Dissertations and Theses

4-2014

Bio-Inspired Mechanism for Aircraft Assessment Under Upset Conditions

Israel A. Moguel
Embry-Riddle Aeronautical University - Daytona Beach

Follow this and additional works at: https://commons.erau.edu/edt

Part of the Aerospace Engineering Commons

BIO-INSPIRED MECHANISM FOR AIRCRAFT ASSESSMENT UNDER UPSET
CONDITIONS

by

Israel A. Moguel

A Thesis Submitted to the College of Engineering Department of Mechanical Engineering in
Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Embry-Riddle Aeronautical University
Daytona Beach, Florida
April 2014

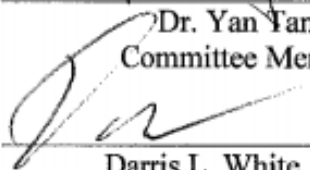# BIO-INSPIRED MECHANISM FOR AIRCRAFT ASSESSMENT UNDER UPSET CONDITIONS
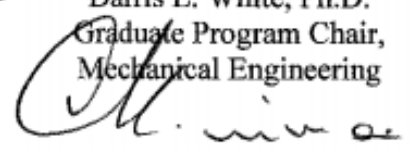
by

Israel A. Moguel

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Hever Moncayo, Assistant Professor, Daytona Beach Campus, and Thesis Committee Members Dr. Yan Tang, Assistant Professor, Daytona Beach Campus, and Dr. Richard Prazenica, Assistant Professor, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering
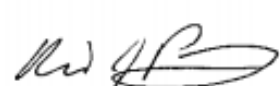
Thesis Review Committee:
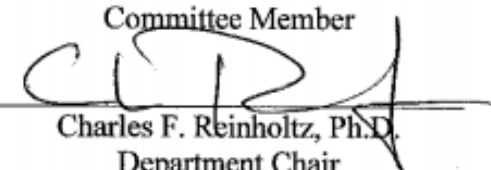
_____
Dr. Hever Moncayo
Committee Chair

_____
Dr. Yan Tang
Committee Member

_____
Dr. Richard Prazenica
Committee Member

_____
Darris L. White, Ph.D.
Graduate Program Chair,
Mechanical Engineering

_____
Charles F. Reinholtz, Ph.D.
Department Chair
Mechanical Engineering

_____
Maj Mirmirani, Ph.D.
Dean, College of Engineering

_____
Robert Oxley, Ph.D.
Associate Vice President of Academics

4-23-2014
Date

Acknowledgements

Abstract

Researcher: Israel A. Moguel

Title: Bio-Inspired Mechanism for Aircraft Assessment Under Upset Conditions

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Mechanical Engineering

Year: 2014

Based on the artificial immune systems paradigm and a hierarchical multi-self strategy, a set of algorithms for aircraft sub-systems failure detection, identification, evaluation and flight envelope estimation has been developed and implemented. Data from a six degrees-of-freedom flight simulator were used to define a large set of 2-dimensional self/non-self projections as well as for the generation of antibodies and identifiers designated for health assessment of an aircraft under upset conditions. The methodology presented in this paper classifies and quantifies the type and severity of a broad number of aircraft actuators, sensors, engine and structural component failures. In addition, the impact of these upset conditions on the flight envelope is estimated using nominal test data. Based on immune negative and positive selection mechanisms, a heuristic selection of sub-selves and the formulation of a mapping- based algorithm capable of selectively capturing the dynamic fingerprint of upset conditions is implemented. The performance of the approach is assessed in terms of detection and identification rates, false alarms, and correct prediction of flight envelope reduction with respect to specific states. Furthermore, this methodology is implemented in flight test by using an unmanned aerial vehicle subjected to nominal and four different abnormal flight conditions instrumented with a low cost microcontroller.

Table of Contents

List of Tables

List of Figures

Nomenclature

English

H	Altitude
V	Ground Speed
M	Mach Number
a_x, a_y, a_z	Linear Accelerations
p, q, r	Roll Rate, Pitch Rate and Yaw Rate
$p_{ref}, q_{ref}, r_{ref,}$	Reference Roll, Pitch and Yaw Rates
$p_{cmd}, q_{cmd}, r_{cmd,}$	Commanded Roll, Pitch and Yaw Rates
NN_p, NN_q, NN_r	Neurally Estimated Roll, Pitch and Yaw Rates
d_e	Longitudinal Stick Control
d_a	Lateral Stick Control
d_r	Directional Stick Control
d_T	Throttle Control
$\dot{p}, \dot{q}, \dot{r},$	Roll, Pitch and Yaw Rate Accelerations
δ_e	Elevator/Stabilator Deflection
δ_a	Aileron Deflection
δ_r	Rudder Deflection
δ_T	Throttle Valve Positions
C_L	Lift Coefficient
C_M	Pitch Coefficient
T	Thrust
U_p, U_q, U_r	Roll, Pitch and Yaw Pseudo Accelerations

Greek

α	Angle of Attack
β	Angle of Sideslip
γ	Flight and Angle and/or Learning Rate
θ	Pitch Angle
φ	Bank Angle
ψ	Heading Angle

Acronyms

ADALINE	Adaptive Linear Network
AIS	Artificial Immune System
ANN	Artificial Neural Network
APC	Antigen Presenting Cells
APM	ArduPilot Mega
BIS	Biological Immune System
CI	Computer Intelligence
CSP	Clonal Selection Principle
CSUM	Cluster Set Union Method
DC	Dendritic Cells
DIV	Directly Involved Variable
DOF	Degrees of Freedom

DQEE	Decentralized Quadratic Estimation Error
DR	Detection Rate
DT	Danger Theory
EA	Evolutionary Algorithm
EDIV	Equivalent Direct Involved Variable
ENSA-RV	Enhanced Negative Selection Algorithm for Real-Values
ERAU	Embry-Riddle Aeronautical University
ERV	Envelope Related Variable
FA	False Alarms
FDIE	Failure Detection, Identification and Evaluation
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GPS	Global Positioning System
HMS	Hierarchical Multi-Self
IMU	Inertial Measurement Unit
INM	Immune Network Models
MIMO	Multiple Input- Multiple Output
MPHF	Maneuverability, Performance Handlin Qualities and Flight Envelope
MQEE	Main Quadratic Estimation Error
NLDI	Nonlinear Dynamic Inversion
NS	Negative Selection
OQEE	Output Quadratic Estimation Error
PS	Positive Selection
SNSA	Structured Non-Self Approach
TN	True Negative
TP	True Positive
WVU	West Virginia University

Chapter 1 Introduction

Aircraft operational safety is the most important objective in the aerospace engineering community [1, 2]. The improvements in post-failure flight integrity and safety have become an essential asset of high importance in military and civil aircrafts. In recent years, numerous research efforts have focused on the development of fault-tolerant flight control systems that offer a comprehensive, highly effective and integrated solution to the aircrafts sub-system failure detection, identification and evaluation (FDIE) problem [3,4]. One of the most promising candidates that offers a solution to this problem is the Artificial Immune System (AIS) model [4, 5]. Integrated high-performance AIS-FDI schemes have shown in the past their capability of handling several categories of aircraft sub-system upset conditions over extended areas of the flight envelope while maintaining high levels of performance and control of the aircraft [6-8].

The AIS approach for FDIE investigated in this research effort is based on a Hierarchical Multi-Self Strategy (HMS), which dictates that a specific set of parameters or "features" may favor the detection of some particular failures [9]. This approach relies on the availability of large amounts of experimental data at nominal conditions that represent the "self" in the hyper-space. These data are also used to generate sets of "detectors" that cover the remaining regions of the hyper-space called "non-self". A predefined set of features or variables must be determined prior to the generation of "detectors". These features must be capable of capturing the dynamic fingerprint for various sub-system failures. The selection of appropriate features to build the "self" and "non-self" in the hyper-space is a crucial step for the effectiveness of the AIS-FDIE scheme. It has been shown in previous research efforts that the selection of features that define

the self/non-self can be reduced to aircraft variables, pilot inputs, variables generated within the control laws and derived variables

The "detection" phase represents the process of declaring a generic failure in one or more of the aircraft sub-systems. The systems can be classified as actuators, sensors, propulsion or structural elements. The detection phase must be designed thoroughly and it must cover aspects such as the aircraft's operational envelope, the sub-system targeted, and the nature and type of the abnormal conditions that are expected to be detected. With all this considered, a detection logic must be designed for real time operation with high detection rates and low false alarm rates.

Once a failure has been detected, the "identification" phase starts. The identification phase determines which sub-system has failed by analyzing which of the detectors has been activated through a positive selection-type (PS) scheme. In this phase, all the detectors are labeled in a previous offline process in order to assign specific detectors to particular categories of failures. This off-line process or "structuring" consists of outlining which non-selves are activated under a specific failure. Depending on the complexity of the targeted systems, the identification process can determine which sub-systems have failed (i.e. actuator, sensor, structural, etc.).

The evaluation phase can be divided into two steps, namely direct evaluation and indirect evaluation. Furthermore, the direct evaluation phase can be classified into qualitative and quantitative evaluation. The direct qualitative evaluation phase isolates and determines the specific subsystem that has failed (i.e. Left Aileron, Right Aileron, Right Engine, Left Wing etc.). The quantitative evaluation phase determines the severity or magnitude of a failed subsystem. In other words, it can determine at which position a control surface is locked or the

percentage of structural damage that a wing has suffered. Finally, the indirect evaluation phase determines the effect on the flight envelope maneuverability and performance of the system after a failure has affected the system.

In this research effort, two-dimensional projections (meaning two features per self) of the self/non-self were created and investigated. Extensive sets of experimental data were clustered, fused and later processed into detectors by a process called the "Cluster Set Union Method" (CSUM) [10]. The process of generating detectors requires an adequate numerical representation of the self/non-self and adequate data processing such that they are manageable given the hardware limitations [10]. A total of 496 two-dimensional selves were used to generate the corresponding sets of detectors. Their performance was analyzed in terms of detection rate (DR) and false alarms (FA). The detection rate was computed as a percentage of all data samples that triggered a detector in the presence of an abnormal condition. Similarly, false alarms were computed as the percentage of all data samples that triggered a detector at nominal conditions. Similarly, the identification phases investigated presents a Structured Non-Self approach in which the direct evaluation stage is integrated into a single algorithm in which the performance of the algorithms is analyzed in terms of correct and incorrect identification [10].

In addition to this research effort, the application of artificial neural networks (ANN) for FDIE purposes in real time is also investigated through flight tests. A "model following" architecture is used to generate reference flight commands that satisfy desired handling qualities. The reference signals are processed by the ANN, which calculates and estimates angular rates on the three body axes (roll, pitch, and yaw) within an "open-loop" approach. The ANN investigated possess' the capability to capture the dynamic fingerprint of the aircraft as well as the dynamic fingerprint produced by sub-systems failures. This characteristic gives the ANN an

immense potential for self definition and FDI. The output parameters of the ANN combined with other features are considered in order to define and generate "selves".

The neural network (NN) algorithm investigated in this effort is an "Adaptive Linear Neuron (ADALINE) network. In this effort, two networks working in parallel have been implemented on each of the three channels. This approach enhances the performance of the network without the computational burden.

This thesis was part of a research project funded by DARPA as a group effort between Embry-Riddle Aeronautical University (ERAU) and West Virginia University (WVU) graduate students and faculty.

The thesis document is organized as follows: Chapter 2 describes the methodology and algorithms used for the generation of detectors as well as the simulation environment. Chapter 3 presents the methodology and algorithms used in the identification phase. Chapter 4 is dedicated to the definition and the process of estimating the flight envelope reduction process. In Chapter 5, the definition and description of the ANN investigated and the model following control is presented. Chapter 6 describes the flight test program and the results obtained. A conclusion of the project and recommendations for future work is provided in Chapter 7.

The research effort presented in this thesis has resulted in the publication or submission of:

Journals

Moguel, I., Moncayo, H., Perhinschi M. G., Perez, Andres., Al Azzawi, D., Togayev, A., *Structured Non-Self Approach for Aircraft Failure Identification within an Immunity-Based Fault Tolerance Architecture*, Submitted to EIEEE Journal Transaction in Aerospace and Electronics Systems, 2014.

Perhinschi M. G., Moncayo H., Al Azzawi, D., Moguel, I., *Generation of Artificial Immune System Antibodies Using Raw Data and Cluster Set Union*, International Journal of Immune Computation, Vol. 2, No. 1, pp. 1 ~15, 2014

Conference proceedings

Moguel I., Moncayo H., Perhinschi M. G., Perez A. E., Al Azzawi D., Togayev A., *Bio-Inspired Approach for Aircraft Health Assessment and Flight Envelope Estimation*, abstract submitted to ASME Annual Dynamic Systems and Control Conference, San Antonio, Texas, Oct. 2014

Moguel I., Moncayo H., Perhinschi M. G., Al Azzawi D., Perez A. E., Togayev A., *Biologically-Inspired Approach for Aircraft Management under Upset Conditions*, accepted for presentation at International Conference and Exhibition on Mechanical and Aerospace Engineering, Philadelphia, Pennsylvania, Sept. 2014

Perez A. E., Moguel I., Moncayo H., Chong C. May; *Low Cost Autopilot System for an Autonomous Unmanned Aerial System*, accepted for presentation at International Conference and Exhibition on Mechanical and Aerospace Engineering, Philadelphia, Pennsylvania, Sept. 2014

Al Azzawi D., Perhinschi M. G., Togayev A., Moncayo H., Moguel I., Perez A. E., *Evaluating Aircraft Abnormal Conditions Using an Artificial Dendritic Cell Mechanism*, accepted for presentation at International Conference and Exhibition on Mechanical and Aerospace Engineering, Philadelphia, Pennsylvania, Sept. 2014

Togayev A., Perhinschi M. G., Al Azzawi D., Moncayo H., Moguel I., Perez A. E., *Immunity-Based Abnormal Condition Accommodation of Aircraft Sub-system Failures*, abstract submitted to ASME Annual Dynamic Systems and Control Conference, San Antonio, Texas, Oct. 2014

Perez A. E., Moncayo H., Moguel I., Perhinschi M. G., Al Azzawi D., Togayev A., *Immunity-based Adaptive Control Laws for Aircraft Fault Tolerance*, abstract submitted to ASME Annual Dynamic Systems and Control Conference, San Antonio, Texas, Oct. 2014

Lyons, B., Moncayo, H., Noriega, A., Moguel, I., Perhinschi, M., *Hardware-in-the-Loop Simulation of an Extended Non-linear Dynamic Inversion Augmented with an Immunity-Based Adaptive Control System*, AIAA Guidance, Navigation, and Control Conference, Boston, Massachusetts, August 2013.

1.1 Literature Review

Computational Intelligence (CI) methodologies inspired by the biological immune system (BIS) emerged in the early nineties as a new approach to address complex real-world problems in several fields of study [4]. This bio-inspired approach has led to the development of several models such as: genetic algorithms (GA), artificial neural networks (ANN), fuzzy logic algorithms and artificial immune systems (AIS). The AIS is a fairly recent paradigm for which several models such as Immune Network Models (INM), Danger Theory (DT), Clonal Selection Principle (CSP), Negative Selection algorithm (NS) and Dendritic Cell (DC) Algorithms, among others, have been developed [11-14]. These AIS mechanisms have shown a promising potential in a variety of applications such as pattern recognition [6], computer security [13,14], data mining [15,16], adaptive controls [17,18] and anomaly detection [19, 20].

The biological immune system is a complex and adaptive system that protects organisms from invading pathogens [12]. Historically, the term *immunity* refers to the condition in which an organism can resist or repel infectious diseases [12]. In other words, immunity is the ability of an organism to react against foreign or dangerous substances. The immune system has many characteristics such as uniqueness, autonomy, distributed detection, noise tolerance, foreigners recognition and pattern recognition. These characteristics can be used to distinguish between foreign cells (non-self) and the body cells (self) that may damage the organism. Immune systems usually work based on two mechanisms: innate and adaptive immunity. Innate immunity is directed against general pathogens while the adaptive immunity launches attacks against invader cells that the innate system cannot remove.

The artificial immune system can be defined as a computational paradigm that is inspired by theoretical immunology, observed immune functions, principles and mechanisms. The AIS uses computational models that mimic the behavior of the biological immune system. The fundamental idea of this computational paradigm is that an abnormal condition (i.e. sub-system failure) can be declared when a current configuration of features does not match with any other configuration of features in nominal conditions. A shown in Figure 1, several techniques have been developed to achieve an integrated and effective framework that mimics the BIS [21].
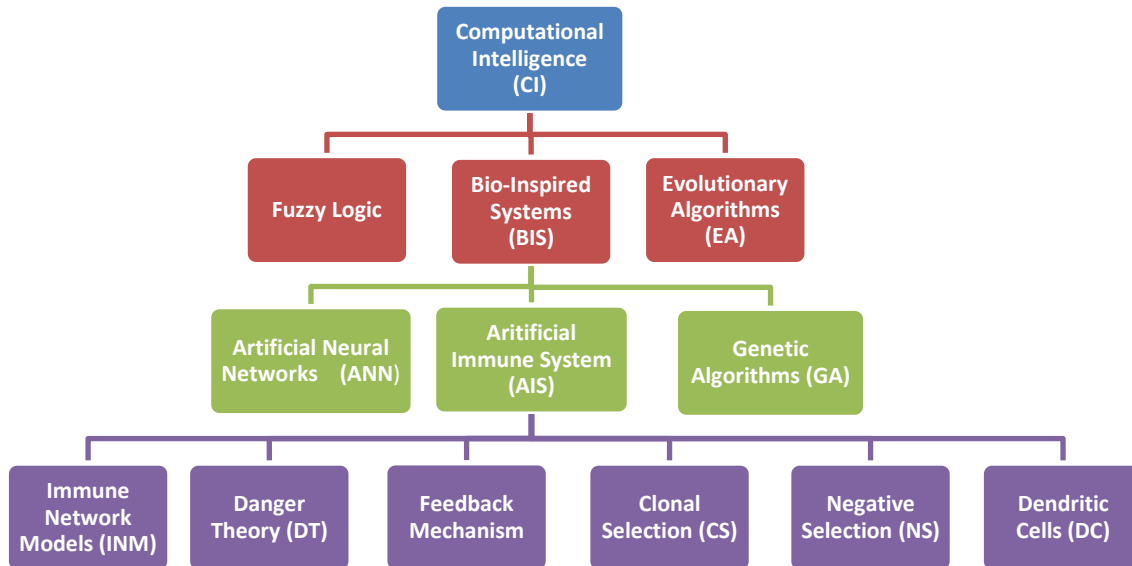
Figure 1- Subdivisions of Computational Intelligence

Immune Network Model

This theory was proposed in the mid-seventies by Jerne. The hypothesis establishes that the immune system maintains an idiotypic network of interconnected B-cells for antigen recognition. These cells stimulate and suppress each other in a way that allows the network to be stable. Two B-cells are connected if the affinities they share exceed a certain threshold, and the strength of the connection is directly proportional to the affinity they share [21].

18

Danger Theory

This theory was primarily developed by Matzinger. The central idea dictates that the immune system does not respond to non-self but to danger. Thus, it supports the need for discrimination. In other words, instead of responding to foreignness, the immune system reacts to danger. Danger is measured by the damage to cells indicated by distress signals that are sent out when a cell suffers an unnatural death. The distress signal in this case is not an abstract representation of danger but a grounded signal [22].

Feedback Mechanism

This approach is inspired by the feedback mechanism of the immune system. The immune system exhibits two types of responses, namely: humoral response and cellular response. In the humoral response, antibodies are produced by B-cells to neutralize antigens. In the cellular response, killer T-cells capture infected cells to kill them. After foreign materials are detected by antigen presenting cells (APC), the APCs transfer information to the helper T-cells. The helper T-cells then stimulate B-cells, killer T-cells and suppressor T-cells. The regulation between B-cells and killer T-cells is considered the main feedback mechanism of the immune system. The suppressor T-cells are an inhibitive mechanism. The helper T-cells and the foreign material activate the suppressor T-cells, which inhibit the other cells by creating a tranquilizing action in the immune systems. This second feedback is called the inhibitive mechanism. The cooperation between these two feedback mechanisms allows the rapid response of the immune system to foreign materials and also it quickly stabilize the immune system when needed [18].

Clonal Selection

The clonal selection principle is the whole process of antigen recognition, cell proliferation and differentiation into memory cells [11,12, 21]. This principle describes the basic

features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigen proliferate, thus being selected against those that do not. Its main features are that:

- The new cells are clones of their parents subjected to a mutation mechanism with high rates.

- Elimination of newly differentiated lymphocytes carrying self-reactive receptors.

- Proliferation and differentiation on contact of mature cells with antigens.

Negative Selection

This method has inspired most of the existing AIS. The objective of the negative selection is to provide tolerance for self cells. It deals with the immune system's ability to detect unknown antigens while not reacting to the self cells. During the maturation of T-cells, if a T-cell in the thymus recognizes any self cell, it is eliminated before deploying it for immune functionality. The negative selection algorithm generates detectors set by eliminating any detector candidate that match elements from a group of self samples [5, 11, 19, 21, 23].

Dendritic Cells

Dendritic cells are responsible for some of the initial pathogenic recognition process, sampling the environment and differentiating depending on the concentration signal, or perceived misbehavior, in the host tissue cell. They provide a vital link between the innate and adaptive immune system. DCs perform the role of coordinating T-cells based immune responses, both reactive and for generation tolerance. The DCs collect antigens from pathogens and host cells tissues, and present multiple antigen samples to naive T-cells in the lymph node [22].

1.1.1 Artificial Immune System and the Failure Detection and Identification Problem

Damage and failures of aircraft sub-systems is a leading source of aircraft accidents throughout the world. Such events, although not very common, have led to a large number of fatalities [2, 3]. Incidents such as the DC-10 American Airlines Flight 191 in 1979 (272 fatalities) [25], B747 Japan Airlines Flight 123 in 1985 (520 fatalities) [24] or the A310 Air Transat Flight 961 in 2005 (0 fatalities) [26] have inspired effort toward the development of fault tolerant systems [3]. Although not all the examples presented above resulted in catastrophes, it should be noted that some of them could have been recovered if a nonstandard set of procedures had been used.

In recent years, the development of fault-tolerant flight controls emerged as a new methodology to increase safety and enhance performance of civilian and military aircraft. Failure detection and identification for aircraft over extended areas of the flight envelope presents a multi-dimensional and highly complex challenge that needs to be addressed by an integrated solution. Previous approaches to the fault detection and identification problem focused on individual classes of failures at isolated and constrained flight conditions [27].

In response to the lack of effectiveness of previous methodologies, the AIS emerged as a comprehensive and integrated solution to the FDIE problem for aircraft sub-systems, showing promising capabilities [7-9, 21]. The application of an integrated high performance AIS-based FDIE for a wide variety of aircraft sub-systems failures based on a hierarchical multi-self strategy has been proposed and tested by several researchers in the aerospace community. The AIS paradigm can address directly the complexity and the multi-dimensionality of aircraft

21

dynamic response to abnormal conditions with a proven record of high detection rates and low false alarm rates. The hierarchical multi-self strategy is capable of detecting and identifying several categories of abnormal conditions over extended areas of the flight envelope. Its effectiveness in terms of false alarms and detection rate has been tested in simulation.

The basic idea of AIS-based FDIE is that an abnormal condition can be declared when a current configuration of features does not match with any set of known normal conditions previously generated through simulation [27-31]. These features can include various sensor outputs, state estimates, stability and control derivatives, pilot inputs and derived variables. The combination of these features should possess enough information to clearly describe the behavior of the system as well as to capture the dynamic signature of abnormal situations. Extensive experimental data are necessary to determine the self and the non-self in the hyperspace. Computational and storage limitations of the available hardware are a key factor in the processing of the data in order to maintain an adequate numerical representation of the self/ non-self.

The processes of detection and identification must be performed in subsequent phases to increase the efficiency and reliability of the FDIE solution. Detection is the process of declaring that a generic malfunction of the system has occurred, assuming that any or several subsystems can be subjected to failure. The subsystems can be actuators, sensors, propulsion, structural elements and any other component of the aircraft not limited to hardware (i.e. software, human pilot, atmospheric conditions, etc.). The identification process has two or more phases depending on the complexity of the subsystems. The first phase of identification consists of determining which subsystem has failed. In other words, in which category does the failure fall. The second phase of identification determines the failed element (e.g. aileron actuator, pitch sensor or right

22

wing). In some cases, an intermediate step could be defined in order to distinguish among groups within the subsystem. For example, if an actuator failure is declared, the intermediate phase of identification would determine which of the controls has failed.

The aircraft subsystem AIS-based FDIE can be considered to include two main processes, namely 1) preprocessing of information and flight data and 2) online detection and identification. The preprocessing of data consists of data recording from simulation to the generation of detectors for the various phases of the FDIE. It includes activities such as the definition of identifiers and detectors, data acquisition, data reduction and detector generation and optimization. The online FDIE process includes the development and application of FDIE schemes. At this stage, detectors are compared against sets of current values of identifiers measured in flight at a certain sampling rate. At each sample, a binary output (i.e., 0 for normal or 1 for abnormal) determines if the current values are inside a detector (abnormal condition) or outside a detector (normal condition). The FDIE scheme utilizes sets of output values over moving time windows, reducing the number of false alarms. Figure 2- illustrates a general flowchart of the FDI process.
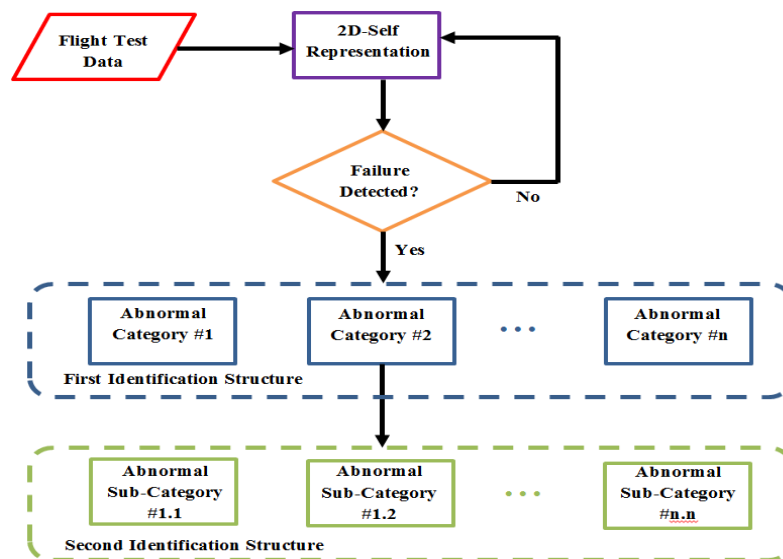
Figure 2- FDI Algorithm

23

1.1.2 Artificial Neural Networks and the FDI Problem

Artificial Neural Networks (ANN) emerged in the field of computer science inspired by the examination of the central nervous system in the early 1940's [32]. ANNs are computational mathematical models that mimic the behavior of neurons located in the brain. Natural neurons receive signals through synapses located on the membranes of the neurons. The neuron is activated and emits a signal through the axon. This signal might be sent to another synapse or may activate other neurons [33, 34]. Similarly, artificial neurons or "nodes" are non-linear functions that act as multiple input-output (MIMO) systems that receive signals, process information, produce results and transmit such results to all its outputs. ANNs are data-driven, self-adaptive models that can capture subtle functional relationships among noisy data, they can capture complex non-linear relationships and primarily, they can automatically adjust themselves to optimize their behavior [32, 35, 36]. ANNs have been utilized for system identification, clustering, vector quantification, pattern association/recognition, feature detection, optimization and control.

The application of Neural Networks in the aerospace industry has grown vastly in the past several years. Its applications evolved from the simplest model validation techniques to more advanced adaptive control law algorithms. Some of the areas in which ANNs have been used in aerospace applications are [32-36]:

- Control Systems

- Stability Augmentation Systems

- Process Modeling

- Radar Signal Processing

- Sensory Information Processing

- Computer Vision

In recent years, researchers and engineers have shown interest in the utilization of ANNs for the development of intelligent flight control systems that enhance aircraft control during physical failure or aerodynamic changes resulting from failures in modeling errors. Commonly, the neural networks are used to generate command augmentation signals to compensate for errors caused by un-modeled dynamics. Other applications of ANN in control law design consists of augmenting Non-Linear Dynamic Inversion (NLDI) controllers with pre-trained neural networks (NLDI+NN), providing the values of the aerodynamic and stability derivatives within the whole flight envelope while cancelling the errors introduced by the dynamic inversion [32].

In addition to this approach, reference control models have been introduced in order to filter command inputs in order to obtain desired handling qualities while the NLDI+NN is used to determine necessary control surface deflections. The neural network system accommodates large errors that are not anticipated in the nominal control law by recognizing patterns in the behavior of the error. The on-line learning capability of ANNs also provides additional potential for adapting to changes in aircraft dynamics due to damage or failure.

Previous research efforts have shown that the utilization of variables generated within the control laws can be used as parameters for the self/non-self definition for AIS-FDIE purposes [7, 37]. Also, it has been shown that adaptive control laws with a "model predictor" architecture based on a NLDI+NN also possess a great potential for FDIE purposes. Generally, the ANN used in this approach operates as a "closed-loop" architecture, thus the aircraft must be equipped with such a control system.

Furthermore, a new FDIE scheme in which the ANNs are included as part of the data processing algorithm has been proposed in order to avoid this limitation. In this configuration the ANNs operate in an "open-loop" configuration and are no longer part of the control laws. This configuration guarantees the portability of the FDIE scheme to various platforms regardless of their control law system.

Chapter 2 Immunity-Based Failure Detection

The biological immune system relies on the generation of specialized cells, called antibodies (e.g. T-cells), that do not match the chemical marker of cells belonging to the host organism [10]. This characteristic allows the antibodies to detect, bind and mark for destruction almost any antigen without damaging "self" cells. Aircraft sub-system failures are considered abnormal conditions analogous to the antigens present in biological systems. The general idea of the Artificial Immune System for Failure Detection, Identification and Evaluation (AIS-FDIE) is that an abnormal condition can be declared when a certain configuration of "features" does not match any pre-determined configuration of "features" at normal conditions. These "features" include several sensor outputs, state estimates, statistical parameters or any other variables that contain information about the dynamic fingerprint of the aircraft and are able to capture the fingerprint of abnormal conditions. These "features" are analogous to the chemical markers that represent the encoding of the self in biological systems.

In general, the detection problem must be generated in great detail and cautiously. The detection logic must be designed for real-time operation with a high rate of failure detection and a low number of false alarms. In order to properly identify the features that capture the signature of failure conditions and to obtain a successful AIS detection scheme, the algorithm must include information about the system's operational envelope, the targeted sub-systems and the nature and type of abnormal conditions expected.

For example, let us assume that an aircraft possess four major sub-systems, with their components, corresponding to: actuators, sensors, propulsion systems and structural elements. Each set of actuators may include two ailerons, one rudder, two throttles and one elevator. The

sensors considered may be the angular rate gyros. The propulsion system may consist of two engines and, for the structural elements, only the wings might be considered. In this example, the total number of sub-systems N_s results to be $N_s = 8 + 3 + 2 + 2 = 15$.

For each sub-system k, a list of Nft_k abnormal conditions types must be formulated. For example, actuator abnormal conditions may be locked control surfaces or an actuator moving freely. For the sensor list, the failures could be constant output and sensor bias. If we assume that two types of failures are considered for each sub-system, the total number of failure results would be $N_F = \sum_{k=1}^{N_s} Nft_k = 26$. The definition and analysis of the failure is important for the process of selecting and defining the features. The feature variables should completely define the targeted systems and achieve the self/non-self discrimination. Table 1 presents an example of features, or states, which can be used for the self/non-self definition.

Table 1- List of Feature for AIS-FDI Design

H	=	Altitude	$\dot{r}$	=	Yaw Acceleration
V	=	Ground Speed	d_e	=	Longitudinal Stick Displacement
M	=	Mach Number	d_a	=	Lateral Stick Displacement
a_x	=	Longitudinal Acceleration	d_r	=	Pedal Displacement
a_y	=	Lateral Acceleration	d_T	=	Throttle Command
a_z	=	Vertical Acceleration	p_{ref}	=	Roll Rate Command
α	=	Angle of Attack	q_{ref}	=	Pitch Rate Command
β	=	Sideslip Angle	r_{ref}	=	Yaw Rate Command
φ	=	Roll Angle	NN_p	=	Roll Rate Estimate
θ	=	Pitch Angle	NN_q	=	Pitch Rate Estimate
ψ	=	Yaw Angle	NN_r	=	Yaw Rate Estimate
p	=	Roll Rate	$MQEE$	=	Main Quadratic Estimation Error
q	=	Pitch Rate	$OQEE$	=	Output Quadratic Estimation Error
r	=	Yaw Rate	$DQEE_p$	=	Decentralized Quadratic Roll Rate Estimation Error
$\dot{p}$	=	Roll Acceleration	$DQEE_q$	=	Decentralized Quadratic Pitch Rate Estimation Error
$\dot{q}$	=	Pitch Acceleration	$DQEE_r$	=	Decentralized Quadratic Yaw Rate Estimation Error

The list includes 32 different features that could be processed to obtain a 32-dimensional hyper-space that defines the self/non-self projection. The high dimensionality representation presents a significant computational cost that should and can be avoided by means of the "Hierarchical Multi-Self (HMS) Strategy", which reduces the order of the projections into sub-selves. The HMS strategy relies on the assumption that some features may be relevant in capturing some types of failures and may lack relevance with respect to others. Therefore, lower dimensional projections of the self/non-self may be enough to detect and identify specific failures. It should be noted that lower dimensional projection may reduce the detection capabilities under certain circumstances. In this research effort, sets of 2-dimensional and 3-dimensional projections are investigated. The total number of sub-selves N_{SS} that can be built for a complete set of $N = 32$ features and $N_{max} = 2$ feature combinations is:

$$N_{SS-2D} = C_N^{Nmax} = \frac{N!}{N_{max}!\,(N - N_{max})!} = \frac{32!}{2!\,30!} = 496 \tag{1}$$

Similarly for 3-dimensional projection ($N_{max} = 3$):

$$N_{SS-3D} = C_N^{Nmax} = \frac{N!}{N_{max}!\,(N - N_{max})!} = \frac{32!}{3!\,27!} = 4960 \tag{2}$$

It should be noted that, even though 4960 projections are possible, it was decided that it would be lengthy and somewhat unnecessary to generate such a large number of projections. Therefore, only 45 3-dimensional projections were created. The 3-dimensional projections were not included in this effort.

2.1 Generation of Detectors

The process of generating detectors, or antibodies, is an exhaustive and lengthy procedure that requires adequate computational data processing capabilities and adequate numerical representations of the self/non-self. In this research effort, 2-dimensional and 3-dimensional selves were generated for the self/non-self representation using the "Cluster Set Union Method" (CSUM). In the CSUM, extensive experimental data are necessary to properly determine the nominal "hyper-space" representation. For each combination of features corresponding to a particular projection segments of data are processed separately to produce sets of detectors that cover the non-self hyperspace.
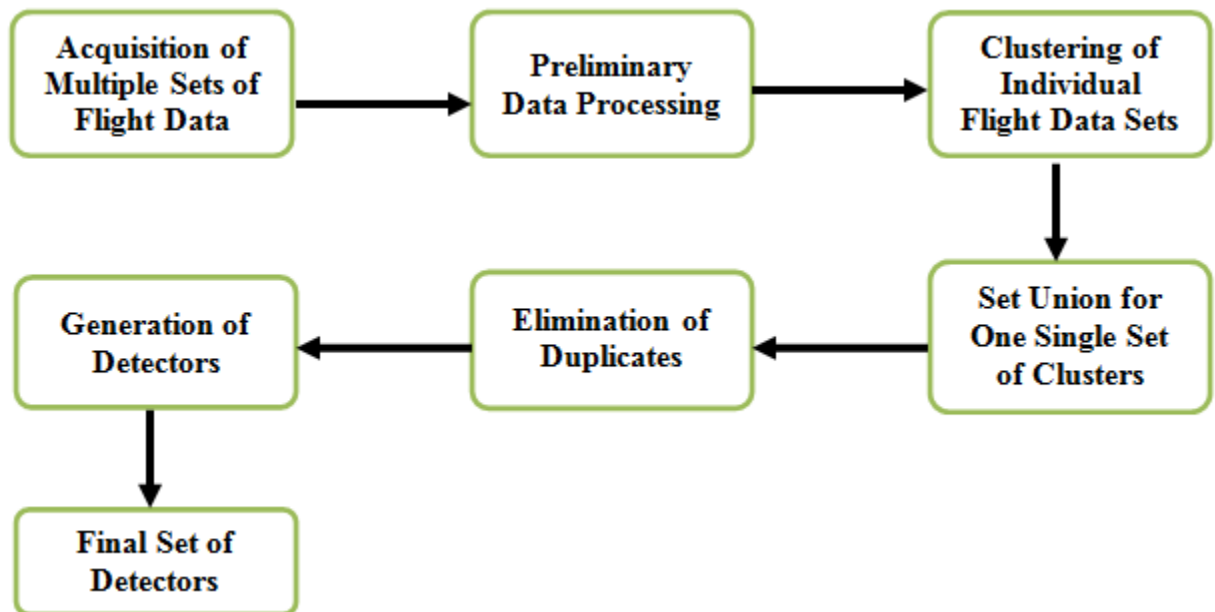
Figure 3- Cluster Set Union Based Logic

2.1.1 Cluster Set Union Method

The CSUM is a 5 phase process that uses an optimized algorithm to fuse different sets of clusters generated from single sets of flight or simulation data. This methodology is particularly convenient when the computer characteristics used for pre-processing are not powerful enough for an efficient application of the algorithms. The methodology is described next:

1) **Preliminary Data Processing:** The prerecorded data are split into subsets for memory allocation convenience. Then, the individual sets of data are normalized in order to obtain values between 0 and 1, leading to a hypercube space for the selected features. The normalization factor for each dimension is determined by 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. It is important to remark that the same normalization factors must be used in all data sub sets.

2) **Clustering of Individual Data Sets:** The normalized data generated in the previous step define self points that need to be represented by a definite number of geometric hyper-bodies called clusters. The clustering mechanism is based on a modified version of a "k-means" vector quantization algorithm.

3) **Cluster Set Union:** Once several sets of clusters have been generated, a fusion process is performed. The fusion process consists of the unions of cluster sets as well as overlapping elimination.

4) **Elimination of Duplicate Clusters:** The clusters generated in the previous step may overlap each other causing an unnecessary use of computing power that can be

eliminated. 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 radii of the overlapping clusters minus the permitted overlapping threshold. This approach favors clusters with larger radii and clusters with a more efficient coverage. Also, this approach allows the update of the database when new flight test data are available by clustering only the newly acquired data sets and adding the results to the older sets while eliminating any duplicates.
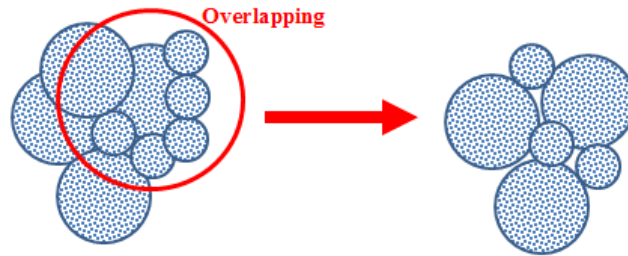
Figure 4- Elimination of Duplicates

5) **Generation of Detectors:** Self clusters are used to generate detectors by covering the non-self hyper-space with hyper-bodies similar to the clusters. An enhanced negative selection algorithm for real-valued representation with variable non-self radius (ENSA-RV) is used in this research effort. This algorithm ensures that there is no overlapping with the self and that the non-self is covered to the desired predefined amount. It should be noted that the algorithm requires certain parameters that must be selected carefully. The detector generation process can be stopped when a desired coverage of the non-self is achieved or after a prescribed number of iterations when a preset maximum number of detectors is reached. In general, the following optimization criteria may be considered when generating detectors:

- No overlapping among self and non-self

- Minimum empty space in the self clusters

- Minimum uncovered non-self space

- Minimum overlapping among self clusters

- Minimum overlapping among non-self detectors

- Minimum number of detectors

Figure 5 shows an example of a 2-dimensional projection of the dynamics of an aircraft throughout the investigated flight envelope. The variables shown are a reference roll rate command and a neural network roll rate estimate for the x and y axes respectively. It should be noted that the blue clusters represent the self and the red clusters represent the non-self.
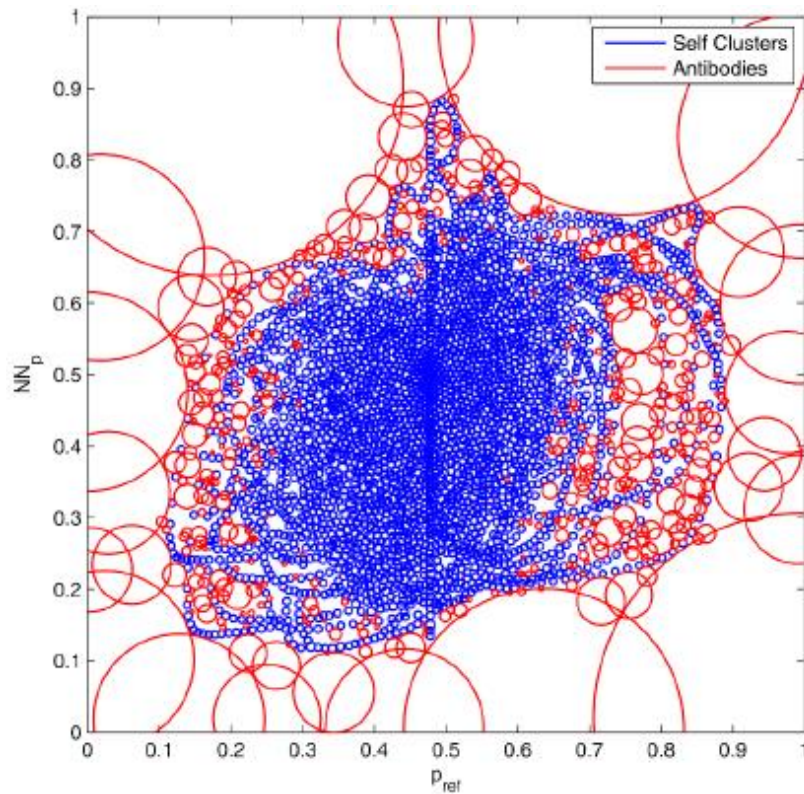
Figure 5- Self/Non-Self Projection

2.1.2 Detection Performance Analysis

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

- True Positive (TP): A failure is detected and declared as failure

- True Negative (TN): Nominal conditions are declared as nominal

- False Positive (FP): Nominal conditions are declared as failures

- False Negative (FN): Failure condition is not detected

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

$$DR = \frac{TP}{TP + FN} \times 100 \tag{3}$$

Similarly, the FA rate can be calculates as follows:

$$FA = \frac{FP}{TN + FP} \times 100 \tag{4}$$

It can be seen that both rates are ratios of the corresponding condition divided by the total amount of data points considered to be of that particular condition respectively.

2.2 Simulation Environment

The flight test data utilized for the generation of detectors were obtained using a 6DOF motion based flight simulator (shown in Figure 6) at West Virginia University (WVU). The flight simulator has been interfaced with an external computer on which the research aircraft dynamic model can be run within the MATLAB/Simulink environment. The Simulink model used for this research effort is customized in such a way that the flight conditions could correspond to either nominal or upset conditions. It should be noted that the flight test data were also used in the identification and direct evaluation phase of this research effort. Therefore, all mentions of flight test data should reference back to this section.
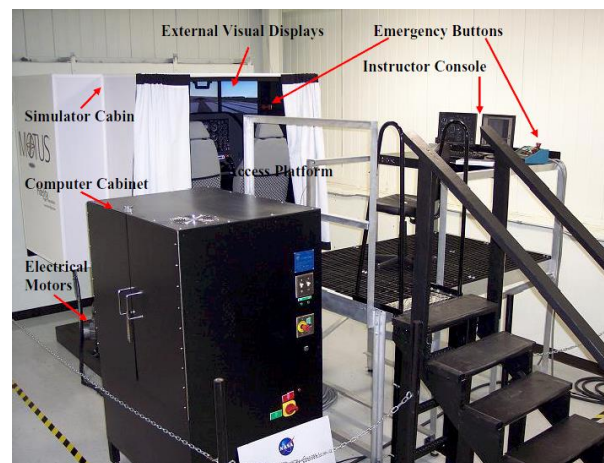
Figure 6- WVU 6-DOF Motion-Based Flight Simulator

The aircraft model used includes a model following adaptive control based on a non-linear dynamic inversion and an artificial neural network (ANN) augmentation architecture, which also produces estimates of the aircraft angular rates and angular acceleration errors. Figure 6 shows the top level Simulink model of the aircraft interfaced with the flight simulator.

Figure 7- Top level Simulink Model for the WVU Flight Simulator Interface

A broad range of flight test scenarios was performed over a wide range area of the flight envelope. The flight envelope is split into nine specific points for Mach number ranging between 0.6 to 0.9 and altitudes between 9000 ft and 31000 ft. The flight scenarios are simulated under nominal flight conditions lasting between 10 and 20 minutes each. These nominal flights include steady state flight conditions and transitions from steady state to mild and moderate flight maneuver conditions. These flights are then repeated under one failure at a time to capture the dynamic fingerprint of each type of failure. Figure 8 presents the nine specific points of the flight envelope used for all the flight scenarios.

Figure 8- Investigated Flight Envelope

2.3 Detection Results

The detection rates and false alarms were calculated for 26 different failures varying in subsystem type and severity using Equations (3) and (2). Table 2 shown below presents a sample set of projections among the 496 generated sub-selves that were selected as an example of the relevant combination of features utilized in this effort.

Table 2- Features of Selected Projections

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

Table 3 presents the performance analysis of those selves under four different failures:

left aileron locked at 2.5 deg., left stabilator locked at 2 deg., 6% left wing loss and 5 deg bias in

the pitch rate sensor output. A sample set of 20 projections is presented. The detection rate as well as the false alarm rate is shown.

Table 3- Detection Performance

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

The results from Table 3 show that certain selves favor the detection of certain types of failures while showing poor detection rates for others. This fact has been used in previous research efforts. The presented mechanism uses low order projections to build sub-selves using a specific hierarchy of features relevance with respect to each type of failure. Figures 9-12 show the individual detection rate obtained for different projections and four different failures.
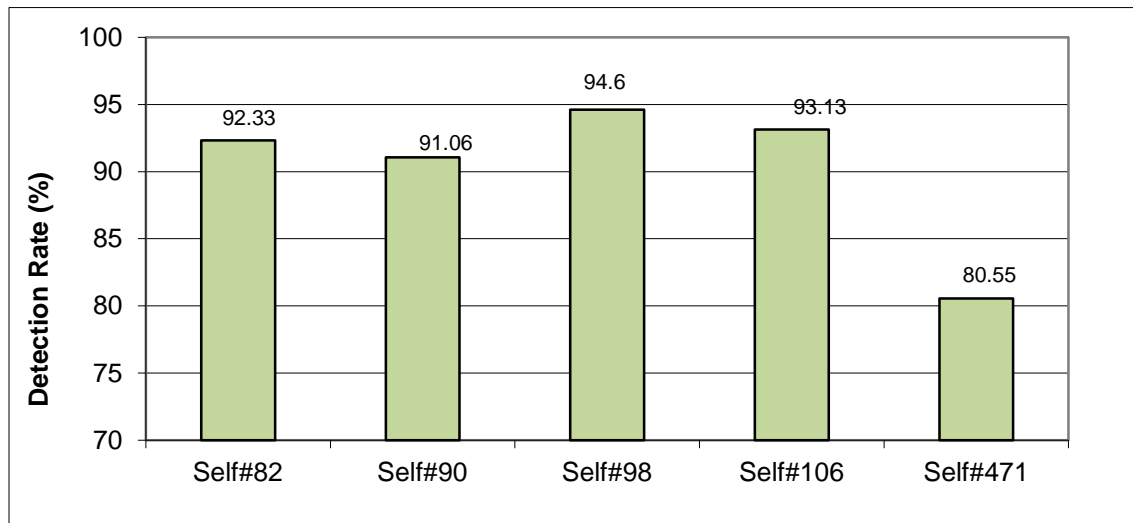
Figure 9- Detection Rate of Left Aileron Locked at 2.5 deg Failure
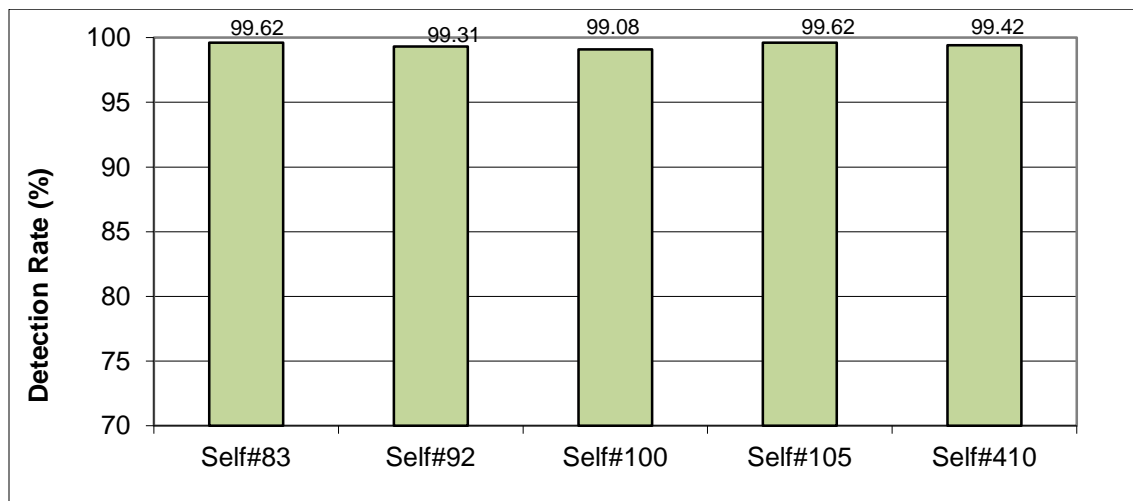
Figure 10- Detection Rate of Left Stabilator Locked at 2 deg Failure
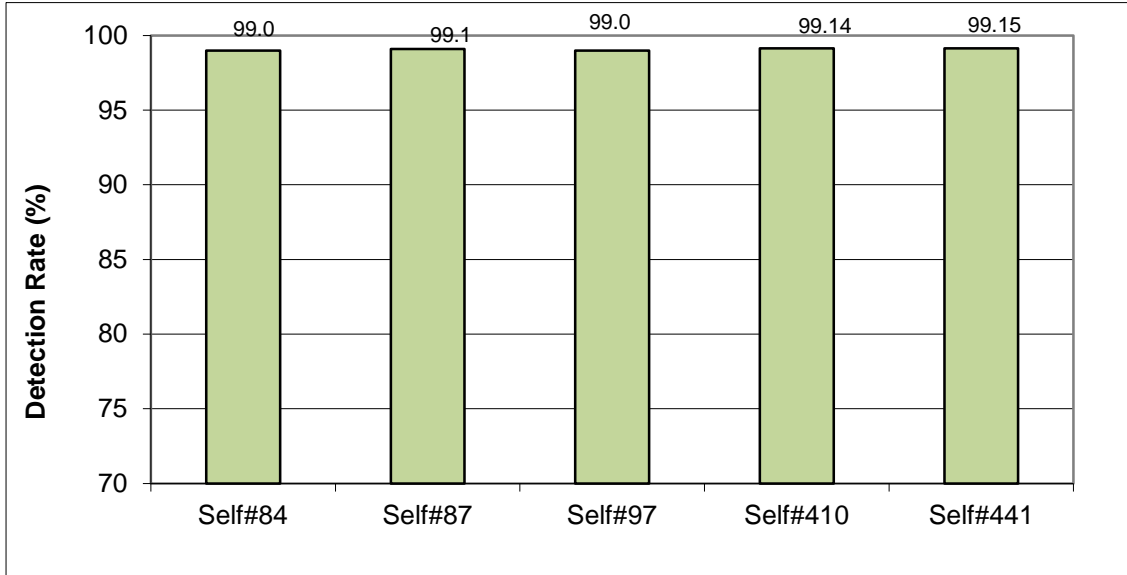
Figure 11- Detection Rate of 6% Left Wing Loss Failure
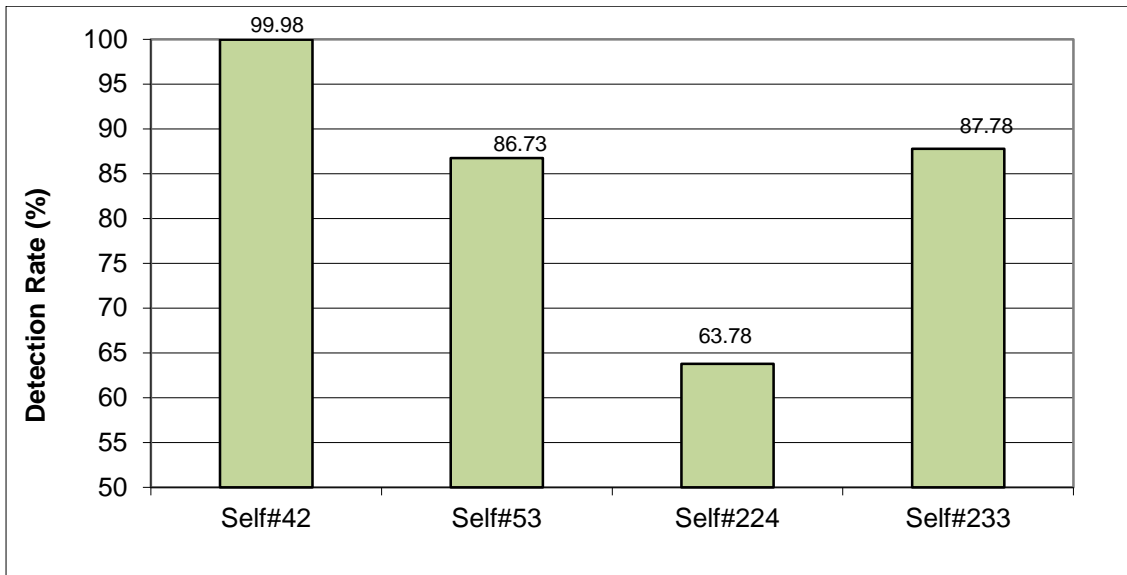
Figure 12- Detection Rate of Pitch Sensor Bias of 5 deg/sec Failure

Figure 12 presents the detection rate that a single projection can attain when tested against four different failures. It should be noted how a single projection is able to detect three different

failures successfully while showing poor performance for a sensor failure. This is due to the fact that the dynamics of such failure do not have great impact on the features corresponding to Self 87.
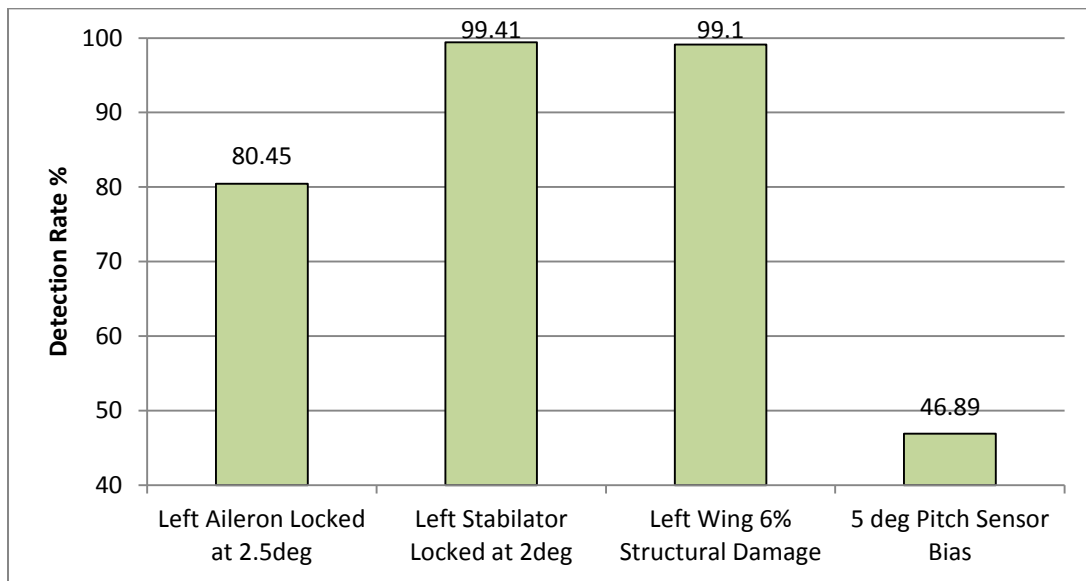
Figure 13- Detection Rate of Self 87 for 4 Types of Failure

Chapter 3 Immunity-Based Failure Identification

The identification phase is the second step in the aircraft health assessment algorithm. The identification phase consists of the correct classification of the sub-system, type and severity of the failure the aircraft is suffering at any particular time during flight. The identification algorithm is only activated when the detection algorithm declares that an abnormal condition is present. The identification phase assumes that the output of the detection phase is correct.

In this research effort a novel structured non-self approach (SNSA) has been developed within a HMS strategy [38]. This approach is based on a structuring process, or arrangement, of non-self projections and intends to reduce the computational effort required and facilitate the real-time application of the AIS approach without compromising the FDIE performance. The SNSA consists of a dual- phase algorithm where 2-dimensional self/non-self projections, previously generated using a negative selection mechanism and tested in simulation under several abnormal conditions, are selected according to the ability to detect failures at a predefined detection rate (DR) percentage. Then, by using a positive selection-type mechanism, the resulting projections are processed in order to generate identifiers capable of differentiating similar dynamic prints among several abnormal conditions and declaring correct failure types, and magnitudes.

For example, within a first phase of the SNSA, a total of 496 self/non-self projections were generated based on the availability of 32 features to capture the dynamic print of abnormal conditions. After extensive experimentation it was possible to reduce the number of self/non-self projections to 183 possible candidates with a DR equal or larger than 70%. These projections are

considered to possess the ability to capture the dynamic print of several sub-system failures and, more importantly, facilitate the process of characterizing the projections that perform better during the identification of specific failures. Table 4 presents a sample set of 2-dimensional projections investigated within the identification phase.

Table 4- Self/Non-Self Projections

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

It is important to note that the dynamic fingerprint of several failures may produce a very similar effect on the features of self/non-self projections. This characteristic presents a more complex problem in which incorrect identification may be produced if the identification problem is not defined appropriately. For example, let us assume that an identification algorithm, only consisting of *Self#3* (p_{ref}, NN_p), is tested for two sub-system failures (i.e. right wing structural failure and left aileron stuck failure). This particular pair of failures will produce an undesired roll rate that can be successfully perceived and detected by *Self#3*. The dynamic fingerprint produced by both abnormal conditions in the selected projection may look very similar, increasing the complexity of the identification problem. Now, let us assume that the same identification algorithm is augmented with *Self#30* (q_{ref}, NN_p), which can also capture the

abnormal condition dynamic print of the mentioned failures. Due to the fact that *Self#30* also captures dynamic changes in pitch rate, it is possible to identify and distinguish between the two mentioned failures. Within a second phase of the SNSA, positive selection applied to the 183 self/non-self projections is performed in order to address the mentioned identification problem. Figures 14a, 14b and 15a, 15b shown below present the similarity of the dynamic print of two different failures in a 2-dimensional projection.

The combined identification capabilities of the projections utilized within the two phases of the SNSA (see Figure 16) provides a more robust system capable of not only correctly identifying the detected failure but also providing information regarding the magnitude of the investigated failures. With the correct combination of projections and their corresponding identifiers, it is possible to discard incorrect failures and ultimately determine which abnormal condition is affecting the system.
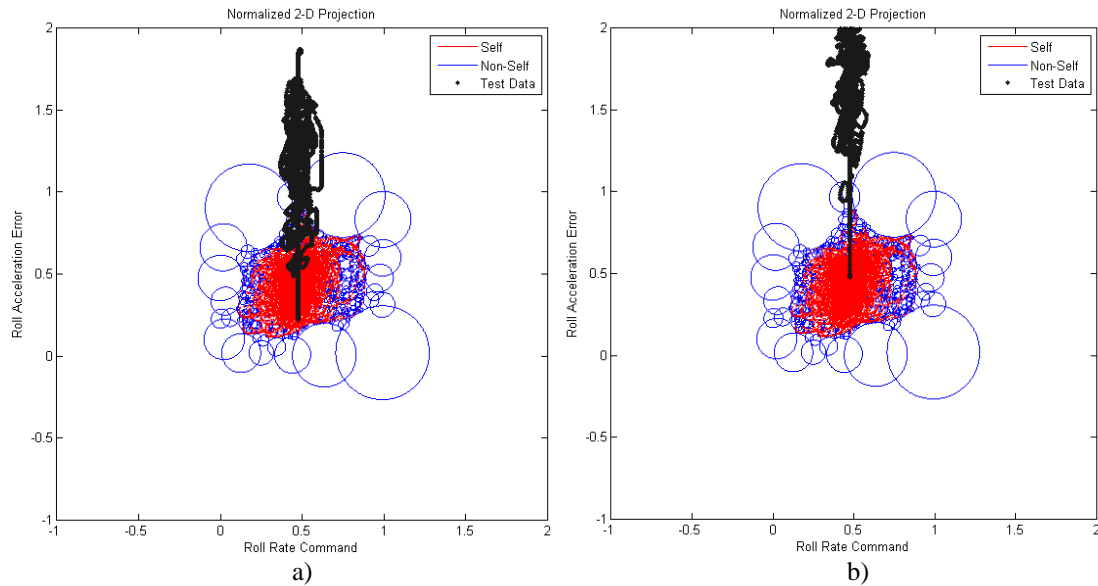
Figure 14- a) Self#3 with Left Aileron Failure, b) Self#3 with Right Wing Structural Damage
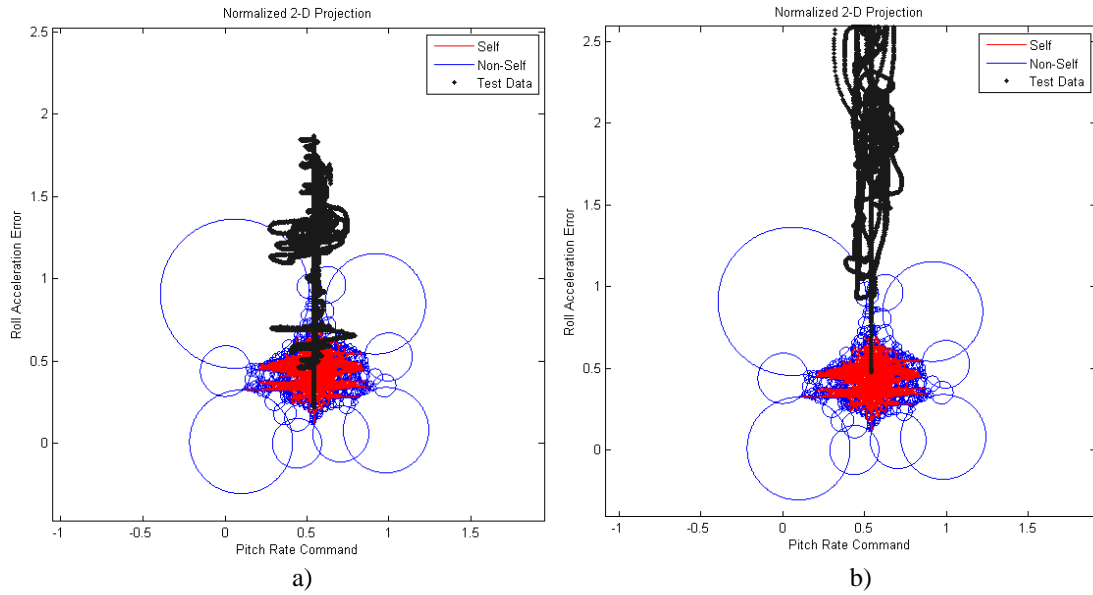
Figure 15- a) Self#30 with Left Aileron Failure, b) Self#30 with Right Wing Structural Damage
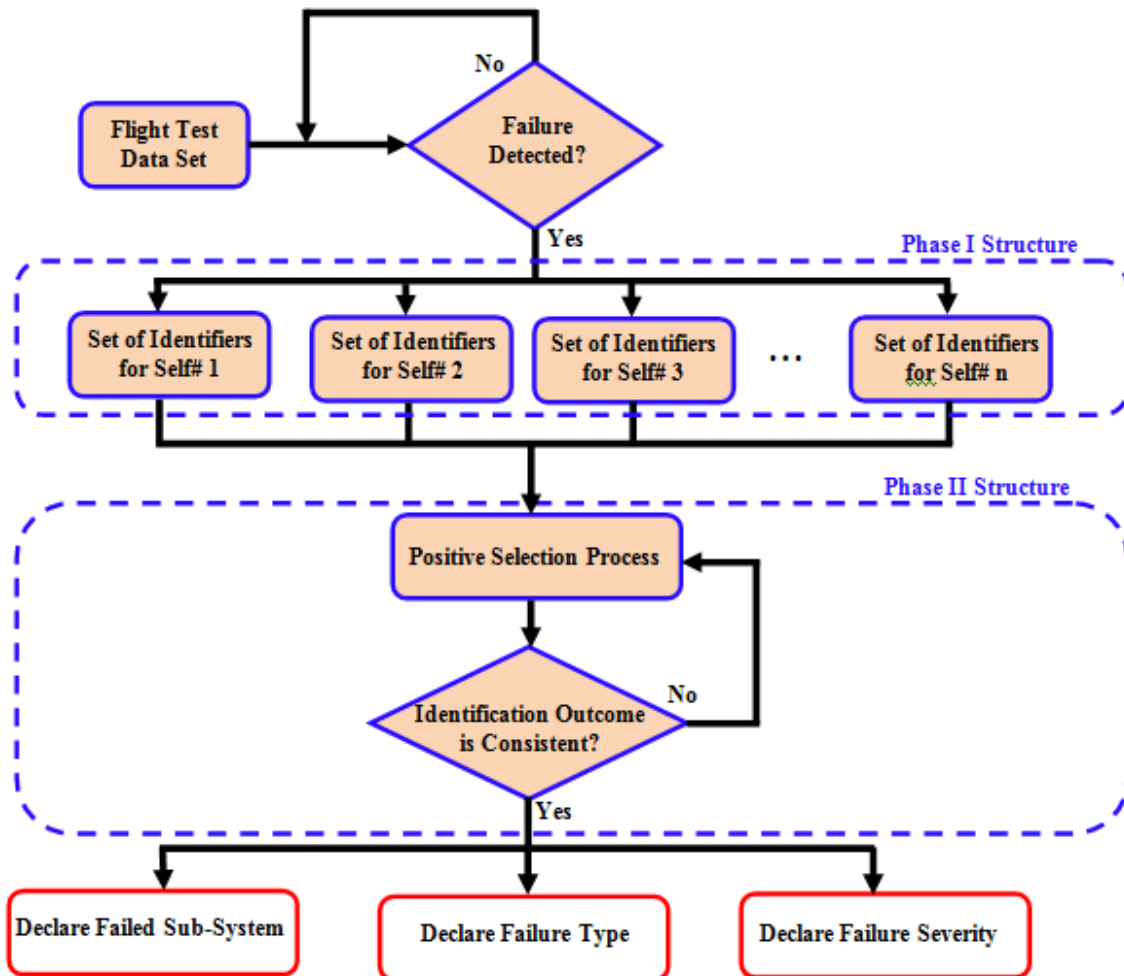
Figure 16- Structured Non-Self Approach Logic

45

3.1 Generation of Identifiers

The process to generate identifiers is very similar to the detector generation algorithm. The generation of identifiers is based on a positive selection algorithm in which failure flight test data are reproduced in order to record the dynamic fingerprint of a failure throughout the entire flight envelope. The generation of identifiers consists of a multi-step process that optimizes the generation and where the radii of the identifiers depend mostly on their distance to the self rather than a greater coverage of the non-self. Figure 17 presents the generation logic for the generation of identifiers.
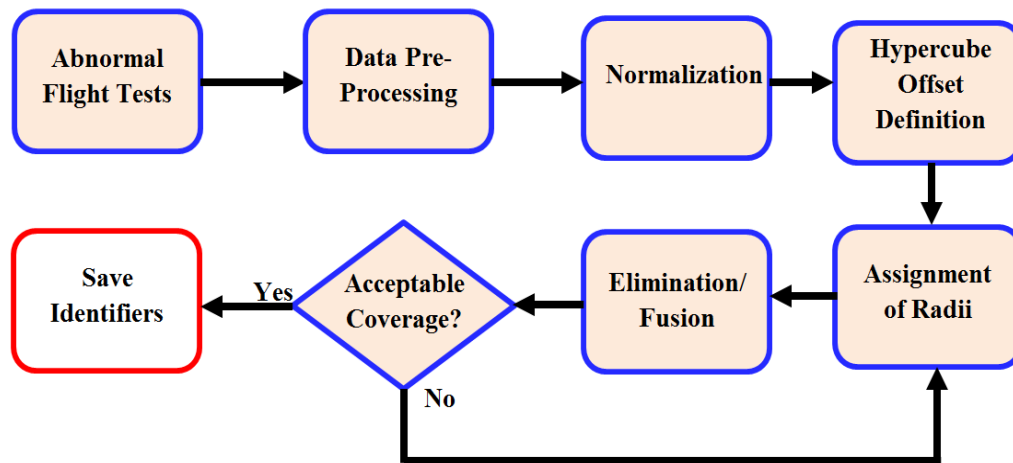
Figure 17- Identifier Generation Logic

Abnormal Flight Tests: Several flight tests at different abnormal conditions throughout the entire flight envelope were performed. Previously selected features corresponding to the self/non-self definition as shown in Table 1 are recorded for future processing and identifier definition. Section IV provides more details on the flight testing environment and conditions.

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

Offset Hypercubes: The unit hypercube determined during the normalization process delimits the hyperspace of the nominal condition flight tests. High magnitude failures may contain data points that lay far away from the unit hypercube of the self/non-self projection. Therefore, outward concentric hypercubes are defined in order to determine the distance of the abnormal condition point from the self, which subsequently allows the algorithm to determine the magnitude of the corresponding failure.
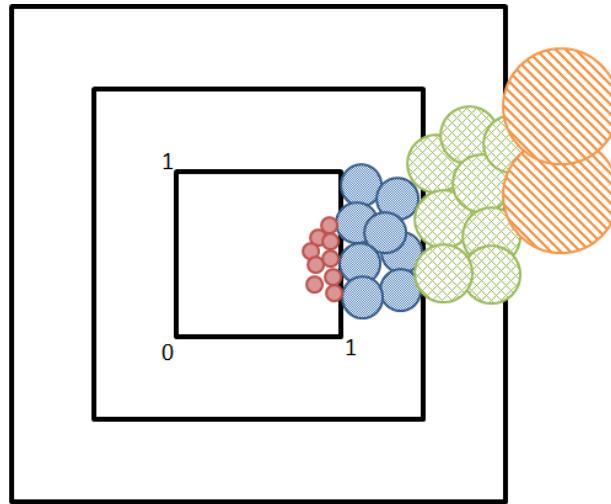
Figure 18- Radii Variation With Respect to Distance from the Self

Radii Assignment: The radius of any identifier is predetermined and it is assigned depending on the location of its center with respect to the offset hypercubes. The radius of an identifier increases as the position of its center lies within an outward hybercube. In other words, the radii of all identifiers increase as their distance to the self increases. Figure 17 shows an example of the offset hypercube and the radii assignment.

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

3.2 Structured Non-Self Phase I: Non-Self 2D-Projections Selection

The first phase of the SNSA is the result of the failure detection testing within the HMS strategy. As mentioned previously, 496 2-dimensional self/non-self projections were generated for failure detection algorithm experimentation. These projections were then tested against over 20 different failures including several sub-systems under different failure magnitudes. Extensive experimentation was required in order to determine which projections could substantially detect a failure with good detection rates and minimum false alarms within a negative selection approach. It was determined that a total of 183 projections were able to fulfill the objectives of a DR equal or higher than 70%. This process is referred to as the Phase I Non-Self Structure. The

selected projections were chosen as potential candidates for identification included sensor outputs, state estimates and statistical parameters among other features. The set of abnormal conditions involved sensor failures, structural damage on the wings, engine failures and control surface failures. Table 5 shown below presents a list of the failures investigated in this research effort.

Table 5- Investigated Sub-System Failures

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

Several failures presented similar dynamic prints on several 2D-projections, which subsequently led to the repetition of several projections with the ability to detect multiple failures. On the other hand, certain failures that are difficult to detect, such as rudder failure, only resulted in the activation of a small number of projections. The negative selection logic behind the Phase I Structure resulted in the reduction of the number of the original projections into a smaller set, reducing the complexity and the hardware requirements for its implementation. Table 6 presents a sample of the projections that are considered to be adequate for abnormal condition identification based on a detection performance equal or higher than 70%.

Table 6 presents the detection capability of a sample set of projections for five types of failures. This analysis was performed on the 496 original projections under 26 failures varying in sub-system categories, failures types and magnitudes. Various projections present the ability to capture the dynamic fingerprint of several abnormal conditions while others can only capture the

dynamics of a small set or just a single abnormal condition. For example, *Self#3* demonstrated its ability to capture the dynamic fingerprint of a left aileron locked, a right wing structural damage and a left stabilator locked failure. On the other hand, *Self#4* only demonstrated the ability to capture the dynamic fingerprint of a left stabilator locked type of failure.

Table 6- Detection Performance of a Sample Set of Projections

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

Within this analysis it was possible to isolate the projections that can be used for identification purposes. From the Phase I Non-self structure analysis, it was possible to determine which specific projections correspond to every specific failure investigated. Furthermore, it is also possible to determine how many projections capture the dynamic fingerprint of an abnormal condition. Table 7 below presents the number of projections from the

50

183 total selected that have the potential ability to be used as a projection for identification purposes.

Table 7- Total Number of Projections Activated per Failure

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

The outcome of the Phase I Non-Self structure reduces the total amount of projections needed to perform the FDIE algorithm. Alternately, its outcome allows the adequate design of the mapped-based positive selection algorithm utilized in the second phase of the SNSA by reducing the number of possible projections for the generation of identifiers as well as the reduction of identifiers required in the identification algorithm for each individual projection.

3.3 Structured Non-Self Phase II: Positive Selection Algorithm

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

The identifiers generated during Phase I and II are then loaded into an identification function and organized in a single array such that the index of each identifier corresponds to a

51

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

The approach investigated in this paper covers not only a general identification logic, but also a qualitative and quantitative evaluation logic integrated into a single less complex algorithm. This novel approach intends to reduce the computational processing for real time application of the solution to the FDIE problem. The proposed mapping-based positive selection logic targets a multi-dimensional problem by means of a simpler but effective logic that can result in a more efficient real time algorithm.

3.3.1 Direct Evaluation

In previous efforts, the FDIE algorithm consisted in a step-by-step process in which a failure was first detected. The following step was the identification of the subsystem under failure followed by the qualitative evaluation that determines the type of failure. Then the severity of the failure was estimated through a process called quantitative evaluation. The qualitative and quantitative evaluation steps are known as "Direct Evaluation". Finally, the estimation of the flight envelope reduction, or Indirect Evaluation, is performed.

The evaluation process suggested in previous efforts was a three-step process in the FDIE algorithm. In this work, the determination of the type and severity of a failure, also known as

direct evaluation, is incorporated in the identification stage as a single process, leaving the indirect evaluation as an individual stage in the SNSA algorithm proposed.

The identification process mentioned in section 3.3 is capable of declaring the subsystem under failure, the type of failure and the magnitude. Figure 19 demonstrates the novel architecture used in this effort.
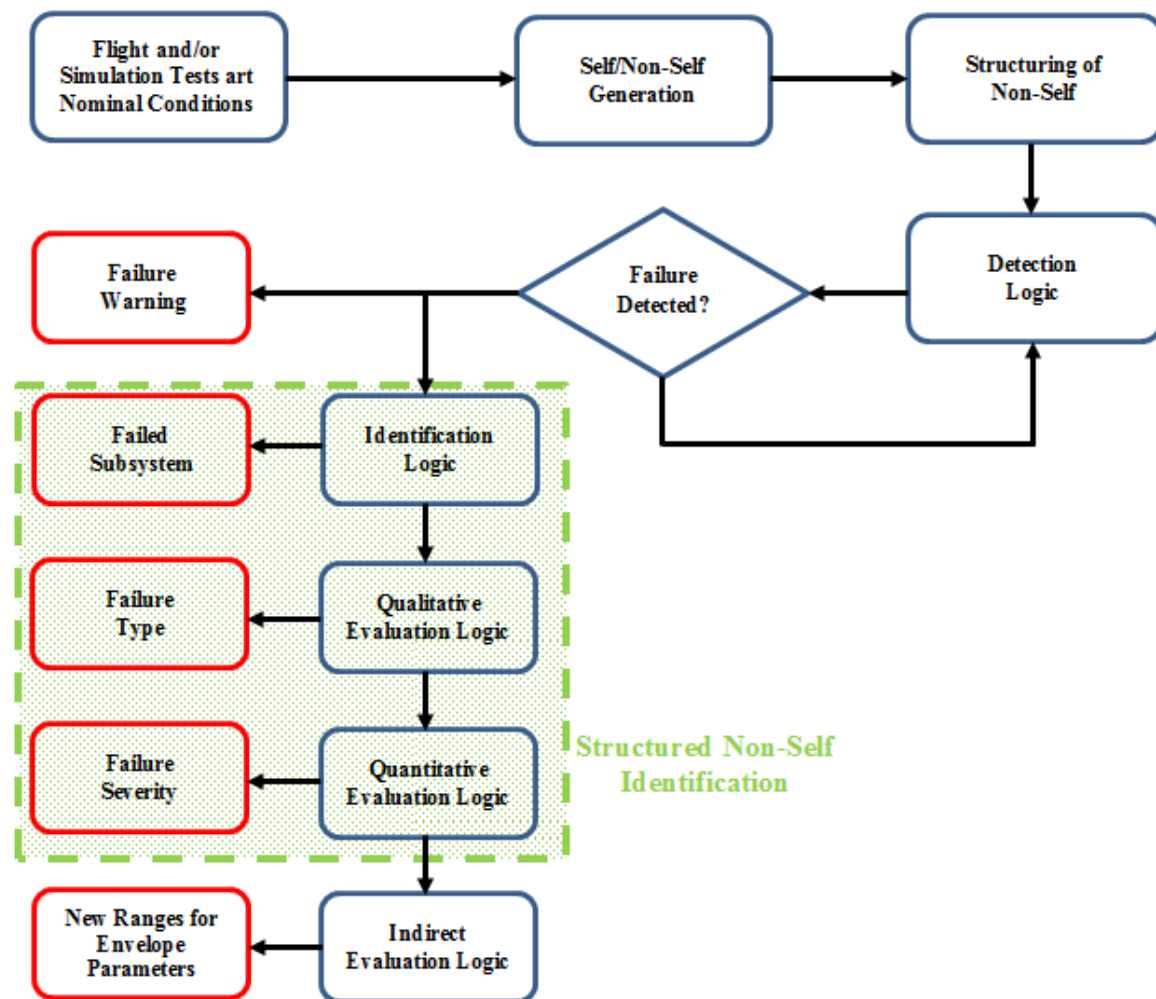
Figure 19- Integration of Direct Evaluation and SNS Identification

3.4 Identification Results

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

Each set of identifiers generated per failure contains on average 36 identifiers. Considering that every set of identifiers for all failures is integrated into each projection, an approximate total of 324 identifiers per projection are used for the identification positive selection process.

After an initial analysis, the algorithm was optimized and it was determined that out of the 183 projections, a total of 93 projections were enough to correctly identify the investigated failures. The reduction of the total number of projections required for identification has reduced the computational complexity of the algorithm considerably. Table 8 presents a sample set of projections used for failure identification.

Table 8- Projections Used for Identification

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

Further analysis was carried out to reduce the number of projections required to produce desirable identification outputs. In some cases, the use of a single projection was enough to obtain favorable identification rates. On the other hand, other failures require more projections in order to obtain desirable identification results and also to redeuce misidentification rates. Table 9 shown below present the number of projections required for a correct failure identification output.

Table 9 Total Number of Projections Used for Identification

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

The identification algorithm was tested under 16 different failures (refer to Table 9). Table 10 shown below presents the results for identification rate analysis. The table below includes the false identification percent for other types of failures. It should be noted that in some cases the dynamic fingerprint of a failure fell outside an identifier. For these particular cases, the identification algorithm output a 0% identification rate. For simplicity, a "no identified failure" column was not included in Table 10. It should be noted that Table 10 presents the identification results in a horizontal fashion. For example, failure #1 is output correctly 99.7% of the time but presents confusion with failures 3, 4 and 14 for 0.1% of the time respectively.

Table 10- Identification Results for 16 Different Failures.

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

Chapter 4 Flight Envelope Reduction

The indirect evaluation of a failure, or flight envelope estimation, is a re-assessment of the maneuverability, performance, handling qualities and flight envelope (MPHF) limits of an aircraft under upset conditions. Evidently, after an aircraft has suffered a failure of any kind the MPHF limits are affected and probably impaired significantly. It is of great importance that an estimation of the post-failure capabilities of the aircraft is calculated in order to avoid actions that may jeopardize the safety of the system and/or the mission. In other words, the achievable operational limits are calculated based on all or some dynamic parameters necessary for the completion of a task or mission.

The indirect evaluation process assumes that the failure detection, identification and direct evaluation stages are successfully completed. The post-failure flight envelope reduction estimation is performed offline in order to train the system and to generate a database that can be loaded into a flight computer. The online application of this methodology will only require interpolation between values of the database of failures of the same nature that are not contemplated in the investigation. This reduces the computational requirements for this process, resulting into a more efficient and simpler algorithm.

The re-assessment of the MPHFs is an analytical process that requires that the effects of a failure are related to at least one feature of the variables set. The features used to generate the 2-dimensional projections capture the limits of the states related to the dynamics of the aircraft at nominal conditions. Therefore, a set of variables that relate to the dynamics of the abnormal condition, the aircraft controls and its flight envelope variables must be formulated. These

variables are divided into three categories, namely, directly involved variables (DIV), equivalent directly involved variables (EDIV) and envelope relevant variables (ERV).

The DIVs are variables that are directly and significantly affected by an abnormal condition. The DIV's used in this investigation are control surface deflections, measured sensor outputs, lift coefficients, side force coefficients and moment coefficients. These variables are used to define the upset conditions and they may or may not be part of the features set. Table 11 presents a list of DIVs used in this analysis.

<div align="center">

Table 11- Directly Involved Variables

Stabilator Deflection	δ_e
Aileron Deflection	δ_a
Rudder Deflection	δ_r
Throttle Valve Position	δ_T
Measured Roll Rate	p_{meas}
Measured Pitch Rate	q_{meas}
Measured Yaw Rate	r_{meas}
Lift Coefficient	C_L
Pitch Moment Coefficient	C_M
Engine Thrust	T

</div>

If the variables used to define the upset conditions are not part of the set of variables a relationship between the DIVs and some other variables in the feature set (EDIVs) must be established. The EDIVs are variables that are in fact in the features set and possess a direct relation to the abnormal conditions investigated. Table 12 presents a list of EDIVs used for this analysis.

<div align="center">

Table 12- Equivalent Directly Involved Variables

Longitudinal Stick	d_e
Lateral Stick	d_a
Pedals	d_r
Throttle Command	d_T

</div>

It is clear that the relation between DIVs and EDIVs corresponds to the pilot controller commands as a function of control surface actuation, throttle valve position and measured angular rates depending on the failure investigated. This relation is demonstrated below such that:

$$EDIV = f(DIV)$$

In which f can be a model reference first order transfer function, a model reference second order transfer function or even a linear function.

Finally, the ERVs are variables in the feature set that contain information about the dynamics of the aircraft along the entire flight envelope. These variables can be linear accelerations, angular velocities, angular accelerations, Euler angles, altitude, velocity etc. Table 13 presents a list of relevant envelope variables used for the flight envelope reduction analysis.

Table 13- Envelope Relevant Variables

Altitude	H	Vertical Acceleration	a_z
Mach Number	M	Pitch Acceleration	$\dot{q}$
Velocity	V	Roll Acceleration	$\dot{p}$
Pitch Rate	q	Yaw Acceleration	$\dot{r}$
Roll Rate	p	Angle of Attack	α
Yaw Rate	r	Sideslip Angle	β
Longitudinal Acceleration	a_x	Pith Angle	θ
Lateral Acceleration	a_y	Bank Angle	φ

The indirect evaluation process requires specific customization depending on the subsystem, type of failure and affected parameters of the flight envelope. The strategy relies on analytical methods that can be validated to graphical means using the 2-dimensional projections. A general framework can be formulated and applied for all the investigated failures.

4.1 Quantitative Indirect Evaluation Methodology

The analysis of the flight envelope reduction estimation used in this research effort consists of the analytical calculation of the effective control commands after a failure is present in the system, and then the estimated ranges for ERV's are determined graphically through the 2-dimensional projections that are related to the failure's dynamics. In other words, the effective control stick is calculated based on predefined equations. Then, the 2-dimensional projections involving the control stick affected and any other envelope relevant variable is plotted. Next, the new values for stick commands are located in the nominal projections. Finally, based on the current maximums and minimums of the control stick values, the maximums and minimums of the ERVs are found. Figure 20 below illustrates the graphical analysis of the flight envelope reduction.
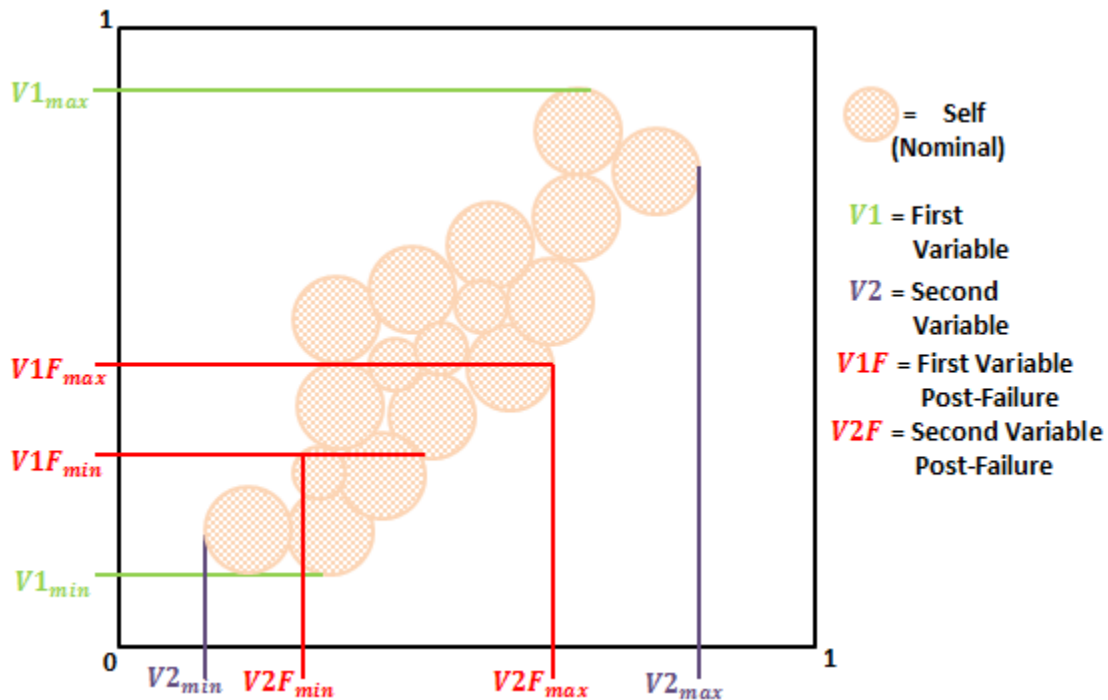
Figure 20- Indirect Evaluation Methodology

As mentioned before, the methodology must be tailored depending on the affected subsystem and the type of failure. In this research effort, 16 different failures varying in severity and type were analyzed for detection and identification. The analytical methods for the control stick reduction can be reduced to 9 different failures corresponding to 9 different subsystems. This is possible because the variation on severity and/or the changes between left and right subsystems does not change the logic behind the indirect evaluation concept. The subsystems investigated for the indirect evaluation analysis are stabilators, ailerons, rudders, throttle efficiency, angular rate sensors, structural damage to a wing and engine failures.

4.1.1 Locked Stabilator

A locked stabilator failure flight envelope reduction can be analyzed through the following process. It is important to note that the differentiation between left or right control surface failure is negligible. The flight envelope reduction analysis requires the following variables:

$$DIV = \delta_e$$

$$EDIV = d_e$$

$$ERVs = [a_x, a_z, H, M, V, q, p, \dot{q}, \alpha, \theta]$$

The relationship between DIV and EDIV can be calculated by a linear function such that:

$$d_e = f(\delta_e)$$

$$d_e = k_e \delta_e = k_e \frac{\delta_{eL} + \delta_{eR}}{2} \tag{5}$$

In which the suffixes "L" and "R" denote the left or right control surface and k_e is a proportional value.

The nominal ranges for the EDIV are calculated as follows:

$$d_{emax} = k_e \frac{\delta_{eLmax} + \delta_{eRmax}}{2} \tag{6}$$

$$d_{emin} = k_e \frac{\delta_{eLmin} + \delta_{eRmin}}{2} \tag{7}$$

where typically $d_{emax} = -d_{emin}$.

After a failure is present in the system the new ranges for the EDIVs can be calculated as follows:

$$d_{emaxF} = k_e \frac{\delta_{eLF} + \delta_{eRmax}}{2} \tag{8}$$

$$d_{eminF} = k_e \frac{\delta_{eLF} + \delta_{eRmin}}{2} \tag{9}$$

where the value δ_{eLF} is the failure magnitude (the position in which the control surface is locked) determined by the direct quantitative evaluation process. It is important to note that the same concept can be applied to a right stabilator lock with no further modifications to the analysis.

The calculated ranges of the EDIV are considered to be virtual ranges since the actual control stick in the cockpit is still capable of moving freely as in nominal conditions. However, the difference is that the effectiveness of the control stick is impaired. The new calculated ranges of the EDIV are then used to determine the new ranges for all pertinent ERVs.

It should be considered that a stabilator failure may have an effect on the ability to produce rolling rates. The asymmetry induced by a stabilator failure may produce undesired roll moments that must be compensated by an aileron deflection. This required deflection reduces the control authority of the roll channel and thus the production of roll rates. This indirect effect

must be then calculated in order to determine the influence on the roll command channel and subsequently the corresponding aileron deflection.

The deflection necessary to compensate for the rolling effect produced by a locked stabilator failure is considered a pseudo-failure and can be approximated using the balance of rolling moments as follows:

$$C_{l\delta e_L}\delta_{eLF} = -C_{l\delta a}\delta_{aPF} \tag{10}$$

$$\delta_{aPF} = -\frac{c_{l\delta eL}}{c_{l\delta a}}\delta_{eLF} \tag{11}$$

in which δ_{aPF} is the non-zero trim aileron deflection necessary to compensate the investigated failure.

If we assume a generic aileron deflection convention to be the following:

$$\delta_a = \frac{\delta_{aL} + \delta_{aR}}{2} \tag{12}$$

and the relationship with its corresponding EDIV:

$$d_a = k_a\delta_a \tag{13}$$

the nominal lateral control ranges are:

$$d_{amax} = k_a\delta_{amax} \tag{14}$$

$$d_{amin} = k_a\delta_{amin} \tag{15}$$

Under stab failure, this becomes:

$$d_{amaxF} = k_a(\delta_{amax} - \delta_{aPF}) \tag{16}$$

$$d_{aminF} = k_a(\delta_{amin} - \delta_{aPF}) \tag{17}$$

and finally:

$$d_{amaxF} = k_a(\delta_{amax} + \frac{c_{l\delta eL}}{c_{l\delta a}}\delta_{eLF}) \tag{18}$$

$$d_{aminF} = k_a(\delta_{amin} + \frac{c_{l\delta eL}}{c_{l\delta a}}\delta_{eLF}) \tag{19}$$

It is important to note that the same concept can be applied to a right stabilator lock with no further modifications to the analysis. Also, the aileron pseudo-failure range is the same if the opposite stabilator is locked in the opposite direction.

4.1.2 Locked Aileron

A locked aileron failure flight envelope reduction can be analyzed through the following process. It is important to note that the differentiation between a left or right control surface failure is negligible. The flight envelope reduction analysis requires the following variables:

$$DIV = \delta_a$$

$$EDIV = d_a$$

$$ERVs = [a_y, p, r, \dot{p}, \dot{r}, \beta, \varphi]$$

The relationship between DIV and EDIV can be calculated by a linear function such that:

$$d_a = f(\delta_a)$$

$$d_a = k_a \delta_a = k_a \frac{\delta_{aL} + \delta_{aR}}{2} \tag{20}$$

in which the suffixes "L" and "R" denote the left or right control surface and k_a is a proportional value.

The nominal ranges for the EDIV are calculated as follows:

$$d_{amax} = k_a \frac{\delta_{aLmax} + \delta_{aRmax}}{2} \tag{21}$$

$$d_{amin} = k_a \frac{\delta_{aLmin} + \delta_{aRmin}}{2} \tag{22}$$

where typically $d_{amax} = -d_{amin}$.

After a failure is present in the system the new ranges for the EDIV can be calculated as follows:

$$d_{amaxF} = k_a \frac{\delta_{aLF} + \delta_{aRmax}}{2} \tag{23}$$

$$d_{aminF} = k_a \frac{\delta_{aLF} + \delta_{aRmin}}{2} \tag{24}$$

where the value δ_{aLF} is the failure magnitude (the position in which the control surface is locked) determined by the direct quantitative evaluation process. It is important to note that the same concept can be applied to a right aileron lock with no further modifications to the analysis.

The calculated ranges of the EDIV are considered to be virtual ranges since the actual control stick in the cockpit is still capable of moving freely as in nominal conditions. However, the difference is that the effectiveness of the control stick is impaired. The new calculated ranges of the EDIV are then used to determine the new ranges for all pertinent ERVs.

4.1.3 Locked rudder

A locked rudder failure flight envelope reduction can be analyzed through the following process. It is important to note that the differentiation between the left or right control surface is negligible. The flight envelope reduction analysis requires the following variables:

$$DIV = \delta_r$$

$$EDIV = d_r$$

$$ERVs = [a_y, p, r, \dot{p}, \dot{r}, \beta, \varphi]$$

The relationship between DIV and EDIV can be calculated by a linear function such that:

$$d_r = f(\delta_r)$$

$$d_r = k_r \delta_r = k_r \frac{\delta_{rL} + \delta_{rR}}{2} \tag{25}$$

in which the suffixes "L" and "R" denote the left or right control surface and k_r is a proportional value.

The nominal ranges for the EDIV are calculated as follows:

$$d_{rmax} = k_r \frac{\delta_{rLmax} + \delta_{rRmax}}{2} \tag{26}$$

$$d_{rmin} = k_r \frac{\delta_{rLmin} + \delta_{rRmin}}{2} \tag{27}$$

where typically $d_{rmax} = -d_{rmin}$.

After a failure is present in the system the new ranges for the EDIV can be calculated as follows:

$$d_{rmaxF} = k_r \frac{\delta_{rLF} + \delta_{rRmax}}{2} \tag{28}$$

$$d_{rminF} = k_r \frac{\delta_{rLF} + \delta_{rRmin}}{2} \tag{29}$$

where the value δ_{rLF} is the failure magnitude (The position in which the control surface is locked) determined by the direct quantitative evaluation process. It is important to note that the same concept can be applied to a right rudder lock with no further modifications to the analysis.

The calculated ranges of the EDIV are considered to be virtual ranges since the actual control stick in the cockpit is still capable of moving freely as in nominal conditions. However, the difference is that the effectiveness of the control stick is impaired. The new calculated ranges of the EDIV are then used to determine the new ranges for all pertinent ERVs.

4.1.4 Locked Throttle

A locked throttle valve failure flight envelope reduction can be analyzed through the following process. It is important to note that the differentiation between left or right throttle valves is negligible. The flight envelope reduction analysis requires the following variables:

$$DIV = \delta_{TL}$$

$$EDIV = d_T$$

$$ERVs = [a_x, a_z, r, H, M, V, \alpha, \theta]$$

The relationship between DIV and EDIV can be calculated by a linear function such that:

$$d_T = f(\delta_{TL})$$

$$d_T = k_T \delta_{TL} = k_T \frac{\delta_{TL} + \delta_{TR}}{2} \tag{30}$$

In which the suffixes "L" and "R" denote the left or right throttle valve and k_T is a proportional value.

The nominal ranges for the EDIV are calculated as follows:

$$d_{Tmax} = k_T \frac{\delta_{TLmax} + \delta_{TRmax}}{2} = 100\% \tag{31}$$

$$d_{Tmin} = k_T \frac{\delta_{TLmin} + \delta_{TRmin}}{2} = 0\% \tag{32}$$

where typically $d_T \in [0,100\%]$

After a failure is present in the system the new ranges for the EDIV can be calculated as follows:

$$d_{TmaxF} = k_T \frac{\delta_{TLF} + \delta_{TRmax}}{2} = k_T \frac{\delta_{TLF} + 100}{2} \tag{33}$$

$$d_{TminF} = k_T \frac{\delta_{TLF} + \delta_{TRmin}}{2} = k_T \frac{\delta_{TLF}}{2} \tag{34}$$

where the value δ_{TLF} is the failure magnitude (the position in which the throttle valve is locked) determined by the direct quantitative evaluation process. It is important to note that the same concept can be applied to a right throttle valve failure with no further modifications to the analysis.

The calculated ranges of the EDIV are considered to be virtual ranges since the actual throttle stick in the cockpit is still capable of moving freely as in nominal conditions. However, the difference is that the effectiveness of the throttle stick is impaired. The new calculated ranges of the EDIV are then used to determine the new ranges for all pertinent ERVs.

It should be considered that a throttle actuator failure may have an effect on the capability to produce yawing rates. The asymmetry induced by a throttle failure may produce undesired yaw moments that must be compensated by a rudder deflection. This required deflection reduces the control authority of the yaw channel and thus the production of yaw rates. This indirect effect must then be calculated in order to determine the influence on the yaw command channel and subsequently the corresponding rudder deflection.

The deflection necessary to compensate for the yawing effect produced by a locked throttle failure is considered a pseudo-failure and can be approximated using the balance of yawing moments as follows:

$$C_{n\delta TL}\delta_{TLF} = -C_{n\delta r}\delta_{rPF} \tag{35}$$

$$\delta_{rPF} = -\frac{c_{n\delta TL}}{c_{n\delta r}}\delta_{TLF} \tag{36}$$

in which δ_{rPF} is the non-zero trim rudder deflection necessary to compensate the investigated failure.

If we assume a generic rudder deflection convention to be the following:.

$$\delta_r = \frac{\delta_{rL} + \delta_{rR}}{2} \tag{37}$$

and the relationship with its corresponding EDIV:

$$d_r = k_r\delta_r \tag{38}$$

the nominal lateral control ranges are:

$$d_{rmax} = k_r \delta_{rmax} \tag{39}$$

$$d_{rmin} = k_r \delta_{rmin} \tag{40}$$

Under throttle failure, this becomes:

$$d_{rmaxF} = k_r(\delta_{rmax} - \delta_{rPF}) \tag{41}$$

$$d_{rminF} = k_r(\delta_{rmin} - \delta_{rPF}) \tag{42}$$

and finally:

$$d_{rmaxF} = k_r(\delta_{rmax} + \frac{C_{n\delta TL}}{C_{n\delta r}}\delta_{TLF}) \tag{43}$$

$$d_{rminF} = k_r(\delta_{rmin} + \frac{C_{n\delta TL}}{C_{n\delta r}}\delta_{TLF}) \tag{44}$$

It is important to note that the same concept can be applied to a right throttle lock with no further modifications to the analysis. Also, the rudder pseudo-failure range is the same but different direction if the opposite throttle valve is locked.

4.1.5 Roll Sensor Bias

A roll sensor bias failure flight envelope reduction can be analyzed through the following process. The process relies on the specifics of the control system that uses the sensor information. The control laws of the aircraft determine a "desired" angular rate response determined by a reference model designed to satisfy handling qualities criteria. The flight envelope reduction analysis requires the following variables:

$$DIV = p_{meas}$$

$$EDIV = d_a$$

$$ERVs = [a_y, p, r, \dot{p}, \dot{r}, \beta, \varphi]$$

The relationship between DIV and EDIV can be calculated such that:

$$p_{ref} = TF(d_a) \tag{45}$$

in which TF represents a first order transfer function obtained from the roll channel reference model. It is possible to avoid the use of a transfer function, if it is not available, by finding a linear relation between the DIV and the EDIV.

As a result of the roll sensor failure, an undesired roll rate command is produced in order to compensate for the "fake" roll seen by the controller. In other words, the aircraft will rotate in the opposite direction of the sensor bias. This behavior must be compensated using an aileron command, reducing the authority of the roll channel. The excursion of the lateral stick control necessary to compensate for the control law roll command is equivalent to a bias in the form:

$$d_{aF} = TF^{-1}(p_{bias}) \tag{46}$$

in which d_{aF} is the necessary excursion of the lateral stick command in order to counteract the induced roll rate bias. It is important to note that the transfer function can be substituted by a linear relation between the DIV and the EDIV.

Assuming a linear relation between the stick input d_a and the control surface actuation δ_a, the nominal ranges for the EDIV when a positive roll ($p_{meas} > 0$) rate bias affects the system are calculated as follows:

$$d_{amaxF} = d_{amax} - d_{aF} \tag{47}$$

$$d_{aminF} = d_{amin} \tag{48}$$

Similarly, for a negative roll rate bias ($p_{meas} < 0$):

$$d_{amaxF} = d_{amax} \tag{49}$$

$$d_{aminF} = d_{amin} - d_{aF} \tag{50}$$

With the new ranges of effective lateral stick commands, it is possible to calculate the corresponding ranges for all the ERVs using the 2-dimensional projections.

4.1.6 Pitch Sensor Bias

A pitch sensor bias failure flight envelope reduction can be analyzed through the following process. The process relies on the specifics of the control system that uses the sensor information. The control laws of the aircraft determine a "desired" angular rate response determined by a reference model designed to satisfy handling qualities criteria. The flight envelope reduction analysis requires the following variables:

$$DIV = q_{meas}$$

$$EDIV = d_e$$

$$ERVs = [a_x, a_z, H, M, V, q, \dot{q}, \alpha, \theta]$$

The relationship between DIV and EDIV can be calculated such that:

$$q_{ref} = TF(d_e) \tag{51}$$

in which TF represents a second order transfer function obtained from the pitch channel reference model. It is possible to avoid the use of a transfer function, if it is not available, by finding a linear relation between the DIV and the EDIV.

As a result of the pitch sensor failure, an undesired pitch rate command is produced in order to compensate for the "fake" pitch seen by the controller. In other words, the aircraft will rotate in the opposite direction of the sensor bias. This behavior must be compensated using a stabilator/elevator command, reducing the authority of the pitch channel. The excursion of the longitudinal stick control necessary to compensate for the control law pitch command is equivalent to a bias in the form:

$$d_{eF} = TF^{-1}(q_{bias}) \tag{52}$$

in which d_{eF} is the necessary excursion of the longitudinal stick command in order to counteract the induce pitch rate bias. It is important to note that the transfer function can be substituted by a linear relation between the DIV and the EDIV.

Assuming a linear relation between the stick input d_e and the control surface actuation δ_e, the nominal ranges for the EDIV when a positive roll ($q_{meas} > 0$) rate bias affects the system are calculated as follows:

$$d_{emaxF} = d_{emax} - d_{eF} \tag{53}$$

$$d_{eminF} = d_{emin} \tag{54}$$

Similarly, for a negative pitch rate bias ($q_{meas} < 0$):

$$d_{emaxF} = d_{emax} \tag{55}$$

$$d_{eminF} = d_{emin} - d_{eF} \tag{56}$$

With the new ranges of effective longitudinal stick command, it is possible to calculate the corresponding ranges for all the ERVs using the 2-dimensional projections.

4.1.7 Yaw Sensor Bias

A yaw sensor bias failure flight envelope reduction can be analyzed through the following process. The process relies on the specifics of the control system that uses the sensor information. The control laws of the aircraft determine a "desired" angular rate response determined by a reference model designed to satisfy handling qualities criteria. The flight envelope reduction analysis requires the following variables:

$$DIV = r_{meas}$$

$$EDIV = d_r$$

$$ERVs = [a_y, p, r, \dot{p}, \dot{r}, \beta, \varphi]$$

The relationship between DIV and EDIV can be calculated such that:

$$r_{ref} = TF(d_r) \tag{57}$$

in which TF represents a second order transfer function obtained from the pitch channel reference model. It is possible to avoid the use of a transfer function, if it is not available, by finding a linear relation between the DIV and the EDIV.

As a result of the yaw sensor failure, an undesired yaw rate command is produced in order to compensate for the "fake" yaw seen by the controller. In other words, the aircraft will rotate in the opposite direction of the sensor bias. This behavior must be compensated using a rudder command, reducing the authority of the yaw channel. The excursion of the directional pedal control necessary to compensate for the control law yaw command is equivalent to a bias in the form:

$$d_{rF} = TF^{-1}(r_{bias}) \tag{58}$$

in which d_{rF} is the necessary excursion of the directional commands in order to counteract induced yaw rate bias. It is important to note that the transfer function can be substituted by a linear relation between the DIV and the EDIV.

Assuming a linear relation between the stick input d_r and the control surface actuation δ_r, the nominal ranges for the EDIV when a positive roll ($r_{meas} > 0$) rate bias affects the system are calculated as follows:

$$d_{rmaxF} = d_{rmax} - d_{rF} \tag{59}$$

$$d_{rminF} = d_{rmin} \tag{60}$$

Similarly, for a negative yaw rate bias ($r_{meas} < 0$):

$$d_{rmaxF} = d_{rmax} \tag{61}$$

$$d_{rminF} = d_{rmin} - d_{rF} \tag{62}$$

With the new ranges of effective directional commands, it is possible to calculate the corresponding ranges for all the ERVs using the 2-dimensional projections.

4.1.8 Wing Structural Damage

The flight envelope reduction estimation for a wing structural failure is discussed next. This failure requires more attention due to the shift in center of gravity and also the fact that the equilibrium of forces and moments on the body axis is compromised after the failure affects the system. It is important to note that the differentiation between left or right wing is negligible. The flight envelope reduction analysis requires the following variables:

$$DIV = [\Delta C_L \ \Delta C_M]$$

$$EDIV = [d_a, d_e, d_r, d_T]$$

$$ERV's = [H, M, V, p, q, r, a_x, a_y, a_z \ \dot{p}, \dot{q}, \dot{r}, \alpha, \beta, \theta, \varphi]$$

For nominal conditions during a steady-state flight, the forces and moments that act on the aircraft are shown in the figure below.
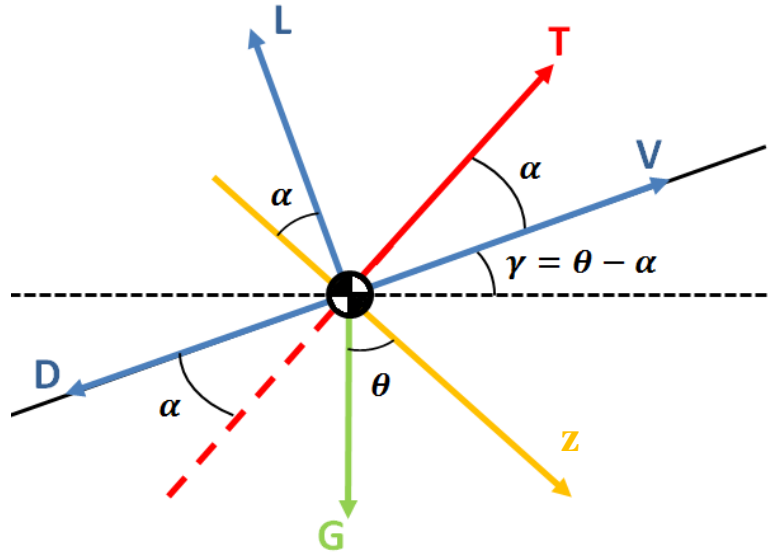

Figure 21- Forces and Moments on an Aircraft

The equilibrium of longitudinal forces and moments in the body axis satisfy the following statements:

$$\sum F_x = 0 = T + L \sin \alpha - D \cos \alpha - G \sin \theta \tag{63}$$

$$\sum F_z = 0 = -L \cos \alpha - D \sin \alpha + G \cos \theta \tag{64}$$

$$\sum M_y = 0 = M_W = -M_{TH} \tag{65}$$

$$C_L = C_L(C_D) \tag{66}$$

$$C_{lw} = 0 \tag{67}$$

in which:

$$T(M, H, d_T), L(M, \alpha), D(M, \alpha), M_W(M, \alpha), M_{HT}(M, \alpha, \delta_e), C_L(M, \alpha), C_D(M, \alpha)$$

Let us assume that certain flight conditions namely, Mach number, altitude and flight path angle, are required to be unchanged at post-failure conditions. The failure will invariably produce changes, or deltas, on all other variables leading to:

$$\sum F_x = 0 = T + \Delta T + (L + \Delta L) \sin(\alpha + \Delta\alpha) - (D + \Delta D) \cos(\alpha + \Delta\alpha) - G \sin(\theta + \Delta\theta) \tag{68}$$

$$\sum F_z = 0 = -(L + \Delta L) \cos(\alpha + \Delta\alpha) - (D + \Delta D) \sin(\alpha + \Delta\alpha) + G \cos(\theta + \Delta\theta) \tag{69}$$

$$\sum M_y = 0 = (M_W + \Delta M_W) = (-M_{HT} - \Delta M_{HT}) \tag{70}$$

$$C_{LF} = C_{LF}(C_D) \tag{71}$$

In which l_{AL} is the distance between the plane of symmetry of the aircraft and the aerodynamic center of the wing after failure.

The differential lift between wings will produce an undesired rolling moment that must be compensated using aileron deflection. In other words, a pseudo-failure of the aileron exists.

The aileron deflection required to compensate for the undesired rolling moment can be approximated using the balance of rolling moments such that:

$$-\Delta C_{LW} \cdot l_{AL} = \varepsilon_a b C_{L\delta a} \cdot \delta_{aPF} \tag{72}$$

$$\delta_{aPF} = -\frac{\Delta C_{LW} \cdot l_{AL}}{\varepsilon_a b C_{L\delta a}} \tag{73}$$

where b is the wingspan and ε_a is an alteration factor that depends on how the aileron is affected due to the wing failure. For example, if $\varepsilon_a = 1$ the aileron is in perfect conditions after the failure and its actuation is nominal, and if $\varepsilon_a = 0.5$ the aileron is completely damaged and unusable.

Assuming the relationship with its EDIV:

$$d_a = k_a \delta_a \tag{74}$$

the nominal lateral control ranges are:

$$d_{amax} = k_a \delta_{amax} \tag{75}$$

$$d_{amin} = k_a \delta_{amin} \tag{76}$$

Under stab failure, this becomes:

$$d_{amaxF} = k_a (\delta_{amax} - \delta_{aPF}) \tag{77}$$

$$d_{aminF} = k_a (\delta_{amin} - \delta_{aPF}) \tag{78}$$

This eventually becomes:

$$d_{amaxF} = k_a (\delta_{amax} + \frac{\Delta C_{LW} \cdot l_{AL}}{\varepsilon_a b C_{L\delta a}}) \tag{79}$$

$$d_{aminF} = k_a (\delta_{amin} + \frac{\Delta C_{LW} \cdot l_{AL}}{\varepsilon_a b C_{L\delta a}}) \tag{80}$$

Now that the ranges for the aileron control are known, it is possible to determine the effects of the wing failure on $p,\ r, a_y, \dot{p},\ \dot{r}, \beta, \varphi$ using the 2-dimensional projections.

A similar analysis can be performed for the pitching moment coefficient as well. If the alteration is known, the reduction in the longitudinal control authority can be determined using the pitching moment balance as follows:

$$\Delta M_{WF} = -\Delta M_{HT} \tag{81}$$

$$\Delta C_{mW} = -C_{m\delta e} \cdot \Delta \delta_{ePF} \tag{82}$$

$$\Delta \delta_{ePF} = -\frac{\Delta C_{mW}}{C_{m\delta e}} \tag{83}$$

The new ranges for d_e when $\Delta \delta_{ePF} > 0$ are:

$$d_{emaxF} = d_{emax} - k_e \cdot \Delta \delta_{ePF} \tag{84}$$

$$d_{eminF} = d_{emin} \tag{85}$$

Similarly, for $\Delta \delta_{ePF} < 0$:

$$d_{emaxF} = d_{emax} \tag{86}$$

$$d_{eminF} = d_{emin} - k_e \cdot \Delta \delta_{ePF} \tag{87}$$

Now that the ranges for the stabilator/elevator control are known, it is possible to determine the effects of the wing failure on $q, a_x, a_z, \dot{q}, \alpha, \theta$ using the 2-dimensional projections.

4.1.9 Engine Reduced Effectiveness

The reduction estimation of the flight envelope of an aircraft due to engine deficiencies can be calculated using the following process. It is important to note that the differentiation between left or right control engine is negligible. The flight envelope reduction analysis requires the following variables:

$$DIV = T_L$$

$$EDIV = d_T$$

$$ERVs = [a_x, a_z, r, H, M, V, \alpha, \theta]$$

The relationship between DIV and EDIV can be calculated by a linear function such that:

$$d_T = f(T_L + T_R) \tag{88}$$

in which the suffixes "L" and "R" denote the left or right engine and f is a linear function.

The nominal ranges for the EDIV are calculated as follows:

$$d_{Tmax} = f(T_{Lmax} + T_{Rmax}) = 100\% \tag{89}$$

$$d_{Tmin} = f(T_{Lmin} + T_{Rmin}) = 0\% \tag{90}$$

where typically $d_T \in [0,100\%]$.

After a failure is present in the system the new ranges for the EDIV can be calculated as follows:

$$d_{TmaxF} = f(T_{LmaxF} + T_{Rmax}) \tag{91}$$

$$d_{TminF} = f(T_{LminF} + T_{Rmin}) \tag{92}$$

Note that the minimum values for thrust do not change after a failure. It is important to note that the same concept can be applied to a right engine failure with no further modifications to the analysis.

The new calculated ranges of the EDIV are then used to determine the new ranges for all pertinent ERVs except for yaw rate. An engine failure will produce an undesired yawing moment that needs to be explored with more depth. The undesired yawing moment must be corrected by a rudder deflection which subsequently reduces the authority of the yaw command channel. This rudder pseudo-failure is a function of the throttle command and can be assumed to be a linear relation. Therefore, the total yawing moment can be expressed such that:

$$C_{n\delta TL} \cdot \delta_{TL} + C_{n\delta TR} \cdot \delta_{TR} = 0 \tag{93}$$

$$(C_{n\delta TL} + C_{n\delta TR})d_T = 0 \tag{94}$$

Note that for a healthy system the control derivatives satisfy the expression $C_{n\delta TL} = -C_{n\delta TR}$. On the other hand if a failure is present in the system the statement is no longer true.

$$(C_{n\delta TL} + C_{n\delta TR})d_T \neq 0 \tag{95}$$

Therefore, the rudder actuation necessary to counteract the yawing moment due to an engine failure can be calculated through

$$(C_{n\delta TL} + C_{n\delta TR})d_T = -C_{n\delta r} \cdot \delta_{rPF} \tag{96}$$

$$\delta_{rPF}(d_T) = -\frac{(C_{n\delta TL} + C_{n\delta TR})}{C_{n\delta r}}d_T \tag{97}$$

where δ_{rPF} is the rudder deflection necessary to compensate for the undesired yawing moment due to an engine failure.

The relationship with its corresponding EDIV for this pseudo-failure is:

$$d_r = k_r \delta_r \tag{98}$$

The nominal directional control ranges are:

$$d_{rmax} = k_r \delta_{rmax} \tag{99}$$

$$d_{rmin} = k_r \delta_{rmin} \tag{100}$$

Under stabilator failures, this becomes:

$$d_{rmaxF} = k_r(\delta_{rmax} - \delta_{rPF}(d_T)) \tag{101}$$

$$d_{rminF} = k_r(\delta_{rmin} - \delta_{rPF}(d_T)) \tag{102}$$

and finally:

$$d_{rmaxF} = k_r(\delta_{rmax} - \frac{(C_{n\delta TL} + C_{n\delta TR})}{C_{n\delta r}}d_T) \tag{103}$$

$$d_{rminF} = k_r(\delta_{rmin} - \frac{(C_{n\delta TL} + C_{n\delta TR})}{C_{n\delta r}}d_T) \tag{104}$$

It is important to note that the same concept can be applied to a right engine failure with no further modifications to the analysis. Also, the pseudo failure is the same as in the throttle

failure case except that the range varies with respect to throttle input d_T. The new ranges of the directional control can be used to determine the effects on yawing moment.

4.2 Flight Envelope Reduction Results

The flight envelope reduction algorithm was implemented for 8 failures with several ranges of severity. The proposed algorithm presents a simple analytical approach to estimate the effects on the maneuverability, performance, handling qualities and flight envelope reduction under upset conditions. For the most part, the results obtained from the analysis of the simulation tests fell within the expected limits. The reduction of relevant variables directly related to the flight dynamics characteristics namely angular rates, linear accelerations and angular rate accelerations among others are often estimated correctly. On the other hand, the reduction in variables such as altitude, velocity, angle of attack and angle of sideslip present more inaccurate results.

It is confirmed that the estimated values obtained from the analytical analysis that define the limits of the ERVs are actual bounds of the expected flight envelope reduction for the vast majority of ERVs. In other words, the results from the analytical analysis do delimit the flight envelope variables when compared with the 2-dimensional projections of the investigated selves/non-selves.

The graphical method used to determine the ranges of ERVs mentioned earlier in this chapter was applied to 5 failures. Examples of this method applied to actual sensor failures are presented in figures 22 and 23.
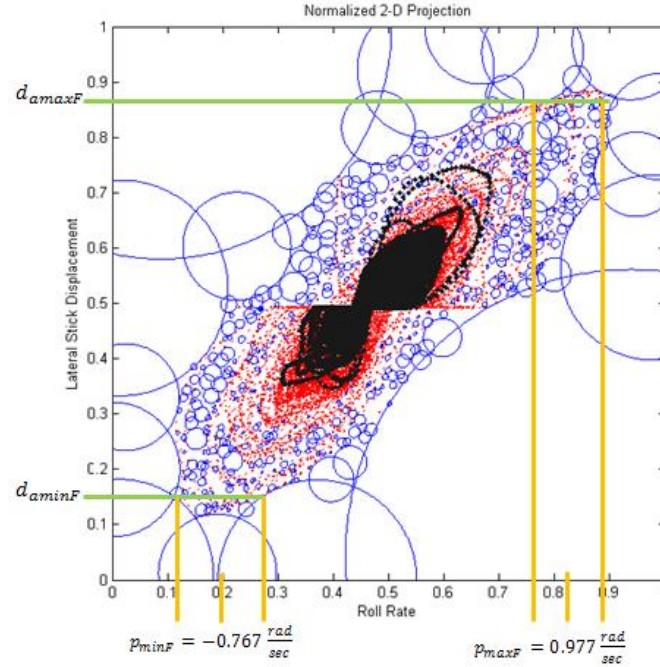
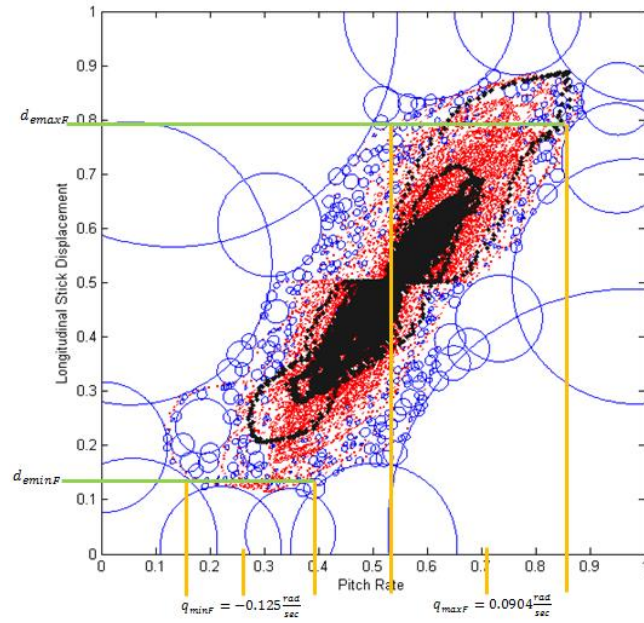**Figure 22-Reduction in Roll Rate After Roll Sensor Bias Failure**

Figure 23- Reduction in Pitch Rate After Pitch Sensor Bias Failure

It should be noted that the red clusters represent the self and the blue clusters represent the non-self. Also, the black points represent the dynamics of the aircraft after a failure.

Tables 14 through 18 present the results for the estimated reduced ranges of ERVs for several flight test scenarios that include 5 types of failures of varying magnitudes. It should be noted that for simplicity and to avoid excessive length the results presented below are just a sample set of the investigated failures.

Table 14- Estimated Reduced Ranges for a Roll Rate Sensor Bias of 5deg/sec Failure

ERV	Nominal Ranges	Estimated Reduced Range
p (rad/sec)	[1.08194, -0.9373]	[0.9779 +/- 0.018, -0.7677+/- 0.018]
r (rad/sec)	[0.0621, -0.0637]	[0.0237 +/- 0.007, -0.0209+/- 0.0015]
a_y (g's)	[0.6658, -0.8634]	[0.159 +/-0.15, -0.228 +/-0.15]
$\dot{p}$ (rad/sec^2)	[3.0897, -2.9513]	[1.5613 +/-0.31, -1.894 +/- 0.21]
$\dot{r}$ (rad/sec^2)	[0.1264, -0.1173]	[0.029 +/-0.008, -0.0123 +/-0.008]
β (rad)	[0.1712, -0.1675]	[0.0173 +/-0.002, -0.0142 +/-0.002]
φ (rad)	[0.6027, -1.0177]	[0.235 +/-0.01, -0.355 +/- 0.01]

Table 15- Estimated Reduced Ranges for a Pitch Rate Sensor Bias of 1 deg/sec Failure

ERV	Nominal Range	Estimated Reduced Range
a_x (g's)	[0.5973, -0.3437]	[0.2586 +/-0.04, -0.102 +/-0.04]
a_z (g's)	[3.0488, -2.4616]	[2.26 +/- 0.29, -1.376 +/-0.29]
H (ft)	[9929.16, 2012.33]	[7936.5 +/-1500, 2911.5 +/- 700]
M	[0.95, 0.5261]	[0.916 +/-0.017, 0.724 +/-0.047]
V (ft/sec)	[301.32, 157.69]	[245.33 +/- 43, 191.2 +/- 23]
q (rad/sec)	[0.1831, -0.2247]	[0.0904 +/-0.023, -0.125 +/-0.023]
$\dot{q}$ (rad/sec^2)	[0.3857, -0.3825]	[0.195 +/-0.12, -0.12 +/-0.082]
α (rad)	[0.2339, -0.1013]	[0.152 +/-0.014, -0.059 +/-0.014]
θ (rad)	[0.3985, -0.3253]	[0.253 +/-0.004, -0.127 +/-0.004]

Table 16- Estimated Reduced Flight Envelope for Left Throttle Command Stuck at 90%

Variable	Nominal Range	Estimated Reduced Range
a_x	[0.5973, -0.3437]	[0.3421,-0.2979]
a_z	[3.0488, -2.4616]	[2.4856,-1.5402]
r	[0.0621, -0.0637]	[0.0399,-0.031556]
H	[9929.16, 2012.33]	[9927.97,2013.52]
M	[0.95, 0.5261]	[0.94961,0.5595]
V	[301.32, 157.69]	[291.9707,163.1548]
α	[0.2339, -0.1013]	[0.1622,-0.09015]
θ	[0.3985, -0.3253]	[0.39322,-0.32331]

Table 17- Estimated Reduced Flight Envelope for Left Throttle Command Stuck at 60%

Variable	Nominal Range	Estimated Reduced Range
a_x	[0.5973, -0.3437]	[0.3715,-0.2157]
a_z	[3.0488, -2.4616]	[2.585,-2.2852]
r	[0.0621, -0.0637]	[0.03851,-0.0335]
H	[9929.16, 2012.33]	[9881.67,2059.83]
M	[0.95, 0.5261]	[0.95016,0.553]
V	[301.32, 157.69]	[301.399,171.36]
α	[0.2339, -0.1013]	[0.21014,-0.0905]
θ	[0.3985, -0.3253]	[0.39416,-0.2550]

Table 18- Estimated Reduced Flight Envelope for Left Throttle Command Stuck at 10%

Variable	Nominal Range	Estimated Reduced Range
a_x	[0.5973, -0.3437]	[0.50607,-0.27689]
a_z	[3.0488, -2.4616]	[2.0128,-2.0703]
r	[0.0621, -0.0637]	[0.040149,-0.0466]
H	[9929.16, 2012.33]	[9881.659,2214.211]
M	[0.95, 0.5261]	[0.9474,0.53141]
V	[301.32, 157.69]	[283.662,161.6317]
α	[0.2339, -0.1013]	[0.192,-0.094945
θ	[0.3985, -0.3253]	[0.394,-0.26450]

Chapter 5 Neurally-Augmented Detection and Identification

Previous research efforts in fault tolerant systems have used ANNs as nonlinear approximators that rely on analytical redundancies among inputs, states and outputs of a system. The ANN used for this efforts are strictly treated as a dynamic system with inputs, outputs, and states in which the dynamics of the approximation process are considered an essential part of the system. In particular, these ANNs give emphasis to the approximation results rather than the approximation process. The ANNs were designed for real-time applications as part of control systems. Thus, the speed, efficiency, overall performance and the computational requirements of the ANN code provide a suitable tool for flight test implementation.

Former AIS-based FDIE schemes relied on parameters computed by fault tolerant control laws augmented with ANNs. This characteristic limits the application of such algorithms only to aircraft equipped with such control systems. In this research effort an alternate design of ANNs which are not aircraft dependent provides a viable solution. The architecture uses an open-loop online training ANNs system that augments a direct adaptive control law design with a "model following" architecture. The open-loop characteristic implies that the ANNs are no longer part of the control laws. This ensures portability of the FDIE scheme for several platforms regardless of their control systems.

The ANN's outputs represent estimates of the angular acceleration on the three body axes with proven potential for self/non-self definitions due to their excellent capabilities to capture the dynamic fingerprints of upset conditions. In particular, the ANN architecture used in this research effort is the "Adaptive Linear Element" (ADALINE) network, which consists of a single neuron of the McCulloch-Pitts type.

5.1 Model Following Controller

As mentioned before, a "model following" type of controller is used in order to provide inputs to the ANNs that subsequently will output the estimates for angular rates used for the definition selves/non-selves projections and finally FDIE. The controller is maintained as an open-loop system and it will not serve as a control augmentation mechanism. The architecture of the model following is presented.

The pilot inputs are expressed in terms of longitudinal, lateral and directional stick/pedal displacements:

$$\delta_{cmd} = [\ \delta_{lon_{stick}} \delta_{lat_{stick}}\ \delta_{dir_{pedal}}]$$

These flight commands are then converted into preliminary command states:

$$p_{cmd} = k_{lat}\delta_{lat_{stick}} \tag{105}$$

$$n_{z\,cmd} = k_{lon}\delta_{lon_{stick}} \tag{106}$$

$$n_{y\,cmd} = k_{dir}\delta_{dir_{pedal}} \tag{107}$$

where p_{cmd} is a roll rate, $n_{z\,cmd}$ is a vertical load factor, $n_{y\,cmd}$ lateral load factor and $k_{lat}, k_{lon}, k_{dir}$ are stick and pedal gains.

Next, the commanded angular rates are obtained such that:

$$p_{cmd} = p_{cmd} \tag{108}$$

$$q_{cmd} = \frac{g}{V}n_{z\,cmd} \tag{109}$$

$$r_{cmd} = \frac{g}{V}(n_{y\,cmd} + \sin\phi) \tag{110}$$

where g is gravity, V is velocity and ϕ is the bank angle.

These commanded angular rates are then converted to filtered reference angular rates $(p_{ref}, q_{ref}, r_{ref})$through first and second order transfer functions such that:

$$p_{ref}(s) = \frac{1}{\tau_p s + 1} p_{cmd}(s) \tag{111}$$

$$q_{ref}(s) = \frac{\omega_q^2}{s^2 + 2\zeta_q \omega_q + \omega_q^2} q_{cmd}(s) \tag{112}$$

$$r_{ref}(s) = \frac{\omega_r^2}{s^2 + 2\zeta_r \omega_r + \omega_r^2} r_{cmd}(s) \tag{113}$$

The tracking errors between actual and reference angular rates are given by $e_x = x_{ref} - x$ in which $x = p, q, r$ are calculated. The tracking errors are used to provide proportional, integral and derivative compensation.

Next, the pseudo-controls in terms of acceleration command are calculated. These values are only used as inputs the ANN and are obtained through:

$$U_p = \left(K_{p_p} + \frac{K_{i_p}}{s} + K_{d_p} s \right) \cdot e_p + s \cdot p_{ref} \tag{114}$$

$$U_q = \left(K_{p_q} + \frac{K_{i_q}}{s} + K_{d_q} s \right) \cdot e_q + s \cdot q_{ref} \tag{115}$$

$$U_r = \left(K_{p_r} + \frac{K_{i_r}}{s} + K_{d_r} s \right) \cdot e_r + s \cdot r_{ref} \tag{116}$$

where K_p, K_i, K_d are the constant values of proportional-integral-derivative type controller, respectively. Since the outputs of the ANN are not used for control purposes, all the parameters mentioned before are generic. The difference between nominal and abnormal condition outputs is the main variable used for FDIE.

5.2 ADALINE Network

In general an adaptive linear neuron (ADALINE) network consists of a single neuron of the McCulloch-Pitts type in which weights integrated in the architecture are determined by delta rule also known as the least mean square training law. However, the ADALINE network used in this research effort utilizes a steepest descent gradient to minimize the error between measurements and estimates. This characteristic allows the system to adapt and update constantly. The architecture of the ADALINE network is shown in Figure 24.
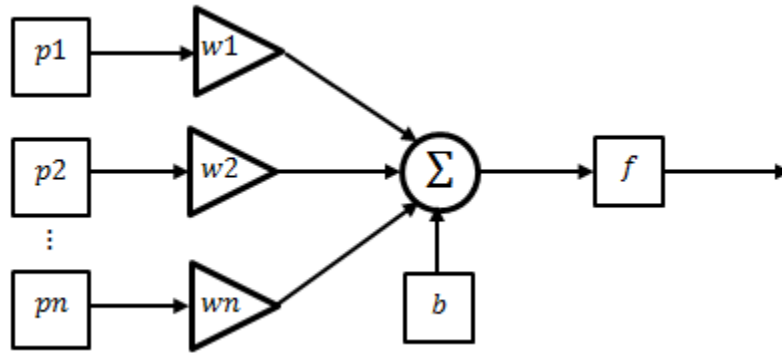

Figure 24- ADALINE Network

The ADALINE network approximates a vector signal $y \in R^n$ as a linear combination of M input vectors $x \in R^m$ during its training phase. For example, assume that $y = Wx$ where $W \in R^{m \times n}$ is a matrix of real numbers that is constantly updated by a steepest descent gradient method. Then, the update law results in:

$$W(k + 1) = W(k) - \gamma e(k)x(k) - mod(e, W)$$

$$(117)$$

where γ is the learning rate, $mod(e, W)$ is the modification that stabilizes the update law and $e(k)$ is the current estimation error of the form $e(k) = y(k) - y_s(k)$.

The inputs to the ANN include pseudo control acceleration commands (U_p, U_q, U_r) obtained from the reference model controller, sensor feedback and the previous step network outputs $(U_{p_{ad}}, U_{q_{ad}}, U_{r_{ad}})$. In general, for each channel (pitch, roll and yaw) the ANN's inputs are altitude, velocity, angle of attack and sideslip angle. Furthermore, each channel in particular receives the following inputs:

$$Roll\ Channel = bias, \frac{1 - e^{\left(-U_p - U'_{p_{ad}}\right)}}{1 - e^{\left(-U_p - U'_{p_{ad}}\right)}}, p, r, p_{ref} \tag{118}$$

$$Pitch\ Channel = bias, \frac{1 - e^{\left(-U_q - U'_{q_{ad}}\right)}}{1 - e^{\left(-U_p - U'_{q_{ad}}\right)}}, q, q_{ref} \tag{119}$$

$$Yaw\ Channel = bias, \frac{1 - e^{\left(-U_r - U'_{r_{ad}}\right)}}{1 - e^{\left(-U_r - U'_{r_{ad}}\right)}}, p, r, r_{ref} \tag{120}$$

Ultimately, for on-line learning the ANN uses PID acceleration errors in the form of:

$$U_{x\_error} = \left(K_{p_{x\_error}} + \frac{K_{i_{x\_error}}}{s} + K_{d_{x\_error}} s \right) \cdot e_x \tag{121}$$

where

$$x = p, q, or\ r$$

Chapter 6 Flight Test

An initial performance analysis of the proposed FDIE solution has been assessed through flight tests. A set of two nominal flights that included figure-8-maneuvers as well as control surface doublets in all three axes were performed in order to capture the entire flight envelope dynamics of the test aircraft. The platform used for this experimentation was the Skywalker RC aircraft instrumented with an APM 2.5 micro controller. The RC aircraft was flown manually by a pilot on the ground and the maneuvers were implemented in a sequential order. The nominal flight tests consisted of the following stages:

1. Manual flight until an altitude of 80 meters

2. Trim flight at constant speed

3. Figure 8 maneuver

4. Elevator Doublet

5. Figure 8 maneuver

6. Aileron Doublet

7. Figure 8 maneuver

8. Rudder Doublet

9. Figure 8 maneuver

The flight test sequence was performed twice and the data were saved into a flash memory for later processing. The data was recorded at a rate of 50Hz and included angular rates, linear accelerations, neural network angular rate estimations, reference model commands, Euler angles, altitude, stick inputs and velocity. A total of 21 features were recorded. Using Equation

(1) it can be determined that a total of 210 Self/Non-Self projections can be generated for the performance assessment through flight test implementation. The reduced flight envelope limits of the Skywalker as well as the length of the flights yield a reduced amount of data points. This should be taken into consideration during the generation of projections in order to obtain proper coverage of the Self.

For validation purposes, four different types of failures were injected into the system at later flights in order to capture the dynamic fingerprint of abnormal conditions on the test platform. The failures investigated in flight test included low and high magnitude aileron failures. The failures were injected manually by the pilot through a PWM signal sent from an RC transmitter. Once a failure was injected, the sequence of maneuvers presented above was attempted by the pilot. Table 19 presents the injected failures in the system.

Table 19- Failures Injected in Flight Tests

Failure #1	Left Aileron Locked at Wings Level Trim
Failure #2	Right Aileron Locked at Wings Level Trim
Failure #3	Left Aileron Locked at Trim during Bank Turn
Failure #4	Right Aileron Locked at Trim during Bank Turn

6.1 Test Platform

The RC airplane chosen for experimentation was the "New Skywalker 1880". This platform offers a stable and low-cost system that is able to satisfy the needs of the flight tests. Figure 25 shows the actual system used for the flight tests. This platform was used in previous projects for which system identification techniques were performed. The physical characteristics of the system are presented in Table 20.

90

Figure 25- Skywalker 1880 RC

Table 20- Skywalker Dimensions and Mass Properties

Wing Area (m^2)	0.41143
Wing MAC (m)	0.22647
Wingspan (m)	1.88
Horizontal Tail Span (m)	0.5626
Horizontal Tail MAC (m)	17.1
Vertical Tail Span (m)	0.244
Vertical Tail MAC (m)	19.5
Total Length (m)	1.183
Weight (Kg)	0.9525

6.2 Hardware Instrumentation

The Skywalker 1880 was equipped with a set of analog and digital sensors that provide essential variables for the generation of Selves and Non-selves. Primarily, the onboard microcontroller is an APM 2.5 with an "Atmel ATMEGA 2560" processor. The APM 2.5 board includes embedded sensors such as an IMU, magnetometer and a 4MB data flash chip as well as digital and analog ports for GPS, telemetry and a pitot tube sensor. Figure 26 shows the APM and the setup inside the fuselage of the Skywalker.
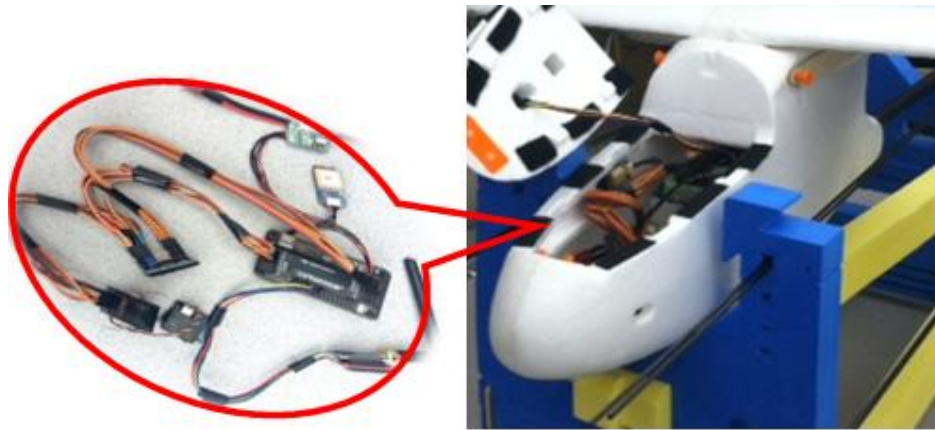
Figure 26- Onboard APM 2.5

6.2.1 APM 2.5

The Ardupilot Mega (APM) 2.5 is an out-of-the-shelf low-cost autopilot solution produced by 3D Robotics. Its dimensions are approximately 66x40 mm. and with the enclosure it weighs approximately 20 grams. It uses an 8-bit, 16Mhz "Atmel AT Mega 2560" processor which has 54 digital I/O pins for which 14 can be used for PWM signals. Figure 27 shows a close view of the board.
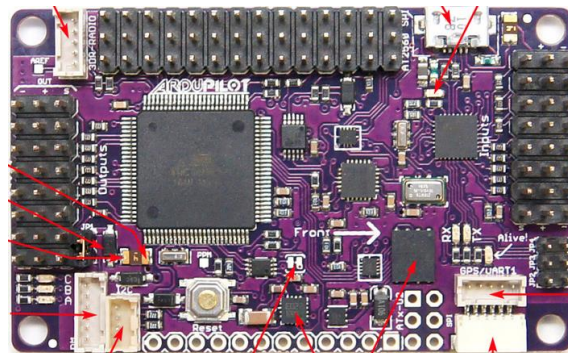

Figure 27- APM 2.5 Close View

6.2.2 InvenSense MPU-6000 Inertial Sensor

The MPU-600 is a 6-axis motion tracking device. It combines a 3-axis gyroscope and a 3-axis accelerometer in a 4x4x0.9mm QFN footprint and it communicates through a serial interface in an IC2 protocol. Figure 28 shows an image of the MPU-6000 inertial sensor.

Figure 28- MPU-6000

6.2.3 MEAS Switzerland MS5611 Barometric Pressure Sensor

This barometric pressure sensor offers a high resolution altimeter sensor with SPI and I2C bus interfaces up to 20MHz. It offers a factory calibrated sensor with a resolution of 10 cm. and its dimensions corresponds to a 5x3x1mm QFN footprint. Figure 29- shows an image of the MS5611 sensor.

Figure 29- MS5611-01BA093 Barometric Pressure Sensor

6.2.4 Data Flash

The data flash card is a 4Mb chip embedded in the APM 2.5 board. Previous efforts have shown that recording 20 floating point parameters at 50Hz it is possible to record approximately 17 minutes of data recording. Figure 30 shows an image of the flash chip on the APM 2.5 board.

Figure 30- Data Flash Memory Chip

6.2.5 MediaTek MT3329 GPS

The MT3329 is a 66 channel single chip solution with a binary output protocol with an update rate of 10Hz. Its sensitivity can be up to -165dB tracking, a position accuracy of less than 3meters and USB/UART interfaces. Its dimensions are 38x16x7.8mm and it weighs 9.45g. Figure 31 shows an image the GPS.

Figure 31- MediaTek MT3329

6.2.6 Freescale MPXV7002DP Differential Pressure Sensor

This analog sensor was connected directly to a miniature pitot tube located on the right wing. Its maximum rating for pressure is 2kPa at 60° C. This sensor provides true air speed measurements that are used for the definitions of selves. Figure 32 shows the analog sensor and the pitot tube.
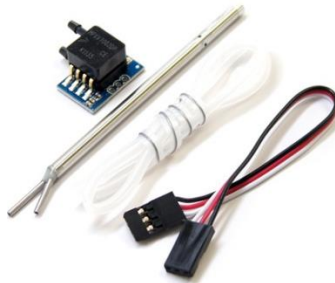

Figure 32- Unassembled Pitot tube and Pressure Sensor

6.2.7 Spektrum DX7s RC Transmitter

The DX7s is a 7-channel, 2.4 GHz remote control transmitter and receiver used for manual control of the aircraft and control of aileron failures in the system. Five channels were used for the control of ailerons, rudder, throttle and elevator and a sixth channel commanded the aileron to lock at a given position. Figure 33- shows an image of the RC equipment.

Figure 33- DX7s Transmitter and Receiver

6.2.8 Thunder Power TP3300-5SPL25 5 Cell LiPo Battery

The power source chosen for the system was a G6 Pro Lite Thunder Power 5 cell LiPo 18.5V battery. Figure 34- shows an image of the battery.

Figure 34- Thunder Power LiPo Battery

6.2.9 Turnigy D3542/6 Brushless Motor

The electric motor chosen for the Skywalker was a 1000Kv RPM Turnigy brushless motor. Its maximum current is 38A and it has a maximum power of 665W. Its weight is approximately 130g. and it dimension are 35x42mm with a shaft diameter of 5mm. Figure 35 shows the motor used.

Figure 35- Turnigy Brushless Motor

6.3 Simulink Models

The onboard microcontroller has the ability to be targeted through Simulink and the APM 2.0 Block set for Simulink. This feature provides a great advantage for any effort involving low cost autopilots and sensor fusions boards. Several Simulink models including model reference control, artificial neural networks and Kalman filter models, were designed in Simulink and later loaded into the APM 2.5 board for flight test implementation. Figure 36 shows the top level of the Simulink model loaded into the APM 2.5 board.
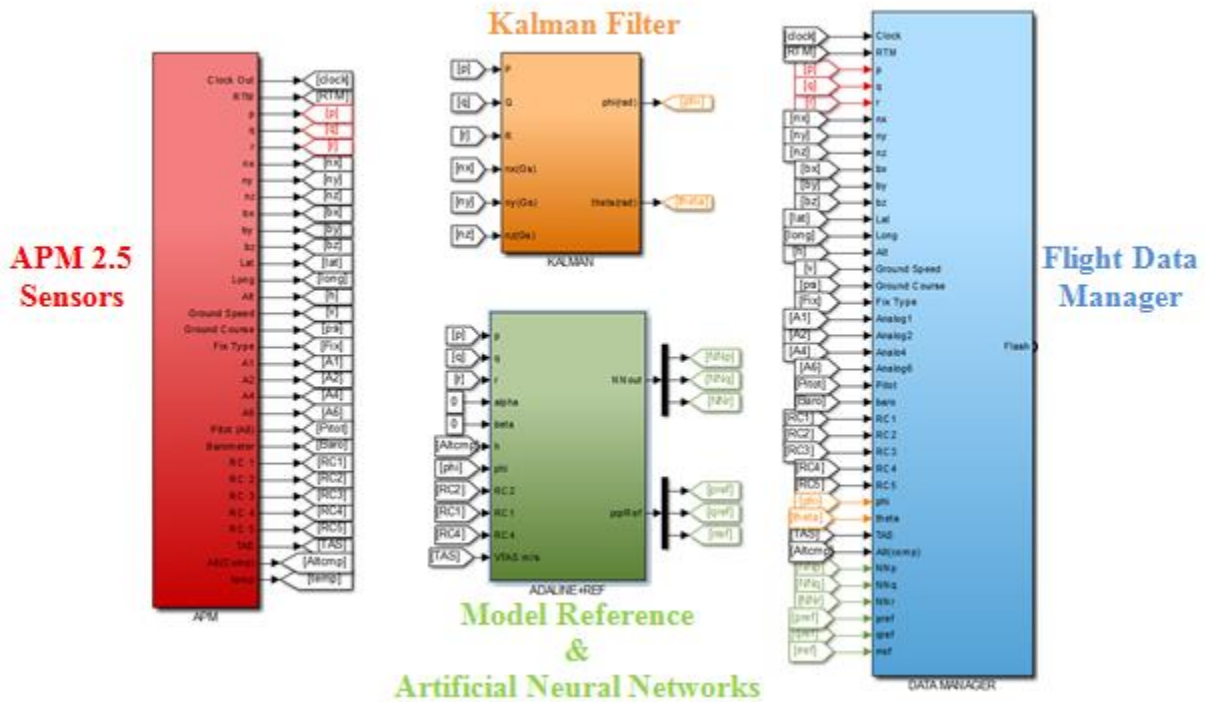

Figure 36- Simulink Model Top Level

In general, the Simulink model allows the APM board to read several sensors embedded in the board while recording flight test data in real time. Once the data is recorded in the flash memory, it can be downloaded and processed off-line for the generation of projections. Most of

the features are obtained from the sensors embedded in the onboard computer. On the other hand, the bank and pitch angles as well as the ANN and model reference control outputs are generated by separate Simulink models. The following sections will briefly describe the Simulink models for the mentioned systems.

6.3.1 APM 2.5 Sensors

It is possible to auto-generate code through the Run-On-Target-Hardware tool in the Simulink environment into the APM 2.5 board. The APM 2.0 Simulink blockset allows users to read data from the sensors embedded in the board, command PWM signals to servos and to run guidance, navigation and control algorithms. Figure 37 shows the library mentioned before.
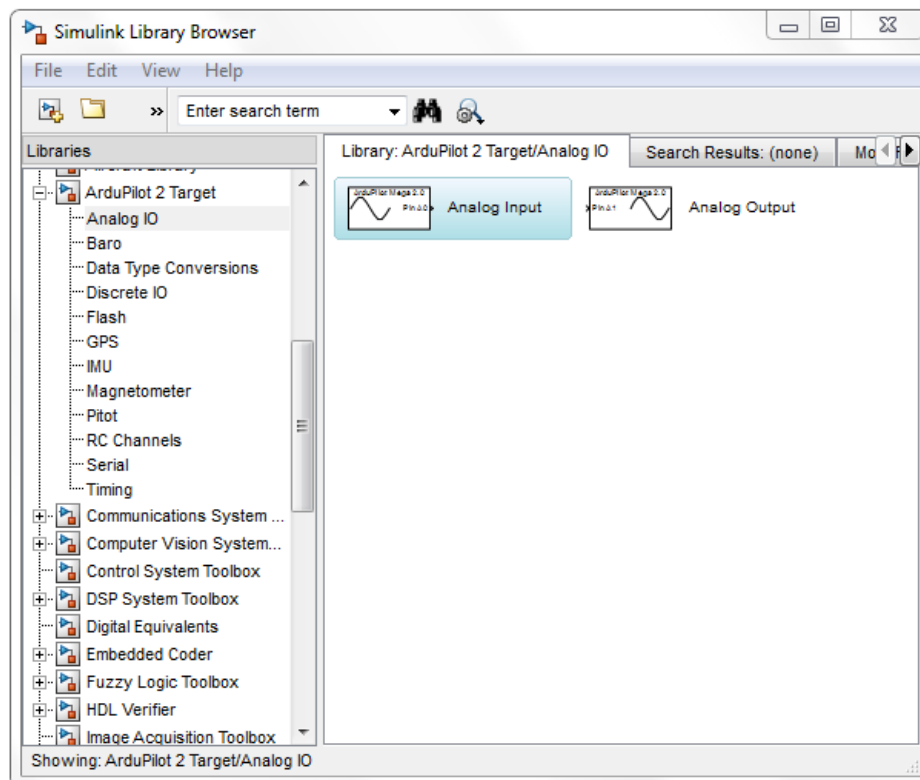

Figure 37- APM 2.0 Blockset

This library allows users to read the sensors embedded in the microcontroller. Therefore, a Simulink model that reads the data from such sensors was generated and loaded into the onboard computer. Figure 38 shows a sample model of the sensors.
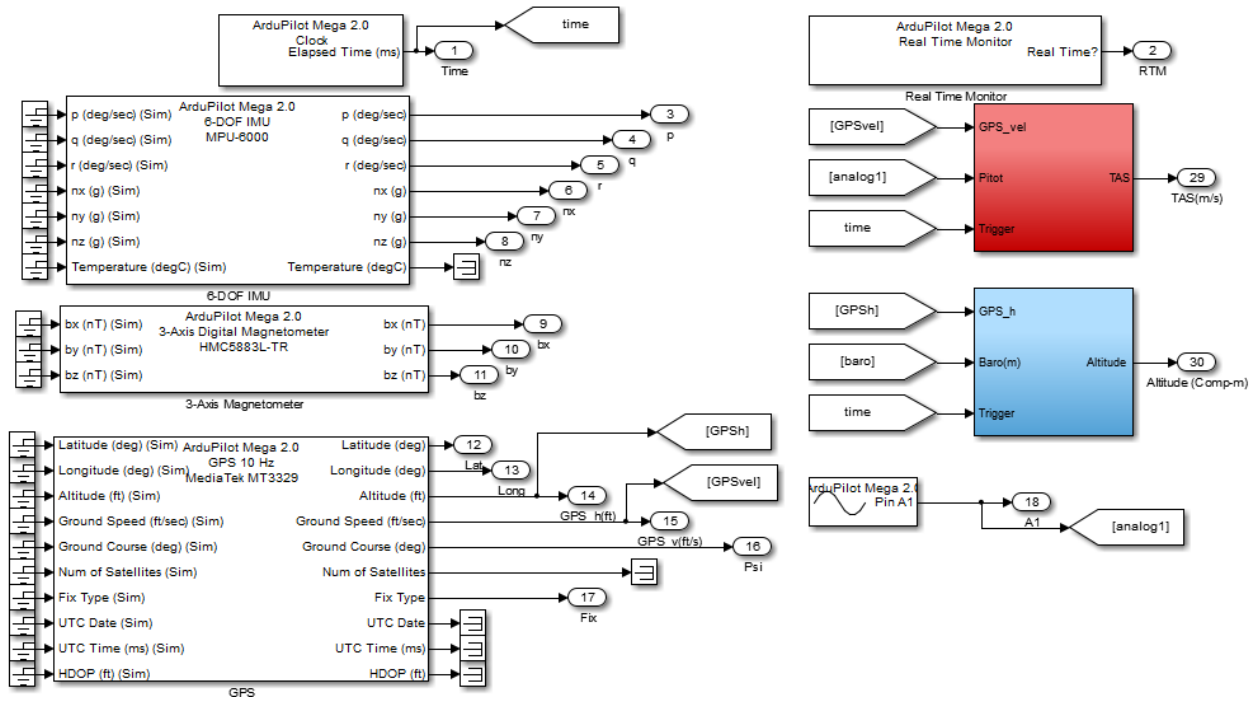

Figure 38-Simulink Sensor Model

6.3.2 Kalman Filter

In previous efforts, the Skywalker 1880 platform was used as an autonomous platform. For this effort, a sensor fusion solution was developed. A discrete Kalman filter was designed in Simulink in order to determine bank and pitch angles of the platform. The Kalman filter received inputs from a complementary filter that integrated data from the gyroscopes and the accelerometer. The Kalman filter demonstrated excellent performance and accuracy in the estimation of the desired Euler angles. Based on these characteristics, the estimator was included in the Simulink model with the objective to generate important variables for the definition of

99

Self/Non-self projections. Figure 39 shows a Simulink model with the basic architecture of the Kalman filter implementation.
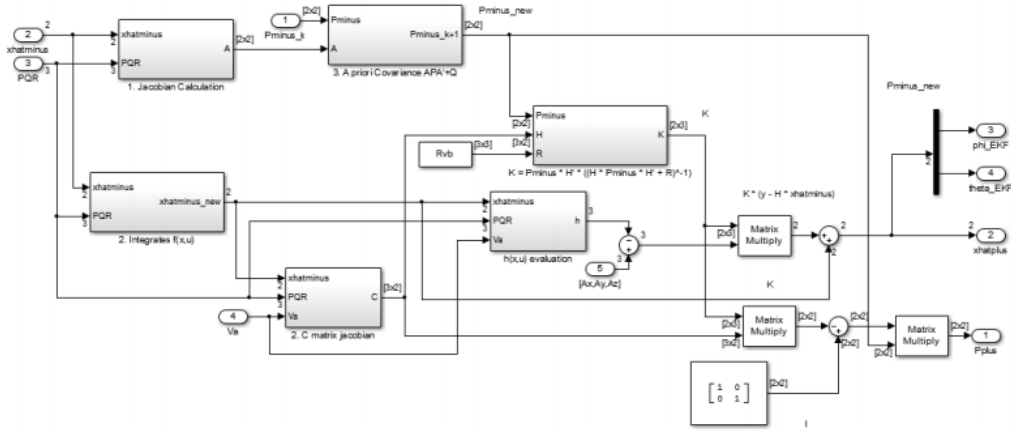

Figure 39- Kalman Filter Architecture

6.3.3 Model Reference and ANN

As mentioned in Chapter 5, a model reference controller and ANNs were incorporated into the system in order to generate features for the definition of projections. The Simulink model presents a relatively complex architecture that will not be shown in detail for simplicity. Regardless, Figure 40 presents the top level architecture of the model reference-artificial neural network model (MR+ANN).
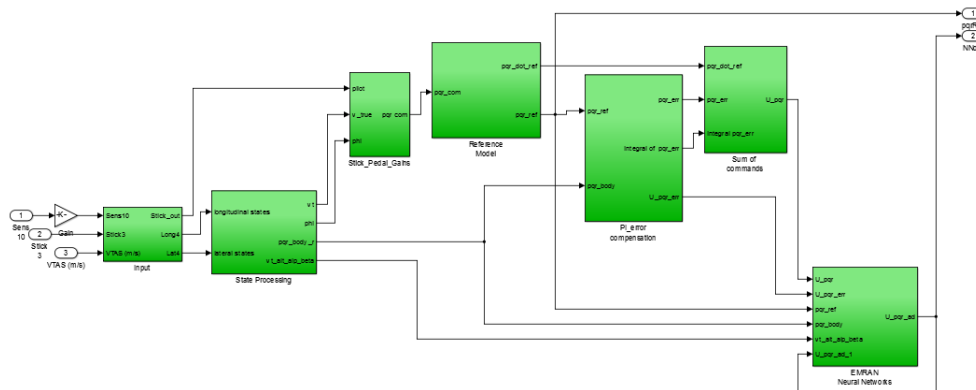

Figure 40- MR+ANN Top Level Architecture

The ANN used was an ADALINE network. The design requires that eight different variables are input to the network. Figure 37 shows the Simulink model of a pitch channel neural network estimator.
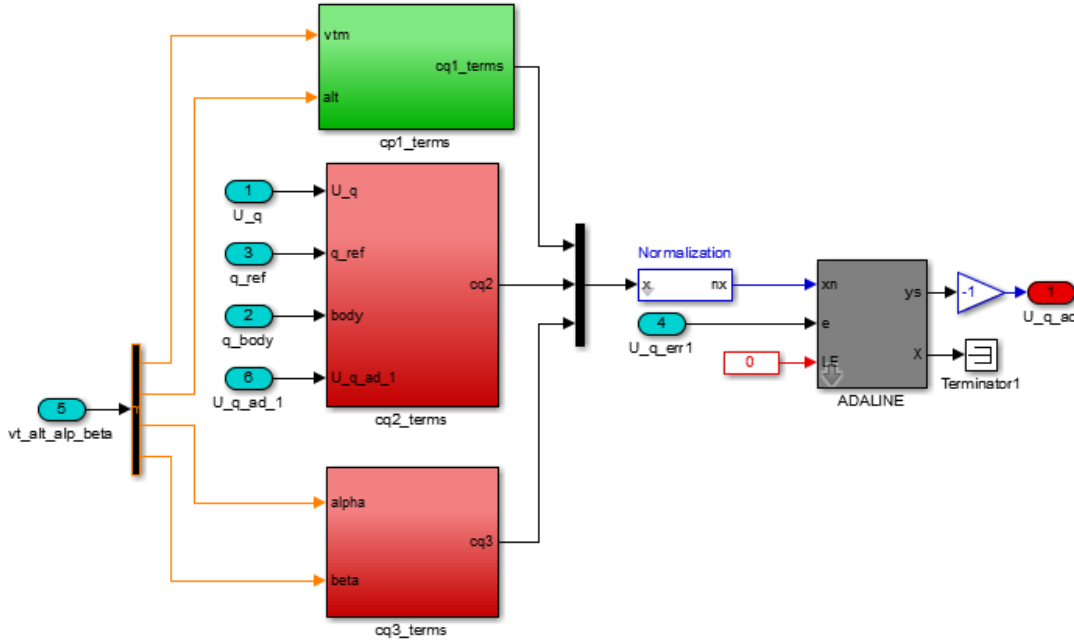
Figure 41- ADALINE Simulink Model

6.4 Flight Test Results

As mentioned before, two nominal flights and four failure flights were performed. A total of 38 different channels were recorded into the flash memory. Out of those, only 21 were selected for the definition of projections. Considering that only 2-dimensional projections were generated, a total of 210 projections are possible. Table 21 presents the features that were recorded and selected for projection definition.

Table 21- Flight Test Features

p	NN_q	ψ
q	NN_r	V
r	a_x	H
p_{ref}	a_y	d_e
q_{ref}	a_z	d_a
r_{ref}	φ	d_r
NN_p	θ	d_T

Once the 210 projection were defined, it was decided that projections that contained variables related to the inertial reference frame (i.e. heading, altitude ground speed) were not necessary for detection of abnormal conditions. Therefore, 153 projections were tested against flight test data in order to determine the detection performance of the generated selves. The selves analyzed the data from four different failure flights. The analysis determined that 30 different self projections were able to capture the dynamic fingerprint of abnormal conditions. Table 22 presents these projections.

Table 22- Skywalker Self Projections

Self#	Features	Self#	Features
1	p_{ref}, q_{ref}	16	p_{ref}, θ
2	p_{ref}, r_{ref}	17	p_{ref}, φ
3	p_{ref}, p	18	r_{ref}, a_x
4	p_{ref}, q	19	p, d_a
5	p_{ref}, r	20	q, d_a
6	p_{ref}, NN_p	21	r, a_x
7	p_{ref}, NN_q	22	NN_p, d_a
8	p_{ref}, NN_r	23	NN_q, d_a
9	p_{ref}, a_x	24	a_x, φ
10	p_{ref}, a_y	25	a_x, θ
11	p_{ref}, a_z	26	a_x, d_a
12	p_{ref}, d_a	27	a_x, d_T
13	p_{ref}, d_e	28	θ, d_a
14	p_{ref}, d_T	29	φ, d_a
15	p_{ref}, d_r	30	d_a, d_e

The selected projections obtained an average 21.11% and 30.68% DR for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver respectively. These detection rate perceived seems to be low. However, due the nature of the flight tests and the process of the detection analysis it is expected to obtain such results for individual selves. For example, for a wing level aileron locked a trim, the pilot injects the failure into the system manually when the aircraft is flying hands free. At the moment of injection it is considered that a failure is present in the system. Nevertheless, the system will not perceive a change in the dynamics until the pilot attempt a new maneuver. Table 23 presents the DR and FA rate of a sample set of projections.

Table 23- Detection Rate and False Alarm Rate for 4 Failures

Self#	Right Aileron Locked at Wings Level		Right Aileron Locked at Bank Maneuver		Left Aileron Locked at Wings Level		Left Aileron Locked at Bank Maneuver	
	FA	DR	FA	DR	FA	DR	FA	DR
1	0.00	25.59	0.00	28.67	0.00	12.40	0.00	23.75
12	0.00	25.08	0.00	34.77	0.00	19.15	0.00	34.11
13	0.00	32.46	0.00	36.36	0.00	31.55	0.00	39.25
15	0.00	24.56	0.00	44.10	0.00	14.75	0.00	25.46
22	0.00	10.15	0.00	15.59	0.00	11.28	0.00	30.52
30	0.00	20.56	0.00	23.33	0.00	26.20	0.00	32.23
Average	0.00	23.07	0.00	30.47	0.00	19.22	0.00	30.89

As shown before the individual selves present a low DR but if they are integrated into a single detection mechanism the DR improves greatly. For example, if the 30 selves selected are used for failure detection, then the DR percentage improves to 71.86% and 90.68% for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver, respectively. This configuration is able to obtain a significant improvement in DR but it also increases the FA rates approximately to 5.2%. For this reason, other configurations were tested in order to obtain an acceptable tradeoff between DR and FA.

Two more configurations of selves were tested. The second configuration utilizes only the 6 selves shown in Table 23. This configuration obtained an average of 43.26% and 58.27% DR for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver respectively, with FA rates of approximately 1% for both cases. The third configuration of selves utilizes only projections that have 0% FA rate for each failure disregarding FA for other types of failures. This configuration guarantees that the FA rate will be low and it also offer the possibility to use this method for identification purposes. The third configuration presents an average of 59.57% and 77.25% DR for ailerons stuck at wings level trim and ailerons stuck during a bank maneuver, respectively and less than 0.5% FA. Table 24 presents a summary of these results.

Table 24- Detection Rates and False Alarms for 3 Configurations of Selves

Method	R. Aileron Locked at Wings Level		R. Aileron Locked at Bank Maneuver		Left Aileron Locked at Wings Level		Left Aileron Locked at Bank Maneuver	
	FA	DR	FA	DR	FA	DR	FA	DR
1	5.2	74.34	1.54	90.01	4.3	69.39	2.33	91.36
2	0.52	47.17	0.12	60.23	1.0	39.35	0.02	56.32
3	0.3	61.18	0.0	80.94	0.2	57.97	0.0	73.56

It should be noted that the four failures investigated in flight test are considered to be low magnitude. Also, the duration of all flights with an injected failure lasted approximately 20 seconds. These two characteristics yield small sets of data in which great portions of data fall inside the self which subsequently led to the low detection rates presented in Table 24. More importantly, the detection rate calculation is considered to be very conservative for the nature of the test flights under failure. Each failure flight test is considered to start as soon as the pilot injects the failure and end when the switch is moved back to nominal position. However, the dynamics of the aircraft will not change until the pilot commands maneuvers that require the

actuation of the damaged control surface. In other words, a failure is not visible to the mechanism until the flight attitude is not at trim condition.

Table 23 shows that single projections can only obtain a maximum of 30.89% detection rate. Nevertheless, when the projections are integrated as a single mechanism the DR increases greatly (see Table 24). This occurs because each projection at certain time periods of the flight test only captures abnormal dynamics when excited by certain maneuvers and commands. On the other hand, integrating several projections allows the mechanism to capture abnormal dynamic fingerprints at different periods of time during the flight tests. Figures 42-44 show the detection activity of single projections and Figure 45-46 shows the detection activity when the projections are integrated into a single mechanism. The value of 1 represents that a detector was been activated and alternatively a value of 0 determines that none of the detectors have been activated. The first 5 seconds are nominal flight test conditions while the remainder of the time corresponds to flight test data under an upset condition.
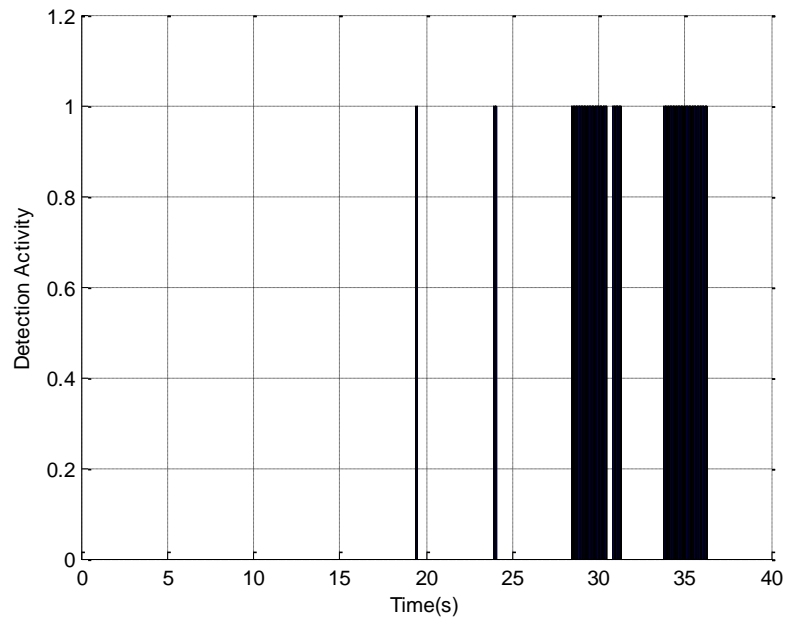

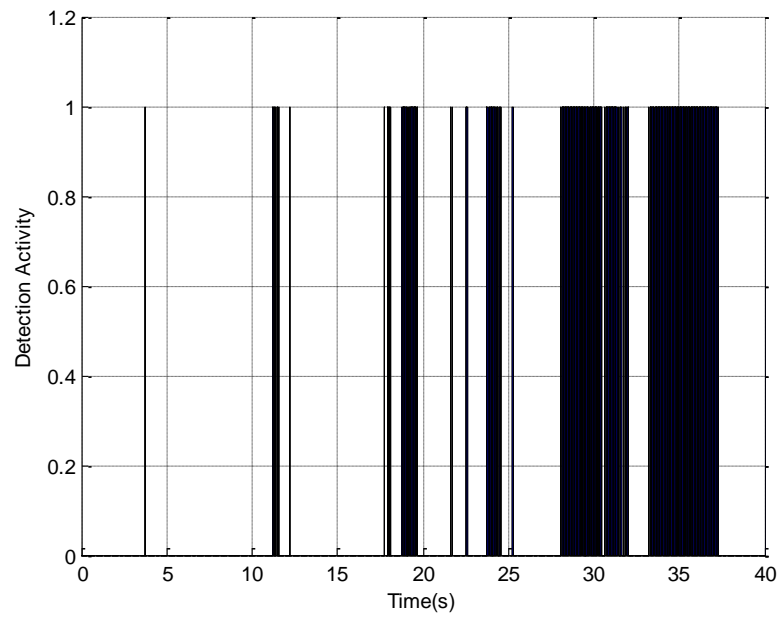**Figure 42- Self #1 Detector Activity for Left Aileron Locked at Bank Maneuver**

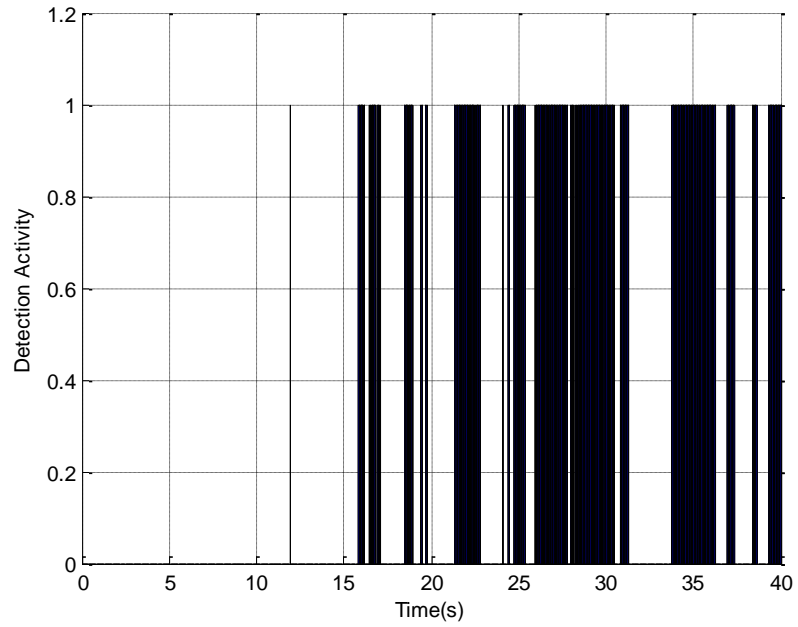**Figure 43- Self #17 Detector Activity for Left Aileron Locked at Bank Maneuver**

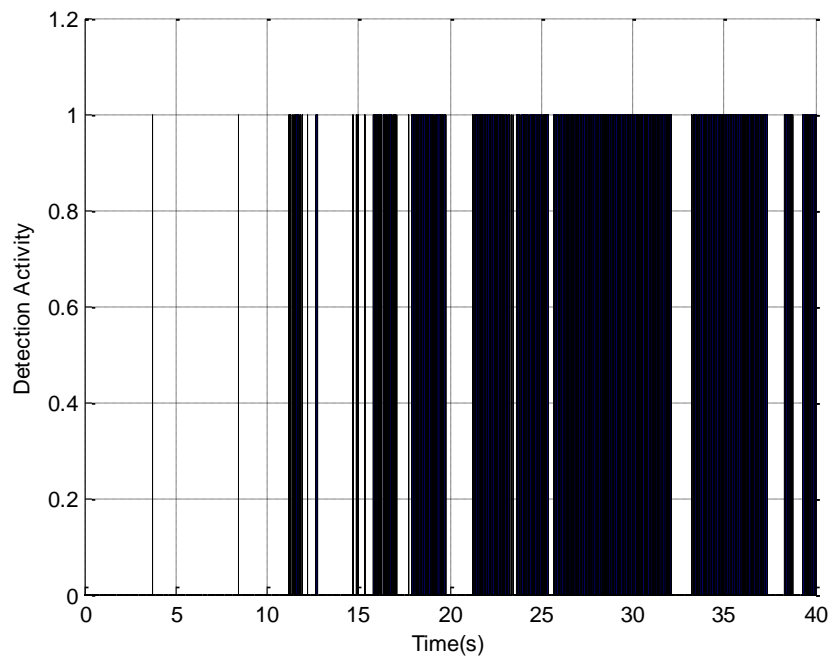**Figure 44- Self #18 Detector Activity for Left Aileron Locked at Bank Maneuver**

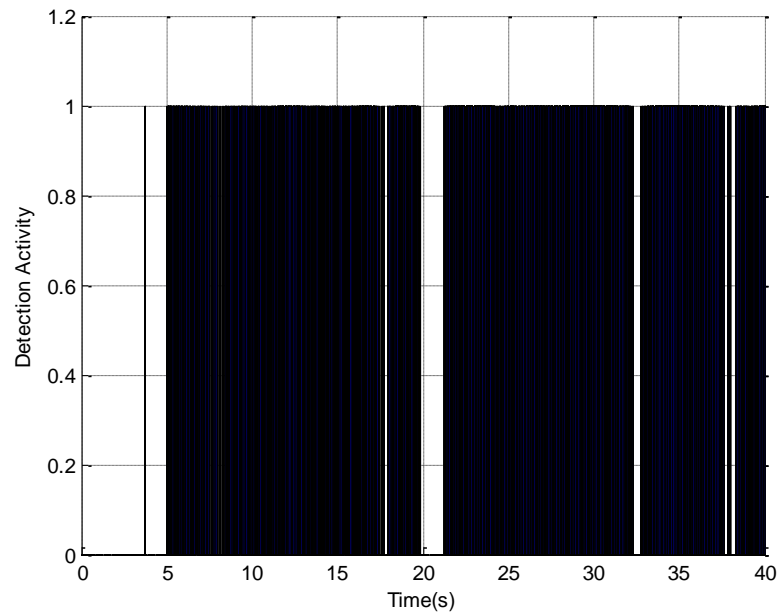**Figure 45- Method 2 Detector Activity for Left Aileron Locked at Bank Maneuver**

Figure 46- Method 1 Detector Activity for Left Aileron Locked at Bank Maneuver

Chapter 7 Conclusions

A set of comprehensive algorithms inspired by the artificial immune system paradigm capable of performing detection, identification and evaluation of aircraft subsystems failures has been developed and tested. The performance of the mechanism has been assessed through simulation and flight tests showing excellent performance in all phases of the process. The aircraft health management framework presented provides the premises for an integral and comprehensive solution to the problem of aircraft subsystems FDIE. The simulation tests were implemented for a fighter aircraft while the flight test experimentation was performed on a RC aircraft.

The architecture of the mechanism presented is inspired in the artificial immune system paradigm within a hierarchical multi-self strategy. Vast amounts of flight test data were obtained from a 6DOF flight simulator located in WVU. A total of 496 2-dimensional Self/Non-Self projections were generated for FDIE purposes. The Cluster Set Union Method for antibodies/detectors generation is presented in this thesis. Furthermore, the failure detection capabilities of each projection were tested within the Matlab/Simulink Environment. A total of 183 projections exhibited excellent results with detection rates reaching up to 99% while false alarm rate were kept down to 0%.

A Structured Non-Self Approach for the identification and direct evaluation methodology is presented. This methodology is based on an arrangement of non-self projections and intends to reduce computational efforts for real-time application. It offers a simple yet reliable algorithm capable of correctly identifying specific failure types, failed subsystems as well as the severity of the failure affecting the system.

Furthermore, an indirect evaluation algorithm for the estimation of reduced flight envelope variables at post-failure conditions is presented. This analytical process takes advantage of the graphical

representation of the dynamics of the aircraft to determine the reduction in performance, maneuverability and other limits of investigated aircraft parameters.

The mechanism presented in this thesis is also implemented in flight test. A Skywalker RC aircraft was instrumented with an APM 2.5 micro controller that has embedded digital sensors that provided essential variables for the definition of self/non-self projections. The onboard computer was loaded with a Kalman filter, a model reference controller and an ANN Simulink model. Four different failures were injected during flight. The detection process was performed offline and presented acceptable detection rates and false alarms rates.

7.1 Future Work

Future work regarding this topic includes:

- Flight test implementation with other aircraft systems

- Injection of more severe failures during flight tests

- Use of more powerful computers for online detection and identification experimentation

- Use this approach for aircraft crash investigation in addition to classical methods

- Utilize an "Extended Minimum Resource Allocation Network" in parallel with the ADALINE network

References

[1]   International Civil Aviation Organization, "2011 State of Global Aviation Safety", ICAO, Montréal, Canada, 2011

[2]   Belcastro C. M., Foster J. V., "Aircraft Loss-of-Control Accident Analysis", Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, CA, August 2010.

[3]   Belcastro C. M., Jacobson S. R., "Future Integrated Systems Concept for Preventing Aircraft Loss-of–Control Accidents", Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, Aug. 2010.

[4]   Dasgupta D. (Editor), Artificial Immune Systems and Their Applications, Springer Verlag, New York, USA, 1999.

[5]   Dasgupta D., KrishnaKumar K., Wong D., Berry M., "Negative Selection Algorithm for Aircraft Fault Detection", in Proceedings of ICARIS 2004, edited by G. Nicosia et al., LNCS 3239, pp.1–13, 2004.

[6]   Dasgupta D., Nino L. F. Immunological Computation – Theory and Applications, CRC Press, Auerbach Publications, Taylor & Francis Group, Boca Raton FL, USA, 2009.

[7]   Moncayo H., Perhinschi M. G., Davis J., "Artificial Immune System – Based Aircraft Failure Detection and Identification Over an Extended Flight Envelope", The Aeronautical Journal, Vol. 115, No. 1163, 2011.

[8]   Moncayo H., Perhinschi M. G., Davis J., "Artificial Immune System – Based Aircraft Failure Evaluation Over Extended Flight Envelope", AIAA Journal of Guidance, Control, and Dynamics, Vol. 34, No. 4, pp.989-1001, 2011.

[9]   Moncayo H., Perhinschi M. G., Davis J., "Aircraft Failure Detection and Identification Using an Immunological Hierarchical Multi-Self Strategy", AIAA Journal of Guidance, Control, and Dynamics, Vol. 33, No. 4, pp.1105-1114, 2010.

[10] Perhinschi M. G., Moncayo H., Al Azzawi D., Moguel I., "Generation of Artificial Immune System Antibodies Using Raw Data and Cluster Set Union", International Journal of Immune Computation, March 2014

[11] Al-Enezi J.R., Abbod M.F., Alsharhan, S., "Advancement in Artificial Immune Systems: A Perspective of Models, Algorithms and Applications", GCC Conference & Exhibition, 2009 5th IEEE. IEEE, 2009.

[12] Duch W., "What is Computational Intelligence and Where is it going?" Challenges for computational intelligence. Springer Berlin Heidelberg, 2007. 1-13

[13] Hofmeyr S. A., Somayaji A., Forrest S., "Intrusion detection using sequences of system calls", J. of Computer Security, 6, pp.151–180, 1998.

[14] Twycross J., Aickelin U., "Libtissue-Implementing Innate Immunity", in the Proceedings of IEEE World Congress on Computational Intelligence, Vancouver, Canada, 2006.

[15] Timmis J., Knight T., "Artificial immune systems: Using the immune system as inspiration for Data Mining", in Abbass H. A., Sarker R. A., Newton C. S., (Eds.), "Data Mining: A Heuristic Approach", Chapter XI, Group Idea Publishing, pp. 209–230, 2001.

[16] Serapiao A. B. S., Ricardo J., Mendes P., Miura K., "Artificial immune systems for classification of petroleum well drilling operations", in Proceedings of the 6th International Conference on Artificial Immune Systems (ICARIS), Brazil, 2007.

[17] Karr C., Nishita K., Graham, K., "Adaptive Aircraft Flight Control Simulation Based on an Artificial Immune System", *Applied Intelligence,* Vol. 23, No. 3, pp.295–308, 2005.

[18]   Takahashi K., Yamada T., "Application of an Immune Feedback Mechanism to Control Systems", The Japan Society of Mechanical Engineers, JSME International Journal, Series C, Vol. 41, No. 2, 1998.

[19]   Gonzalez F., Dasgupta D., Kozma R, "Combining negative selection and classification techniques for anomaly detection", in Proceedings of the 2002 Congress on Evolutionary Computation CEC2002, IEEE Press, Honolulu, HI, pp.705–710, 2002.

[20]   Guzella T. S., Mota-Santos T. A., Caminhas W. M., "A novel immune inspired approach to fault detection", in Proceedings of ICARIS 2007, Lecture Notes in Computer Science, Springer, Santos, 2007.

[21]   Dasgupta D., "Advances in Artificial Immune Systems" in IEEE Computational Intelligence Magazine, November 2006

[22]   Greensmith J., Aickelin U., Cayzer S., "Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection". Artificial Immune Systems. Springer Berlin Heidelberg, 2005. 153-167.

[23]   Dasgupta D., Nino, F., "Comparison of Negative and Positive Selection Algorithms in Novel Pattern Detection", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp.125-130, 2000.

[24]   National Transportation Safety Board Report AAR-79-17, Washington DC, 1979.

[25]   Job, Macarthur, Air Disaster, Volume 2, Aerospace Publications, 1996, pp.136-153.

[26]   Transportation Safety Board of Canada Report A05F0047, Gatineau, Quebec, Canada, 2005

[27]   Perhinschi M. G., Moncayo H., Davis J., "Integrated Framework for Artificial Immunity-Based Aircraft Failure Detection, Identification, and Evaluation", AIAA Journal of Aircraft, Vol. 47, No. 6, pp.1847-1859, 2010.

[28]   Perhinschi M. G., Moncayo H., Al Azzawi D., "Development of Immunity-Based Framework for Aircraft Abnormal Conditions Detection, Identification, Evaluation, and Accommodation", in Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, 2013.

[29]   Moncayo H., Perhinschi M. G., Davis J., "Simulation Environment for the Development and Testing of Immunity-Based Aircraft Failure Detection Schemes", in the Proceedings of the AIAA Modeling and Simulation Technologies Conference, Portland, OR, 2011

[30]   Perhinschi M. G., Moncayo H., Wilburn B., Bartlett A., Davis J., Karas O., "Testing of Immunity-Based Failure Detection Scheme with the NASA Generic Transport Model", in the Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, 2011.

[31]   Perhinschi M. G., Porter J., Moncayo H., Davis J., Wayne W. S., "Artificial Immune System-Based Detection Scheme for Aircraft Engine Failures," AIAA Journal of Guidance, Control, and Dynamics, Vol. 34, No. 5, pp.1423-1440, 2011.

[32]   Zurada J. "Applications of Neural Networks for Aerospace-Related Technologies", Aerospace Application Conference, 1996. Proceedings, 1996 IEEE. Vol. 2. IEEE, 1996.

[33]   Kanesige J., Bull J., Totah J., "Generic Neural Flight Control and Autopilot System" AIAA Guidance, Navigation and Control Conference and Exhibit Vol. 4281.

[34]   Rysdyk R., Calise A., "Fault Tolerant Flight Control via Adaptive Neural Network Augmentation", AIAA Guidance, Navigation and Control Conference and Exhibit, 1998.

111

[35] Burken J., Williams-Hayes P., Kaneshige J., Stachowiak S., " Adaptive Control Using Neural Network Augmentation for a Modified F-15 Aircraft. Control and Automation, 2006, MED'06. 14th Mediterranean Conference on. IEEE,2006.

[36] Mackall D., Nelson S., Scumman J., "Verification & Validation of Neural Networks for Aerospace Systems". National Aeronautics and Space Administration, Ames Research Center, 2002.

[37] Moncayo H., Perhinschi M. G., Wilburn B., Wilburn J., Karas O., "UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity-based Mechanism", in Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, 2012.

[38] Moguel, I., Moncayo, H., Perhinschi M. G., Perez, Andres., Al Azzawi, D., Togayev, A "Structured Non-Self Approach for Aircraft Failure Identification within an Immunity-based Fault Tolerance Architecture", Under review by the IEEE Journal of Aerospace and Electronic Systems.