxford University Press, Oxford, UK, 2001.

[69] Ko, Albert, H. Y. K. Lau, and T. L. Lau, "An Immuno Control Framework for Decentralized Mechatronic Control," *Internation Journal of Unconventional Computing*, Vol. 1, No. 3, 2005, pp. 255–280.

[70] Luh, Guan-Chun, Chun-Yin Wu, and Wei-Chong Cheng, "Artificial Immune Regulation (AIR) for Model-Based Fault Diagnosis," Proceedings of the 3rd International Conference on Artificial Immune Systems, Catania, Sicily, Sept. 2004, pp. 28–41, doi: 10.1007/978-3-540-30220-9_3.

[71] Guzella, Thiago S., Tomaz A. Mota-Santos, and Walmir M. Caminhas, "A Novel Immune Inspired Approach to Fault Detection," Proceedings of the 6th International Conference on Artificial Immune Systems, Santos, Brazil, Aug. 2007, pp. 107–118, doi: 10.1007/978-3-540-73922-7_10.

[72] KrishnaKumar, Kalmanje, A. Satyadas, and J. C. Neidhoefer, "An Immune System Framework for Integrating Computational Intelligence Paradigms," Computational Intelligence, A Dynamic Perspective, IEEE Press, Washington, 1995.

[73] KrishnaKumar, Kalmanje, and J. C. Neidhoefer, "Immunized Neurocontrol," *Journal of Expert Systems Application*, Vol. 13, No. 3, 1997, pp. 201–214.

[74] KrishnaKumar, Kalmanje, and J. C. Neidhoefer, *Immunized Adaptive Critics*, ICNN, Houston, TX, 1997, doi: 10.1109/ICNN.1997.614406.

[75] KrishnaKumar, Kalmanje, and J. C. Neidhoefer, *Immunized Adaptive Critic for an Autonomous Aircraft Control Application*, Dipankar Dasgupta, editor, Artificial Immune Systems and Their Applications, Springer, 1999, pp. 221–241, doi: 10.1007/978-3-642-59901-9_12.

[76] Ootsuki, J. T., and T. Sekiguchi, "Application of the Immune System Network Concept to Sequential Control," Vol. 3, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, 1999, pp. 869–874, doi: 10.1109/ICSMC.1999.823342.

[77] Kaneshige, John T., and Kalmanje KrishnaKumar, "Artificial Immune System Approach for Air Combat Maneuvering," Proceedings of SPIE 6560, Intelligent Computing: Theory and Applications V, Orlando, Florida, 2007, doi: 10.1117/12.718892.

[78] Perhinschi, Mario G., Hever Moncayo, and Jennifer Davis, "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, doi: 10.2514/1.45718.

[79] Moncayo, Hever, Mario G. Perhinschi, and Jennifer Davis, "Aircraft Failure Detection and Identification Using an Immunological Hierarchical Multiself Strategy," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, July 2010, pp. 1105–1114, doi: 10.2514/1.47445.

[80] Moncayo, Hever, Mario G. Perhinschi, and Jennifer Davis, "Aircraft Failure Detection and Identification Over an Extended Flight Envelope Using an Artificial Immune System," *The Aeronautical Journal*, Vol. 115, No. 1163, Jan. 2011, pp. 43–55.

[81] Moncayo, Hever, Mario G. Perhinschi, and Jennifer Davis, "Artificial Immune System-Based Aircraft Failure Evaluation Over Extended Flight Envelope," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, July-Aug. 2011, pp. 9890–1001, doi: 10.2514/6.2010-8038.

[82] Moncayo, Hever, Mario G. Perhinschi, Brenton Wilburn, Jennifer Wilburn, and Ondrej Karas, "UAV Adaptive Control Laws Using Nonlinear Dynamic Inversion Augmented with an Immunity-based Mechanism," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, Aug. 2012, doi: 10.2514/6.2012-4678.

[83] Davis, Jennifer, *The Design of an Evolutionary Algorithm for Artificial Immune System Based Failure Detector Generation and Optimization*, Master's thesis, West Virginia University, Aug. 2010.

[84] Davis, Jennifer, Mario G. Perhinschi, and Hever Moncayo, "Evolutionary Algorithm for Artificial Immune System-Based Failure Detector Generation and Optimization," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, March-April 2010, pp. 302–320, doi: 10.2514/1.46126.

[85] Oates, Robert, Julie Greensmith, Uwe Aickelin, Jonathan Garibaldi, and Graham Kendall, "The Application of a Dendritic Cell Algorithm to a Robotic Classifier," Proceedings of the 6th International Conference on Artificial Immune Systems, 2007, pp. 204–215, doi: 10.1007/978-3-540-73922-7_18.

[86] Ogundipe, Oluropo, Julie Greensmith, and Gethin Roberts, "Multipath Detection Using the Dendritic Cell Algorithm," XXIV FIG International Congress, 2010, pp. 1–14.

[87] Bendiab, Esma, and Mohamed K. Kholladi, "A Method for Plant Classification Based on Artificial Immune System and Wavelet Transform," Proceedings of the International Conference on Digital Information and Communication Technology and Its Applications, Dijon, France, June 2011, pp. 199–208, doi: 10.1007/978-3-642-21984-9_17.

[88] Amaral, Jorge L., "Fault Detection in Analog Circuits Using a Fuzzy Dendritic Cell Algorithm," Proceedings of the 10th International Conference on Artificial Immune Systems, Cambridge, UK, July 2011, pp. 294–307, doi: 10.1007/978-3-642-22371-6_26.

[89] Pinto, Jose Carlos L., and Fernando J. Von Zuben, "Fault Detection Algorithm for Telephone Systems Based on the Danger Theory," Proceedings of the 4th International Conference on Artificial Immune Systems, Banff, Alberta, Aug. 2005, pp. 418–431, doi: 10.1007/11536444_32.

[90] Moguel, Israel, Hever Moncayo, Mario G. Perhinschi, Andres Perez, Dia Al Azzawi, and Adil Togayev, "Structured Nonself Approach for Aircraft Failure Identification Within an Immunity-Based Fault Tolerance Architecture," *IEEE Transactions on Aerospace and Electronic Systems*, Jan. 2014, submitted, Manuscript ID: TAES-201400058.

[91] Sagoo, Girish K., Srikanth Gururajan, Marcello R. Napolitano, Mario G. Perhinschi, Gu Yu, and Brad Seanor, "Pilot-in-the-Loop Assessment of Neurally Augmented Dynamic Inversion Based Fault Tolerant Control Laws in a Motion-Based Flight Simulator," AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, Aug. 2008, doi: 10.2514/6.2008-6843.

[92] Meyer, Austin, and H. Van Kampen, *X-Plane Online Instruction Manual*, Laminar Research Inc., Columbia, SC, 8th ed., 2002.

[93] Perhinschi, Mario G., and Marcello R. Napolitano, "Teaching Aircraft Health Management: A Simulation-Based Approach," *Computers in Education Journal*, Vol. 26, No. 1, Oct.-Dec. 2009, pp. 1–10.

[94] Antoniewicz, Robert F., Eugene L. Duke, and Brian P. Patterson, "User's Manual for Interactive LINEAR, a Fortran Program to Derive Linear Aircraft Models," 1988.

[95] Perhinschi, Mario G., Marcello R. Napolitano, Giampiero Campa, and Mario L. Fravolini, "A Simulation Environment for Testing and Research of Neurally Augmented Fault Tolerant Control Laws Based on Nonlinear Dynamic Inversion," Proceedings of the AIAA Modeling and Simulation Technologies Conference, Providence, RI, Aug. 2004, doi: 10.2514/6.2004-4913.

[96] Moncayo, Hever, and Mario G. Perhinschi, *Aircraft Fault Tolerance: A Biologically Inspired Immune Framework for Subsystem Failures*, VDM Verlag Dr. Muller GmbH & Co. KG, VDM Publishing House ltd., Saarbruecken, Germany, 2011.

[97] Perhinschi, Mario G., Giampiero Campa, Marcello R. Napolitano, M. Lando, L. Massotti, and Mario L. Fravolini, "Modeling and Simulation of a Fault Tolerant Control System," *International Journal of Modelling and Simulation*, Vol. 26, No. 1, Jan. 2006, pp. 1–10, doi: 10.2316/Journal.205.2006.1.205-4044.

[98] Perhinschi, Mario G., Marcello R. Napolitano, Giampiero Campa, Mario L. Fravolini, and Brad Seanor, "Integration of Sensor and Actuator Failure Detection, Identification, and Accommodation Schemes within Fault Tolerant Control Laws," *Control and Intelligent Systems*, Vol. 35, No. 4, Dec. 2007, pp. 309–318, doi: 10.2316/Journal.201.2007.4.201-1657.

[99] Moncayo, Hever, Mario G. Perhinschi, and Jennifer Davis, "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, Aug. 2011, doi: 10.2514/6.2011-6697.

[100] Perhinschi, Mario G., Hever Moncayo, Dia Al Azzawi, and Israel Moguel, "Generation of Artificial Immune System Antibodies Using Raw Data and Cluster Set Union," *IC: International Journal of Immune Computation*, Vol. 2, No. 1, March 2014, pp. 1–15, http://www.aicit.org/ic/global/paper_detail.html?jname=IC&q=82.

[101] Moncayo, Hever, *Immunity-Based Detection, Identification, and Evaluation of Aircraft Subsystem Failures*, Ph.D. thesis, West Virginia University, 2009.

[102] Steinman, Ralph M., and H. Hemmi, *Dendritic Cells: Translating Innate to Adaptive Immunity*, Bali Pulendran and Rafi Ahmed, editors, Current Topics in Microbiology and Immunology, Vol. 311, Springer-Verlag, Berlin, Germany, 2006, pp. 17–58, doi: 10.1007/3-540-32636-7_2.

[103] Tarbell, Kristin V., Sayuri Yamazaki, and Ralph M. Steinman, "The Interactions of Dendritic Cells with Antigen-Specific, Regulatory T-Cells that Suppress Autoimmunity," *Seminars in Immunology*, Vol. 18, No. 2, April 2006, pp. 93–102, doi: 10.1016/j.smim.2006.01.009.

[104] Al Azzawi, Dia, Mario G. Perhinschi, and Hever Moncayo, "Dendritic Cell Mechanism for Aircraft Immunity-Based Failure Detection and Identification," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, Aug. 2013, doi: 10.2514/6.2013-5185.

[105] Al Azzawi, Dia, Mario G. Perhinschi, and Hever Moncayo, "Artificial Dendritic Cell Mechanism for Aircraft Immunity-Based Failure Detection and Identification," *AIAA Journal of Aerospace Information Systems*, Vol. 11, No. 7, July 2014, pp. 536–550, doi: 10.2514/1.I010214.

[106] Perhinschi, Mario G., Hever Moncayo, and Dia Al Azzawi, "Development of Immunity-Based Framework for Aircraft Abnormal Conditions Detection, Identification, Evaluation, and Accommodation," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, Aug. 2013, doi: 10.2514/6.2013-5184.

[107] Al Azzawi, Dia, Mario G. Perhinschi, Hever Moncayo, Adil Togayev, Andres Perez, and Israel Moguel, "Evaluating Aircraft Abnormal Conditions Using an Artificial Dendritic Cell Mechanism," 2nd International Conference and Exhibition on Mechanical & Aerospace Engineering, Philadelphia, PA, Sept. 2014.

[108] Perhinschi, Mario G., Jaclyn Porter, Hever Moncayo, Jennifer Davis, and Scott Wayne, "Artificial-Immune-System-Based Detection Scheme for Aircraft Engine Failures," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5, Sep.-Oct. 2011, pp. 1423–1440, doi: 10.2514/1.52746.

[109] Kohavi, Ron, and George H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, Vol. 97, No. 1-2, Dec. 1997, pp. 273–324, doi: 10.1016/S0004-3702(97)00043-X.

[110] Al Azzawi, Dia, Hever Moncayo, Mario G. Perhinschi, Andres Perez, and Adil Togayev, "Comparison of Immunity-Based Schemes for Aircraft Failure Detection and Identification," *IEEE Computational Intelligence Magazine*, June 2014.

[111] Moguel, Israel, Hever Moncayo, Mario G. Perhinschi, Andres Perez, Dia Al Azzawi, and Adil Togayev, "Bio-Inspired Approach for Aircraft Health Assessment and Flight Envelope Estimation," ASME 2014 Dynamic Systems & Control Conference, San Antonio, TX, Oct. 2014.

[112] Perez, Andres, Hever Moncayo, Israel Moguel, Mario G. Perhinschi, Dia Al Azzawi, and Adil Togayev, "Immunity-Based Adaptive Control Laws for Aircraft Fault Tolerance," ASME 2014 Dynamic Systems & Control Conference, San Antonio, TX, Oct. 2014.

[113] Perhinschi, Mario G., Hever Moncayo, and Dia Al Azzawi, "Integrated Immunity-Based Framework for Aircraft Abnormal Conditions Management," *AIAA Journal of Aircraft*, April 2013, accepted for publication, Manuscript ID: 2013-04-C032381.R1.

[114] Perhinschi, Mario G., and Dia Al Azzawi, "Undergraduate Experiential Learning Lab for Aircraft Parameter Identification," *Computers in Education (CoED) Journal*, Vol. 5, No. 2, April-June 2014, pp. 79–92.

[115] Perhinschi, Mario G., Dia Al Azzawi, Hever Moncayo, Adil Togayev, and Andres Perez, "Immunity-Based Flight Envelope Prediction at Post-Failure Conditions," *AIAA Journal of Aerospace Information Systems*, June 2014.

[116] Togayev, Adil, Mario G. Perhinschi, Dia Al Azzawi, Hever Moncayo, Israel Moguel, and Andres Perez, "Immunity-Based Abnormal Condition Accommodation of Aircraft Subsystem Failures," ASME 2014 Dynamic Systems & Control Conference, San Antonio, TX, Oct. 2014.

[117] Perez, Andres, Hever Moncayo, Mario G. Perhinschi, Adil Togayev, and Dia Al Azzawi, "Pilot-in-the-Loop Evaluation of a Bio-Inspired Adaptive Fault Tolerant Control System in a Motion Based Flight Simulator," AIAA Science and Technology Forum, Kissimmee, Florida, Jan. 2015.