2009

Development and evaluation of a fault detection and identification scheme for the WVU YF-22 UAV using the artificial immune system approach

Sebastian Pablo Sanchez
West Virginia University

Development and Evaluation of a Fault Detection and Identification Scheme for the WVU YF-22 UAV Using the Artificial Immune System Approach

Sebastian Pablo Sanchez

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Aerospace Engineering

Department of Mechanical and Aerospace Engineering
Morgantown, West Virginia
2009

Keywords: Failure Detection and Identification, Artificial Immune System, UAV flight testing

Abstract

Development and Evaluation of a Fault Detection and Identification Scheme for the WVU YF-22 UAV Using the Artificial Immune System Approach

Sebastian P. Sanchez

A failure detection and identification (FDI) scheme is developed for a small remotely controlled jet aircraft based on the Artificial Immune System (AIS) paradigm. Pilot-in-the-loop flight data are used to develop and test a scheme capable of identifying known and unknown aircraft actuator and sensor failures. Negative selection is used as the main mechanism for self/non-self definition; however, an alternative approach using positive selection to enhance performance is also presented. Tested failures include aileron and stabilator locked at trim and angular rate sensor bias. Hyper-spheres are chosen to represent detectors. Different definitions of distance for the matching rules are applied and their effect on the behavior of hyper-bodies is discussed. All the steps involved in the creation of the scheme are presented including design selections embedded in the different algorithms applied to generate the detectors set. The evaluation of the scheme is performed in terms of detection rate, false alarms, and detection time for normal conditions and upset conditions. The proposed detection scheme achieves good detection performance for all flight conditions considered. This approach proves promising potential to cope with the multidimensional characteristics of integrated/comprehensive detection for aircraft sub-system failures.

A preliminary performance comparison between an AIS based FDI scheme and a Neural Network and Floating Threshold based one is presented including groundwork on assessing possible improvements on pilot situational awareness aided by FDI schemes. Initial results favor the AIS approach to FDI due to its rather undemanding adaptation capabilities to new environments. The presence of the FDI scheme suggests benefits for the interaction between the pilot and the upset conditions by improving the accuracy of the identification of each particular failure and decreasing the detection delays.

DEDICATION

To my family for being always supportive in every decision I made and for helping me become who I am today.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Prof. Perhinschi and Prof. Napolitano for giving me the opportunity to further extend my college education. Specials thanks go to Prof. Perhinschi for letting me take part of this groundbreaking research giving me the best guidance one can expect from an Advisor. I would also like to thank Dr. Bojan Cukic for eagerly agreeing to form part of my committee and for providing positive feedback towards the successful completion of this work.

My parents, Olga and Vicente, deserve a special thank for always letting me choose my own path and giving me an education one cannot learn from books; their generosity is out of imaginable limits. I would also like to express my sincere appreciation to my siblings Javier and Mariana for always believing in me even more than myself.

I would also like to extend my gratitude to my friends, now spread throughout the world, for being the siblings one gets to choose to walk side by side through life, I am sure I would not have made it this far without all of you. A special thank goes to Juan, Santi, and Natalia for creating a real home away from home for me as soon as I set foot in Morgantown. One gigantic thanks goes to my girlfriend Sophie for keeping me grounded and bearing my constant trivia for the past year.

Being part of this research group allowed me to get to know a great group of people. I want to acknowledge Sergio, Kerri, Jason, and Marco for making my office such an interesting and enjoyable place to work in. I would also like to extend my gratefulness to the senior members of the group and, in particular to Srik, for giving me all necessary support whenever I found myself in situations that exceeded my knowledge.

I also want to acknowledge Hever for putting together the first versions of the Matlab codes to implement the various algorithms necessary for the AIS implementation that gave me an invaluable head start to begin working on my own codes. It was a pleasure to work side by side with such a kind and smart person.

Last but not least the financial support provided by NASA through a grant within the EPSCOR program coordinated by the NASA West Virginia Space Grant Consortium is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Acronyms

AIAA	American Institute of Aeronautics and Astronautics
AIS	Artificial Immune System
ARMA	Auto-Regressive Mean Average
ATM	Automated Teller Machine
BPA	Back Propagation Algorithm
CARAT	Center for Advanced Research in Autonomous Technologies
CI	Computational Intelligence
CS	Control Switch
DC	Direct Current
DNN	Decentralized Neural Network
DR	Detection Rate
ECU	Electronic Control Unit
EMI	Electro-Magnetic Interface
ENSA-RV	Enhanced Negative Selection Algorithm for Real Valued representation
EPSCoR	Experimental Program to Stimulate Competitive Research
FA	False Alarms
FAA	Federal Aviation Administration
FDI	Failure Detection and Identification
FDII	Failure Detection, Isolation and Identification
FI	False Identification
FL	Floating Limiter
FN	False Negative
FP	False Positive
FuA	Full Autonomous mode
GA	Genetic Algorithm
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
GUI	Graphic User Interface
ICARIS	International Conference on Artificial Immune Systems
IFC	Intelligent Flight Control
IFCS	Intelligent Flight Control System
IMU	Inertia Measurement Unit

IR	Identification Rate
LNCS	Lecture Notes in Computer Science
M	Manual mode
MBSF	Motion Base S-Function
MNN	Main Neural Network
MQEE	Mean Quadratic Estimation Error
NASA	National Aeronautics and Space Administration
NIS	Natural Immune System
NN	Neural Network
NS	Negative Selection
OBC	On Board Computer
OBES	On Board Excitation System
OQEE	Output Quadratic Estimation Error
PA	Partial Autonomous mode
PCM	Pulse Code Modulation
PID	Parameter Identification
PS	Positive Selection
PSE	Positive Selection Enhancement
RC	Research Computer
SC	Server Computer
TN	True Negative
TP	True Positive
UAV	Unmanned Air Vehicle
UDP	User Datagram Protocol
V&V	Verification and validation
VSF	Visuals S-Function
WV	West Virginia
WVU	West Virginia University

List of Symbols

$\vec{c}$	Center of hyper-body – vector
$\vec{D}$	Vector defining hyper-body characteristics
D_{thre}	Detection threshold
$d\left(\vec{x},\vec{y}\right)$	Distance function evaluated between the points $\vec{x}$ and $\vec{y}$
M_j	Maximum value for dimension j of the nominal data
m_j	Minimum value for dimension j of the nominal data
N	Non-self Set
N_{RD}	Set of random centers for detectors
P	Sample size for Monte Carlo computations
p, q, r	Pitch, roll, and yaw angular rates (respectively)
r	Radius of hyper-sphere
S	Self Set
T_W	Time window
U	Universe Set
$\overline{x}_{ij}$	Raw data point
$\tilde{x}_{ij}$	Intermediate step for normalization
x_{ij}	Normalized data point
$\vec{x}$	Vector

Greek:

λ	Minkowski's distance defining variable
η	Confidence level ($1-\eta$)
ε	Integration error
δ	Deflection of a control surface

Subscripts:

AR, AL	Right Aileron, Left Aileron
F	Flight
I	Identification
i	Index for number of points defining the self – from 1 to m
j	Index for dimension – from 1 to n

k	Time step
$\min$	Minimum
PP	Point to Point
RD	Random variable
RR, RL	Right Rudder, Left Rudder
ref	Output of the reference model
S, A, R	Stabilator, Aileron, Rudder
SR, SL	Right Stabilator, Left Stabilator

Chapter 1: INTRODUCTION

Flight safety has concerned mankind since the very first day a man-designed machine lifted off by its own means. The first formal attempts to make flying machines safer started in the mid 1920s when USA started regulating civil aviation via the Air Commerce Act of May 20, 1926. This effort was a first step in setting up standards that would diminish the chances of fatal accidents caused by unsafe designs and/or operations of air vehicles. This would lead to what is now known as the Federal Aviation Administration (FAA) created in 1958 and to many other organizations working towards safer skies [1].

As expected, evolution of flying machines motivated an increase in the number and complexity of the constituent sub-systems. Even complying with all the present regulations, these subsystems are to certain extent susceptible to failures. Depending on how critical a particular subsystem is for flight safety, a single subsystem failure may result in very dangerous situations, frequently with disastrous outcomes. According to [2], more than 25% of fatal accidents from 1950 to the present day involved some type of mechanical failure. Several of these cases were attributed to failures causing surfaces to jam or move to extreme positions followed sometimes by worsening actions performed by the uninformed flight crew. Most of these incidents ended up in the worst way causing catastrophes involving not only people inside the plane but, occasionally, also people on ground.

In some cases pilot skills, helped by particularities of the malfunction, allowed the flight to continue and land safely as was the case of an EMB-145, which in April 2001 experienced a jammed horizontal stabilizer in trim position during cruise flight. Fortunately, the pilot was able to perform a landing without incidents. Other flights, however, ended up the worst way like United Airlines 585 (March 1991) and USAir 427 (September 1994) both involving Boeings 737 and rudder failures that caused the control surface to deflect in opposite direction to pilot inputs [3]. Could the pilots have recovered the aircraft if an onboard Failure Detection and Identification (FDI) system would have warned them of the nature of the failure? Could information from the first flight have been used to train the FDI scheme to improve detection or even accommodation in the event of such a failure? These and many other questions remain unanswered but the usefulness of comprehensive FDI systems as an aid to flight safety can hardly be argued.

1.1. Literature Review

Failures, faults, upset conditions, abnormal conditions among others, are terms used in literature to refer to similar or equivalent situations in which some system experiences behavior that is not expected and/or desired. In this context, any real world system is susceptible of some type of failure and thus, preventing, detecting, correcting, and/or adapting to such situations has been the focus of many research efforts in the most varied fields. Nevertheless, areas of technology such as aerospace, where the occurrence of these undesirable situations represents a substantial increase in risks or costs, pioneered the research of tools to cope with the inevitable likelihood of facing failures.

In the early days, physical redundancy of components was proposed to grant the failsafe concept by which if one component was to fail the redundant one/s would be equally capable of allowing the plane's safe operation. This approach can be acceptable up to some extent for some subsystems; however, it poses many drawbacks that have made it highly impractical to deal with the evolving number and variety of subsystems present in an airplane. The first important consequence of physical redundancy is an increase in airplane's weight which translates in a decrease of its payload, an increase in fuel consumption and so on. A second effect of physical redundancy is that it can hide the effects of some failures thus hindering the possibility of taking corrective action; this could lead to more catastrophic combinations of failures.

Increasing the safety of aircraft operation has become in recent years a major objective for the aerospace engineering community and particularly for NASA's Aviation Safety Program [4]. Considerable attention has been paid to failures for which the plane had the potential of continuing the flight safely if the correct actions would have been taken. Malfunctions that can be listed in this group are control surface failures and sensor failures. Research has pointed on technologies capable of providing at least the chance for a safe emergency landing even after some subsystems have stopped working.

Part of the efforts to achieve this goal has been aimed to develop control systems capable of accommodation/adaptation during abnormal conditions. NASA Ames Research Center started a program called Intelligent Flight Control (IFC) in 1992 to "*examine alternate sources of control power to accommodate in-flight control system failures*" [5; 6]. The IFC program focused on development and evaluation of control systems capable of taking advantage of the remaining healthy subsystems in the event of a failure to allow an increase in survivability rates for various failures. IFC research made wide use of a variety of Neural Network architectures to create control systems capable of overcoming mathematical errors in the modeling of the plant and at the same time robust enough to allow adaptation to new dynamic conditions of the plant produced by several factors [5].

Analytical techniques examined for IFC and in general for fault detection and/or accommodation can be divided in two large groups:

- Model based approach
- Knowledge based approach

Both approaches have proven successful depending on the particular application.

Model based techniques make use of an explicit mathematical model of the system. This model is used to obtain estimated behavior of the plant to the given inputs. The outputs of the model allow comparison with the actual outputs of the system and decisions can be taken upon evaluating the difference between the two (residual) [7]. Failures are said to have definite signatures that present as combinations of particular behavior of these residuals. Failure detection consists then in two stages, residuals generation and decision making. Different techniques can be used to generate these residuals from which the following is a partial list of the most common ones.

- Observer-based approaches rely on the use of Kalman Filters or other types of filters to generate such residuals. Chen and Saif presented in [8] an implementation of Thau's and sliding-mode observers for detecting actuator failures in the presence of superimposed nonlinearities. Wang et al. applied a similar approach in [9] for detecting (and accommodating) failures on a simulation of an aerospace vehicle. These classes of systems and particularly those relying in Kalman filters have been shown in [10] to experience problems when dealing with strong nonlinearities and uncertainties. The concept of multiple models, switching, and tuning appears as an extension to this approach and was successfully implemented to provide adaptive capabilities in the event of control surface blockage failures in [11] and also for sensor failures [12]; both references used airplane simulations to evaluate the strengths of the created schemes.
- Parity relations approaches are based on analytical redundancy and can take two forms, direct redundancy when dealing with instantaneous sensors outputs, or, temporal redundancy when involving relationships among time histories of sensors and actuators [13]. Residuals are obtained from these comparisons as measures of the discrepancy between the reigning conditions and the expected behavior for normal conditions. This approach served in [14] to detect failures in a nonlinear model of generic servoactuator

used for manipulating robots; this work showed how the use of nonlinear parity equations avoids modeling errors and the need for linearization of nonlinear models. Kabbaj et al. presented in [15] an application of this technique to detect failures in greenhouse processes for both sensors and actuators failures and described the used of fixed threshold as an issue of their implementation. This method, however, as was presented in presented by Mercadal in [16] can suffer of high sensitivity to modeling error and noise.

- Parameter identification (PID) is also used to generate residuals by producing estimations of characteristic coefficients of the system that experience unexpected changes when in presence of upset condition. Broussard and Trahan illustrated in [17] the use of two such techniques for detecting failures on both the armature winding resistance of a Direct Current (DC) servomotor and an analog second order-system presenting as the main problem the need for persistent excitation for successful parameter identification. Meyer and Zakrajsek applied PID techniques to detect failures in rocket engines test firing data in [18] using a limited database of failed and nominal firings with no sensor failures present; it is described in this work that the presence of sensor failures may trigger false alarms and would require extra tuning to cope with this abnormalities. Also in an aerospace context, Melody et al. compared in [19] the performance of three different PID techniques for in-flight detection and identification of aircraft icing and presented a detailed analysis of accuracy and velocity of detection from which the use of the H^{∞} algorithm for low levels of excitation of the system.

Knowledge based techniques become useful when a detailed or accurate model of the system is not available or simply too complicated to obtain. Artificial Intelligence (AI) techniques such as Neural Networks (NNs) and Fuzzy Logic have been used among knowledge based systems for failure detection and identification. NNs have the capacity of adaptation to changing environments by changing the weights of the different components of the network. The adaptation process of NNS is referred to as learning and allows the NN to adjust its performance to a desired behavior. NNs turn out to be very useful tools when dealing with highly non-linear problems [10]. Fuzzy Logic gives a tool to translate human thought process to computer processes thus allowing operators experience to be included in the detection logic. These two techniques have been used to develop logic maps that can detect failure signatures based on NNs estimations and expert assessment.

Neural Networks have been proposed to deal with failure conditions as early as the work by Elsley et al. [20] and Dietz et al. [21] in the end of the 80s. Napolitano et al. have shown in many research papers such as [22; 23; 24; 25] the validity of using NNs for failure detection and accommodation for upset sensors and actuators for airplane systems. Perhinschi et al. have also published papers in the same area like [26; 27] in which more robust thresholds and logic schemes are implemented to allow more flexibility to the detection process. Most of this work focused on high performance aircraft or Unmanned Air Vehicles (UAVs); however, Pesonen et al. proposed in [28] a NN based controller for general aviation airplanes capable of adaptation to faulty conditions requiring little or no pilot compensation. Nevertheless, this early work was based on PC based simulations and did not include lateral failures nor pilot in the loop interactions. The following two references show the versatility of the NN approach. First, the work proposed by Liu et al. in [29] using a two-stage improved Elman Neural Network model to perform failure detection on a hydraulic servo system shows the strengths of NNs approaches to overcome strong nonlinearities. Lastly, the work developed by Tarng et al. in [30] employed a multi-layer feed-forward neural network with Back-Propagation Algorithm (BPA) to detect abnormal situations in milling processes.

The concept of Fuzzy Logic isolated from any NN structure has not been widely used and one of the few works that can be referenced is that of Curry et al. [31] that applied a robust l_1 estimator to calculate residuals. These residuals were judged using fuzzy thresholds that allow for an extra evaluation of the abnormal conditions. Most research has focused on using Fuzzy Logic as a combination with NN scheme thus creating the so-called Neuro-fuzzy schemes. Chen et al. created in [32] a sensor fusion algorithm based on "*applying fuzzy logic to give a neural network real time adaptability to compensate for faulty sensors*" [32]. In general, the inclusion of fuzzy components in the NN detection schemes tends to increase the flexibility and robustness.

Most of the research detailed throughout this section has focused on individual classes of failures, not dealing with the need for comprehensive FDI schemes that can detect both expected/known failures and have the potential of provide acceptable detection capabilities for completely unforeseen situations. Integrating a variety of failures plus making the scheme robust enough to deal with wide ranges of the operational point, forces the FDI scheme to be capable of dealing with an increasing number of dimensions. It is in this context that the Artificial Immune System (AIS) concept emerged as a promising tool for developing more comprehensive failure detection schemes. This rather new biologically inspired technique has specific characteristics that allow it to deal with complex multidimensional problems and large amounts of information.

The mammalian immune system provided inspiration to many recent biologically motivated techniques that can be grouped under the so called Artificial Immune System techniques that emerged in the 1990s. The first scientific meeting with immunity based models as a center of attention was the international workshop “Immunity-based Systems” held in Japan in December 10, 1996. Two years after this conference, Dasgupta put together a set of publications regarding this novel branch of CI in the form of a book [33], that served to establish more formally some of the theories and principles being used by researchers around the globe. By 2002, the AIS had gained such importance that a conference exclusive on AIS related topics was created called International Conference on Artificial Immune Systems (ICARIS) that continues to operate regularly once a year since then.

One of the first applications of AIS principles for failure detection purposes in contained in [34] for a particular application to intrusion detection on a computer network. This early work presents the use of a Negative Selection algorithm based on the way the mammalian immune system allows only maturation of antibodies that do not attack the body’s own cells. Since these early stages, the theories referred to the AIS have been perfected and the applications have been diversified. Dasgupta and Forrest presented in [35] in 1998 a more formal view to the application of the NS approach to intrusion detection; however it took 10 years from the first appearance of the NS algorithm for Dasgupta to propose a real-valued failure detection scheme for application to aircraft subsystem failures in [36].

Some authors like Stepney et al. [37] have made considerable efforts to lay more formality onto AIS application in general, while others as Pasek in [38] have tried to formalize the theory behind a particular application such as the Negative Selection. Most failure detection applications of the AIS techniques such as [39; 40; 41] have used benchmark type of data to assess the detection capabilities of different algorithms to generate antibodies. Some of the practical applications of the AIS are the implementation in [42] of an FDI scheme for milling operation tool breakage, the application of danger theory for detection of erroneous communications inside a telephone network presented in [43] and the Negative Selection algorithm using binary representation used in [44] to assess the correct performance of refrigeration systems. The work by Ayara et al. [45] deserves a more extensive comment as it explores a combination of AIS techniques tending to create a system capable of immunizing a network of Automated Teller Machines (ATMs). This proposed system presents real time adaptation that intends to mimic the constant learning process of the natural immune system, while it also implements a vaccination concept via which one ATM can provide the rest of the network with information about local abnormalities to improve the global strength of the network to possible succeeding occurrences.

Besides the already mentioned application to airplane failure detection, two more works were focused on airplane subsystem failures. In 2004 Dasgupta et al. published [46], a research work primarily focused in showing the validity and importance of the presence of an accurate FDI scheme on top of any IFC available to reduce dead bands caused by adaptation times to unexpected situations. It was stated in this paper that knowing beforehand that the airplane is not working under healthy conditions can help trigger faster and more accurate responses of the IFC system. This effort used high level simulations of a C17 transport aircraft to test the validity of the approach "*The parameters considered for the fault detection study included body-axes commanded rates, actual aircraft body-axes rates, and corresponding neural network outputs.*" [46]. One year later, Wong et al. presented in [47] an extension of the previous year's work. In this case, the focus was in implementing a top layer to the system that allowed correct adaptation without the need for supervised identification. This automation layer provides this scheme with the capability to recognize the effects of the failures that it detects and thus modify the IFC accordingly. It is stated here that such a system can also provide accurate information to the pilot that can improve his/hers situational awareness therefore increasing the possibilities of a successful completion of the mission. As the preceding one, this paper also presents results using state of the art simulations of a C17 transport aircraft.

1.2. Research Objectives

The present work forms a constituent part of an extensive venture among West Virginia University's Center for Advanced Research in Autonomous Technology (CARAT) research group titled "Design, Simulation, Validation, and Flight-Testing of Adaptive Fault-Tolerant Flight Control Systems". This project is part of the NASA Experimental Program to Stimulate Competitive Research (EPSCoR). This project represents a continuation of the effort by the West Virginia University (WVU) team to advance both the state-of-the-art in fault-tolerant flight control theory and the state-of-the-practice in control system validation and testing methods. The primary objective of this research is the development of an integrated adaptive flight control system capable of detecting, identifying, and accommodating for both sensor and actuator failures in real-time. A complete system development cycle will be defined and demonstrated including conceptual and detailed design, PC-based and motion-based simulations, model-based software verification and validation (V&V), and incremental flight-testing validation.

Over the previous decade researchers at WVU have extensively investigated different adaptive architectures for on-line sensor and actuator fault diagnosis and accommodation. In addition, the WVU team has worked closely and effectively with researchers from NASA Dryden Flight Research Center towards developing and flight-testing several flight control laws for the NASA Intelligent Flight Control System (IFCS) program. This effort is aligned in the general area of fault-tolerant systems dealing with failures to the actuators of primary control surfaces as well as to different sensors in the flight control system.

The present research is a continuation of this sequence of research efforts that also included implementation of different detection schemes and assessment of their performance. Previous work has dealt mainly with data generated using state of the art simulations; however, this work represents the first effort among this research group that deals with real flight data for FDI implementation.

The present thesis describes the design and implementation of an immunity based FDI scheme for actuator failures using real flight data from a small jet Unmanned Air Vehicle. The first objective is to check the power of the Artificial Immune System paradigm for FDI using the available flight data. The second objective is to assess the performance for known failures and estimate its extension to unknown failures – this is one of the main advantages of this particular technique as is explained in the following chapters. Finally, a comparison of different FDIs is carried out to investigate the benefits and drawbacks of different FDI techniques.

1.3. Overview of the Thesis

The next chapter outlines the AIS paradigm starting from the biological inspiration and following to its application to failure detection in aircraft sub-systems including details of the different algorithms involved in the creation of the FDI scheme. Chapter 3 contains a detailed description of the UAV used for flight testing as well as the depiction of the totality of the data used. This chapter also presents the motion based simulator used for the comparison of failure detection schemes and explanations of each scheme's particular details. The design process of the FDI scheme using the AIS paradigm is discussed in Chapter 4 detailing the particular design decisions involved. The results obtained using the produced FDI scheme are summarized in Chapter 5. A simple comparison of FDI approaches is also included in this chapter. Conclusions are summarized in Chapter 6 together with possible future extensions of the present work.

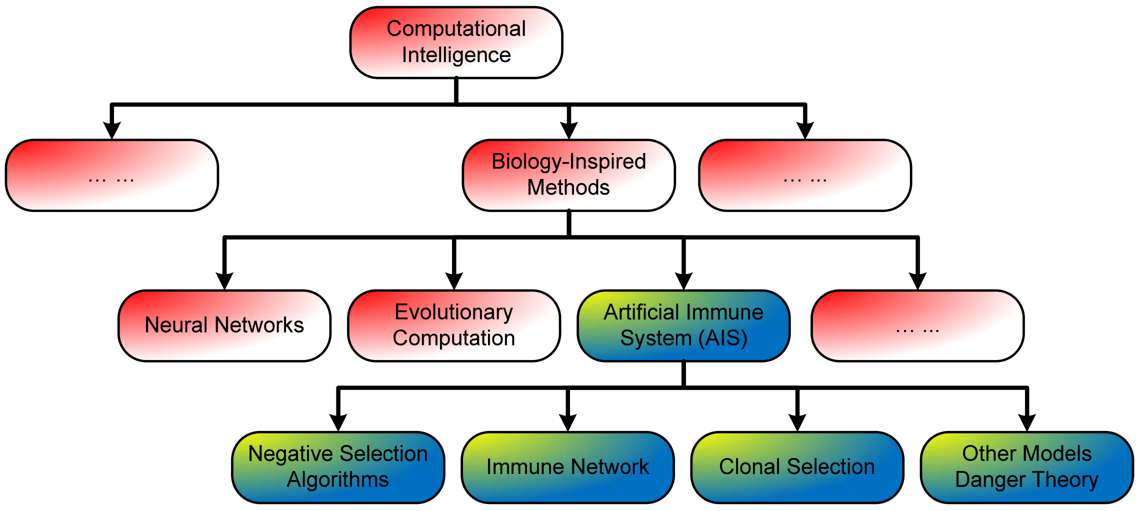
Chapter 2: IMMUNE SYSTEM

"*Immunology can be defined as the study of the defense mechanisms that confer resistance against diseases. The system whose main function is to protect our bodies against the constant attack of external microorganisms is called the immune system.*" [48 p. 9]

All vertebrates are equipped with a protection structure that helps them defend against dangers such as viruses, bacteria, etc. This is the immune system and it consists of various types of cells that fulfill specific functions; nevertheless, it depends also on a series of organs for its maintenance and evolution.

The Natural Immune System (NIS) constitutes a complex adaptive system that work in a decentralized way to detect and isolate possible threats to the organism, moreover, among other features, it also induces actions against these menaces (immune responses) and retains in memory previous positive detections to increase detection speed in the future. The NIS provided inspiration to many recent biologically motivated techniques that can be grouped under the so called AIS techniques.

Computer science helped develop many models in order to replicate biological phenomena. Moreover, these phenomena have also been used as metaphors that serve as inspiration to create new computational techniques capable of solving problems in a variety of disciplines [48]. These techniques are contained in the broader field of Computational Intelligence (CI), and form a separate branch referred to as Biology-Inspired Methods. Neural networks, evolutionary computation, and fuzzy systems can be included in this branch [49]. The AIS is one of the latest additions to this area and as such has experienced a great boost to try to explore its strengths as a CI technique. The previous is shown in the following figure showing also some of the different branches inside the AIS.



Computational Intelligence
… …
Biology-Inspired Methods
… …
Neural Networks
Evolutionary Computation
Artificial Immune System (AIS)
… …
Negative Selection Algorithms
Immune Network
Clonal Selection
Other Models Danger Theory

Figure 1: Artificial Immune System as a branch of Computational Intelligence [49].

2.1. Natural Immune System

This section does not intend to be a complete description of the processes and components involved in the NIS, however it is intended to give a broad view in all the aspects involved in such a complex and multilayered system. Although particularly clarified in some paragraphs, the main references for this section are [48] and [50].

Immunology is a rather new science that can track its roots back to the end of the 18th century, when Edward Jenner discovered the principles of vaccination as a way to immunize animals to certain diseases, even decades before science gained a decent understanding of the processes triggered by this phenomenon. The concept of antibodies was introduced in the early years of the 19th century; however, even 50 years after their first appearance, questions were still to be answered as to the processes involved in their creation and their actual role in the protection of the body. The theories of clonal-selection, negative selection and immune network were all formulated in the second half of the past century. These latter theories together with others not mentioned in this extremely brief historical account and introduced in the same time frame clarify the youth of this field in science. [48]

The immune system consists of an intricate arrangement of cells and molecules designed to protect the host's body against the constant attacks of antigens. It is constituted of two layers of defense, the innate immune system and the adaptive immune system, both of which depend upon activity of white blood cells, called leukocytes. These leukocytes can be divided considering the presence of granules into granulocytes and agranulocytes and the latter can also be subdivided into lymphocytes, monocytes, and macrophages. Granulocytes and macrophages are mainly involved in the innate immune system while lymphocytes constitute the main part of the adaptive immune system. [1]

The cells involved in the innate immune system are conceived to deal with many antigens even before any exposure of the body to these particular threats constituting a front line that directly attacks these extraneous cells. It is called innate because the reactions to particular hazards are equivalent in different healthy individuals. This initial immune response also helps triggering a sequence of reactions that activate the adaptive immune system.

The adaptive immune system implies a learning capacity of the immune system and thus requires the body to be exposed to particular antigens. This exposure triggers the creation of more specialized cells that provide a faster reaction to future exposures to the same antigen. This learning process consumes time and thus requires the presence of innate responses; however, in the future it fastens the response to already known antigens. Lymphocytes are responsible for both recognition and elimination of these alien entities. There are two main types of Lymphocytes, T-cells (T-lymphocytes) and B-cells (B-lymphocytes). Each naïve lymphocyte (lymphocytes before maturation process) carries surface antigen receptors of single specificity (monospecificity). This specificity is determined during the creation of lymphocytes and consists of a gene rearrangement that can create millions of different variants of the encoding genes. The surface antigen receptors of B-cells are particularly called antibodies while their T-cells counterparts do not have a specific name and are generally referred to as T-cell receptors.

The first major responsibility of the immune system is to differentiate all the cells within the body and classify these cells as self or non-self (pattern recognition). The antigen receptors mentioned above are basically surface molecules capable of recognizing antigens binding to determined protein chains found as well on the surface of the antigens. While lymphocytes are said to be monospecific, antigen might present several different types of protein chains in their surface, which implies that several lymphocytes can bind to each particular antigen. The binding process can be understood as a key and lock phenomenon, where binding occurs only when the two components match.

[1] Some authors [50] distinguish a third layer of defense to be the anatomic barrier consisting of skin, the mucous membranes, and bony encasements.

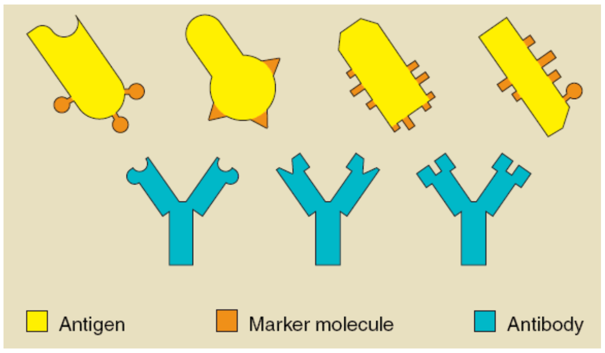


Figure 2: Simplified view of the pattern recognition mechanisms involved in the NIS. [51]

The production, evolution and distribution of the described cells, specially the lymphocytes, involve a set of organs called lymphoid organs. These lymphoid organs can be divided into primary or central organs, responsible for the production and maturation of lymphocytes, and secondary or peripheral organs, where lymphocytes experience stimulation inducing the adaptive immune responses. The following figure shows the main lymphoid organs.

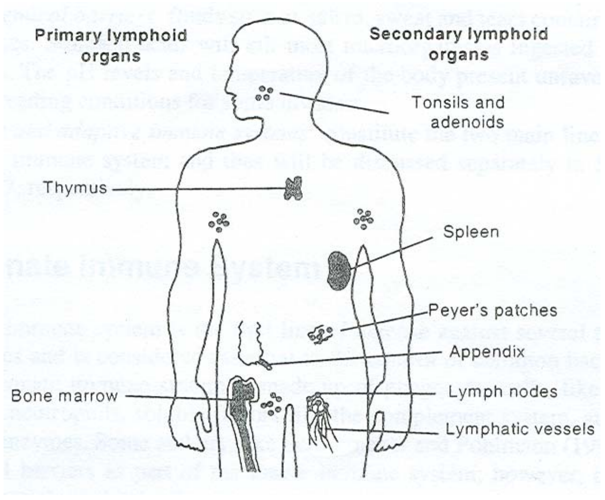


Figure 3: Anatomy of the immune system (lymphoid organs) [48]

Describing each of these organs is beyond the scope of this thesis, however, the two primary lymphoid organs deserve a short description of their role in the immune system:

- Bone marrow: Is a soft tissue found inside the most elongated bones. It houses stem cells that then become white and red blood cells. Is the main supplier of blood cells.
- Thymus: It is a glandule located behind the sternum, above and in front of the heart. It provides the required environment for T-cells maturation.

Both B-cells and T-cells are created in the bone marrow; however, the maturation process of B-cells occurs also in the bone marrow, while T-cells mature inside the thymus.

"*The immune system in its ability to recognize antigens is complete. The antibody molecules and T-cell receptors produced by the lymphocytes of an animal can recognize any molecule, either self or non-self, even those artificially synthesized.*" [48 p. 36]. The previous phrase simply states that the immune system is assumed to comply with the completeness axiom and means that part of the initially random generated lymphocytes reacts to the own body cells. This is not acceptable considering that, as was mentioned before, lymphocytes are the main pattern recognition tool in the self-non-self discrimination. Lymphocytes then experience a censoring or selection processes to avoid the phenomena of autoimmunity. There are two main such mechanisms used by the immune system:

- Positive Selection (PS): Impedes the natural process of cell death for necessary lymphocytes.
    - T-cells experience what is called thymic positive selection (in the thymus) by which cells that bind with specific molecules (self-MHC [48]) are selected and their lifespan is increased.
    - B-cells experience a similar process, selecting cells that can recognize particular non-self molecules in the presence of T-cell activation signals.
- Negative Selection (NS): Results in the death of a lymphocyte due to binding with self cells. For this purpose, the primary lymphoid organs contain big sets of self-cells.
    - T-cells NS can occur in the thymus or in the secondary lymphoid organs. The process basically consists in exposing the T-cells to a collection of self-cells and eliminating the ones that experience activation.
    - B-cells NS occurs in the bone marrow or in the secondary lymphoid organs and is similar to the above explained process; however, the cells that are eliminated are the ones that get activated without the necessary presence of an activated T-cell.

These two censoring and selection mechanisms have as a consequence that no matured lymphocyte activates immune responses against self cells. Once the lymphocytes are matured, they are sent throughout the body to perform their described tasks locally.

There are two types of responses the Immune System experiences when the adaptive immune system detects an antigen:

- Humoral Immune Response (antibody mediated) is triggered by an activated B-cell secreting antibodies that bind to the detected antigen to mark it thus summoning other active cells such as phagocites to destroy the aggressor.
- Cellular Immune Response (cell mediated) occurs after a T-cell has been activated becoming what is called effector. Effectors trigger the proliferation of similar specificity receptors in the area surrounding the antigen. It also produces secretion of chemicals with the purpose of eliminating the threat.

The number of lymphocytes that can bind to a particular antigen is limited and thus depends on adaptive processes to favor the reproduction of successful lymphocytes. This adaptive process is called clonal selection theory (or clonal expansion principle) and involves mainly B-cells. After a B-cell becomes active it is helped by other accessory cells to allow stimulation by the antigen. This stimulation results in the proliferation (division) of such B-cells and their following maturation into antibody secreting cells called plasma cells.

Learning in the context of the immune system involves increasing the number of clones in the lymphocytes population of the antibodies that have experienced positive successful activations thus generating a biased distribution in contrast with the original ideal random distribution. The global population of lymphocytes circulating throughout the body is considerably large and is kept somewhat constant; therefore an increase in the number of clones on one type of antibody produces in consequence a drop in the occurrence of other antibodies considered less important. This does not mean that after an infection, the immune system maintains a large number of clones of these successful antibodies; on the contrary, it only maintains in memory a small set of the best fit antibodies for this particular infection. This small set of cells kept latent are called memory cells and help speed up the response of the immune system after a first infection on account of their availability for cloning purposes in case the same antigen is detected in the future.

There is a somewhat parallel theory formally proposed by Jerne in [52] in 1974 named Immune Network Theory (or Idiotypic Network Theory). This theory presents a different approach to many of the processes involved in the immune system such as memory, learning, self/non-self discrimination, etc. The basis of this theory states that the immune system is composed mainly of a regulated network of molecules and cells that recognize themselves even in the absence of antigens. As can be easily inferred, this theory is clearly in conflict with the clonal selection theory. The details of this concept are beyond the scope of this brief introduction to immunology.[2]

2.2. Artificial Immune System

The set of techniques inspired in the behavior of the Natural Immune System are referred to as Artificial Immune System techniques. This section presents first an overview of the characteristics of the NIS that are valuable for other fields of science and then introduces the models created to mimic these features. A special subsection is dedicated to the Negative Selection principle that represents the main technique used for this thesis.

2.2.1. Computational aspects of the NIS

"*From an information-processing perspective, the NIS is a remarkable parallel and distributed adaptive system with (partial) decentralized control mechanism.*" [49]. There are several characteristics that make the NIS a good metaphor for solving engineering problems, and thus, from a computational point of view, the following is a list of such characteristics:

- Recognition: Self-non-self discrimination in the immune system represents the NIS's ability to recognize particular patterns and act accordingly. This concept is greatly related with the amount of information available a priori to the NIS. [33] [53].
- Diversity: To be able to deal with either known or unknown antigens, the creation of the lymphocytes relies in part on a genetic process that allows that any antigen at least binds with some of these lymphocytes improving robustness. [33]
- Learning: The NIS learns from experience increasing the chances of detecting an antigen after the first infection by regulating the proliferation of specialized lymphocytes.
- Memory: After positive activation, keeps a small amount of the most specialized cells that detected the threat as memory cells to be used in the event of a second infection.
- Distributed detection: As the lymphocytes perform the detection processes locally, there is no central control unit necessary for leading the immune responses. This also implies that local failures in the detection do not produce global failure of the system.

[2] Details of this theory can be found in [48] in section 2.11, page 41.

- Threshold mechanism: Immune responses depend not only on a single cell binding but on the reaction of adjacent cells after the first activation that depends on the strength of the chemical binging. [33]
- Dynamic protection: The NIS is constantly discarding cells and creating new ones. Clonal selection, as well as other such mechanisms, is used to favor exploration over exploitation.
- Probabilistic detection: The detection is approximate; a lymphocyte can bind with different antigens with similar molecular structures.
- Robustness: Is a consequence of the NIS being diverse, distributed, and error tolerant. [54]

Other features of NIS important from a computational point of view are: adaptability, specificity, reinforcement learning, parallel processing, multi layered, no centralized control, self tolerance, co-stimulation, self regulation, etc.

2.2.2. AIS techniques

All the characteristics described in the previous section have allowed researchers to develop models of one or many such mechanisms to be implemented not only as models of the NIS itself but as metaphors to solve other science problems. There is a vast diversity of models based on the NIS and thus included in the AIS techniques that have been explored, however this section includes a brief review of the most applied ones only.

The immune network model is based primarily on the theory first expressed by Jerne in [52] by which B-cells produce a network that produces bindings even in the absence of stimulatory antigens. This model generates a starting population and then uses cloning and mutation processes based on the interconnections experienced when the initial population is exposed to training data. At each new step, the algorithm attempts to incorporate new cells to the remaining network using affinity criteria. Lack of this affinity produces the elimination of such individuals. Immune network models have proven their use in problems that involve learning such as pattern recognition [55], clustering [56], and data mining [57], among others.

The bone marrow model uses a gene library concept as information basis for the generation of individuals. This gene library contains pieces of a solution that has been determined a priori [53]. The production of such antibodies uses random concatenation and the newly created individuals are evaluated for affinity using a fitness function. This model has not been extensively used in literature; however, [58] makes use of the gene library concept to solve scheduling problems.

The Clonal selection algorithm (based on the principle of the same name) is another of the AIS techniques and has some similarities with the genetic evolution. The main difference between genetic evolution and the clonal selection principle is the time frame. The NIS relies on this principle for the generation of significant changes in rather short periods of time in the order of hours (to help in the adaptive immune response). The developed algorithm selects well fit individuals, performs cloning and in the process favors the maturation of the best fit ones. Moreover, during the maturation process it performs hypermutation (high levels of mutation) to promote exploration of spaces close to the best fit individuals. This algorithm provides a successful method for searching complex spaces [53]. Among other areas, this algorithm has been used for learning and optimization [59], pattern recognition [39], and anomaly detection [60].

Positive selection was informally described by Ebner et al. in [61] and formally by Stibot et al. in [62]. This technique has been applied both to binary-string as well as real-valued cases using self data as detectors and applying some particular constant matching rule. The benefits of this technique according to [63] is that no training is required while the main drawback is said to be the computational cost in view of the amount of self data that is used. Moreover, Ji and Dasgupta state in [64] that "*when the number of self samples is larger than the number of detectors by more than one order of magnitude, such positive selection method like Self-Detector is not a realistic solution*". Not many applications of this method can be found in literature applied to real world problems. The above mentioned work by Stibor et al. [62] uses KDD Cup 1999 data set for intrusion detection on a computer network while Ji and Dasgupta used Iris-Fisher and Biomedical datasets for their analysis in [63]. All the mentioned datasets are usual benchmarks for comparing novelty detection methods.

Negative selection algorithm was initially proposed by Forrest et al. in 1994 [34] using binary representation and it tries to emulates the maturation process of T-cells in the thymus in which cells that get activated in the presence of self cells are discarded in what is called self/non-self discrimination. Due to the importance of this technique for the present AIS implementation, a more detailed description of the algorithm is presented in the following subsection. The main applications of this method are in the area of anomaly or novelty detection in timed series [35] and computer virus detection [65].

2.2.2.1. Overview of negative selection algorithm

The NS algorithm in general can be summarized in three main steps [35]:

1. Definition of the self as a collection *S* of normal patterns/activities corresponding to stable behavior of the process/system that needs to be monitored. This data needs to be normalized into a finite feature space *U*.
2. Generation of a set *R* of detectors, each of which fails to match any of the elements of *S*. A way to generate these detectors is by randomly generating individuals inside *U* and only using the ones that comply with the criteria. This random generation can also serve as a first step to the use of other growth algorithms to optimize detector distribution.
3. Monitor new observations for changes by continually checking for matches of new incoming data with the set *R*.

This absolutely general description of the NS algorithm applies to almost every implementation of this algorithm found in literature. The differences arise when dealing with the details in each of these steps. First, it is important to formally define the sets mentioned earlier. *U* is the universe of possibilities for the data observed from the process being monitored and is partitioned into two subsets which are called from now on Self (S) and Non-self (N) with the following properties [54]:

$$U = S \cup N \tag{2.1}$$

$$S \cap N = \varnothing \tag{2.2}$$

The type of data that forms *U* produces the first main division. Most NS algorithms have dealt either with binary string or real valued representations, however, some hybrid approaches have also used integers, categorical information, boolean values, text information, etc. [66]. The principal consequence of the choice for representation would be the possible matching rules that can be implemented to check a new data point as included in *S* or in *N*. The matching rule represents a way to interpret the relative distance between an arriving data instance and a specific data point. Matching rules allow for partial matching, in which case, the incoming data only needs to be “close” to the detector to establish a match. Matching rules can be combined or replaced with a matching threshold.

There are three main binary strings matching rules usually reported in literature, this being r-contiguous bits, Hamming distance and r-chunk matching rules. The definitions are not included in this document as the implementation is based on a real-valued approach.[3]

For the case of real-valued representations (or vector representations), each data point is a vector of size n – one component per dimension used. Distance in an n-dimensional space can be addressed in many different ways; however, for a distance definition $d\left(\vec{x},\vec{y}\right)$ between the points $\vec{x}$ and $\vec{y}$ to be valid it must comply with the following metric properties [67]:

- Non-negativity: $d\left(\vec{x},\vec{y}\right)\geq 0$
- Reflexivity: $d\left(\vec{x},\vec{y}\right)=0$ iff $\vec{x}=\vec{y}$
- Symmetry: $d\left(\vec{x},\vec{y}\right)=d\left(\vec{y},\vec{x}\right)$
- Triangle inequality: $d\left(\vec{x},\vec{y}\right)+d\left(\vec{y},\vec{z}\right)\geq d\left(\vec{x},\vec{z}\right)$

One distance that complies with this definition is the Euclidean distance that can be expressed as:

$$d\left(\vec{x},\vec{y}\right)=\sqrt{\left(x_1-y_1\right)^2+\cdots+\left(x_n-y_n\right)^2} \tag{2.3}$$

However, Euclidean distance can be interpreted as a particular case of Minkowski distance (λ-norm distance) for λ=2. Minkowski's distance is defined as:

$$d_\lambda\left(\vec{x},\vec{y}\right)=\left(\sum_{i=1}^{n}\left|x_i-y_i\right|^\lambda\right)^{1/\lambda} \tag{2.4}$$

This definition of distance complies with the requirements stated before for any value of λ. Reference [66] presents a thorough discussion about a variety of definitions of distance for real valued representations. On the limit, when $\lambda\rightarrow\infty$, Minkowski distance tends to:

$$\lim_{\lambda\rightarrow\infty} d_\lambda\left(\vec{x},\vec{y}\right)=d_\infty\left(\vec{x},\vec{y}\right)=\max\left\{\left|x_i-y_i\right|,\text{ for } i=1,\cdots,n\right\} \tag{2.5}$$

[3] A simplified explanation of these methods as well as further references can be found in [66].

All these definition are useful for the following section where the particular implementation is described.

The set N is usually considerably larger than S and using the entire set for detection purposes is often impractical. To cover N it is necessary to define detectors that can act as the antibodies computational counterparts. These detectors, represented by circumscribed geometrical hyper-bodies, report a match whenever an incoming data instance is found to be inside its boundaries. Any of the previously shown distance definitions can be used to describe the limits of hyper-bodies; however, the choice produces changes in the physical image of such a geometrical body. The hyper-bodies most commonly used for real-valued NS algorithms are:

- Hyper-Rectangles are defined as a vector $\vec{D}$ composed of two other vectors $\vec{c}$ and $\vec{s}$:

$$\vec{D} = (\vec{c}, \vec{s}) \tag{2.6}$$

  Where $\vec{c}$ is a vector of size n that represents the center and $\vec{s}$ is a vector of the same size that contains the side of the hyper-rectangle for each dimension. Hyper-cubes are a particular case where $\vec{s}$ is actually a scalar. In this type of detectors, no overlapping is usually allowed greatly simplifying calculations such as coverage [68].

- Hyper-spheres are defined as a vector, as shown in (2.6), where $\vec{s}$ is a scalar and corresponds to the radius. Different definitions of distance cause changes in the shape of the spheres from the intuitive (Euclidean) idea of such geometric body. The effects of these changes are analyzed and explained in following sections.
- Hyper-ellipsoids of rotation are described also by a vector containing the center $\vec{c}$ and two values for the axes included in $\vec{s}$.
- Generalized hyper-ellipsoids are described by a vector containing the center $\vec{c}$ and n values for the different axes included in $\vec{s}$.

There are other hyper-bodies that would be valid for NS detection; however favoring computational simplicity the four above mentioned have been the most used ones. Hyper-spheres are used as detectors in the present implementation and so a more detailed description of the particularities involved is included in following chapters.

The desire when creating detectors is to maximize the coverage of *N*. If any of the last three shapes described above is used, it is necessary to allow some overlapping between detectors or there would not be a possibility to cover "holes" in between them. This overlapping is a burden when calculations such as volume of detectors or coverage are needed; however, the benefit of using a simpler hyper-shape pays off for this inconvenience.

Coverage calculation in higher dimensions is a challenging task. Stibor et al. presented in [67] a brief description of the use of Monte Carlo method for obtaining a measure of both coverage and overlapping between detectors. The method has probabilistic basis and consists mainly of the following steps:

1. Given a subset A of the universe U that is defined using n dimensions
2. Generate a set T of P random points using a uniform distribution.
3. Calculate the size of $A \cap T$ as P_A
4. The coverage is calculated as P/P_A

In the case of this AIS application, the subset A is a set of hyper-bodies, U is the hypercube in which the totality of the self data is contained and the calculation of P_A is performed using the concept of distance to check how many of the P points in T are contained inside the bounds of at least one of the hyper-shapes in A. Stibor et al. also state in [67] that this method has an error independent of n and of order $1/\sqrt{P}$. It also describes that "*…specifying a confidence level $1-\eta$, one can determine the smallest sample size P that guarantees an integration error no larger than ε*" [67]. This is called the "*…(ε,η) absolute criterion that leads to the worst-case sample size*" [67]:

$$P := \left\lceil \frac{1}{4 \cdot \eta \cdot \varepsilon^2} \right\rceil \tag{2.7}$$

Overlapping calculations are performed in a similar fashion. In this case, after a point in T has been found to belong to the dominions of one hyper=body in A, the rest of A is checked to find if the same point belongs also to another. The size of the subset of points in T that are found to belong to more than one element in A is P_O. The overlapping is then calculated as P_O/P_A.

Most NS based algorithms perform the generation of detectors offline because of the complexity and computational cost of such a process; however, they benefit of the relatively low cost of monitoring an already defined set of detectors for online operation.

As the NS method relies on the use of geometrical tools in high dimensional space, it is important to mention the phenomenon known as "*curse of dimensionality*". This phenomenon that was first mentioned by Bellman [69] is responsible for many issues regarding NS algorithms using high dimensional real-valued spaces. The following is an extract from [67] with a simple example regarding this phenomenon: "*For example, given a function defined on a unitary hypercube of dimension n, in each dimension 10 discrete points are considered for evaluating the function. In dimension n = 2, this results in 100 evaluations, whereas in dimension n = 10, 10^{10} function evaluations are required.*" This has undesirable consequences in the first step of random generation of detectors as the number of evaluations needed to accomplish a certain resolution follows similar trends as the ones presented in the example. Moreover, this phenomenon has implications in the properties of hyper-spheres in higher dimensional spaces with the most important being that the volume tends to 0 if n goes to infinity. This means that for a constant radius, hyper-spheres generally cover less detection space N as n increases. Stibor et al. showed in [67] that for a determined radius, there is a dimension for which the volume is a maximum and any decrease or increase in dimension would produce a decrease in the space covered. They also showed that hyper-spheres with radius of 1 or less experience their maximum volume in dimensions less than 6. A consequence of high-dimensional spaces is that to maintain a constant coverage percentage of N (constant detection resolution) using a determined hyper-body, exponentially more detectors need to be defined [68].

Dasgupta presents in [49] a list of factors that play an important role in the successful performance of an NS algorithm:

- Algorithm for generating detectors: Detectors quality depends upon the success of this algorithm.
- Number and coverage of detectors: More detectors with less overlapping increase the chances of detecting data that belongs to N. The need for more detectors undermines the algorithms efficiency in generating the detectors and in monitoring activation.
- Applicability of scenario: In general, problems that are suitable for this technique require availability of a large amount of self data (in contrast with problems with more existing upset data).
- For real-valued NS algorithms:

- Curse of dimensionality: Its effect in the particular application needs to be understood.
- Estimation of coverage: In dimensions above 3, the accuracy of the coverage estimation needs to be addressed as no exact measure can be implemented.
- Selection of distance measure: As explained before, consequences are expected of different choices of distance definition.

The theoretical foundation of this method is still a work in progress; however, Ji and Dasgupta describe in [66] the main advantages of a NS approach as being:

- No prior knowledge of non-self is required.
- Detection is distributed not requiring communication between detectors.
- It can hide the self concept. This benefit is mostly acknowledged when privacy is an issue and the negative selection aids in only storing negative representation of the information for querying purposes [70].

2.2.3. AIS techniques for aircraft sub-system failure detection

This implementation is based mainly on the use of a Real-valued NS algorithm. For the aircraft AIS-based FDI, the self is defined as a collection of all the possible combinations of parameters that are representative of normal operations. A subset of these parameters needs to be specified such that this division contains dynamic signatures that can show difference with or without failures. These variables are called identifiers and play a similar role to the molecules that form the antigen receptors of Lymphocytes. Candidate identifiers can be grouped in the following categories [68]:

1. Aircraft state variables
2. Pilot input variables
3. Stability and control derivatives
4. Variables generated within control laws
5. Derived variables

For the test-bed used, and as is explained in the following chapter, categories 1 and 2 are available and variables in category 5 could be experimented. Aircraft state variables represent the first intuitive choice; however, Perhinschi et al. showed in [68] that similar combinations of state variables can be achieved either by faulty conditions or by coupled pilot inputs and so stated that pilot input information is needed for correct failure detection.

Selection of identifiers is not a simple process; however, knowledge of the system is the major instrument for a successful selection. The critical process of selecting the identifiers is performed through a limited "trial and error" approach as shown in Figure 4. Several combinations of candidate identifiers were tested to define alternative selves. These selves were used as detector sets using a Positive Selection approach to assess their validity using a limited set of test data. The best set of identifiers is then chosen to design the FDI scheme.

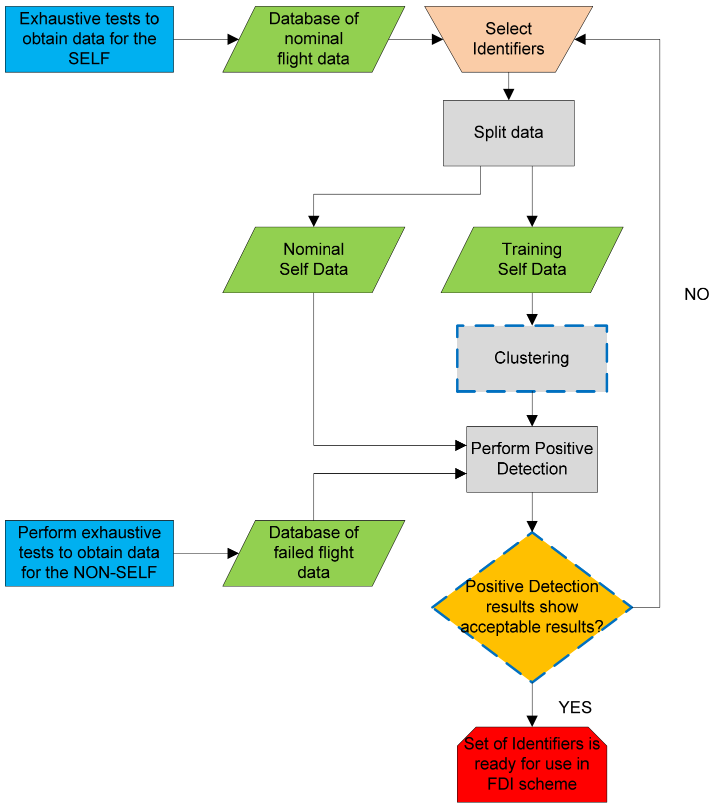

Exhaustive tests to obtain data for the SELF
Database of nominal flight data
Select Identifiers
Split data
Nominal Self Data
Training Self Data
NO
Clustering
Perform Positive Detection
Perform exhaustive tests to obtain data for the NON-SELF
Database of failed flight data
Positive Detection results show acceptable results?
YES
Set of Identifiers is ready for use in FDI scheme

Figure 4: Selection of identifiers for AIS-Based FDI - General block diagram.

Positive detection consists of monitoring incoming data points to check if they are within the limits of at least one cluster thus detecting this point as belonging to the self. Acceptable results for this evaluation imply that a percentage of at least 50% of the validation data is correctly detected while no more than 2% of the failed data is mistakenly detected. The clustering process is described in subsection 2.2.3.2 and, as is explained later, includes an embedded design loop.

2.2.3.1. Normalization

Once the identifiers have been selected, the database consists of a cloud of data points in an n-dimensional space. The distance between the points is critical for the detection process and different scales of the identifiers are not desired; therefore, the data must be normalized. This normalization is made such that the self data is contained inside the hypercube $[0,1]^n$, where n is the number of dimensions involved. Moreover, the normalization is such that the self data is located around the center of the hypercube.

First the minimum and maximum of all data for each dimension is calculated as:

$$M_j = \max\left\{\overline{x}_{ij},\ \text{for } i = 1,\cdots,n\right\} \tag{2.8}$$

$$m_j = \min\left\{\overline{x}_{ij},\ \text{for } i = 1,\cdots,n\right\} \tag{2.9}$$

Where i is an index that accounts for the different dimensions involved (from 1 to m), j is an index that accounts for the number of points defining the self (from 1 to n), and $\overline{x}_{ij}$ is a raw data point. Then, for each $\overline{x}_{ij}$ point, the following values are calculated:

$$\tilde{x}_{ij} = \overline{x}_{ij} - m_j \tag{2.10}$$

$$x_{ij} = \frac{\tilde{x}_{ij}}{M_j - m_j} \tag{2.11}$$

Where $\tilde{x}_{ij}$ is an intermediate step and x_{ij} is the corresponding normalized value.

2.2.3.2. Clustering

The data points are clustered to reduce the amount of information the following processes have to deal with and also to avoid possible “holes” of healthy data not represented in the self database. For simplicity, the same hyper-body shape and definition of distance that is used for generating the detectors should be used. In this case, hyper-spheres are chosen and Euclidean distance. Clustering is performed using an improved k-means algorithm [71].

The implemented algorithm essentially creates a random set C of N_C centers (points) in the n dimensional space using a uniform distribution. It calculates then the relative distances between these centers and the points in the set S of self data. Each point in S is assigned a membership to the closest element in C. Each element c of the set C is iteratively moved based on the mean of the points S assigned to it. The algorithm converges when the distance d_C from each element c to its assigned points in S is constant. For the case of hyper-spheres, the radius of the cluster is set to the maximum between the converged value of d_C and the minimum radius allowed $r_{\min}$. The algorithm used takes advantage of the method presented in [71].

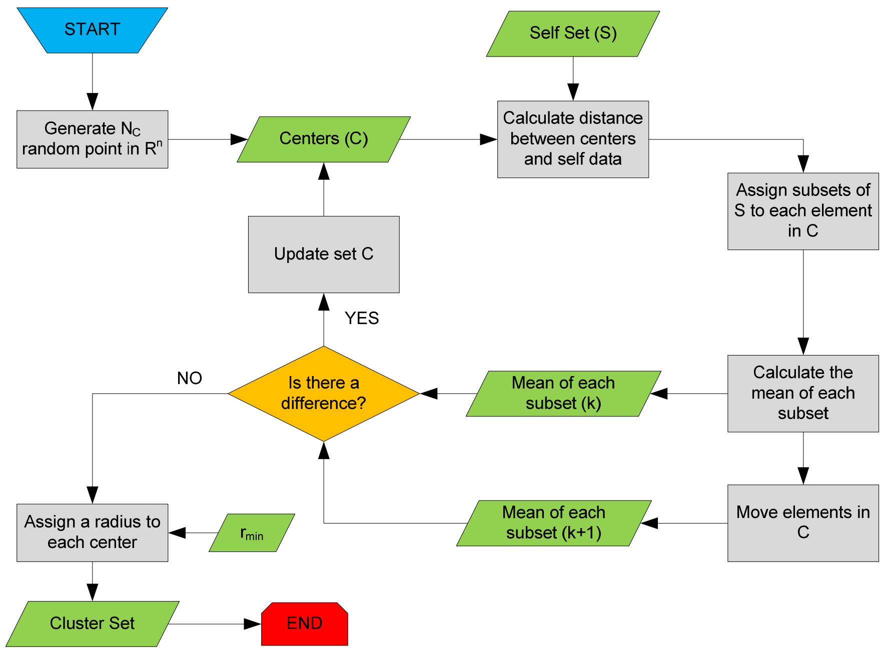

START
Self Set (S)
Generate N_C random point in R^n
Centers (C)
Calculate distance between centers and self data
Assign subsets of S to each element in C
Update set C
YES
NO
Is there a difference?
Mean of each subset (k)
Calculate the mean of each subset
Assign a radius to each center
r_{min}
Mean of each subset (k+1)
Move elements in C
Cluster Set
END

Figure 5: Clustering algorithm

Clusters created with this algorithm include a (potentially) large number of self points and also a certain amount of “empty-space”. If a small number of clusters are generated, the size of each cluster will be large as compared to the distances between points and thus include a big amount of this “empty-space” and potentially non-self space. A consequence of this is an increase in the size of the space effectively considered as self (covered by the clusters). On the other extreme, if there are as many clusters as self data points, the spaces in between clusters may allow detectors to be placed in locations where healthy behavior may fall thus generating false activations. An analysis of the behavior of this algorithm regarding “empty-space” coverage is included in [68].

Once a good trade-off has been accomplished between the number of clusters and "empty-space" coverage, the cluster set is formally referred to as the SELF.

2.2.3.3. Detector generation

Due to the high dimensionality of the problem, the detector generation process may be exposed to several specific issues [68]. One of these is that adequate coverage of the non-self may not be achieved with a reasonable computational effort. Different approaches are used to avoid this situation and are explained in the following.

The present implementation is a side project of an extensive effort among the research group headed by Dr. Mario Perhinschi for creating a comprehensive AIS FDI schemes for the NASA IFCS F-15 simulator [72]. For this broader project, a variety of algorithms have been studied, tested and evaluated and the result is a two phase evolutionary algorithm. The first phase consists of the Enhanced Negative Selection Algorithm for Real-Valued representation with variable detector radius (ENSA-RV) that ensures no overlapping with the self and guarantees certain coverage of the non-self [68]. The second phase uses Genetic Algorithms to optimize the already created set of detectors [73]. For the present implementation, only the first phase is used. The ENSA-RV algorithm steps can be summarized as follows:

1. Starts with a limited initial population of detectors whose centers are placed randomly in the non-self space. The radius of these detectors is set as the maximum possible before overlapping with a self cluster, meaning that the distance between centers is greater than or equal to the radius of the cluster plus the minimum radius $r_{\min}$ allowed for a detector.
2. Detectors are evaluated for overlapping checking how much of a particular detector is inside the rest of the set. The following are the formulas to calculate this overlapping [36]:

$$\delta_{ij} = \frac{r_i + r_j - d_\lambda\left(\vec{c}_i, \vec{c}_j\right)}{2 \cdot r_i} \tag{2.12}$$

$$w_i = \sum_{j=1}^{m} \left(e^{\delta_{ij}} - 1\right)^{n-1} \tag{2.13}$$

A fixed threshold w_{thre} is established. If $w_i \leq w_{thre}$ the detectors are selected as matured while the remaining ones are selected as rejected.

3. Matured detectors experience a cloning process:
    a. Matured detectors with $w_i = 0$ are placed in an area where no other detectors are found and as such are selected to have the most number of clones. A random unitary direction $\vec{d}$ is calculated and a detector center is added at a distance equal to one radius. A number of $N_{CLON} = (n-1)\cdot 2 - 1$ of centers are generated at 90º angles from the first distance in n different planes.
    b. Matured detectors with $0 < w_i \leq w_{thre}$ are placed in an area where not many detectors are found and as such are selected to have only one clone. The distance to all clusters and detectors is calculated and the closest is then selected. A unitary direction $\vec{d}$ is calculated opposite to the direction of the closest element. The distance to the new center, however, is controlled by two user-defined parameters τ_{CLON} and $\eta_{iniCLON}$. The following decay parameter is needed to calculate the new center:

$$\eta_{CLON} = \eta_{iniCLON} \cdot e^{-iter/\tau_{CLON}} \tag{2.14}$$

$$\vec{c}_{clon} = \vec{c}_{mat} + (1+\eta_{CLON})\cdot r_{mat} \cdot \vec{d} \tag{2.15}$$

    τ_{CLON} grants that for the first iterations the cloned detectors will be placed farther away from the original cluster while the latter iterations will generate clones closer to the original detector. $\eta_{iniCLON}$ is an initialization value involved in the calculation of η_{CLON} that directly affects the position of the new detector.

4. A defined number N_{MOV} of rejected detectors are selected to face a moving process. These detectors are selected as the N_{MOV} smallest rejected detectors. The moving is performed using a similar decay factor as (2.14) but replacing τ_{MOV}, η_{MOV} and η_{iniMOV} instead of τ_{CLON}, η_{CLON} and $\eta_{iniCLON}$ respectively. The moved center is calculated as:

$$\vec{c}_{mov} = \vec{c}_{rej} + \eta_{MOV} \cdot \vec{d} \tag{2.16}$$

The old rejected detector is removed from the database and replaced by this new one.

5. A set of N_{RD} centers is inserted to explore areas of the hyperspace that may have not been explored in the first random placement of detectors.
6. The maximum allowable radius is calculated for the cloned, moved and newly inserted centers.
7. Calculate coverage and overlapping using Monte Carlo method with $\eta = 0.98$ and $\varepsilon = 0.01$.
8. Go back and perform steps #2 to #7 iteratively. The process stops when a determined number of iterations is reached, when the desired number of detectors is achieved or when a certain coverage is obtained.

The following flowchart summarizes the process of detectors generation using ENSA-RV algorithm.

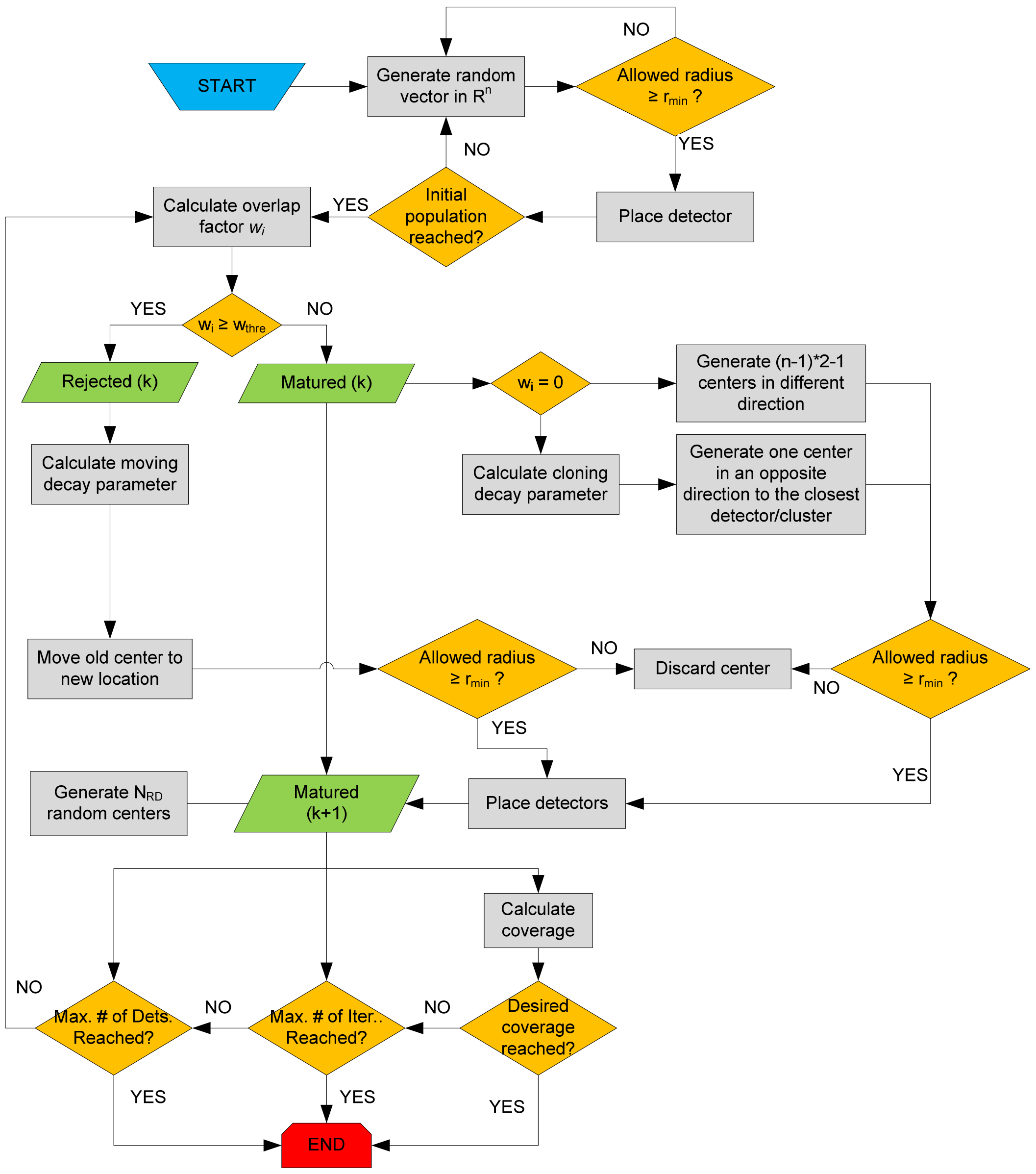

START
Generate random vector in R^n
Allowed radius $\geq r_{min}$?
NO
YES
Place detector
Initial population reached?
NO
YES
Calculate overlap factor w_i
$w_i \geq w_{thre}$
YES
NO
Rejected (k)
Matured (k)
$w_i = 0$
Generate (n-1)*2-1 centers in different direction
Calculate moving decay parameter
Calculate cloning decay parameter
Generate one center in an opposite direction to the closest detector/cluster
Move old center to new location
Allowed radius $\geq r_{min}$?
NO
Discard center
NO
Allowed radius $\geq r_{min}$?
YES
YES
Generate N_{RD} random centers
Matured (k+1)
Place detectors
Calculate coverage
NO
Max. # of Dets. Reached?
NO
Max. # of Iter.. Reached?
NO
Desired coverage reached?
YES
YES
YES
END

Figure 6: ENSA-RV algorithm

This algorithm has been used for the implementation in [72]; however tests for the particular problem trying to solve in this theses showed that the algorithm favored too much big detectors and so very few detectors with radius under 0.1 were obtained. Dasgupta et al. explained in [36] that this is an expected behavior due to the use of the formulas (2.12) and (2.13) to calculate detector overlapping. This is found to be disadvantageous for detecting the failures implemented using the available identifiers.

A second algorithm is then proposed using a vast portion of random generation. The algorithm is a one-shot type of algorithm similar to the V-detector algorithm presented in [74]. The V-detector algorithm is extended and enhanced. This algorithm, that for this implementation is named #2, consists of the following steps:

1. Generates a population of detectors whose centers are placed randomly in the non-self space. The radius of these detectors is set as the maximum possible before overlapping with a self cluster, meaning that the distance between centers is greater than or equal to the radius of the cluster plus the minimum radius allowed for a detector.
   A number of intervals of detectors sizes are defined. For each interval a determined number of detectors (N_{RD}) are forced to be placed and a number of clones (N_{CLON}) is assigned. As is stated before, due to the "curse of dimensionality", randomly obtaining detectors closer to the self is not an easy task and thus forcing the algorithm to provide a certain amount of such detectors increases the chances of obtaining better coverage in the very proximity of the self.
2. The cloning process in this case is performed in a similar fashion as in the ENSA-RV algorithm. A random unitary direction is obtained and a new center is defined at a distance equal to:

$$\vec{c}_{clon} = \vec{c}_{RD} + RD \cdot \vec{d} \tag{2.17}$$

   Where RD is a random number between 0 and 1. This step is repeated for as many times as the number of clones in the above mentioned table states; however, these centers are then evaluated to check that they fall in the non-self space and that the maximum allowed radius is bigger than the minimum threshold $r_{\min}$.
3. This population of detectors contains huge overlapping that needs to be addressed. The set of detectors is now processed using the same maturation process as the ENSA-RV but using a bigger threshold. The detectors that are found to be rejected are eliminated whereas the matured ones formed the final set.

The following flowchart summarizes the process of detectors generation using algorithm #2.

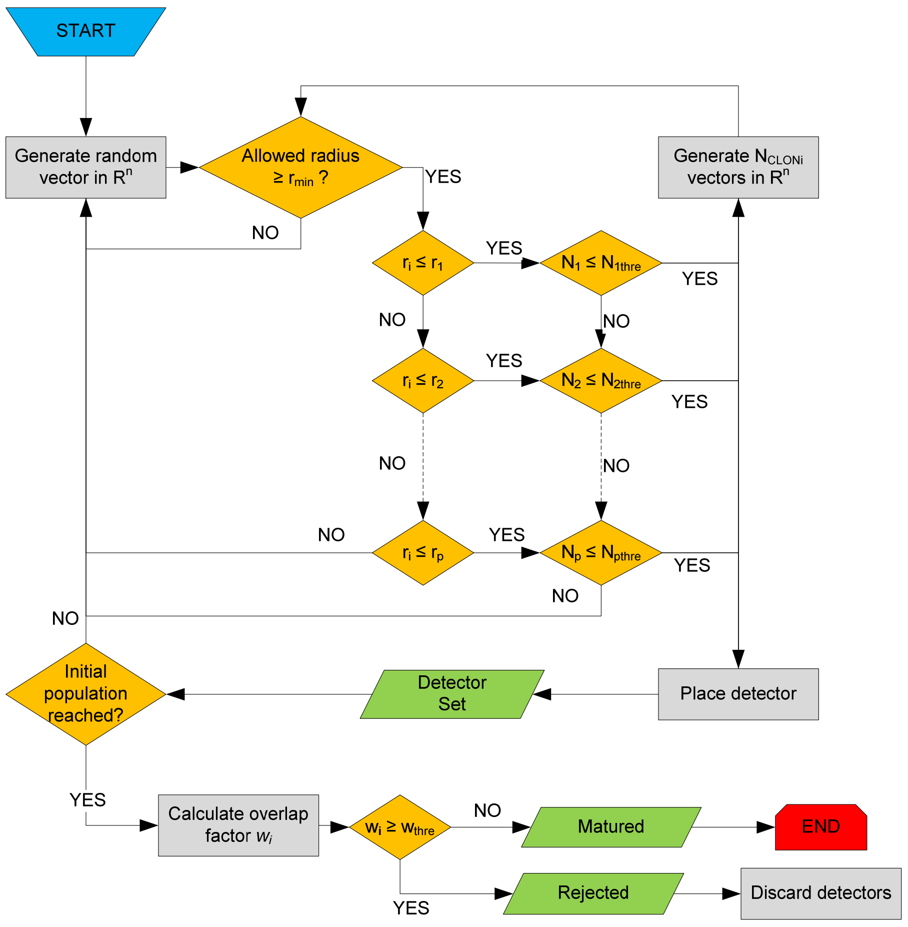


Figure 7: Algorithm #2

The computational costs of algorithm #2 and ENSA-RV are considerably different. For small populations (below 1000 detectors), algorithm #2 is considerably faster; however, if the population is increased, both algorithms become extremely slow. Also, for a population of less than 1000 detectors, the overlapping experienced by the detectors created using Algorithm #2 generates a set with poor coverage of the non-self space.

Finally, as an enhancement tool for any of the two above mentioned algorithms, a positive selection approach is proposed. This method uses failed data to create specialized detectors to perceive a particular failure with 100% detection rate and remove detectors to reduce the false alarms to 0%. This method is used in [75] in a more extensive way changing the size of both clusters and detectors to perform similar tasks. In this case, the algorithm used can be outlined as:

1. Chose a small subset of failed flights to be used for positive selection processes. Use these flights and obtain the detection results point to point, meaning that for each data

point one can obtain one of two situations: one detector is activated or no detectors are activated.

2. Generate specialized detectors at each failed data point not detected.
    a. Generate the biggest detector allowable at the first incoming point that is not detected as faulty.
    b. Rerun the detection.
    c. Go back to point "a" until 100% of failed points are detected as such.
    d. Create a list of these points ordered from biggest to smallest.
    e. Generate the biggest detectors allowed following the order given by point d
3. Eliminate detectors that induce false activations:
    a. Rerun the detection
    b. Remove the first detectors generating false activations
    c. Go back to point a until 0% of false activations is accomplished

Items 2.d and 2.e proved to help reducing the number of detectors needed as in most cases some bigger detectors inserted first could enclose more failed points beforehand. The following flowchart presents this Positive Selection Enhancer (PSE) algorithm.

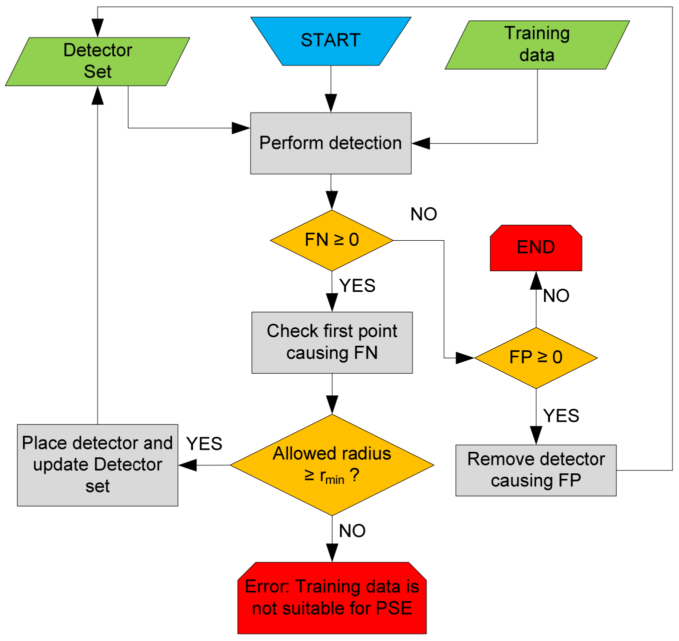

Detector Set
START
Training data
Perform detection
NO
FN ≥ 0
END
YES
NO
Check first point causing FN
FP ≥ 0
YES
Place detector and update Detector set
YES
Allowed radius ≥ r_min ?
Remove detector causing FP
NO
Error: Training data is not suitable for PSE

Figure 8: PSE algorithm.

Even though this algorithm shows to be useful in increasing detection rates, care must be taken if the starting point is too ineffective in detecting the few known failures available for testing and/or training of the scheme. If the original detector set presents very low DR_{PP} for this set of known failures, and it is also found that the failed data points are not located inside the self, a first conclusion is that the coverage of the non-self is unsatisfactory. Moreover, if the PSE is successful in creating detectors to be added to the original set in order to make DR_{PP} reach 100%, it implies that the negative selection algorithm used left unexplored or not properly covered spaces. Even in this case, little can be concluded as to what the behavior of the scheme will be with completely unexpected failures; however, it can be presumed that other unknown upset situations, whose dynamic signature is found in places in the universe comparable to where the known ones are found, may also experience comparable values of DR_{PP}. Detecting unexpected anomalies is the main strength of this technique and so acceptable values of DR_{PP} are desired for known failures before applying the PSE algorithm.

2.2.3.4. Detection, identification, and metrics

Each incoming data instance $\vec{y}$ is monitored versus the entire set of detectors using the definition of distance $d_{\lambda}\left(\vec{y},\vec{c}_h\right)$. The following condition determines activation:

$$d_{\lambda}\left(\vec{y},\vec{c}_h\right) \le r_h \quad \text{for} \quad h=1,\cdots,N_{DET} \tag{2.18}$$

If (2.18) is true for any h, the monitoring stops and the h^{th} detector is said to be activated by this data point. To improve detection speed, the detectors are first ordered from the largest to the smallest. After each activation, the activated detector is moved to the first position of the detector set considering that the following data instance is likely to be located in the vicinity of the previous one.

Outliers can trigger activation; therefore, single activation is not used for detection. To increase the robustness of the detection algorithm, a time window of T_W data points is defined as the time for which information of the activation of a detector is retained. During this time window, the number of activated detectors is added so that if all T_W previous data points activated a detector, there will be T_W active detectors. This concept of simultaneous activation allows the use of a detection threshold D_{thre} that determines what percentage of T_W is needed to be activated to declare a failure. The results show the effect of varying this threshold.

The activation algorithm has four logical outcomes:

- TP - True Positives: abnormal case detected as abnormal
- TN - True Negatives: normal case NOT detected as abnormal
- FP - False Positives: normal case detected as abnormal
- FN - False Negatives: abnormal case NOT detected as abnormal

The following 2D figure shows a simplified schematic for these four situations.

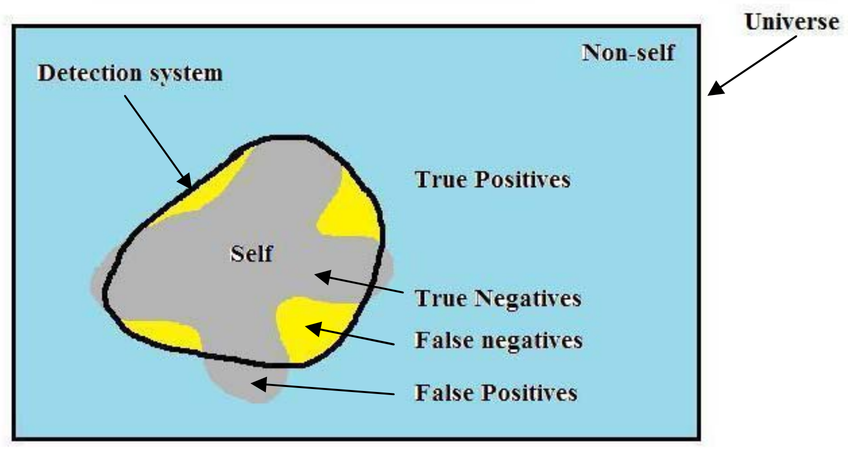


Figure 9: Logical outcomes of FD scheme [54]

For the identification phase, inevitably there is a need for a set of training flights including failures. These flights are used to create a list of the detectors that get activated for each particular failure. The identification only follows a positive detection and when the time comes, it is a matter of comparing the detectors that have been activated versus the detectors contained in the lists. If none of the activated detectors correspond to the lists of known failures, an unknown failure is said to have occurred.

The Point to Point Detection Rate (DR_{PP}) is defined then as the ratio between abnormal data points detected as abnormal divided by the total amount of abnormal data points:

$$DR_{PP} = \frac{TP_P}{TP_P + FN_P} \cdot 100 \tag{2.19}$$

The Point to Point False Alarm rate (FA_{PP}) is defined as the ratio between normal data points detected as abnormal divided by the total amount of normal data points:

$$FA_{PP} = \frac{FP_P}{TN_P + FP_P} \cdot 100 \tag{2.20}$$

Where TP_P, FN_P, TN_P, and FP_P stand for True Positive Points, False Negative Points, True Negative Points and False Positive Points respectively.

The Flight Detection Rate (DR_F) is defined then as the ratio between abnormal flights detected as abnormal divided by the amount of abnormal flights:

$$DR_F = \frac{TP_F}{U_F} \cdot 100 \tag{2.21}$$

The Flight False Alarm rate (FA_F) is defined as the ratio between normal flights detected as abnormal divided by the amount of flights:

$$FA_F = \frac{FP_F}{A_F} \cdot 100 \tag{2.22}$$

Where TP_F, FP_F, U_F, and A_F stand for True Positive Flights, False Positive Flights, Upset Flights, and Available Flights respectively.

The Flight Identification Rate (IR_F) is defined for each particular failure as the ratio between correctly identified flights divided by the total amount of flights:

$$IR_F = \frac{TP_I}{U_F} \cdot 100 \tag{2.23}$$

The Flight False Identification rate (FI_F) is defined for each particular failure as the ratio between incorrectly identified flights divided by the total amount of flights not presenting the particular failure:

$$FI_F = \frac{FP_I}{A_F} \cdot 100 \tag{2.24}$$

Where TP_I and FP_I stand for True Positive Identifications, False Positive Identifications respectively.

Finally, the time to detect (TD) is obtained as the time elapsed from the starting of the abnormal situation until the moment of the first data point for which the detection threshold is met.

Chapter 3: EXPERIMENTAL SETUP

This chapter presents the setup of the flights used in developing the Immunity Based scheme as well as those used for the comparison of FDI approaches. The data collection procedures are also outlined.

3.1. WVU YF-22 Research Aircraft

The aircraft test-bed used to carry out the flight tests involved in the design of the FDI scheme is shown in Figure 10. These aircraft geometrically resembles the USAF YF-22 and considering these similarities, it is named WVU YF-22; however, it is not a scaled version.

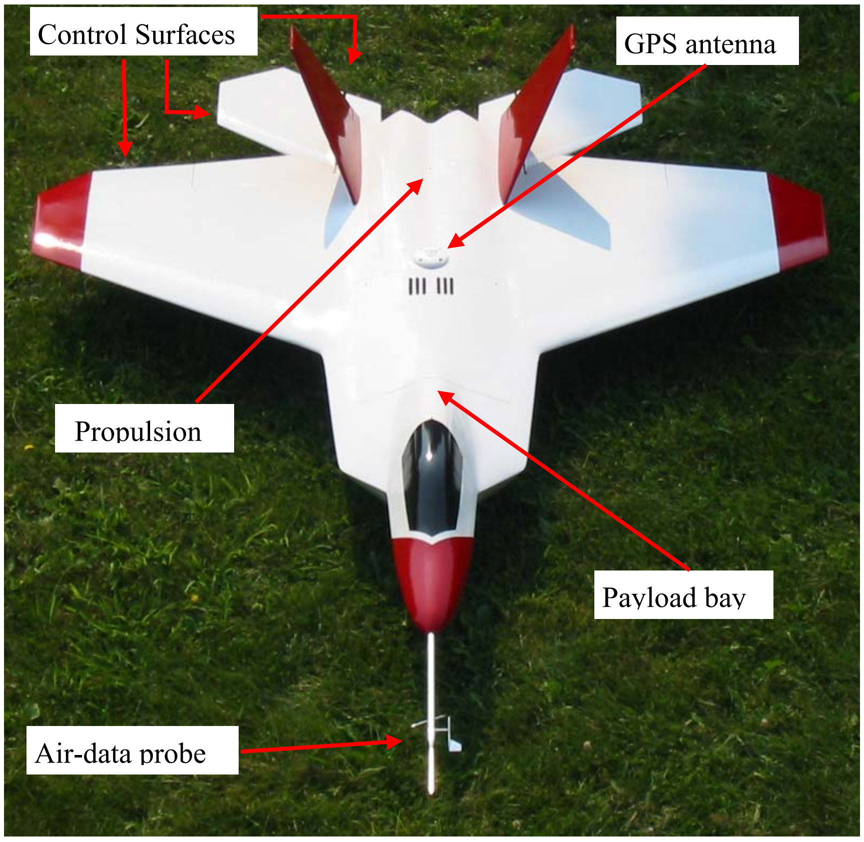


Figure 10: WVU-YF22 used for flight testing – Main features

The following table contains the main specifications of this aircraft and is a partial reproduction of a similar table presented in [76]:

WVU YF-22 Specifications	
Wingspan	1.96 m
Length	3.05 m (including nose probe)
Height	0.61 m
Wing Area	1.37 m^2
Take-off Weight	24 Kg
Fuel Capability	3.5 L
Maximum flight duration	12 minutes
Cruise Airspeed	42 m/s
Take-off Airspeed	30 m/s
R/C Radio	JR 10X 10 channel SRCM

Table 1: Specifications of the WVU YF-22 aircraft. [76]

The airplane's propulsion system consists of a RAM 1000 miniature turbine engine, whose main features are presented in the following table.

RAM 1000 Micro Turbine Engine Main Features	
Thrust	125 N
Fuel	Jet-A
Control	Electronic Control Unit (ECU) indirectly controls RPM adjusting fuel flow
Max RMP	127,000
Fuel consumption @ max RPM	0.35 L/minutes
Fuel consumption @ cruise cond.	0.15 - 0.30 L/minutes

Table 2: Main features of the miniature turbine engine. [76]

The airframe is mainly manufactured with composite materials, foam, and wood. The Remote Control (R/C) system used is based on a JR Pulse Code Modulation (PCM) 10-channel radio package [76]. The pilot has control over the primary control surfaces (stabilators, ailerons, and rudders), secondary control surfaces (flaps), throttle, brakes, and a 'control switch' that allows engaging or disengaging of the autonomous controller.

Electronic payload comprises approximately 5 Kg of the total weight of the plane. This airplane was originally designed to be part of a research effort involving formation flight (with two other matching aircrafts) and this payload was tailored for the specific needs of such experiments. The major components of this payload include the on-board computer (OBC), Data Acquisition card and related sensors, and on-board power supply.

The OBC is based on a PC-104 format computer system with the following main features:

OBC Main Features	
CPU module	MSI-CM588
Processor	NS GX1 (300 MHz)
RAM	128 Mb
Data acquisition module	Diamond-MM-32-AT
Power supply/communication module	Jupiter-MM-SIO

Table 3: Main features of the OBC. [76]

The OBC contains also an IDE compact flash adapter and a set of custom developed controller board and sensor interface boards. The data acquisition system collects flight data both for the flight controller and for offline data analysis. The data comes from a suite of sensors including:

Sensor	Measured variables	Range
SpaceAge Mini Air Data Boom	α, β	$(-25, 25)$ deg
SenSym pressure sensors (connected to the SpaceAge Mini Air Data Boom)	P_A	$(0,103.5)$ KPa
	P_D	$(0, 6.9)$ KPa
Crossbow IMU400 Inertia Measurement Unit (IMU)	A_x, A_y, A_z	$(-10,10)$ g
	p, q, r	$(-200, 200)$ deg/second
Goodrich VG34 vertical gyro	ϕ	$(-90, 90)$ deg
	θ	$(-60, 60)$ deg
Potentiometers attached to control surfaces	δ_{R_R}, δ_{R_L}, δ_{S_R}, δ_{S_L}, δ_{A_R}, δ_{A_L}	$(-15,15)$ deg
Thermistor	T	$(-10, 70)$ deg C
Novatel OEM4 GPS receiver	X_{GPS}, Y_{GPS}, Z_{GPS}	N/A
	V_x, V_y, V_z	N/A

Table 4: Sensors, variables measured, and ranges.

A total of 22 analog channels are measured using 16-bit resolution at a rate of 50 Hz except for the GPS data which is acquired at a rate of 20 Hz.

Six battery packs are used as power supply for all the electronic equipment on board:

- Four 4.8 V, 1600mA NiMh packs for the R/C system (2 for receiver and 2 for servos).
- A 7.2V, 1250mA NiCd pack for the ECU and the starter.
- A 14.8V, 3300mAh Li-Poly battery pack to operate the OBC.

A study of Electro-Magnetic Interference (EMI) among the components was carried out as explained in [76] and the current positioning of the elements was selected in order to reduce EMI effects as well as extra considerations as signal cables shielding. Some extra EMI reduction was accomplished using ferrite RF chokes on the external wiring.

The control system is designed to operate in three modes:

- Manual (M): Pilot has full control over all R/C channels
- Partial Autonomous (PA): The pilot has control over a subset of R/C channels while the on-board controller directs the remaining ones.
- Full Autonomous (FuA): The on board controller has full control over all R/C channels.

The pilot has the choice to switch (using the so called ‘control switch’) to Manual mode at any point during a flight, thus allowing this mode to be used as a safety mode to avoid dangerous situations following failures or extreme maneuvers performed by the OBES. Take off and landing of this aircraft is always performed using this mode.

3.2. Data Collection Procedures

The flight data used within this research effort was obtained as part of a comprehensive flight test program designed to obtain a new non-linear model for the WVU YF-22 aircraft [77]. They were not specifically designed for the development of the FDI scheme which produced additional constraints on the FDI scheme design process. These tests were performed at the WVU Jackson’s Mill facility located approximately 60 miles south of WVU campus in Morgantown WV. This facility features a paved runway allowing the operation of this type of aircrafts and it can be closed to public to make it a safe experimental setting for flight testing.

All flights were performed within close visual range, on an attempted trajectory consisting of two semi-circles connected by two straight legs. All flights consist of the following stages:

1. Manual take-off.
2. 1 lap following the desired path used for trimming purposes and to gain altitude.
3. A series of between 5 or 6 laps following the desired path trying to maintain constant altitude throughout the flight.
4. Manual landing.

The tests consisted in performing standard maneuvers aimed to excite the different airplane dynamic modes. The maneuvers implemented in the tests are:

- Stabilator doublet
- Aileron doublet
- Rudder doublet immediately followed by an aileron doublet (referred to as rudder-aileron combo or combo maneuver).

In all cases, the maneuver is performed midway between the two ends of the path, in the middle of the straight legs at a maximum rate of two per lap. This indicates that ideally the plane is in level flight before the maneuver is injected. Approximately one second before introducing any of these maneuvers, a control signal engaged the PA mode thus positioning all surfaces in trim position, allowing only movements of the ones strictly involved in the maneuver (i.e. stabilators for stabilator doublet). After the maneuver is performed, all surfaces stayed at trim for a "waiting time" between 3 and 5 seconds to allow the plane to exhibit the different dynamic modes. After this waiting time, the PA mode is disengaged. Figure 3 shows a top view of the flight path and the place where the maneuvers/failures were injected.

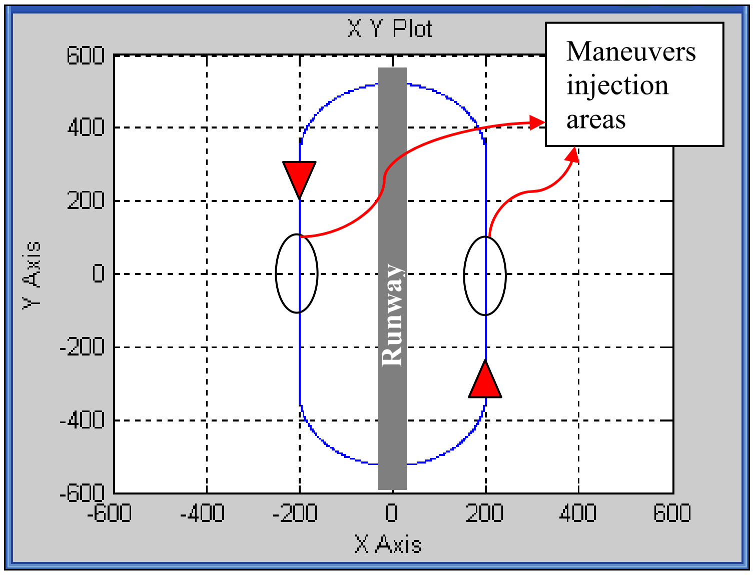


Figure 11: Idealized flight path with failure/maneuver injection areas.

The two red triangles represent the airplane in two positions in the flight path and the apex of the triangles points in the direction of flight. As can be induced from the figure, the path was flown counter-clockwise.

Two types of failures are implemented:

- Left stabilator locked at trim
- Left aileron locked at trim

The failures were implemented only on the left side of the airplane to favor security so that the injected maneuvers would produce rolling effects towards the pilot (inward of the path). This decision is not considered to produce any loss of generality.

As was described earlier, one second before injecting the maneuvers, all surfaces are sent to trim position. The same applies in the case of failed flights, with the only difference being that instead of moving the surfaces involved in the maneuver on both sides of the plane, only one side is moved while the other one remains at trim position.

Two general setups were used during this flight season considering who is in control of the airplane throughout the main part of the laps (item 3 on the description of the flight path):

- Pilot-in-the-loop flights: The pilot controls the aircraft to follow the desired flight path throughout the laps only switching to the PA mode before manually injecting the maneuver. After the waiting time extinguishes, the pilot switches back to Manual mode to regain full control and continue the lap.
- Autonomous flights: The plane is set on FuA mode so that the on board controller follows the desired path. The on board controller is programmed to switch to PA mode before injecting a prerecorded pilot-in-the-loop maneuver in the desired location during the lap. After the waiting time is extinguished, the on board controller switches back to FuA mode to regain full control and continue the lap.

Autonomous flights were used to try to improve the repeatability of the tests. As can be inferred from the explanation above, in both cases the maneuvers injected consisted of pilot-in-the-loop movements. To avoid confusion, the chain of events following the engagement of the PA mode by the on-board controller during autonomous flights is emphasized:

1. The controller is immediately switched off.
2. All the surfaces are set to their respective trim condition.
3. After about a second (enough time to ensure that the above point has been accomplished), a prerecorded maneuver is injected.
4. No actions are performed during the waiting time of 3 to 5 seconds.
5. The controller is turned back on and the mode changed back to FuA.

The available data can then be separated considering who is in full control of the airplane at each moment in time and if there are failures present or not. The following table summarizes the partition of the data available:

Type	Condition	Pilot	Controller	Short description
1	Healthy	Yes	Off	Pilot flying the plane using M mode or using PA mode and injecting maneuvers without failures.
2	Failed	Yes	Off	Pilot manually injecting a maneuver using the PA mode including a failure.
3	Healthy	No	Off	OBES injecting a prerecorded maneuver using the PA mode.
4	Failed	No	Off	OBES injecting a prerecorded maneuver using the PA mode including a failure.
5	Healthy	No	On	On-board controller trying to follow the desired path

Table 5: Breakdown of available data from the 2008 flight season.

Data types 1 and 3 are comparable because no controller is present, and are composed of either just a real pilot input (Type 1) or a prerecorded pilot input (Type 3). The same consideration is applicable for data types 2 and 4. The detector generation as is outlined later is performed using data types 1 and 3. The detection tests are performed using data type 2 and 4.

3.2.1. 2008 flight tests

The 2008 Flight season consisted essentially of about 15 successful flights designed to fulfill the needs of the development of the non linear model. The following is a partial reproduction of a similar table found in [77] showing a summary of the flights performed.

2008 Flight Season			
Flight Date	**Flight #**	**Flight Description**	**Data types available**
5/01/2008-09/01/2008	N/A	Preliminary Test Flights (systems evaluation & software testing)	N/A
9/16/2008	1	Pilot-Injected Elevator Doublet and Rudder-Aileron Combo Doublet; Pilot- Injected Elevator Doublets with Left Elevator Failure at Trim	1, 2
	2	Pilot-Injected Elevator Doublet and Aileron Doublet; Pilot-Injected Aileron Doublets with Left Aileron Failure at Trim	1, 2
10/11/2008	1	Preliminary Virtual Leader Test Flight	1, 5
	2	OBES-Injected Elevator Doublets	1, 3, 5
10/18/2008	1	OBES-Injected Rudder/Aileron Combination Doublets	1, 3, 5
	2	OBES-Injected Elevator Doublets	1, 3, 5
11/01/2008	1	OBES-Injected Elevator Doublets with Left Elevator Failure at Trim	1, 4, 5
11/04/2008	1	OBES-Injected Rudder/Aileron Doublet Combinations with Left Aileron Failure at Trim	1, 4, 5
	2	OBES-Injected Aileron Doublet Combinations with Left Aileron Failure at Trim	1, 4, 5

Table 6: Summary of 2008 Flight Season. [77]

The first row takes account of all the preliminary flights that were used to tune up the on-board data acquisition, on-board controller, on-board excitation system, general flight testing procedures and data reduction techniques. All these tasks were necessary for the successful completion of the remaining flights; however, the data recorded during these sessions lack significance as compared to the rest considering that changes were introduced to the plane during these early stages. As it can be noticed, all flights provide data type 1 as at least the first lap and the final lap before setting up for landing were performed by the pilot using Manual mode.

3.2.2. Flight data

For this implementation, only data from the 9 flights described in the table presented in the previous section are used. The plane operates the large majority of the time at a relatively constant point of its flight envelope, meaning that throughout the laps both the altitude and velocity are almost constant.

This implementation deals with a single point in the airplane flight envelope and thus, the useful data from each flight is extracted as data close to the desired conditions. GPS data for velocity and altitude are used to determine when this point in the flight envelope has been reached.

Once the useful data is extracted from every flight available, the next step consists of isolating each data type to allow its use separately. Data type 1 is the easiest to isolate as all useful data during a flight whenever the 'control switch' is OFF corresponds to this category in addition to the pilot-in-the-loop nominal maneuvers where the 'control switch' is set to ON engaging the PA mode. Data type 2 corresponds to the portions of data in pilot-in-the-loop flights that include maneuvers including failures where the pilot engaged the PA mode by switching the 'control switch' ON. The following figure shows a section of a pilot-in-the-loop flight including an aileron doublet with left aileron locked at trim.

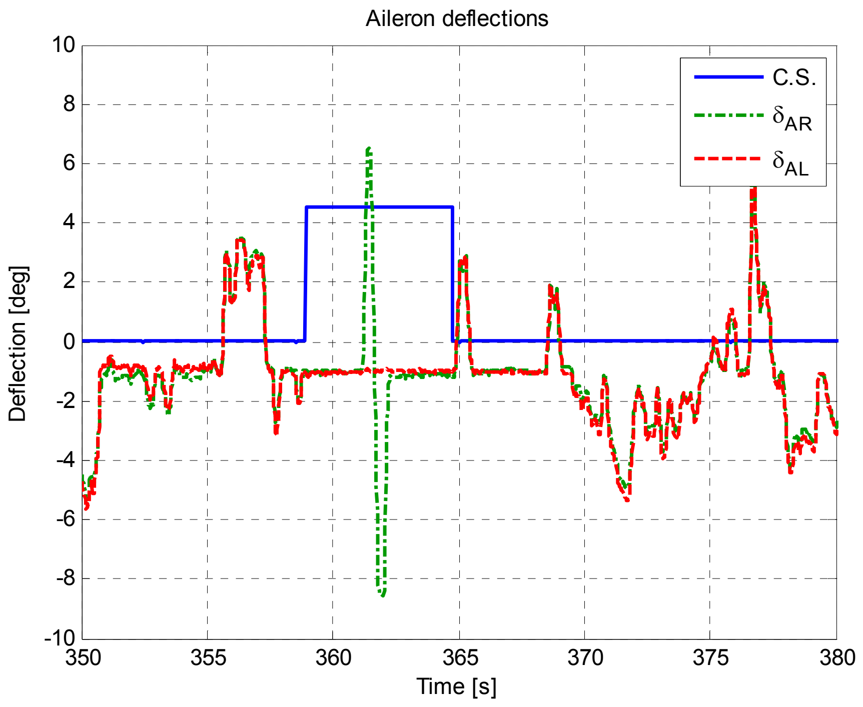


Figure 12: Example of data types 1 and 2.

It is important to clarify that even though the ailerons on both sides show identical measured values, the deflection was opposite for each side as required for rolling purposes. The plot shows the deflections of the two sides and the values of the 'control switch' signal which fluctuates from 0 to 4 (4 is scaling factor chosen to make the plot clear) depending if the PA mode is disengaged or engaged, respectively. All data when the control signal is 0 are data type 1 and for data isolation purposes, all data when the control signal is 4 are assumed as data type 2. Nonetheless, all the data contained in the region where the control signal is 4 do not fall strictly into data type 2. If all surfaces are set to trim and no pilot input is registered, the plane is essentially in a healthy condition yet whenever the maneuver is injected, the plane is said to be failed. This more detailed consideration of the data is used only for detection tests to account for detection rates and false alarms as is explained in following chapters.

Following with the data isolation, data types 3, 4, and 5 need to be separated in a similar fashion as was explained for data types 1 and 2. In this case, the 'control switch' signal cannot be used as a criterion as the control signal remained ON from the moment the pilot engaged the autonomous flight after the first lap, until he disengaged the autonomous flight when getting ready to land the plane. The following figure shows a section of an autonomous flight including a stabilator doublet with left stabilator locked at trim.

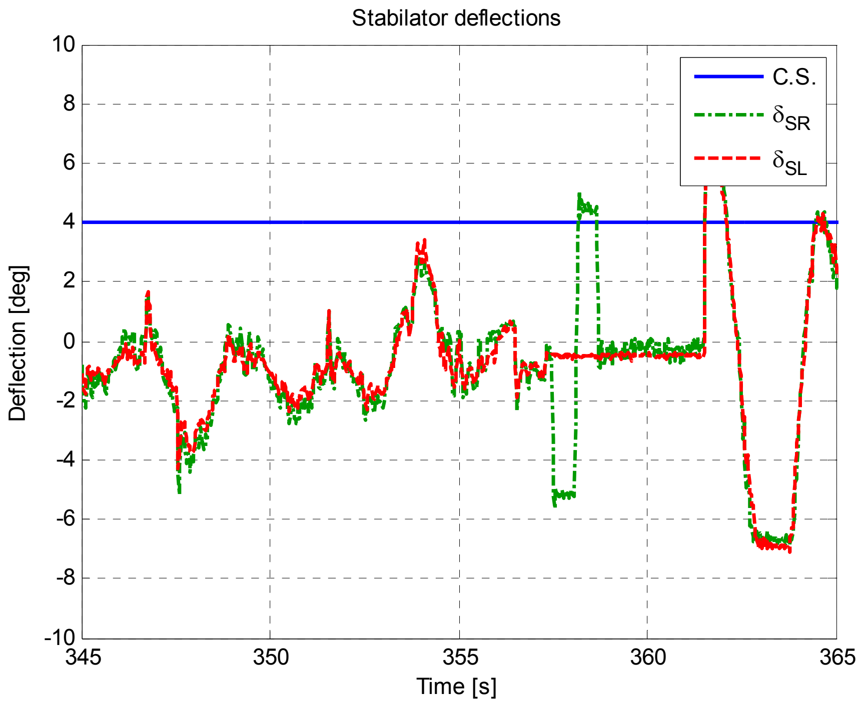


Figure 13: Example of data types 4 and 5.

As is clearly seen, the control signal is always 4 throughout this section of the flight. Each flight and each maneuver was verified and analyzed to obtain the limits between data types 3, 4 and 5. In the particular example shown, data type 5 is assumed from the beginning until 356.8 seconds where the surfaces can be assumed to be at trim position (due to the recent engagement of the PA mode). From 356.8 till 361.4 the surfaces are still at trim (or performing the maneuver) and, for data isolation, this section is assumed to be data type 4. Finally, everything after 361.4 seconds is considered data type 5 again. It is important to note that after the end of the maneuver and the waiting time, the airplane experiences some extreme maneuvers as the controller regains full authority over the plane and tries to take it back to the desired flight path. Data type 3 would correspond to a similar analysis but considering a healthy maneuver.

As was previously detailed, the FDI design is based on pilot-in-the-loop data. Consequently, all data types 1 and 3 that were isolated in the previous steps are considered towards the creation of the self, while data types 2 and 4 as described above are used for evaluation of the FDI scheme. The following figure shows the three standard maneuvers used both in nominal and failed flights. The values are presented after a normalization process that is outlined in the previous chapter.

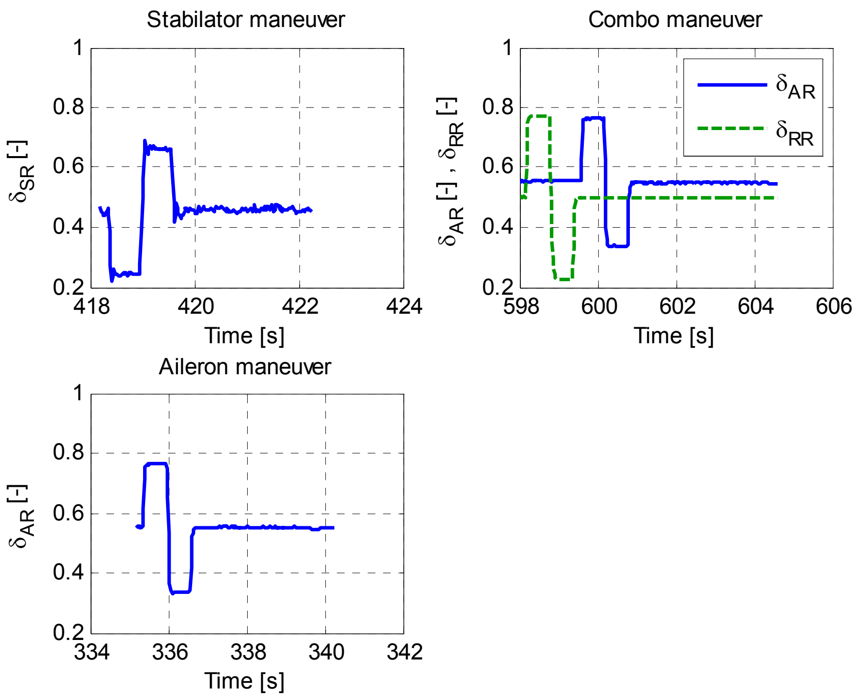


Figure 14: Examples of standard maneuvers

The data in these maneuvers contain healthy data whenever no pilot input is being sent to the control surfaces. The following table summarizes the failures that are used in testing the FDI and includes for each of them the following information:

- “Start” and “End”: The start and end time for the maneuver comprising the delay before performing the maneuver, the maneuver itself and the waiting time.
- “F. Start” and “F. End”: The start and end for the maneuver itself that is the only portion where the failure is present as such

ID #	Maneuver	Start [s]	End [s]	F. Start [s]	F. End [s]
1	Stabilator	357.38	361.46	357.50	358.70
2	Stabilator	387.46	391.80	387.82	389.02
3	Stabilator	418.20	422.24	418.38	419.58
4	Stabilator	448.58	452.50	448.70	449.88
5	Stabilator	478.88	483.10	479.26	480.46
6	Stabilator	509.26	513.50	509.58	510.78
7	Stabilator	540.00	544.00	540.14	541.34
8	R-A Combo	537.14	543.60	538.70	539.90
9	R-A Combo	567.40	574.00	569.02	570.24
10	R-A Combo	598.00	604.60	599.58	600.80
11	R-A Combo	628.34	634.90	629.88	631.10
12	R-A Combo	658.60	665.50	660.46	661.68
13	Aileron + Bias	335.20	340.26	335.20	340.26
14	Aileron + Bias	365.56	370.62	365.56	370.62
15	Aileron + Bias	396.10	401.20	396.10	401.20
16	Aileron + Bias	426.44	431.50	426.44	431.50
17	Aileron + Bias	457.00	462.10	457.00	462.10
18	Aileron	335.20	340.26	335.36	336.62
19	Aileron	365.56	370.62	365.68	366.92
20	Aileron	396.10	401.20	396.24	397.48
21	Aileron	426.44	431.50	426.58	427.80
22	Aileron	457.00	462.10	457.12	458.34

Table 7: Failed maneuvers used for testing the FDI

Maneuvers 13 to 17 correspond to flight #2 performed on November 4 of 2008, which due to an error on the configuration of the data acquisition system, recorded all the data with a considerable constant bias. Maneuvers 18 to 22 correspond to the previous maneuvers after removing this constant bias. The biased flights are used to test detection capabilities of the scheme for real sensor failure. Types 1 and 5 portions of data of the flight mentioned above are used in the development of the FDI to avoid unreliable measures to be introduced in the creation of the Self.

Approximately 600 seconds worth of data valid for healthy operation of the airplane including nominal standard maneuvers are used for the purposes of this paper. The data at upset flight conditions cover a total of 100 seconds. This considerable difference in the available data is produced because the failures were only active for a short period of time, while the plane was mostly flying in healthy conditions.

It is important to mention that the same pilot performed the totality of the flights that are used for this implementation. Also significant to point out is that due to his own flying technique, the pilot rarely uses rudder input to perform any regular maneuver unless the flight test scenario required such an input. This has consequences that are explained in Chapter 5.

3.3. 6 DOF Flight Simulator and WVU IFCS F-15 Model

The aircraft aerodynamic model implemented in this simulator was derived from the non-linear Fortran code based model of a high performance military aircraft distributed by NASA to academic institutions within the 1990 AIAA GNC Design Challenge [78]. WVU's model is a version of the mentioned code with the same functionalities but implemented on a Matlab/Simulink environment. The aerodynamic and thrust characteristics are provided via a set of 42 look-up tables. WVU model also incorporates the aerodynamic model for canard surfaces to resemble the NASA F-15 research aircraft [79]. The aerodynamic look-up tables have been split to obtain the contributions of left and right surfaces.

This Simulink simulation has been interfaced with a Motus 622i motion-based, 6 Degrees-Of-Freedom (6DOF) simulator manufactured by Fidelity Flight Simulation, Inc. The interface created uses a computer (called Research Computer - RC) to run the Matlab/Simulink model and employs S-functions that incorporate User Datagram Protocol (UDP) code to open a communication port and send the necessary data to the Server Computer (SC) that is the main computer of the 6DOF simulator. The data are split and sent via two ports. One port communicates directly with the proprietary Fidelity software called Translator overwriting the motion base related variables to produce the corresponding movements (motion cues), while the other port communicates indirectly with X-plane to provide the visual cues. The visual cues, in fact, communicate with a plugin tailored for this particular application that overwrites the corresponding variables in X-plane [80].

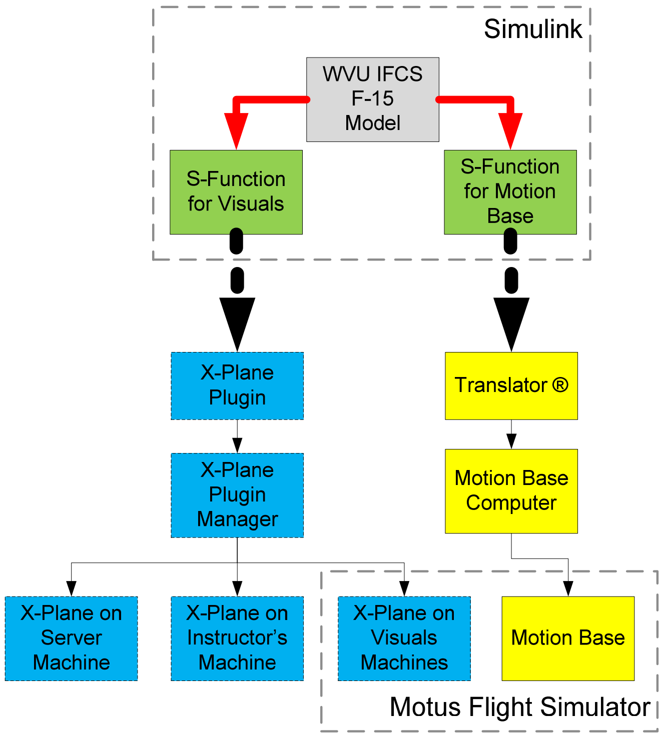
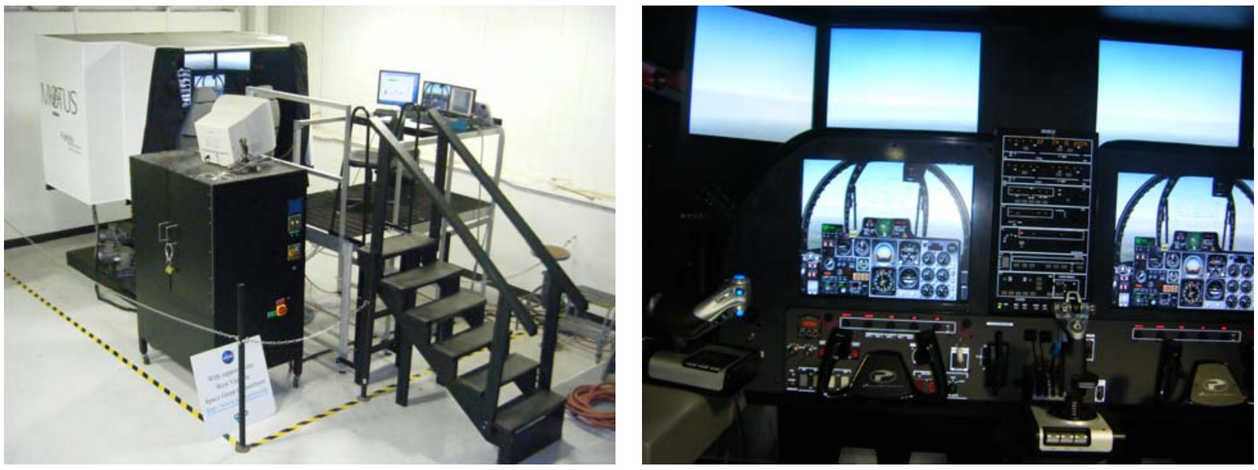


Figure 15: Schematic of Simulink-6DOF simulator interface.

The pilot is provided with a dual flight command consisting of a 3 axis joystick for controlling the stabilators, ailerons, and rudders, and a separate module for the accelerator. Some special accommodation features were included the first time this interface was implemented in [80] to include an external joystick inside the cockpit. The following are some pictures of the simulator.



a) b)

Figure 16: a) Exterior view of simulator; b) View from the pilot seat showing joystick arrangements

3.3.1. Flight tests for comparison of FDI approaches

This Simulink model includes a GUI designed to allow the selection of the different flight scenarios: including failures or not, using different NN inside the controller, piloted flights or recorder flight, etc. For the particular purpose of the tests designed, a separate set of blocks were implemented in Matlab to allow manual insertion of the failure.

The objective of this section is to show preliminary results of comparison between two different approaches to FDI using the same simulated environment and obtain metrics that will assess the pros and cons of each. This is accomplished performing simple short meaningful tests and comparing the detection results of both schemes regarding:

- Flights positive detections versus false alarms considering also the effects of:
  - Maneuvers performed before or after injection of failure
  - Particular piloting skills
  - Particularities of each method for results with determined failures
- Time to detect
- Computational load

The designed flight situation consists of a set of maneuvers performed at constant altitude and mach number (single point of the flight envelope, h=20000 ft ; M = 0.75). The flight scenario consists of:

1. 45° coordinated turn to the left with a bank angle of 20°.
2. Stabilator doublet
3. Aileron doublet
4. Rudder doublet
5. 45° coordinated turn to the left
6. Includes waiting time before and after each maneuver consisting on level flight for 10 seconds.

Two intensities of failure are considered for actuators and sensors:

- Actuator stuck in determined angle (stabilator, aileron, rudder):
  - 5° from trim position
  - 8° from trim position
- Sensor failures (pitch rate, roll rate):

    - Step bias of 5 deg/sec
    - Saturation of sensor via ramp of 8 seconds and saturation value of 10 deg/sec
- Sensor failures (yaw): Reduced to avoid unsafe behavior of the 6 DOF simulator
    - Step bias of (5/3) deg/sec for yaw rate sensor
    - Saturation of sensor via ramp of 8 seconds and saturation value of (10/3) deg/sec for yaw rate sensor

Different failure injection points are considered to understand the effect of the moment of a failure occurrence in the detection process.

- 9 different points were selected corresponding to:
    - 5 points are considered for failure injection during each of the maneuvers
    - 4 points are considered for failure injection in between each of the 5 maneuvers.

All this analysis concludes in the need for 113 flights to complete this research; however, as was stated earlier in this subsection, the analysis presented in this thesis does not intend to be comprehensive. Much on the contrary, the purpose of these sections is to show an interesting line of research for comparing FDI based on completely different schemes. A preliminary analysis is also performed to assess the benefits of having a detection scheme to aid the pilot cope with upset conditions.

3.3.2. NN based FDI including the concept of floating limiter

The FDII scheme as described in [80] is divided into three main steps:

1. Detection: An unspecified abnormal condition is detected
2. Isolation: The detected condition is classified as a sensor or actuator upset situation.
3. Identification: The isolated upset situation is recognized to be:
    - Actuator isolation: stabilator, aileron, or rudder failure.
    - Sensor isolation: roll, pitch or yaw gyro failure.

The FDII scheme is composed of two different sets of NNs with different characteristics. The Main Neural Networks (MNNs) monitor output sensors of the airplane model for a previous time window $k-l$ to generate individual estimates of the three angular rates (one MNN for each channel) for time step k. The Decentralized Neural Networks (DNNs) have an equivalent structure to the MNNs except that the sensors that are used for the estimation do not include the measured value of the particular gyro related to the estimation. The sensors used by these NNs include angle of attack and sideslip, deflections of all 6 surfaces, linear acceleration in the y axis and 3 axis gyro. Both NN sets are composed of a pretrained NN and an online learning NN using an ADALINE and an A+EMRAN Neural Networks [26].

The outputs of these NNs sets are used to calculate a set of cross-correlation functions involving also the angular rates as measured directly from the sensors. Perhinschi et al. make clear that this FDII scheme [26] relies on the basic assumption "*that the occurrence of failures on primary control surfaces can be monitored through the analysis of the aircraft angular rates alone*". The parameters used for this implementation are the following:

- Mean Quadratic Estimation Error (MQEE):

$$\begin{aligned} MQEE(k) = \frac{1}{2}\Big[& \left(p(k) - \hat{p}_{MNN}(k)\right)^2 + \left(q(k) - \hat{q}_{MNN}(k)\right)^2 + \\ & + \left(r(k) - \hat{r}_{MNN}(k)\right)^2 \Big] \end{aligned} \tag{3.1}$$

- Output Quadratic Estimation Error:

$$\begin{aligned} OQEE(k) = \frac{1}{2}\cdot\Big[& \left(\hat{p}_{DNN}(k) - \hat{p}_{MNN}(k)\right)^2 + \\ & + \left(\hat{q}_{DNN}(k) - \hat{q}_{MNN}(k)\right)^2 + \\ & + \left(\hat{r}_{DNN}(k) - \hat{r}_{MNN}(k)\right)^2 \Big] \end{aligned} \tag{3.2}$$

- Autocorrelation parameter of the yaw rate:

$$\overline{R}_{rr}(k) = OQEE(k) + \mu_{rr} \cdot \sum_{i=k-n_R}^{k} R_{rr}(i) \quad ; \quad \mu_{rr} = 0.01 \tag{3.3}$$

- Roll-pitch rate Cross-correlation parameter:

$$\overline{R}_{pq}(k) = \sum_{i=k-n_R}^{k} R_{pq}(i) \tag{3.4}$$

- Angular rate dominance parameter:

$$\omega_{pq}(k) = |\overline{p}| - \mu_{pq} \cdot |\overline{q}| \quad ; \quad \mu_{pq} = 2.2 \tag{3.5}$$

The way this FDII scheme works consists in setting some boundaries for each of these signals that if exceeded changes the value of a flag. This flag activation system is embedded in a logic scheme that accounts for the three steps mentioned before. Initially, this FDI scheme made use of fixed threshold values [27] and thus the logic included waiting times for some parameters in order to allow evolution of slower signals. Perhinschi et al. presented in [26] an evolution of the FDI scheme in which the concept of Floating Limiter was used to create moving boundaries for the signals; however this work was only focused to actuators. In 2008, Sagoo et al. integrated the Floating Limiter to sensor failures as well and preliminary presented it in [81]. The work presented in [80] that constitutes a more complete analysis of this FDII method is used in this analysis as the starting point.

3.3.2.1. Floating limiter concept

The Floating Limiter (FL) concept is applied in [26] in contrast with the fixed threshold to avoid false alarms produced by pilot in the loop interactions with the model. An example of this is that similar values of the cross-correlation parameters can be accomplished by a failure or by pilot maneuvering. The FL bounds can be designed such that can allow large variations at lower rates (ideally associated with pilot inputs) and filter higher rates (associated with different failures). The FL concept as implemented in [26] is based on the following ARMA filter:

$$D(z) = 0.025 \cdot \frac{z^3 + z^2 + z + 1}{z^3 - 0.3 \cdot z^2 - 0.3 \cdot z - 0.3} \tag{3.6}$$

This filter provides a mean average value $\bar{X}$ of the signals using a time window δ of 1 second [26]. The FL threshold is calculated as:

$$X_{(Thd)} = \bar{X} + \beta \cdot \sigma(X) + b \tag{3.7}$$

Where β is a bound factor, $\sigma(X)$ is the standard deviation of the signal in the same considered time window δ, and b is a bias. All these parameters as well as the filter itself admit changes in order to make the FL more or less sensitive. For the implementation in [80] only an upper bound is used ($b > 0$). The following figure illustrates the concept of the FL as implemented in this detection scheme.

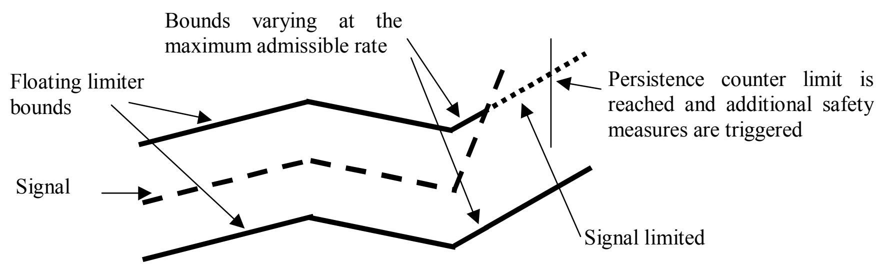

Bounds varying at the maximum admissible rate
Floating limiter bounds
Persistence counter limit is reached and additional safety measures are triggered
Signal
Signal limited

Figure 17: Scheme of floating limiter concept as floating threshold. [26]

3.3.2.2. FDII logic

The logic embedded in this detection scheme is presented in the following flowchart. The subscript SUL stands for Soft Upper Limit.

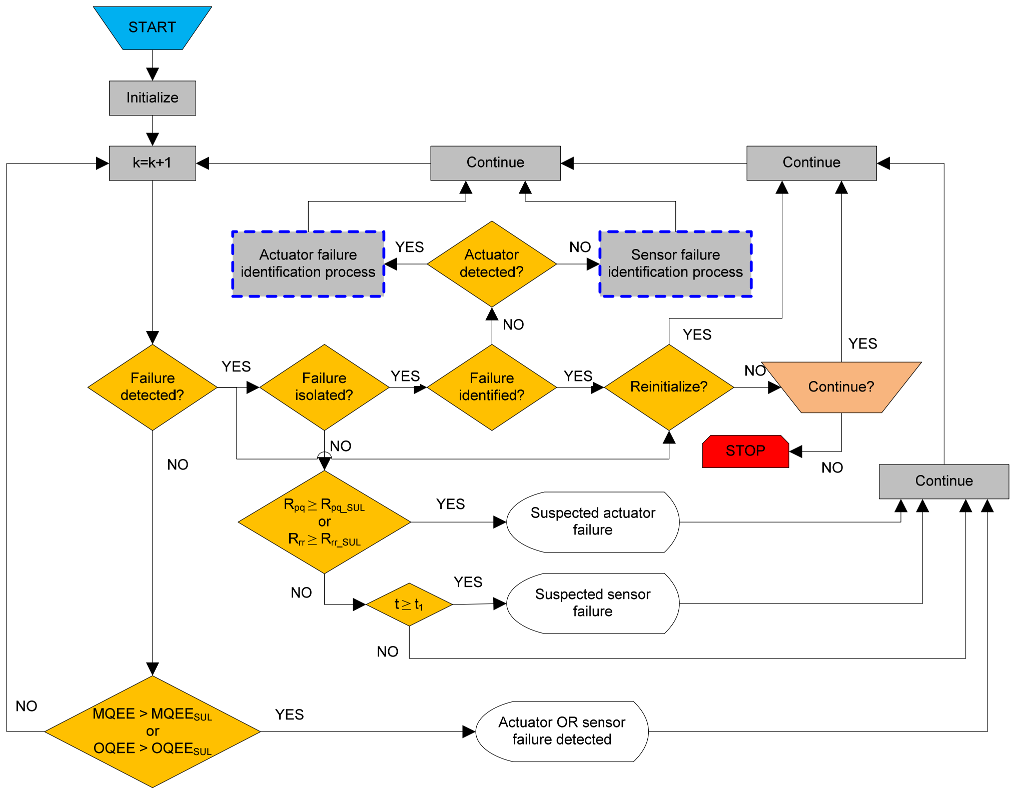

START
Initialize
k=k+1
Continue
Continue
Actuator failure identification process
YES
Actuator detected?
NO
Sensor failure identification process
NO
YES
YES
Failure detected?
YES
Failure isolated?
YES
Failure identified?
YES
Reinitialize?
NO
Continue?
NO
STOP
NO
NO
Continue
$R_{pq} \geq R_{pq\_SUL}$ or $R_{rr} \geq R_{rr\_SUL}$
YES
Suspected actuator failure
NO
$t \geq t_1$
YES
Suspected sensor failure
NO
NO
$MQEE > MQEE_{SUL}$ or $OQEE > OQEE_{SUL}$
YES
Actuator OR sensor failure detected

Figure 18: NN-FL scheme FDII logic – General scheme

The particular logic for actuator and sensor failures identification is outlined in the following two flow charts.

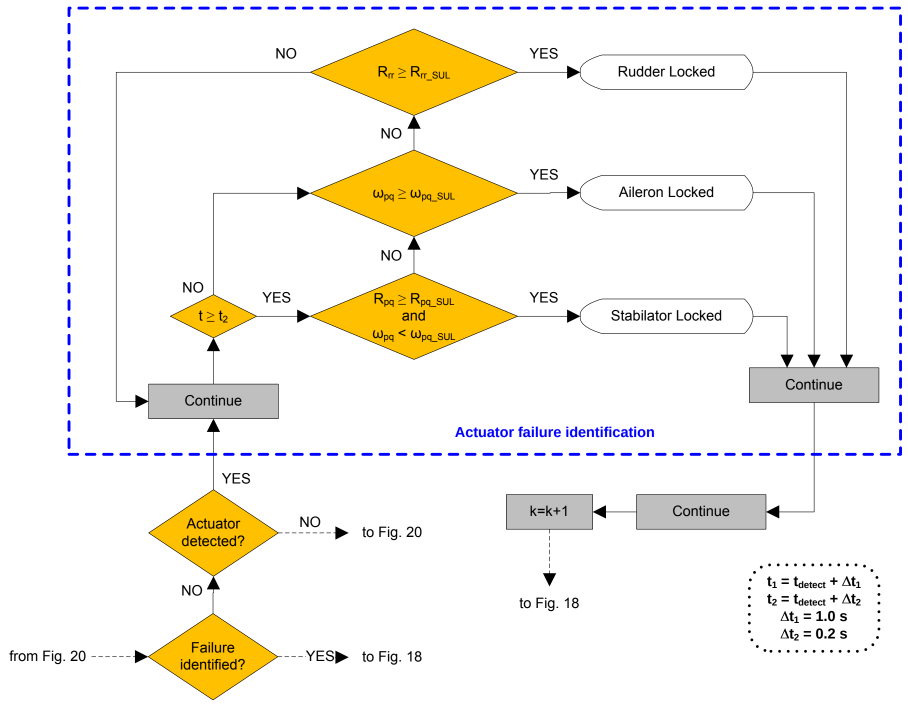


Figure 19: NN-FL scheme FDII logic – Actuator failures

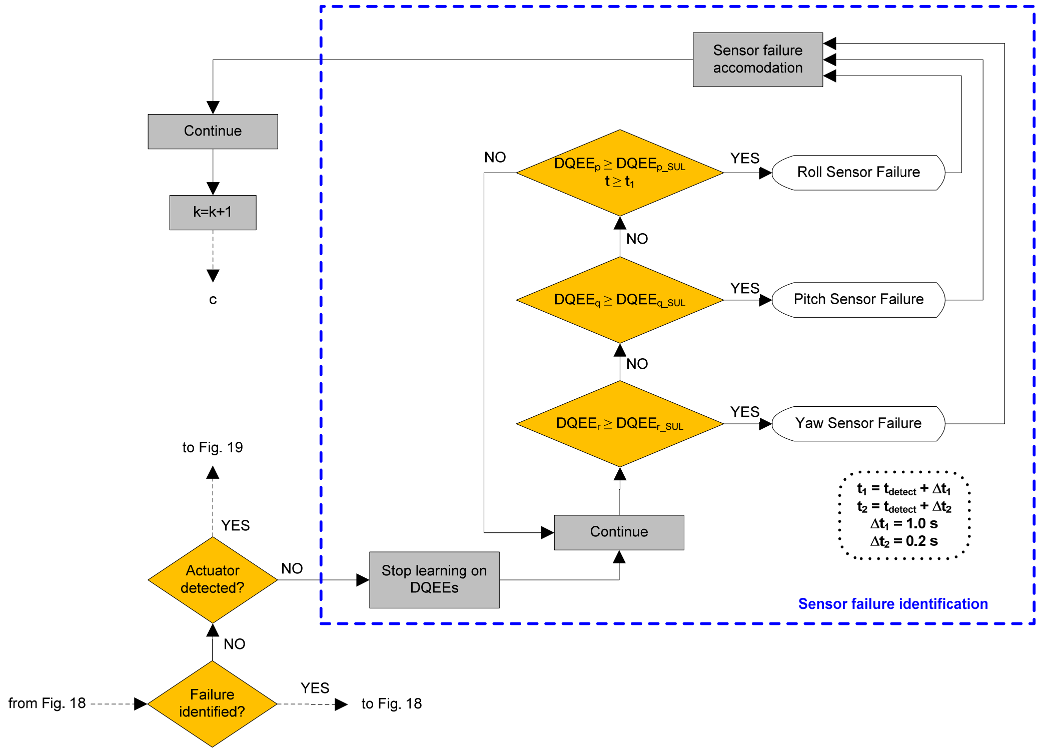


Figure 20: NN-FL scheme FDII logic – Sensor failure

3.3.2.3. FL based FDII scheme parameter tuning

Sagoo in [80] performed the tuning of the FDII system for a simplified flight scenario that consisted basically of a short segment of level flight with pilot in the loop. The inclusion of maneuvers in the present tests requires an intensive retuning of the parameters involved in the FL threshold calculation. This tuning needs to be performed with a complete set of the available failures and maybe using the totality of the flights. This is highly impractical as no auto-regulating tool is yet developed to modify the FL parameters.

The developed AIS based FDI scheme for the WVU YF-22 deals only with stabilator and aileron failures and thus, the FL parameters are modified to deal with these two particular failures. A reduced set of data is used to try to obtain 0% of False Alarms and 100% of Detection Rate and correct Identification. The tuning process is somewhat iterative. The logic order to perform the tuning is the following:

- Adjust the thresholds for $MQEE$ and $OQEE$: These are the variables that declare that a failure is occurring thus triggering the rest of the logic presented.
- Adjust the thresholds for R_{pq} and R_{rr} so that all the stabilator and aileron failures are successfully detected as actuator failures.
- Adjust the threshold for ω_{pq} to avoid misidentifications between aileron and stabilator failures. This step may require also readjustments of the thresholds for R_{pq} and R_{rr} and also the waiting times Δt_1 and Δt_2.

This process may also include slight modifications of the logic embedded for detection if no combination of parameters is found to comply with the training data for the logic given.

3.3.3. FDI using negative selection approach

Perhinschi et al. presented in [68] the framework for creating a comprehensive FDI scheme for airplanes using the Artificial Immune System paradigm. The results presented in the mentioned paper were obtained for the same WVU IFCS F-15 simulation described for this comparison. That implementation made use of many of the variables used for the FL FDI scheme however the use given to the same data is absolutely different.

Hyper-spheres defined using Euclidean distance are chosen to represent the self/non-self space thus used as matching rules. Exhaustive motion-based flight tests were designed and performed in order to obtain the self database. Normalization and clustering were performed using the methods already described in this theses; however due to the amount of information included in the self database, the preprocessing included an extra step consisting in eliminating data points that were closer than a determined threshold. Detector generation was performed using the ENSA-RV algorithm

For this comparison, a self obtained in [68] is used to design an FDI scheme of the same type as the one designed for the YF-22. The identifiers were chosen to be as similar as possible, using the same number of dimensions and the same codes to generate antibodies. In this case and to allow fairness with the FL FDI scheme, a group of flights is selected to train both schemes. The Identification phase as a whole is based on the same concept applied for the WVU YF-22s consisting of creating lists of specialized detectors for each particular known failure.

3.3.3.1. Flight tests used to design the FDI scheme

Flight tests were designed to cover a wide area of the flight envelope; however for the work presented in [72] only a subset of such flights was used. The flight tests applied for the creation of the FDI scheme consisted in:

- Level flight at h=20000 ft and M = 0.75
- Climb at constant Mach to reach h=31000 ft and M = 0.75
- Accelerate at constant altitude to reach h=31000 ft and M = 0.90
- Decelerate at constant altitude to reach h=31000 ft and M = 0.75
- Descend at constant mach to reach h=20000 ft and M = 0.75

The flights lasted between 10 and 20 minutes each. At every different point in the flight envelope, the pilot was required to perform a series of maneuvers consisting in mild to moderate doublets and transitions between steady state conditions. Flights were performed with nominal conditions and with single failed conditions.

3.3.4. Refurbishment/Improvement of Simulink interface for 6DOF flight simulator

The interface as described in [80] is still functional; however, very little documentation was generated and thus, to be able to modify it, efforts were conducted to obtain more flexibility. Figure 15 presented a schematic of the interaction between the different elements involved in this interface.

3.3.4.1. S-functions and UDP data transfer

Two Level 2 S-functions are implemented to receive data from the simulink model, open a UDP port, generate a UDP packet of data and send it. As outlined previously, one S-Function communicates directly with the Translator software that then provides the motion cues, while the other S-Function communicates with the designed plugin to provide the visual cues. These S-functions are referred to as Motion Base S-Function (MBSF) and Visuals S-Function (VSF) respectively. The variables needed as inputs for the MBSF are outlined in the following table [81]:

Message #	Item	Variable	Unit	Convention
170	0	Motion override flag	N/A	Should be set to 1
	1	Total airspeed	knots	N/A
	2	Angle of attack	deg	+ Up
	3	Pitch acceleration	deg/sec^2	+ Up
	4	Yaw acceleration	deg/sec^2	+ Right
	5	Roll acceleration	deg/sec^2	+ Right
	6	Side acceleration	g	+ Right
	7	Normal acceleration	g	+ Up
171	0	Axial acceleration	g	+Forward
	1	Pitch angle	deg	+ Up
	2	Yaw angle	deg	+ Right
	3	Roll angle	deg	+ Right
	4	Not used	N/A	Should be set to -999.0
	5	Not used	N/A	Should be set to -999.0
	6	Not used	N/A	Should be set to -999.0
	7	Not used	N/A	Should be set to -999.0

Figure 21: MBSF variables, units and sign conventions. [80]

All the variables presented in the previous table are defined as IEEE float 32 type for this implementation. Inside the same UDP package, two message structures are sent identified by the numbers 170 and 171 and containing the described variables. This message structures are provided by the manufacturer to be able to successfully communicate with the Translator® software that drives the motion base. The port used to transmit all these data is 49000.

In the case of the VSF, the variables used as input are described in the following table. In addition, the corresponding DataRef (is explained in the following section) that need to be overwritten by the plugin are listed.

Variable	Unit	Convention	DataRef
Latitude	deg	+ North	N/A
Longitude	deg	+ East	N/A
Elevation	m	Above Sea Level	N/A
Quaternions	N/A	N/A	sim/flightmodel/position/q
GPS Velocity in x	m/s	+ East	sim/flightmodel/position/local_vx
GPS Velocity in y	m/s	+ South	sim/flightmodel/position/local_vy
GPS Velocity in z	m/s	+ Up	sim/flightmodel/position/local_vz
Roll rate	deg/sec	N/A	sim/flightmodel/position/P
Pitch rate	deg/sec	N/A	sim/flightmodel/position/Q
Yaw rate	deg/sec	N/A	sim/flightmodel/position/R
Throttle	N/A	0 – No throttle 1 – Full throttle	sim/flightmodel/engine/ENGN_override
Aileron deflection	rad	+ Right Ail. Up	sim/joystick/artstab_roll_ratio
Stabilator deflection	rad	+ Tail Up	sim/joystick/artstab_pitch_ratio
Rudder deflection	rad	+ Tail Left	sim/joystick/artstab_heading_ratio

Figure 22: VSF variables, units and corresponding DataRefs

Latitude, Longitude and Elevation data are used as inputs to the function XPLMWorldToLocal that overrides the values for the three axis position of the plane. The airplane, however, is located initially over Morgantown, more precisely in the point with coordinates 39.64N and 79.91W. This position is then updated for every time frame using the above mentioned X-Plane function. The port used for these data is 49001.

Both S-functions open individual client UDP ports and prepare a binary UDP packet per time step containing the described information. At each time step, one UDP packet of data is sent through each port via the Ethernet connection to a pre-specified IP address corresponding to the Server Computer.

3.3.4.2. X-plane plugin

The data received from the visuals S-Function is then captured by the designed plugin and the values are used to overwrite variables in the X-Plane environment. X-Plane stores variables in a type of tree structure called DataRefs [82]. These DataRefs are organized considering what is the use of the particular value for the simulation environment. An example would be a DataRef like "sim/flightmodel/position/local_vx" that refers to the variable local_vx used to specify the GPS velocity in the world x axis and thus is related to positioning the flight model inside the simulation environment. Other DataRefs are not as intuitive as this one; the description of all DataRefs can be found in [82].

The general process to override a DataRef is as follows:

1. Define a variable inside the plugin as an opaque handle (pointer) to data provided by an external source or another plugin. Example: `XPLMDataRef gP;`
2. Look up the desired opaque handle `XPLMDataRef` defined previously. Example: `gP = XPLMFindDataRef("sim/flightmodel/position/P");`
3. Manipulate the incoming data to assign it to proper variables within the plugin.
4. Write a new value to a DataRef. Example: `XPLMSetDataf(gP,fP);`

After the override is complete, the plugin returns a 1 to the X-Plane Plugin Manager (XPLM) indicating it has successfully completed its execution for the present timeframe.

The following is extracted from [82] on the general behavior of plugins: "*The X-Plane plugin system is built based on DLLs. The central component of this system is the *X-Plane Plugin Manager *(or XPLM). The XPLM is a library of code that manages plugins. The plugin is also a DLL. X-Plane links to the XPLM and the plugin (as well as all other plugins) link to the XPLM. The XPLM then serves as the central hub in the plugin system.*"

The plugin opens a receiving UDP port using the specific number for the visuals and collects the incoming packet of UDP data. This packet is processed to transform it from binary to floating point format in order to allow the use of these values for overriding purposes. It is important to clarify that the plugin will receive one and only one packet of data. If for some reason the connection is lost for one time step, the plugin does not perform the overriding and thus X-Plane generates the visuals based on the previous time step. This situation is experienced as sudden shakes of the visuals (and usually also of the motion base) that alter the quality of the simulation data because it causes confusion to the pilot. This situation, however, is extremely rare.

The main source for documentation as well as examples of plugin programming is [82]. Some simple enhacements proposed for this interface consist in:

- Use Master Warning as visual mean to inform pilot of Failure Detection also triggering a sounding alarm (sim/cockpit/warnings/annunciators/master_warning).
- Display an on screen message with identification results using a more elaborate text box than the one used in [80].
- Allow the pilot a button on the joystick to disengage Master Alarm and/or on screen messages. (sim/cockpit/warnings/annunciators/master_accept)

- Control RPM meters in the cockpit so that whenever an engine failure occurs, the true values are reflected on the instruments.
- Show throttle movements on screen.
- Control stall alarm based on airplane information from the Simulink model. (sim/flightmodel/failures/stallwarning)

Most of these features include simple overriding of particular DataRefs in an equivalent way as explained above. Some others include some special delopment as is the case with the on-screen messages that require the design of the dialog box and probably modifying the control panel image.

Chapter 4: AIS Fault Detection Scheme for WVU YF-22

The entire process needed to obtain the FDI scheme based on the NS algorithm is described in Chapter 2. The present chapter deals with the particular design decisions adopted as well as with results regarding the design process of the scheme.

For this particular application, it was found that a set of identifiers that could work both with longitudinal and lateral failures should include the three angular rates (p, q, r) and information regarding pilot input on the three channels. These are necessary to differentiate pilot induced couplings from failed conditions for several categories of abnormal conditions [68]. Lacking a direct measure of pilot input, the deflections of the healthy control surface (right side deflections: δ_{S_R}, δ_{A_R}, δ_{R_R}) are used to provide the needed information on pilot input.

To illustrate the process, a 2-dimensional simplified example is used next. Figure 4 shows an example of a set of antibodies created using Euclidean distance for a 2D space with pitch rate and rudder deflection as identifiers. Note that these two identifiers have been selected arbitrarily and no special detection capabilities are implied.

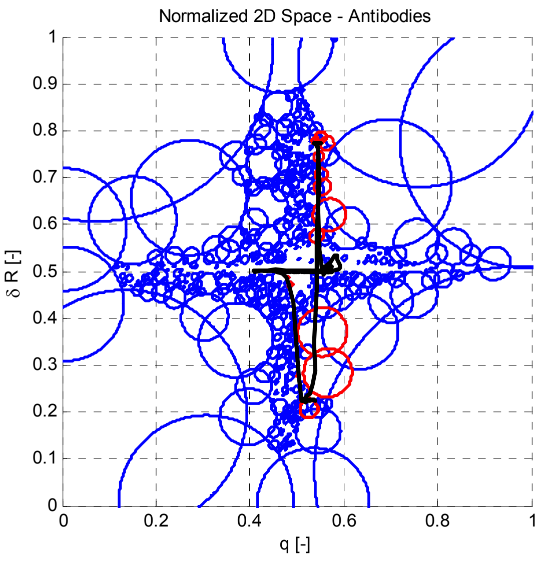


Figure 23: 2-Dimensional solution space using q and δ_R showing faulty data.

As is described in Chapter 2, the values are normalized between 0 and 1. The black line represents flight data for one of the aileron failures followed by the rudder-aileron combo maneuver. The circles correspond to each of the antibodies created. The ones in red dashed line are the ones that get activated for this particular failure while the blue ones are inactive detectors. The empty space not covered by the detectors around q = 0.5 and δ_{R_R} = 0.5, is the area that belongs to the self (healthy conditions) thus covered by the clusters. Clusters are not plotted to favor clarity of the graph.

For comparison, the following plot shows a nominal set of data with the same 2D self as above. No detectors are activated in this case and so all are plotted in blue.

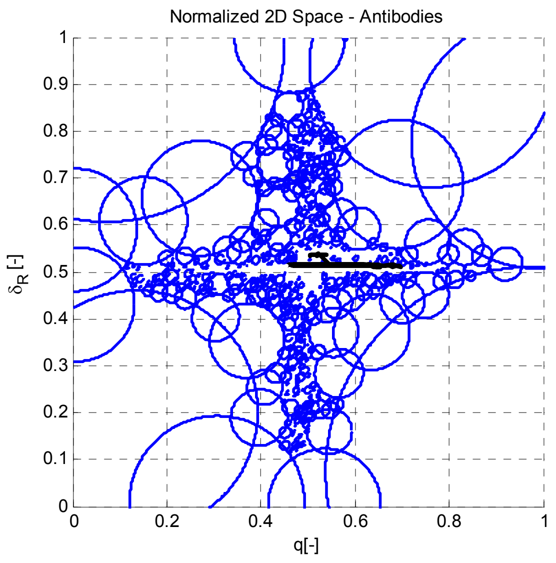


Figure 24: 2-Dimensional solution space using q and δ_R showing healthy data.

The plots shown above correspond to selves (and for consistency, also detectors) generated using 2-dimensional hyper-spheres and Euclidean definition of distance. As was stated before, changes in the definition of distance while keeping the shape constant have consequences on the real shape. The following figure presents 2D spheres using different λ-distance definitions.

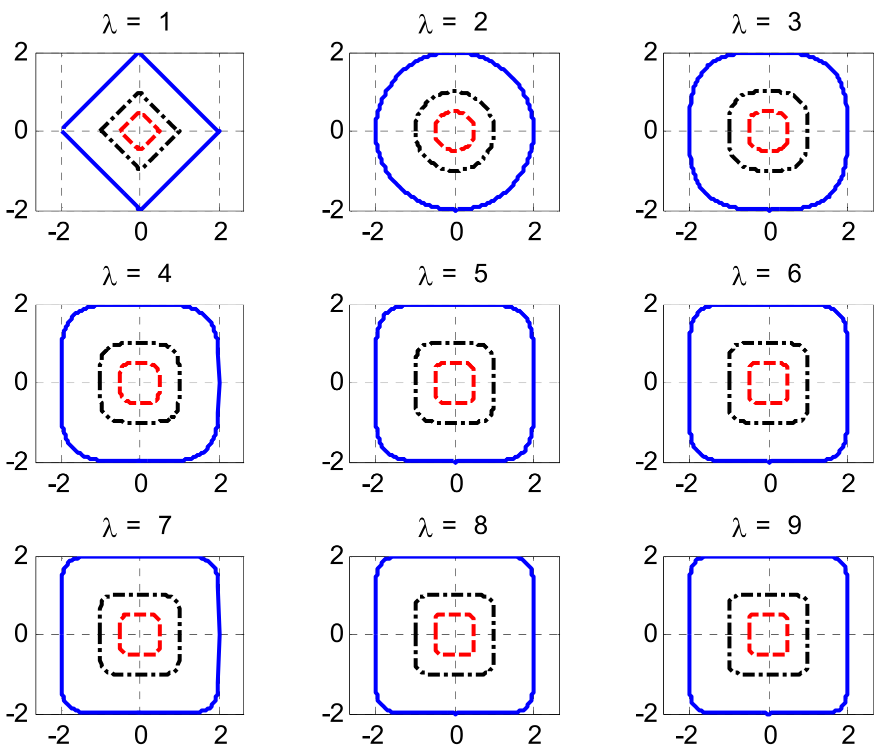


Figure 25: Effect of changing distance definition in 2D spheres

It can be noted from the figure that the sphere tends to a square as λ increases. This is true for any dimension n as was presented in Chapter 2. Moreover, for a given constant radius r and dimension n, the hyper-sphere defined by the distance definition λ_1 encloses completely all hyper-spheres defined using λ_2, given that $\lambda_2 < \lambda_1$. This statement is important for the following design steps and is formally proved in Appendix A.

4.1. Creation of the SELF

The first step in creating the FDI scheme consists in splitting the available self data into data used for creating the self (via clustering) and data used for validation purposes. Self data are obtained using data types 1 and 3 (see table 6) from the following flights (see table 7):

- Flight #1 – 9/16/2008
- Flight #1 – 10/11/2008
- Flight #2 – 10/11/2008
- Flight #1 – 10/18/2008
- Flight #1 – 11/01/2008

- Flight #1 – 11/04/2008

While for validation purposes, the following flights are selected:

- Flight #2 – 9/16/2008
- Flight #2 – 10/18/2008

Self data constitutes approximately 30,000 data points while validation data constitute roughly 5,000 data points. Probably more data should be used for both categories; however, these are all the useful healthy data available.

The self data are used as input data to the clustering algorithm and different sizes of sets of clusters are obtained. The resulting sets are guaranteed to cover all the points used for their creation. To test the validity of this set of clusters, a positive detection approach is used using the validation data. In this case, the generated clusters are used as detectors, then DR_{PP} corresponds to points that fall inside a cluster (inside the SELF).

# of clusters	**1722**	**5799**	**9881**	**14921**
DR_{PP} **[%]**	91.65	68.62	59.72	52.21
Average radius [-]	0.0281	0.0115	0.0063	0.0035
Std. Dev. [-]	0.0132	0.0078	0.0054	0.0034

Table 8: Selection of number of clusters

There is no definite way to chose the number of clusters; however, fewer clusters imply that these are also bigger thus covering more empty-space where detectors could potentially be placed. The average radius and standard deviation are shown here to give an assessment of the size of the generated clusters. The set shown in the last column presents a mean of 0.0035 that implies that a good amount of the 14921 clusters have radii close to the imposed minimum of 0.001. This is not necessarily a bad result but considering the absence of a huge self database, slightly bigger clusters serve as an aid to compensate in part for the lack of information. The set of 9881 clusters is assumed to present an acceptable level of empty-space coverage and thus is used from now on as the SELF.

Once the self is obtained, an interesting test consists in using the same self clusters but changing the distance definition using λ-distance with λ > 2. As was explained before, hyperspheres defined using λ > 2 still enclose at least the points limited by the hypersphere defined using λ = 2. To show the effect of this dimension change, all the available flights are used.

ID #	Maneuver	λ-distance definition											
		λ = 1		λ = 2		λ = 3		λ = 4		λ = 5		λ = 6	
		FA_{pp}	DR_{pp}	FA_{pp}	DR_{pp}	FA_{pp}	DR_{pp}	FA_{pp}	DR_{pp}	FA_{pp}	DR_{pp}	FA_{pp}	DR_{pp}
0	**Validation**	0.00	11.69	0.00	59.72	0.00	65.44	0.00	68.21	0.00	69.72	0.00	70.48
1	**Stabilator**	0.00	0.69	0.00	25.69	0.00	39.58	0.00	42.36	0.00	43.06	0.00	43.75
2	**Stabilator**	0.00	1.27	0.00	29.30	0.00	42.04	0.00	45.86	0.00	49.04	0.00	50.96
3	**Stabilator**	0.00	0.70	0.00	73.94	0.00	93.66	0.00	94.37	0.00	94.37	0.00	95.07
4	**Stabilator**	0.00	0.73	0.00	28.47	0.00	44.53	0.00	47.45	0.00	49.64	0.00	50.36
5	**Stabilator**	0.00	1.99	0.00	58.94	0.00	79.47	0.00	84.11	0.00	85.43	0.00	85.43
6	**Stabilator**	0.00	1.32	0.00	34.21	1.64	44.08	1.64	51.97	1.64	56.58	1.64	58.55
7	**Stabilator**	0.00	4.29	0.00	42.14	0.00	54.29	0.00	58.57	0.00	58.57	0.00	58.57
8	**R-A Combo**	0.00	0.00	0.00	7.98	0.00	19.10	0.00	23.95	0.00	25.86	0.00	26.62
9	**R-A Combo**	0.00	0.74	0.00	12.64	0.00	19.70	0.00	22.30	0.00	23.79	0.00	24.91
10	**R-A Combo**	0.00	1.12	0.00	24.16	0.00	31.23	0.00	33.46	0.00	34.94	0.00	35.69
11	**R-A Combo**	0.00	0.75	0.00	15.73	0.00	23.97	0.00	27.34	0.00	28.46	0.00	28.84
12	**R-A Combo**	0.00	2.82	0.00	30.28	0.00	37.68	0.00	39.08	0.00	39.44	0.00	39.94
13	**Aileron + Bias**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	**Aileron + Bias**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	**Aileron + Bias**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	**Aileron + Bias**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	**Aileron + Bias**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	**Aileron**	0.00	7.37	0.00	72.11	0.00	80.53	0.00	83.16	0.00	85.26	0.00	85.79
19	**Aileron**	0.00	2.62	0.00	65.97	0.00	76.96	0.00	81.68	0.00	85.86	0.00	86.91
20	**Aileron**	0.00	1.04	0.00	38.34	0.00	54.92	0.00	58.03	0.00	59.07	0.00	60.62
21	**Aileron**	0.00	8.85	0.00	72.40	0.00	80.21	0.00	81.25	0.00	82.29	0.00	83.33
22	**Aileron**	0.00	6.19	0.00	57.73	1.61	70.62	1.61	75.26	1.61	76.29	1.61	78.35

Table 9: Effect of changing distance definition in positive detection performance of clusters.

For clarity, FA_{pp} and DR_{pp} in this positive detection approach are explained:

- FA_{pp} : Measure of failed data detected as healthy.
- DR_{pp} : Measure of healthy data detected as healthy.

It can be noticed that there is a monotonic improvement as λ increases. The following plot presents the behavior of DR_{pp} for λ from 1 to 15. The legend corresponds to the flight ID # as is shown in table 9.

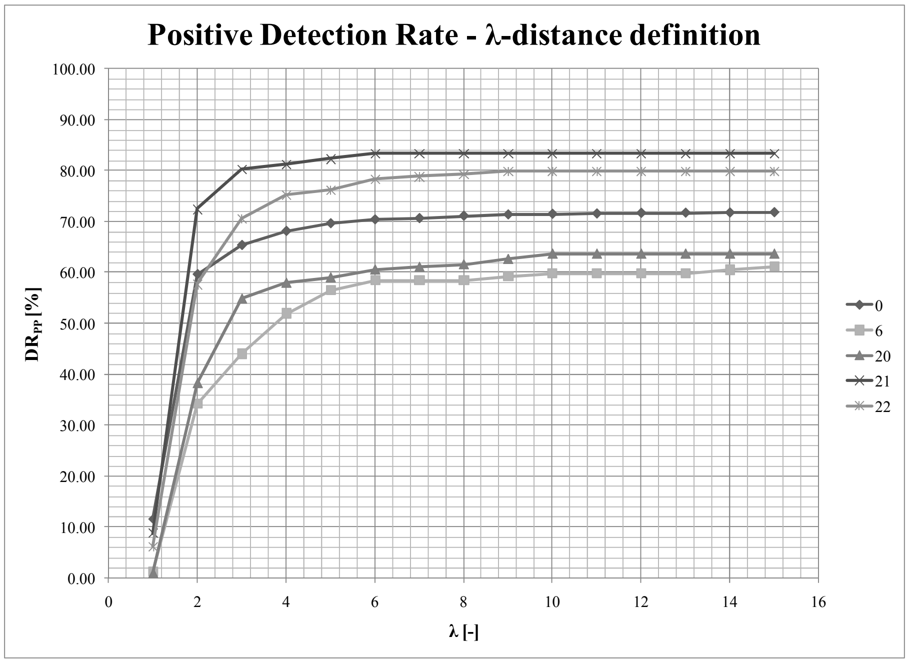


Figure 26: Evolution of Positive DR_{PP} with definition of distance

Also more important to see is the evolution of FA_{PP}. The following plot presents the FA_{PP} for the same flights shown in figure 20. The legend corresponds to the flight ID # as is shown in Table 9.

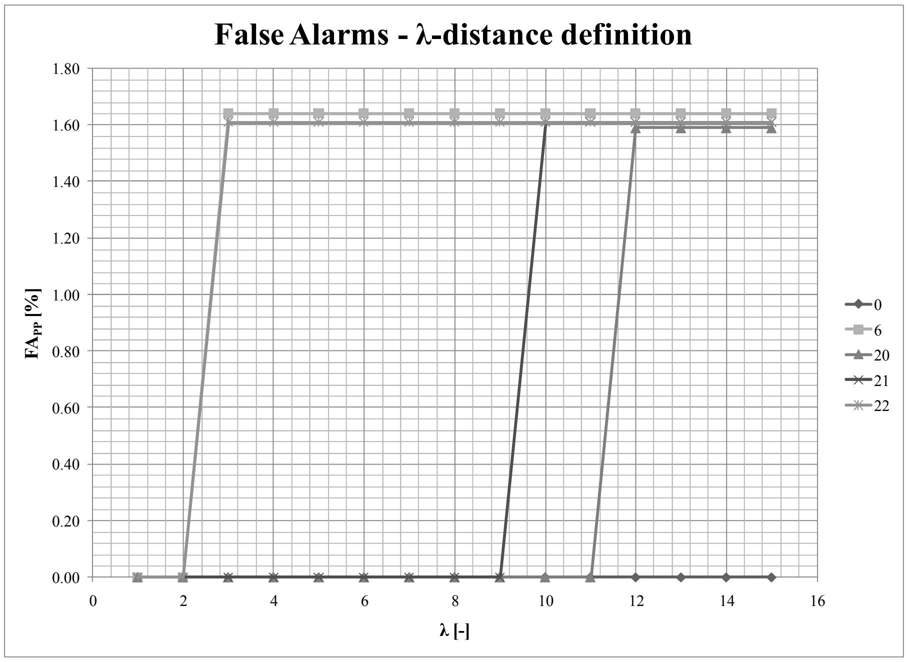


Figure 27: Evolution of Positive FA_{PP} with definition of distance

The flights that are not shown in these two plots experience a similar behavior for DR_{PP} but do not experience increase in FA_{PP} at least until λ = 15. From these plots it can be concluded that changing the distance definition can enhance the performance of the self with little or no cost. Better DR_{PP} accomplished with this Positive Detection approach implies that less FA_{PP} are expected to occur whenever the detectors are created.

By observing the results shown in the last two figures and in the last table, a value of λ = 6 was selected to be used from this point on, as the self does not experience significant improvements after this value, and the FA_{PP} are extremely small (or zero).

4.2. Creation of Detector Set

Different approaches are shown in this section to put into evidence the strengths and weaknesses of each approach. The best result is chosen for the creation of the FDI scheme. It is important to clarify that, from this point on, the results correspond to Negative Selection detection.

4.2.1. ENSA-RV algorithm

Phase 1 of the algorithm presented in [73] is used as the first approach to create a successful detector set. Five detector sets are created and evaluated using the totality of the flights. The results are presented using DR_{PP} and FA_{PP} as evaluation metrics.

ID #	Maneuver	# Detectors									
		500		800		900		1000		1500	
		DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}
0	**Validation**	0.00	0.08	0.00	0.39	0.00	0.12	0.00	0.41	0.00	0.24
1	**Stabilator**	6.56	0.00	9.84	2.08	27.87	3.47	1.64	2.08	60.66	8.33
2	**Stabilator**	1.64	0.00	8.20	0.00	29.51	0.00	1.64	0.00	55.74	0.00
3	**Stabilator**	6.56	0.00	16.39	0.00	29.51	0.00	0.00	0.00	50.82	0.70
4	**Stabilator**	1.67	0.00	31.67	0.00	30.00	0.00	5.00	0.00	86.67	0.73
5	**Stabilator**	3.28	0.00	31.15	0.00	18.03	0.00	1.64	0.00	47.54	0.00
6	**Stabilator**	1.64	0.00	0.00	0.00	11.48	0.00	3.28	0.00	47.54	0.00
7	**Stabilator**	0.00	0.00	21.31	0.00	31.15	0.00	3.28	0.00	50.82	0.71
8	**R-A Combo**	31.15	12.55	57.38	12.93	50.82	13.69	39.34	13.69	67.21	15.21
9	**R-A Combo**	12.90	1.49	9.68	10.41	48.39	4.83	33.87	4.09	58.06	8.92
10	**R-A Combo**	27.42	2.23	48.39	4.46	37.10	8.55	40.32	6.69	72.58	15.61
11	**R-A Combo**	27.42	2.62	51.61	10.11	35.48	6.37	37.10	7.49	64.52	14.61
12	**R-A Combo**	20.97	6.69	38.71	10.92	48.39	11.97	54.84	10.92	72.58	13.73
13	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
14	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
15	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
16	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
17	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
18	**Aileron**	0.00	0.00	1.56	0.00	4.69	0.00	6.25	0.00	4.69	0.00
19	**Aileron**	0.00	0.00	4.76	0.00	1.59	0.00	7.94	0.00	1.59	0.00
20	**Aileron**	0.00	0.00	9.52	0.00	1.59	0.00	34.92	0.00	15.87	0.00
21	**Aileron**	0.00	0.00	14.52	0.00	6.45	0.00	8.06	0.00	14.52	0.00
22	**Aileron**	0.00	0.00	4.84	0.00	8.06	0.00	4.84	1.55	3.23	0.00

Table 10: Different size of detector sets generated with ENSA-RV algorithm

This first table contains a vast amount of information and details that are very important to explain before continuing:

- Validation data shows very low DR_{PP} pointing in the desired direction of obtaining low FA_F.
- All flights show monotonic increasing DR_{PP} with increasing number of detectors except for the 1000 detectors case that presents a considerable drop for stabilator failures. This is attributed mainly to two factors:
  - The reduced randomization processes involved in this method.
  - As was stated before, this method favors generation of bigger detectors considering these as better fit detectors.

The consequence is that the space where these failures are located allows only small detectors thus increasing the size of the search space for the random inclusion of new detectors. The algorithm tries to maximize the coverage; however, the limited space where these failures are located is probably tiny in comparison to the size of the non-self.

- Flights 8 to 12 correspond to the combo maneuver, and the high DR_{PP} is due to the lack of meaningful rudder information. As was stated in Chapter 3, the pilot flies almost without using rudder input. This lack of information allows detectors to be placed in locations that correspond actually to healthy behavior.
- Taking Flight 10 as an example and detector sets with 800, 900 and 1000, the DR_{PP} varies and is actually larger for the smallest set. This can be explained because the algorithm used tries to diminish overlapping of detectors, and even though it can be seen that in the three cases some detectors are placed in this space, the sizes and relative positions may have been modified to reduce overlapping involuntarily reducing DR_{PP}.
- Flights 13 to 17 correspond to the biased flight mentioned in Chapter 3. The data obtained in this flight is located considerably far from the self and thus the algorithm is extremely successful in detecting these failures with 100% DR_{PP}.
- The detector set containing 1500 detectors outperforms the others except for flights 18 to 22 that have better DR_{PP} for the detector set of 1000 detectors. However, in a global view, the 1500 detectors set is determined to be the best set obtained with ENSA-RV algorithm.

To consider that a detector set is promising, a DR_{PP} of at least 50% is expected. This means that the 1500 detector set is only expected to perform successfully in detecting failures contained in flights 1 to 17; however, it is virtually impossible that a DR_{PP} of 4% can be considered successful.

An idea to increase the performance of this algorithm consists in adding more randomization to the process. A bigger initial population of 2000 random detectors is used and then these detectors are run through the censoring, cloning, and moving algorithm. As these 2000 detectors have big overlapping areas, the threshold w_{thre} needs to be set extremely high in order to let more detectors enter the cloning and moving sections of the algorithm. 4000 random centers are also tried at each new iteration but as the iterative process takes no more than a couple of iterations (the initial value is close to the final value and many detectors are cloned and/or moved), not many such detectors make it to the final set.

The initial random detectors generated are forced to be of a maximum possible radius 0.3 which is not necessarily a tiny radius but serves to prove the effect of forcing such detectors. The following table presents results using these 2000 detectors for all the flights except for flights 13 to 17 that, as shown previously, are not difficult to detect.

ID #	Maneuver	# Detectors							
		2000 small		2000 small + ENSA-RV		1500 ENSA-RV		1500 +2000 small	
		DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}
0	**Validation**	0.00	0.41	0.00	0.43	0.00	0.24	0.00	0.53
1	**Stabilator**	32.79	3.47	59.02	3.47	60.66	8.33	75.41	11.11
2	**Stabilator**	8.20	0.64	47.54	0.64	55.74	0.00	60.66	0.64
3	**Stabilator**	42.62	0.00	57.38	0.00	50.82	0.70	82.25	0.70
4	**Stabilator**	56.67	2.92	83.33	2.92	86.67	0.73	93.33	2.92
5	**Stabilator**	36.07	0.00	47.54	0.00	47.54	0.00	77.05	0.00
6	**Stabilator**	13.11	1.32	29.51	1.32	47.54	0.00	54.10	1.32
7	**Stabilator**	32.79	0.00	52.46	0.00	50.82	0.71	72.13	0.71
8	**R-A Combo**	40.98	17.49	45.90	17.49	67.21	15.21	78.69	23.95
9	**R-A Combo**	48.39	10.41	50.00	11.15	58.06	8.92	69.35	16.36
10	**R-A Combo**	48.39	11.90	62.90	11.90	72.58	15.61	88.71	20.45
11	**R-A Combo**	46.77	17.23	58.06	17.23	64.52	14.61	82.26	22.85
12	**R-A Combo**	59.68	17.96	67.74	17.61	72.58	13.73	88.71	21.13
18	**Aileron**	3.13	0.00	3.13	0.00	4.69	0.00	4.69	0.00
19	**Aileron**	4.76	0.00	4.76	0.00	1.59	0.00	6.35	0.00
20	**Aileron**	23.81	0.00	26.98	0.00	15.87	0.00	34.92	0.00
21	**Aileron**	17.74	1.04	17.74	1.04	14.52	0.00	17.74	1.04
22	**Aileron**	3.23	0.00	3.23	0.00	3.23	0.00	4.84	0.00

Table 11: Effect of adding more randomization to ENSA-RV

The first column presents the results for the 2000 random detectors alone. These detectors are not bad even compared to the 1500 generated previously with the entire ENSA-RV algorithm. The second column presents the same 2000 detectors after letting the ENSA-RV algorithm perform cloning and moving on them. The final result of the algorithm is a set of 2138 detectors. The result obtained with this approach gets close to the results obtained with the 1500 detectors previously generated (showed in the third column). The last column is a set of detectors obtained concatenating the 2000 small randomly placed detectors with the 1500 ENSA-RV detectors. The result is a promising detector set that has still very low DR_{PP} for the last five flights presented.

Giving a closer look to the previous table, an interesting fact can be extracted. The 2000 small random detectors produce a DR_{PP} of 17.74 and a FA_{PP} of 1.04 for the Aileron failure of Flight #21 and when these detectors are added to the 1500 ENSA-RV generated, the result is exactly the same. This means that the detectors included in the first detector set that are capable of detecting this particular failure have a large overlapping portion with the ones already in the 1500 ENSA-RV detectors. On the other hand, for a similar flight like 19 also presenting an Aileron failure, the result of the combination is almost equal to the addition of the individual DR_{PP}. In this case, the particular detectors are placed with little or no overlapping.

The next step consists on trying to improve the results of ENSA-RV using the Positive Selection Enhancer algorithm.

4.2.2. ENSA-RV algorithm with Positive Selection Enhancer

The PSE algorithm generates extra detectors using training failed data and also deletes the ones causing FA_{PP}. One flight for each failure/maneuver is selected to use with this algorithm. The chosen flights are:

- Flight 1: Stabilator failure including stabilator maneuver
- Flight 9: Aileron failure including combo maneuver
- Flight 18: Aileron failure including aileron maneuver

ID #	**Maneuver**	**# Detectors**							
		800 ENSA-RV		**800 ENSA-RV + PSE (F#1) = 811**		**800 ENSA-RV + PSE (F#1, F#9) = 830**		**800 ENSA-RV + PSE (F#1, F#9, F#18) = 845**	
		DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}
0	**Validation**	0.00	0.39	0.00	0.39	0.00	0.39	0.00	0.39
1	**Stabilator**	9.84	2.08	100.00	0.00	100.00	1.39	100.00	1.39
2	**Stabilator**	8.20	0.00	78.69	0.00	78.69	1.27	78.69	1.27
3	**Stabilator**	16.39	0.00	91.80	0.00	91.80	0.00	91.80	0.00
4	**Stabilator**	31.67	0.00	88.33	0.00	88.33	0.73	88.33	0.73
5	**Stabilator**	31.15	0.00	86.89	0.00	86.89	0.00	86.89	0.00
6	**Stabilator**	0.00	0.00	54.10	0.00	54.10	0.66	54.10	0.66
7	**Stabilator**	21.31	0.00	86.89	0.00	86.89	0.00	86.89	0.00
8	**R-A Combo**	57.38	12.93	57.38	12.93	68.85	12.17	68.85	12.17
9	**R-A Combo**	9.68	10.41	9.68	6.32	100.00	0.00	100.00	0.00
10	**R-A Combo**	48.39	4.46	48.39	4.46	70.97	2.23	70.97	2.23
11	**R-A Combo**	51.61	10.11	51.61	10.11	72.58	7.12	72.58	7.12
12	**R-A Combo**	38.71	10.92	38.71	10.92	69.35	9.15	69.35	9.15
18	**Aileron**	1.56	0.00	1.56	0.00	1.56	0.00	100.00	0.00
19	**Aileron**	4.76	0.00	4.76	0.00	25.40	0.00	68.25	0.00
20	**Aileron**	9.52	0.00	9.52	0.00	14.29	0.00	69.84	0.52
21	**Aileron**	14.52	0.00	14.52	0.00	16.13	0.00	82.26	0.00
22	**Aileron**	4.84	0.00	4.84	0.00	4.84	0.00	85.48	0.00

Table 12: ENSA-RV detectors including PSE algorithm.

The flights used for training are highlighted in green in the previous table to show how the effect of the PSE consists in generating and/or deleting detectors to finally obtain DR_{PP} = 100 and FA_{PP} = 0. As these flights are used for training the results are obviously perfect; however the other flights of the same type show notorious improvements after the training stage. In fact, for the last column that presents the result of using the PSE algorithm with the three flights chosen, all the failures have DR_{PP} higher than 50% what is the desired minimum limit to consider a set of detectors as successful.

Although the resulting detector set is acceptable to start the design of the detection phase, there is one issue that cannot be overlooked. The original detector set is not satisfactory in detecting this set of known failures, used for creating this FDI scheme. It is desired that the randomly generated detector set presents at least fair detection capabilities for known failures before applying the PSE thus assuming that probabilistically, unknown upset conditions located in comparable spaces in the universe will have higher chances of experiencing adequate coverage (this is explained in more detail in Chapter 2).

Now knowing that a good detection set can be found, it is worthy to take a closer look at the detectors added by the PSE algorithm. The following values are calculated from the 65 specialized detectors added using PSE (20 detectors are removed during the same process).

- Average radius: 0.0506
- Standard deviation: 0.0385
- Min radius: 0.0053
- Max radius: 0.1817

This values show that most of the detectors are added in areas where the maximum size for a detector is less than 0.2. Analyzing the size of the detectors generated for the sets already presented, the following table is obtained:

Det. Set	**500 ENSA-RV**	**800 ENSA-RV**	**900 ENSA-RV**	**1000 ENSA-RV**	**1500 ENSA-RV**	**2000 small rnd dets.**
$r_{\text{det}} \leq 0.20$	264 (51.26%)	551 (68.88%)	613 (67.96%)	721 (71.46%)	1206 (80.29%)	1771 (88.56%)
$r_{\text{det}} \leq 0.10$	68 (13.20%)	185 (23.12%)	216 (23.95%)	286 (28.35%)	497 (33.09%)	1599 (79.96%)
$r_{\text{det}} \leq 0.05$	11 (2.14%)	24 (3.00%)	25 (2.77%)	49 (4.86%)	103 (6.86%)	175 (8.75%)

Table 13: Analysis of size of detectors generated for some of the tested detectors sets.

The presented values clearly illustrate the origin of the difficulty experienced by all previous sets of detectors to fill all the spaces where the failures are located. As is known from the outcome of the PSE algorithm, the majority of failure data is located in areas where the maximum allowable radius is below 0.2, while the average is close to 0.05. 0.2 does not necessarily represent a small detector; however, it was noted earlier that the ENSA-RV algorithm favors detectors with even bigger radii. The need for a major search for smaller detectors is the main motivation for creating Algorithm #2.

4.2.3. Algorithm #2

Six sizes intervals are defined for this case and the variables involved are presented in the following table.

Size	Interval	Number of Dets.	Clones
Size 0	$r_{\min} \leq r < 0.03$	$N_{RD1} = \frac{1350}{3000} \cdot N_{RD}$	$N_{CLON} = 60$
Size 1	$0.03 \leq r < 0.05$	$N_{RD1} = \frac{700}{3000} \cdot N_{RD}$	$N_{CLON} = 25$
Size 2	$0.05 \leq r < 0.10$	$N_{RD2} = \frac{500}{3000} \cdot N_{RD}$	$N_{CLON} = 15$
Size 3	$0.10 \leq r < 0.20$	$N_{RD3} = \frac{300}{3000} \cdot N_{RD}$	$N_{CLON} = 0$
Size 4	$0.20 \leq r < 0.40$	$N_{RD4} = \frac{100}{3000} \cdot N_{RD}$	$N_{CLON} = 0$
Size 5	$0.40 \leq r < 1.00$	$N_{RD5} = \frac{50}{3000} \cdot N_{RD}$	$N_{CLON} = 0$

Table 14: Sizes definition for NS Algorithm #2.

The number of detectors is calculated based on how many detectors would be valuable to include of each size for a 3000 set of detectors (considering that this algorithm does not avoid overlap). A detector set is then generated using the above presented intervals and Algorithm #2. In the following table, the effect of different overlapping threshold w_{thre} is shown. The value in parenthesis in the header of the table is the number of detectors left after the censoring process.

ID #	Maneuver	w_{thre} = 1 (1203)		w_{thre} = 5 (3139)		w_{thre} = 10 (4572)		w_{thre} = 20 (6410)		w_{thre} = 50 (9179)		w_{thre} = N/A (10007)	
		DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}
0	**Validation**	0.00	0.35	0.00	0.61	0.00	0.71	0.00	0.78	0.00	0.78	0.00	0.78
1	**Stabilator**	19.67	14.58	57.38	17.36	57.38	20.41	62.30	20.83	63.93	20.83	63.93	20.83
2	**Stabilator**	24.59	4.46	54.10	5.73	55.74	5.73	55.74	5.73	55.74	5.73	55.74	5.73
3	**Stabilator**	19.67	0.00	44.26	0.00	45.90	0.00	50.82	0.00	50.82	0.00	50.82	0.00
4	**Stabilator**	35.00	2.19	83.33	5.84	83.33	5.84	83.33	6.57	83.33	6.57	85.00	6.57
5	**Stabilator**	4.92	0.00	47.54	0.00	49.18	0.00	52.46	0.00	52.46	0.00	52.46	0.00
6	**Stabilator**	0.00	1.32	49.18	1.32	49.18	1.32	52.46	1.32	52.46	1.32	52.46	1.32
7	**Stabilator**	24.59	0.00	54.10	0.00	59.02	0.00	62.30	0.00	68.85	0.00	68.85	0.00
8	**R-A Combo**	26.23	12.93	81.97	24.33	85.25	25.10	85.25	27.76	85.25	28.90	85.25	29.28
9	**R-A Combo**	25.81	7.81	40.32	16.73	56.45	16.73	61.29	24.16	62.90	24.54	62.90	24.54
10	**R-A Combo**	24.19	3.72	58.06	13.38	82.26	13.75	82.26	21.19	83.87	21.93	83.87	22.68
11	**R-A Combo**	29.03	10.86	69.35	19.85	87.10	20.60	91.94	26.59	91.94	27.72	91.94	27.72
12	**R-A Combo**	30.65	10.56	56.45	20.77	90.32	21.13	90.32	21.13	91.94	23.24	91.94	24.65
13	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
14	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
15	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
16	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
17	**Aileron + Bias**	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00	0.00
18	**Aileron**	50.00	0.00	51.56	0.00	51.56	0.00	51.56	0.00	51.56	0.00	51.56	0.00
19	**Aileron**	31.75	0.00	34.92	0.00	58.73	0.00	58.73	0.00	58.73	0.00	58.73	0.00
20	**Aileron**	41.27	0.00	47.62	0.00	47.62	0.00	47.62	0.00	52.38	0.00	52.38	0.00
21	**Aileron**	53.23	0.00	58.06	0.00	64.52	0.00	64.52	0.00	66.13	0.00	66.13	0.00
22	**Aileron**	46.77	0.00	50.00	0.00	50.00	0.00	50.00	0.00	50.00	0.00	50.00	0.00

Table 15: Variation of overlapping threshold for detector set created using Algorithm #2.

The following figure summarizes the general trend of the flights shown in the table above. The legend corresponds to the flight ID# as shown in table 15.

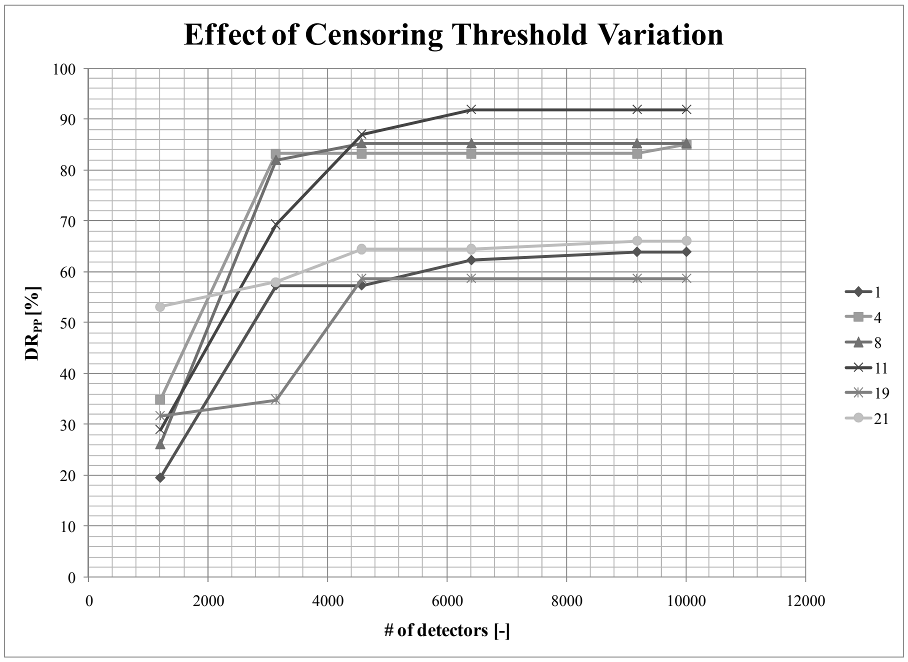


Figure 28: Effect of changing overlapping threshold for Algorithm #2.

As is expected, increasing w_{thre} has as a first consequence the increase of the size of the detector set as more detectors are accepted for maturation. The last column presents the results for the 10007 detectors generated and as can be seen, all the failures have DR_{PP} higher than 50%. An undesirable but expected consequence of imposing small detectors is that, together with the increase in DR_{PP}, there is an increase in FA_{PP}; however, it is important to remember that most of the false alarms for flights 8 to 12 are due to the lack of meaningful information for rudder inputs.

It is assumed that the more extensive the algorithm is run (more detectors are generated), the detector set would have probabilistically more chances of better filling more spaces. Using this Negative Selection approach for detection implies that for each time step, m distance calculations must be performed (one for each of the m detectors). In healthy conditions no detector is activated and so all the calculations are necessary, while for failed conditions, the scheme runs until it finds at least one activated detector and then moves to the next time step. This means that the worst case scenario presents m distance calculations and m comparisons between the distance and the radius, for each time step. Two different Personal Computers (PCs) were used to estimate average time to perform these two operations. Considering a sampling rate of 50 Hz, these calculations should not last longer than 0.02 seconds and thus a maximum number of detectors for real time operation can be roughly estimated.

	PC 1	PC 2
Processor	2.5GHz Intel Core 2 Duo	3.4GHz Intel Pentium IV
RAM memory	2 GB	2 GB
Brand	HP	DELL
Model	Pavilion dv6000	Dimension 5150
Type	Laptop	Desktop
Average time to perform both operations	$8 \cdot 10^{-6}$ seconds	$1 \cdot 10^{-5}$ seconds
Maximum detectors for real time operation	2600	2000

Table 16: Estimation of maximum detectors set size for real time operation.

It can be seen that computational efforts, when dealing with large numbers of detectors, are significant. A max limit of 5000 detectors is imposed to the scheme. This is almost double the maximum detector set for real time operation for the fastest computer described; however, as explained before, the calculations are greatly reduced in the presence of activation. As all the flights used (even the validation ones) present false activations, it is assumed that 2600 is too restrictive.

It must be clarified that the intention of this work is not real time operation; however, this would be an interesting follow up work and thus some consideration to maintain the scheme under certain limits of computational complexity are worth to be included. The results presented in Chapter 5 are obtained using PC1.

A detector set of 25000 detectors is created with the intention of obtaining a final set of detectors that through overlapping censoring can provide a detectors set of about 5000 detectors presenting acceptable DR_{PP} values for the totality of the flights. The set created has the following properties:

Property	Value
Initial random set size	25000
w_{thre}	5
Detector Set Size	5051
Detectors with $r_{\det} \leq 0.20$ **[# (%)]**	4686 (92.77%)
Detectors with $r_{\det} \leq 0.10$ **[# (%)]**	4666 (92.38%)
Detectors with $r_{\det} \leq 0.05$ **[# (%)]**	4049 (80.16%)
Detectors with $r_{\det} \leq 0.03$ **[# (%)]**	2022 (40.03%)

Table 17: Summary of properties of chosen detector set.

The results obtained for the original set as well as the censored one are presented in the following table.

ID #	Maneuver	# Detectors			
		25000 Algorithm #2		25000 Algorithm #2 – Censoring $w_{thre} = 5$	
		DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}
0	**Validation**	0.00	2.04	0.00	1.35
1	**Stabilator**	78.69	22.92	47.54	20.14
2	**Stabilator**	86.89	8.28	57.38	8.28
3	**Stabilator**	77.05	1.41	62.30	1.41
4	**Stabilator**	98.33	12.41	86.67	8.76
5	**Stabilator**	85.25	0.00	37.70	0.00
6	**Stabilator**	62.30	2.63	52.46	2.63
7	**Stabilator**	70.49	0.71	59.02	0.71
8	**R-A Combo**	91.80	28.90	67.21	25.86
9	**R-A Combo**	85.48	31.60	61.29	26.02
10	**R-A Combo**	95.16	26.02	69.35	24.91
11	**R-A Combo**	96.77	34.08	64.52	29.21
12	**R-A Combo**	96.77	28.52	74.19	24.30
13	**Aileron + Bias**	100.00	0.00	100.00	0.00
14	**Aileron + Bias**	100.00	0.00	100.00	0.00
15	**Aileron + Bias**	100.00	0.00	100.00	0.00
16	**Aileron + Bias**	100.00	0.00	100.00	0.00
17	**Aileron + Bias**	100.00	0.00	100.00	0.00
18	**Aileron**	84.38	2.63	82.81	0.00
19	**Aileron**	60.32	0.00	55.56	0.00
20	**Aileron**	85.71	0.52	68.25	0.52
21	**Aileron**	82.62	2.08	59.68	2.08
22	**Aileron**	80.65	0.00	77.42	0.00

Table 18: Detection results for chosen detector set.

The detector set before censoring possesses very good DR_{PP} values for all the flights with a lowest of 60.32%. The censored detector set still presents some very high DR_{PP} values and only two DR_{PP} under 50% (but close). This qualifies as a satisfactory detector set and thus is selected to create the FDI scheme.

4.2.4. Algorithm #2 with Positive Selection Enhancer

Even though the DR_{PP} achieved using just the result of Algorithm #2 are satisfactory, the PSE algorithm has proven to be able to increment them even more at a rather low cost. PSE Algorithm is applied to the chosen detector set in the same way it was applied in subsection 4.2.2 for an ENSA-RV detector set. The following table summarizes the improvements achieved.

ID #	Maneuver	# Detectors							
		5051 A. #2		5051 A. #2 + PSE (F#1) = 5051		5051 A. #2 + PSE (F#1, F#8) = 5024		5000 A. #2 + PSE (F#1, F#8, F#19) = 5044	
		DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}	DR_{PP}	FA_{PP}
0	Validation	0.00	1.35	0.00	1.35	0.00	1.35	0.00	1.35
1	Stabilator	47.54	20.14	100.00	0.00	100.00	0.00	100.00	0.00
2	Stabilator	57.38	8.28	72.13	8.28	72.13	7.64	72.13	7.64
3	Stabilator	62.30	1.41	90.16	1.41	90.16	1.41	90.16	1.41
4	Stabilator	86.67	8.76	95.00	8.76	95.00	8.76	95.00	8.76
5	Stabilator	37.70	0.00	90.16	0.00	90.16	0.00	90.16	0.00
6	Stabilator	52.46	2.63	65.57	2.63	65.57	2.63	65.57	2.63
7	Stabilator	59.02	0.71	75.41	0.71	75.41	0.71	75.41	0.71
8	R-A Combo	67.21	25.86	65.57	25.86	100.00	0.00	100.00	0.00
9	R-A Combo	61.29	26.02	61.29	23.42	61.29	10.04	61.29	10.04
10	R-A Combo	69.35	24.91	67.74	24. 16	79.03	4. 83	79.03	4. 83
11	R-A Combo	64.52	29.21	64.52	26.22	80.65	7.49	80.65	7.49
12	R-A Combo	74.19	24.30	72.58	23.59	87.10	5.28	87.10	5.28
18	Aileron	82.81	0.00	82.81	0.00	82.81	0.00	90.63	0.00
19	Aileron	55.56	0.00	55.56	0.00	55.56	0.00	100.00	0.00
20	Aileron	68.25	0.52	68.25	0.52	68.25	0.52	68.25	0.52
21	Aileron	59.68	2.08	59.68	2.08	59.68	2.08	70.97	2.08
22	Aileron	77.42	0.00	77.42	0.00	77.42	0.00	80.65	0.00

Table 19: Algorithm #2 detectors including PSE algorithm

As before, the flights used to for the PSE algorithm are highlighted in green. The final detector set presents large values of DR_{PP} for all flights with a minimum of 61.29. FA_{PP} values are acceptable though as expected, the highest are experienced for flights 8 to 12.

4.2.5. Failure detection stage

As was explained previously, single activation is not desirable to trigger Failure Detection and instead a time window and a minimum activation threshold are used. Both parameters allow tuning and determine the sensitivity of the FDI scheme. Increasing the size of the time window or decreasing the value of the threshold moves the scheme towards sensitivity and vice versa. The time window T_W is expressed in time steps for which the activation of a detector is kept in memory, while the detection threshold D_{thre} is expressed as a percentage of T_W.

This section presents the results in terms of DR_F (Flight Detection Rate) and FA_F (Flight False Alarm). The following plots present the effects on DR_F and FA_F of changing T_W for different constant values of D_{thre}.

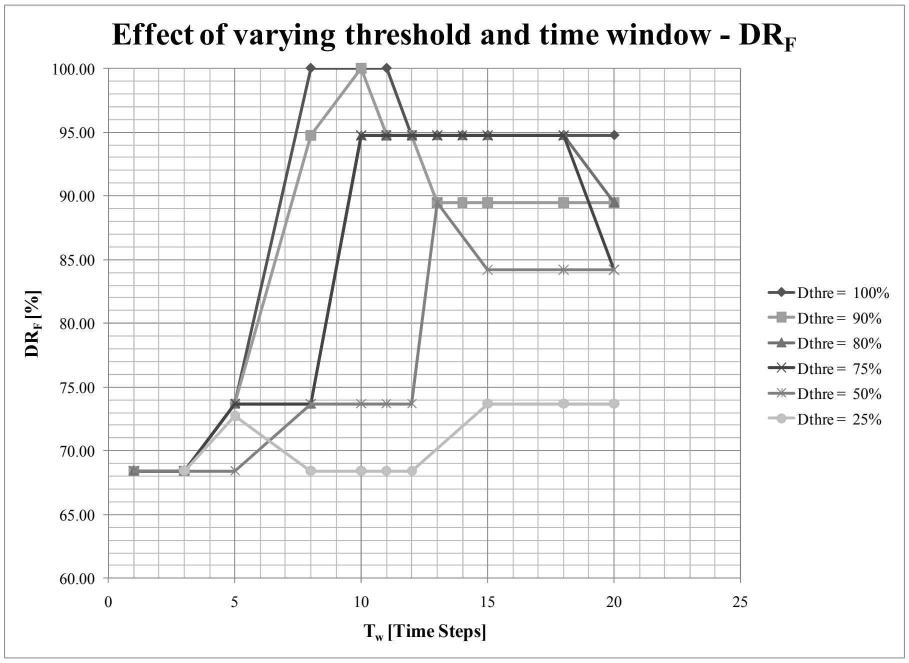


Figure 29: Effect of varying threshold on Flight Detection Rate (DR_F)

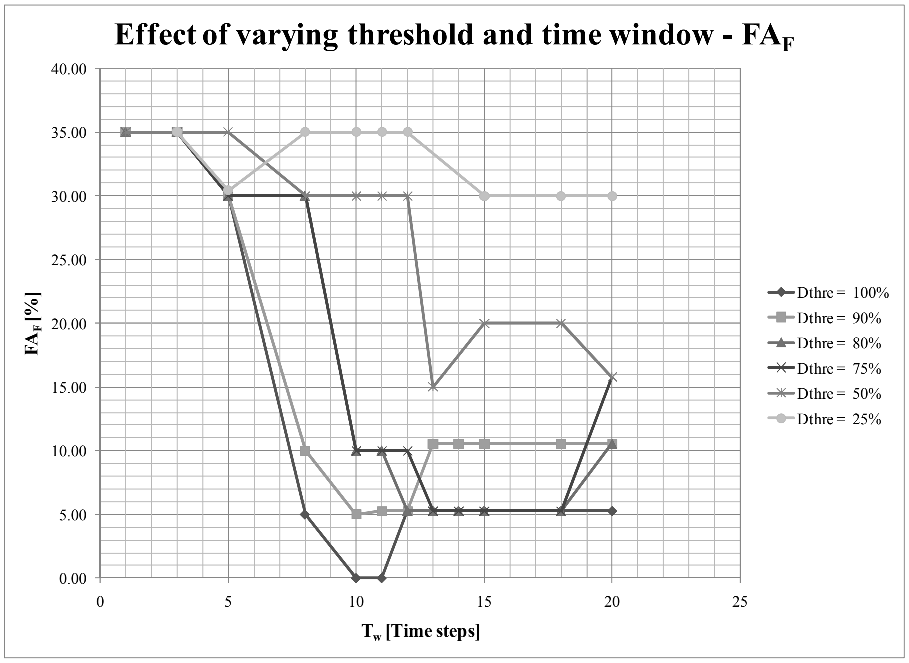


Figure 30: Effect of varying threshold on Flight False Alarms (FA_F)

Most failures only last for about 1.3 seconds (duration of the maneuver) and recalling that this represents 65 data points, an upper limit for T_W is set to 22 that is a third of the duration of the failure. $T_W = 20$ shows already undesirable results (increasing FA_F) and so is the biggest value presented.

Another important aspect of the detection is the time to detect the failures. The following plot presents the average value of TD for all the flights that were correctly detected.

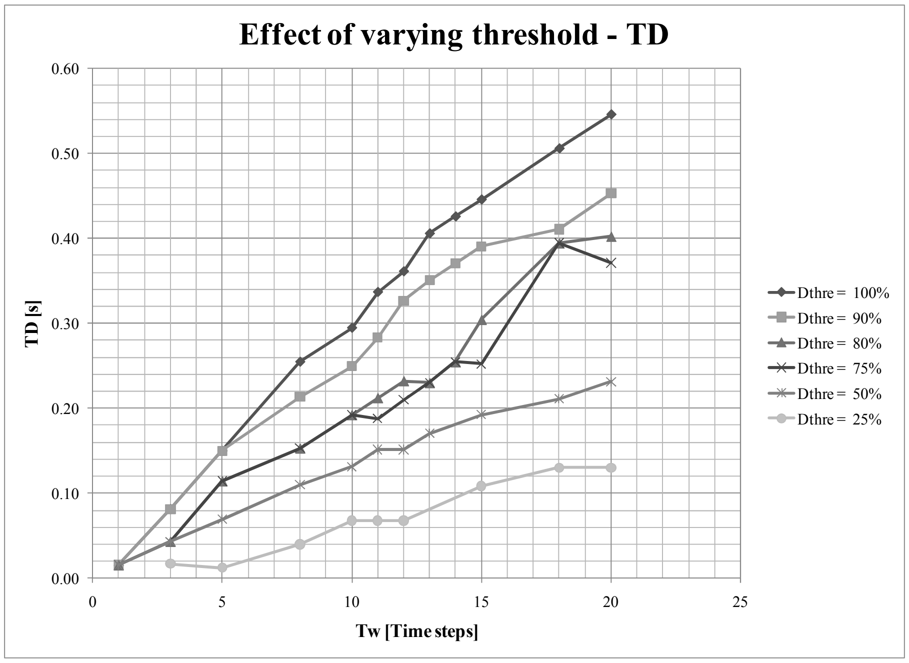


Figure 31: Effect of varying threshold on Time to Detect (TD)

The trends seen in the previous three plots can be summarized as:

- Given a constant D_{thre}, there exists a range of T_W for which a maximum DR_F is reached. The opposite is true for FA_F. This can be explained considering the fact that the detector set implemented contains nonzero FA_{PP} values for most flights. If the time window is made large enough, the activation is kept in memory for longer and then scattered false activations become added and are more likely to trigger false alarms.
- Increases in both D_{thre} and T_W have a general trend of increasing the average TD because more samples need to be considered to declare a failure.

There is an aspect not visible in the graphs and that is important to comment here. The validation data presents a very low DR_{PP} of 1.37%, unfortunately, this false activations are concentrated in two areas of the validation data. This is the cause of bigger values of DR_F corresponding to low T_W or large T_W combined with low D_{thre}.

There are four combinations of T_W and D_{thre} that present 100 DR_F; however, two out of these four ($T_W = 8 \text{ Time Steps}$, $D_{thre} = 100\ \%$ and $T_W = 10 \text{ Time Steps}$, $D_{thre} = 90\ \%$) present FA_F of 5% and particularly present False Alarms for the validation data thus becoming inadequate. The remaining two, require $D_{thre} = 100\ \%$, which is not seen as a desirable quality because it would require that absolutely all the points considered in the time window would need to be activated. Other combinations that present acceptable results both for DR_F and FA_F are:

1. $T_W = 11 \text{ Time Steps}$, $D_{thre} = 90\ \%$, $DR_F = 94.74\%$, $FA_F = 5.26\%$, $TD = 0.28s$
2. $T_W = 12 \text{ Time Steps}$, $D_{thre} = 90\ \%$, $DR_F = 94.74\%$, $FA_F = 5.26\%$, $TD = 0.33s$

Even though these two selections present positive value of FA_F, it is due to a premature activation in flight # 10 produced by the occurrence of some scattered false alarms immediately before the start of the failure. Any of these two combinations could be used; however, number 1 is favored due to having lower average TD.

From the results shown in the plots and the analysis carried out, more test were performed in the vicinity of $T_W = 11 \text{ Time Steps}$, $D_{thre} = 90\ \%$. From these tests, the following values are selected:

- $T_W = 14 \text{ Time Steps}$
- $D_{thre} = 70\ \%$

These values provide exactly the same DR_F and FA_F as options 1 and 2 before but with lower average TD. It is also seen as a benefit the use of a lower value of D_{thre} considering that a less restrictive detection threshold may favor possible detections of unknown failures.

The final result states that the activation is kept in memory for a time window of 0.28 seconds and a failure is declared if 10 detectors are found active at any given time.

4.2.6. Identification stage

Abnormal conditions data is used to create a list of the detectors activated for each one of the two particular failures:

- Left stabilator locked at trim
- Left aileron locked at trim

These lists serve the identification and thus are used to determine if a known or unknown failure has occurred. The identification stage only takes place after a positive detection. If the detector(s) activated is (are) contained in one of the two lists previously described, that particular failure is identified as such. If this is not true, an unknown abnormal condition is called. If an unknown failed condition is found, the system generates a record containing which detector/s activated for this particular situation to use as future reference.

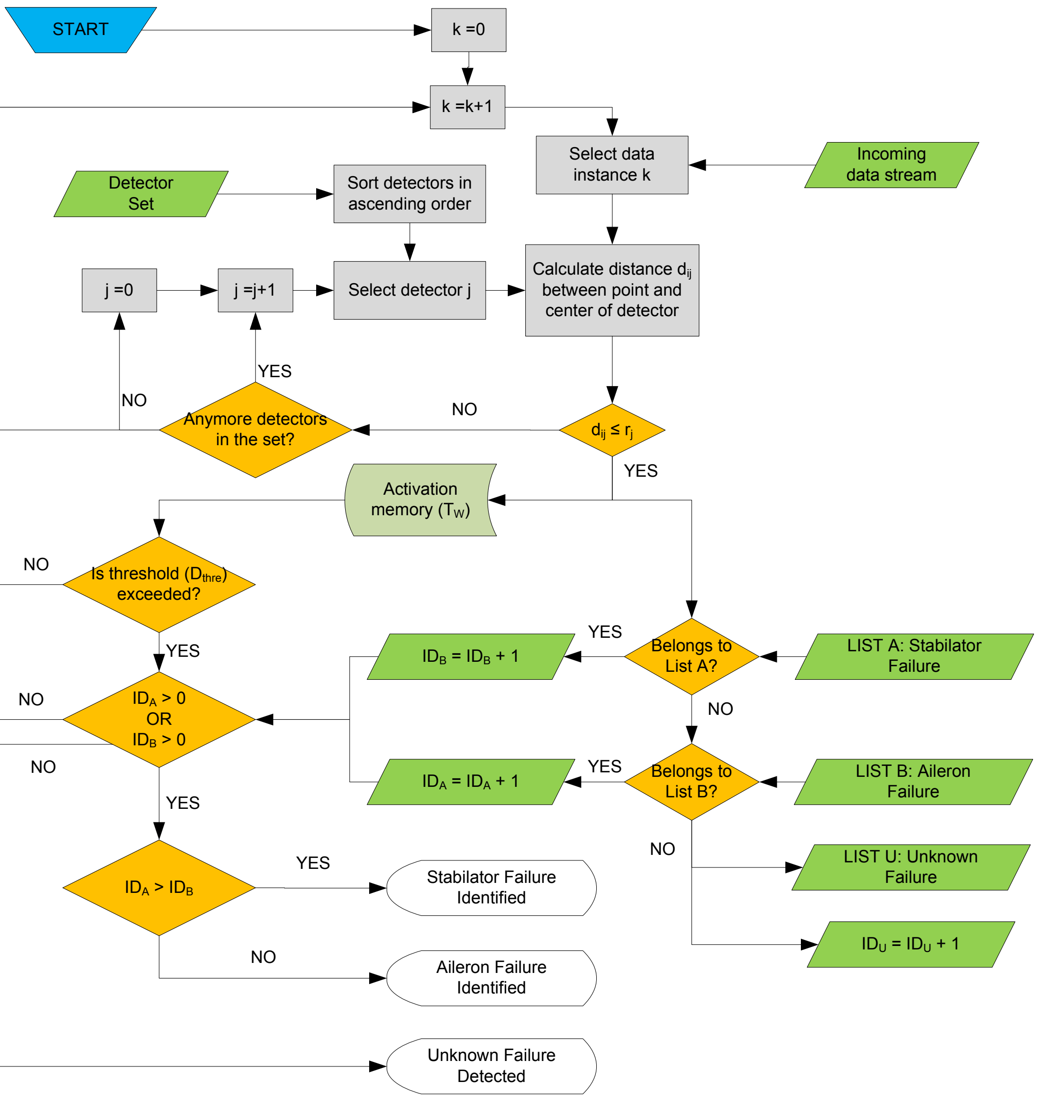


Figure 32: Logic of detection and identification for the created scheme.

The same flights used for the PSE algorithm are used to generate the lists in order to use the least amount of flights in training the system. The identification results are presented in detail in the following chapter.

Chapter 5: Analysis of the AIS FDI Scheme Performance

The flights that were not used for the training of the scheme are analyzed using the FDI scheme developed in the previous chapter. One plot for each particular failure/maneuver combination is presented together with general trends. The results are compiled in the form of a table at the end of this chapter.

5.1. Stabilator FDI

Stabilator failure is presented first to simplify the explanation of the plots used to characterize the results. The following is a figure exhibiting the outcome of the FDI process on Flight #4 consisting of a stabilator failure at trim followed by a stabilator doublet. The top subplot shows the normalized values of the different identifiers, while the bottom one shows the number of activated detectors at any given time. The red area is the part where the failure is present and thus where the scheme should get the most activation. The beginning and end of the failure are marked by darker red stems. Grey stems correspond to areas were all the surfaces are at trim and no failure is present. The black stem marks the detection time, while the blue stem marks the identification time. The dark horizontal line represents the detection threshold.

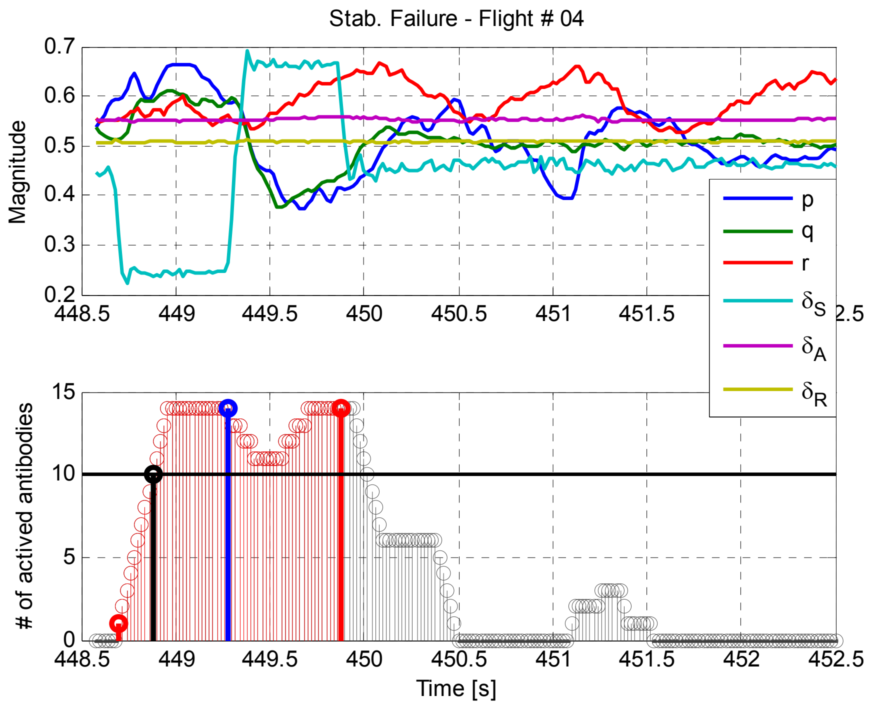


Figure 33: FDI – Stabilator failure – Flight #4.

As can be clearly seen, the results show a big increase in activation as soon as the failure becomes present and a gradual deactivation after the failure disappears. The effect of FA_{PP} is seen close to 451 seconds as some small amounts of activated detectors; however, these are far from reaching the threshold.

Most stabilator failures in contrast with the one shown do not require an extra time for identification, as by the time the activations reach the threshold, some of the activated detectors already coincide with the specialized identification list.

5.2. Validation Data

Validation data are formed using parts of different flights. The following figure presents the result of FDI for this validation data considered as one continuous data stream.

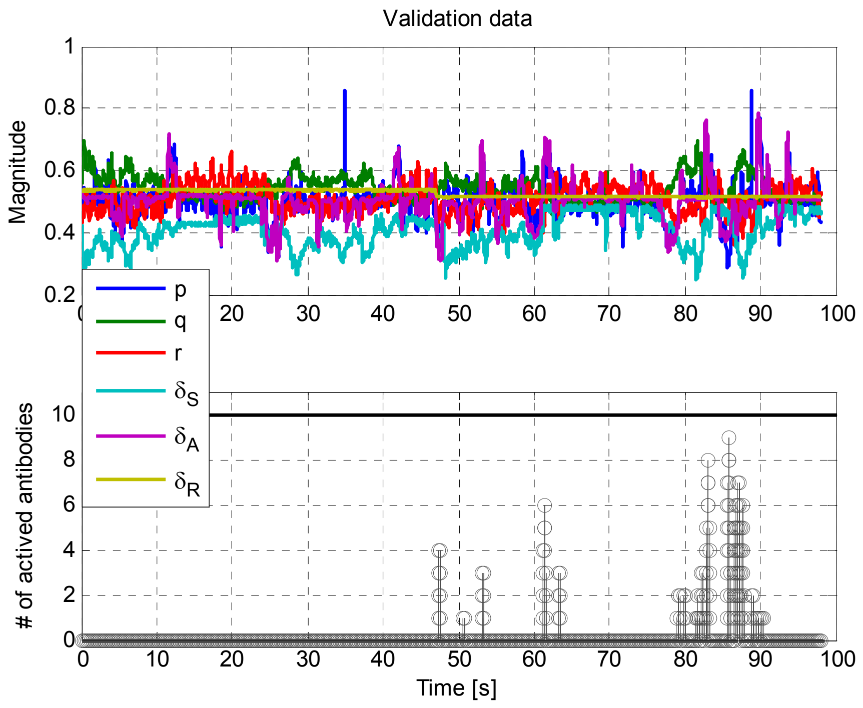


Figure 34: FDI – Validation data – Flight # 0

As stated before, even though the FA_{PP} for this case are rather low (1.37 %), most of the FA_{PP} are concentrated around a few points. This constitutes one of the main reasons for raising the threshold. For the combination of parameters chosen, the threshold is not reached and as can be seen throughout the rest of the flight, the probabilities for exceeding this threshold are somewhat low.

5.3. Aileron FDI

This section presents results separately for cases having a combo maneuver or just an aileron doublet after the aileron failure at trim.

5.3.1. Aileron FDI with aileron doublet

This failure is equivalent to the stabilator one presented previously. The analysis is done based on an equivalent plot.

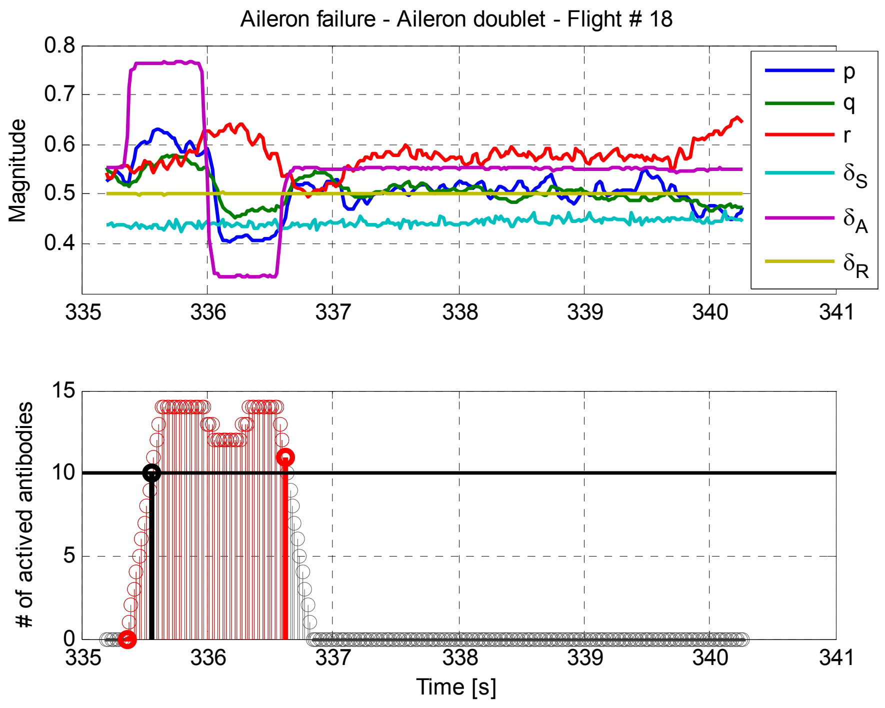


Figure 35: FDI – Aileron failure – Aileron doublet - Flight #18

In this case, the detection and identification are coincident with no delay in between. This is caused because some of the 10 detectors activated before reaching the threshold belong to the specialized aileron failure list. In this case, differently than what was showed for the stabilators, the FA_{PP} are inexistent.

5.3.2. Aileron FDI with combo maneuver

This maneuver is the most difficult to attain good detection rates due to the presence of the rudder input. The following figure shows one test case.

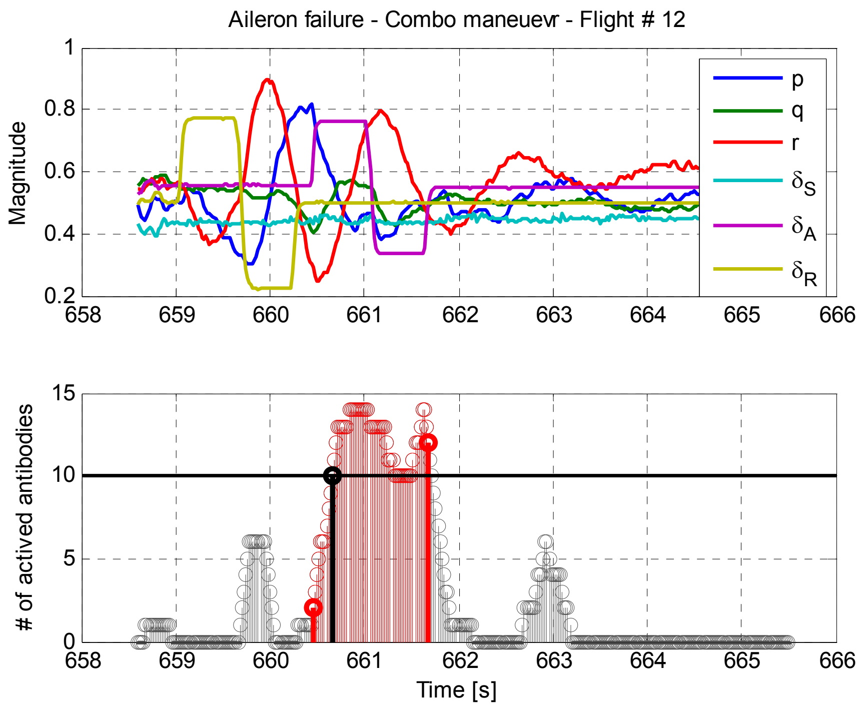


Figure 36: FDI – Aileron failure – Combo maneuver - Flight #12.

Even though the detection is performed in a very short time, the presence of two activation peaks of 6 before and after the aileron maneuver, represent a risk if a lower threshold or a larger time window are used. A more complicated situation can be seen in the following figure for a similar flight.

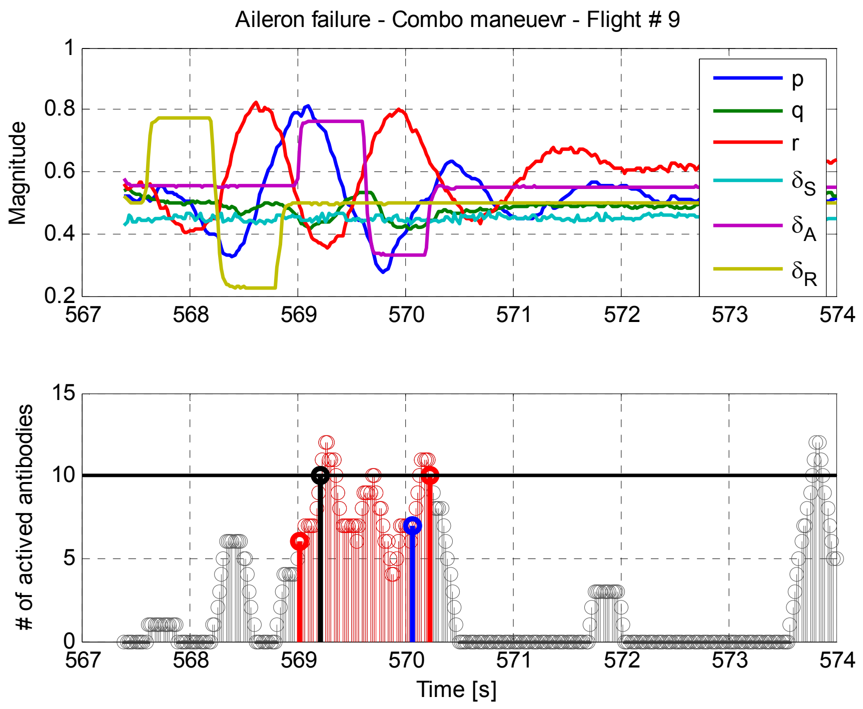


Figure 37: FDI – Aileron failure – Combo maneuver - Flight #9.

In this case, the problem is even more evident as a huge activation peak is found towards the end of the flight. This peak easily exceeds the threshold and thus triggers a false alarm. In the plot it can also be seen the big delay existing between detection and identification. This flight points to emphasize the need for more self data and/or better identifiers to avoid such undesirable qualities.

5.4. Sensor Bias FDI

It was described previously that flights 13 to 17 presented an unexpected bias in all the sensors. The idea of this section is to show the strength of the created FDI algorithm to detect a failure for which the system did not experience any training.

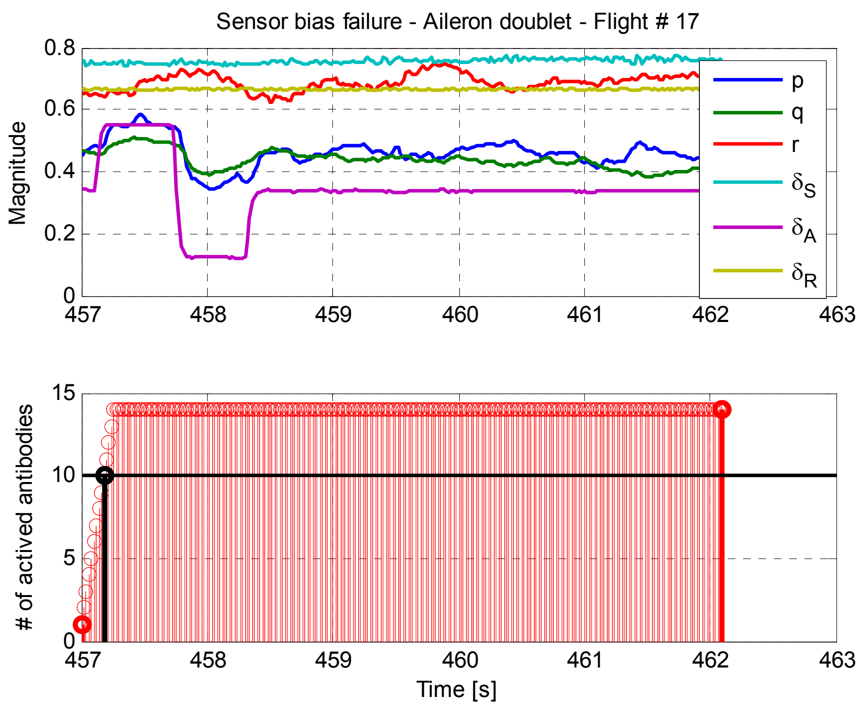


Figure 38: FDI – Sensor bias failure – Aileron maneuver - Flight #17

All five flights available for this type of failure resemble exactly the behavior shown above. The sensor bias is easily detected as soon as the flight starts. The system does not perform any identification due to the lack of specialized lists for this particular failure. The detectors found in this flight can be used to identify other similar sensor failures.

5.5. FDI Results Summary

The following table summarizes the performance of the FDI scheme in detecting abnormal conditions, declaring true failure detections and properly identifying the known failures.

ID #	Maneuver	Failure starts [s]	FDI						
			DR_{PP} [%]	FA_{PP} [%]	Detection		Identification		
					TD [s]	Delay [s]	TI [s]	Delay [s]	Result
0	Validation	N/A	0.00	1.35	ND	N/A	NI	N/A	N/A
1	Stabilator	357.50	100.00	0.00	357.68	0.18	357.68	0.18	Stabilator
2	Stabilator	387.82	72.13	7.64	388.10	0.28	388.10	0.28	Stabilator
3	Stabilator	418.38	90.16	1.41	418.62	0.24	418.62	0.24	Stabilator
4	Stabilator	448.70	95.00	8.76	448.88	0.18	449.28	0.58	Stabilator
5	Stabilator	479.26	90.16	0.00	479.48	0.22	479.48	0.22	Stabilator
6	Stabilator	509.58	65.57	2.63	510.08	0.50	510.08	0.50	Stabilator
7	Stabilator	540.14	75.41	0.71	540.46	0.32	540.46	0.32	Stabilator
8	R-A Combo	538.70	100.00	0.00	538.88	0.18	538.88	0.18	Aileron
9	R-A Combo	569.02	61.29	10.04	569.22	0.20	570.08	1.06	Aileron
10	R-A Combo	599.58	79.03	4. 83	599.74	0.16	599.74	0.16	Aileron
11	R-A Combo	629.88	80.65	7.49	630.06	0.18	630.06	0.18	Aileron
12	R-A Combo	660.46	87.10	5.28	660.66	0.20	660.66	0.20	Aileron
13	Aileron + Bias	335.20	100.00	0.00	335.38	0.18	NI	N/A	Unknown
14	Aileron + Bias	365.56	100.00	0.00	365.74	0.18	NI	N/A	Unknown
15	Aileron + Bias	396.10	100.00	0.00	396.28	0.18	NI	N/A	Unknown
16	Aileron + Bias	426.44	100.00	0.00	426.62	0.18	NI	N/A	Unknown
17	Aileron + Bias	457.00	100.00	0.00	457.18	0.18	NI	N/A	Unknown
18	Aileron	335.36	90.63	0.00	335.56	0.20	335.56	0.20	Aileron
19	Aileron	365.68	100.00	0.00	365.86	0.18	365.86	0.18	Aileron
20	Aileron	396.24	68.25	0.52	396.48	0.24	396.48	0.24	Aileron
21	Aileron	426.58	70.97	2.08	426.80	0.22	426.80	0.22	Aileron
22	Aileron	457.12	80.65	0.00	457.36	0.24	457.36	0.24	Aileron

Table 20: FDI results for the complete set of failed flights.

The results present an average TD of 0.23 seconds and an average TI of 0.34 seconds (excluding data used for training). The IR_F is 100% and the FI_F is 0%. Identification stage for this AIS scheme does not pose a difficulty. All tests run to verify the effects of varying the threshold showed that as long as the failure is detected, it will be always correctly identified as only one specialized detector is needed to declare that a failure has been identified.

Another interesting result is the actual time it takes for the scheme to provide the above presented results. The Real Time Percentage is calculated using the following formula:

$$RT_{\%} = \frac{T_A}{T_F} \cdot 100\% \qquad (5.1)$$

Where T_A is the actual time used by the algorithm and T_F the real duration of the flight. The following table summarizes all these values.

ID #	Maneuver	Flight duration [s]	Algorithm duration [s]	$RT_{\%}$ [%]
0	Validation	98.02	104.71	106.82
1	Stabilator	4.08	3.14	77.04
2	Stabilator	4.34	3.52	81.02
3	Stabilator	4.04	3.08	76.26
4	Stabilator	3.92	2.80	71.46
5	Stabilator	4.22	3.26	77.33
6	Stabilator	4.24	3.58	84.35
7	Stabilator	4.00	3.24	80.95
8	R-A Combo	6.46	5.53	85.56
9	R-A Combo	6.60	5.80	87.88
10	R-A Combo	6.60	5.73	86.77
11	R-A Combo	6.56	5.62	85.69
12	R-A Combo	6.90	5.88	85.17
13	Aileron + Bias	5.06	3.14	61.99
14	Aileron + Bias	5.06	3.17	62.63
15	Aileron + Bias	5.10	3.24	63.46
16	Aileron + Bias	5.06	3.14	62.11
17	Aileron + Bias	5.10	3.25	63.64
18	Aileron	5.06	4.13	81.69
19	Aileron	5.06	4.29	84.78
20	Aileron	5.10	4.38	85.81
21	Aileron	5.06	4.38	86.51
22	Aileron	5.10	4.39	86.11

Table 21: Evaluation of potential real time application.

The previous table shows that for most of the flights, the time spent by the detection and identification algorithm is less than the real time of the flight. However, as was expected, the biggest difference is obtained for nominal data as the activation is very low. Another important aspect that can be seen in the previous table is that flights 13 to 17 are the fastest ones to run as all the points in the flight activate detectors. On the other hand, these same flights after removing the bias (flights 18 to 22) take longer to run due to the reduced activation.

5.6. Comparison of AIS Results with Other FDI Methods

The training of the two schemes is performed using the same limited set of flights. The training set consists of:

- 2 Healthy flights
- 2 Stabilator locked at 5 degrees failed flights
- 2 Stabilator locked at 8 degrees failed flights
- 2 Aileron locked at 5 degrees failed flights
- 2 Aileron locked at 8 degrees failed flights

The training set includes different injection points and also different pilots. The following is a list of the flights used with their particular information.

Flight ID	Failure	Injection Point	Injection time [s]
A001	No failure	N/A	N/A
S001	No failure	N/A	N/A
A010	L. Stab. 5 deg.	Aileron doublet	141.84
A011	L. Stab. 5 deg.	After Aileron doublet	165.10
A015	L. Ail. 5 deg.	Left turn	42.40
A019	L. Ail. 5 deg.	Aileron doublet	118.34
S063	L. Stab. 8 deg.	After Stabilator doublet	167.18
S064	L. Stab. 8 deg.	Aileron doublet	147.26
A073	L. Ail. 8 deg.	Aileron doublet	150.18
S073	L. Ail. 8 deg.	Aileron doublet	232.88

Table 22: Training flights for comparison of FDI schemes.

In the same way, the following is the list of the flights used for testing the two FDI schemes.

Flight ID	Failure	Injection Point	Injection time [s]
A002	No failure	N/A	N/A
A003	No failure	N/A	N/A
A005	No failure	N/A	N/A
A006	L. Stab. 5 deg.	Left turn	57.26
A007	L. Stab. 5 deg.	After left turn	148.70
A018	L. Ail. 5 deg.	After Stabilator doublet	159.44
A021	L. Ail. 5 deg.	Rudder doublet	155.06
A064	L. Stab. 8 deg.	Aileron doublet	166.14
S008	L. Stab. 5 deg.	Stabilator Doublet	106.58
S015	L. Ail. 5 deg.	Left turn	94.32
S021	L. Ail. 5 deg.	Right turn	180.94

Table 23: Testing flights for comparison of FDI schemes.

5.6.1. NN FDI scheme for WVU F-15 simulator

Tuning for this scheme was done using the limited set of nominal and failed flights described above. After performing many trials following the process outlined previously, a successful set of parameters was found. In contrast with [80], in this case only the Soft Bound (SB) values are presented as this is the only threshold used in the applied logic.

Variable	β_{SB}	b_{SB}
$MQEE$	2.50	0.0000060
$OQEE$	3.00	0.0000015
R_{pq}	0.05	0
R_{rr}	2.00	0.001
ω_{pq}	0.60	0.0001

Table 24: Floating limiter tuned parameters

The results obtained using this modified scheme are presented in the following table.

Flight ID	Failure	Injection Point	Detect. [s]	Isol. [s]	Id. [s]	Result
A001	No failure	N/A	N/A	N/A	N/A	N/A
S001	No failure	N/A	N/A	N/A	N/A	N/A
A010	L. Stab. 5 deg.	Aileron doublet	0.22	0.24	0.42	Actuator/Stabilator
A011	L. Stab. 5 deg.	After Aileron doublet	0.24	0.26	0.44	Actuator/Stabilator
A015	L. Ail. 5 deg.	Left turn	0.20	0.48	0.66	Actuator/Aileron
A019	L. Ail. 5 deg.	Aileron doublet	0.14	0.16	0.18	Actuator/Aileron
S063	L. Stab. 8 deg.	After Stabilator doublet	0.14	0.16	0.34	Actuator/Stabilator
S064	L. Stab. 8 deg.	Aileron doublet	0.12	0.14	0.32	Actuator/Stabilator
A073	L. Ail. 8 deg.	Aileron doublet	0.12	0.14	0.20	Actuator/Aileron
S073	L. Ail. 8 deg.	Aileron doublet	0.14	0.16	0.34	Actuator/Aileron
A002	No failure	N/A	N/A	N/A	N/A	N/A
A003	No failure	N/A	N/A	N/A	N/A	N/A
A005	No failure	N/A	N/A	N/A	N/A	N/A
A006	L. Stab. 5 deg.	Left turn	0.22	0.24	0.26	Actuator/Aileron
A007	L. Stab. 5 deg.	After left turn	0.20	0.28	0.40	Actuator/Stabilator
A018	L. Ail. 5 deg.	After Stabilator doublet	0.16	0.42	0.44	Actuator/Aileron
A021	L. Ail. 5 deg.	Rudder doublet	0.48	0.88	0.90	Actuator/Aileron
A064	L. Stab. 8 deg.	Ail. Doublet	0.20	0.22	0.24	Actuator/Aileron
S008	L. Stab. 5 deg.	Stabilator Doublet	0.16	0.18	0.36	Actuator/Aileron
S015	L. Ail. 5 deg.	Left turn	0.16	0.18	0.20	Actuator/Aileron
S021	L. Ail. 5 deg.	Right turn	0.20	0.22	0.24	Actuator/Aileron

Table 25: FDI results for training and testing flights using FL+NN scheme.

The table presents first the training flights, which, as expected, are all detected and identified correctly. The test flights are included below the darker black line. For these flights, the detection is still perfect ($DR_F = 100\%$), however, the identification fails in 3 out of the 8 failed flights shown ($IF_F = 37.5\%$). The Detection, Isolation, and Identification times presented correspond to the delay since the failure is injected. The averages of these delays are:

- Average Detection delay: 0.19 sec
- Average Isolation delay: 0.27 sec
- Average Identification delay: 0.37 sec

The first drawback of this approach is the lack of an automated training system. As the parameters need to be manually tuned using a total of 10 flights, the process is quite time consuming and an optimal set of values is not necessarily found. The results show that the maneuver injection point may affect the identification logic producing confusion. A more comprehensive set of flights should be used for training thus allowing the scheme to learn the different failure signatures in different flight conditions.

The main drawback of this FDI scheme is the lack of a flexible and automated training. In contrast, its benefits are its low computational cost that has already allowed an online implementation as shown in [80].

5.6.2. AIS FDI scheme for WVU F-15 simulator

The identifiers used for this implementation are the three measured angular rates (p , q , r) and the reference angular rates (p_{ref} , q_{ref} , r_{ref}). This later variables are the output of a reference model contained inside the simulation. This model consists of a linear model of the F-15 and its inputs are directly pilot joystick movements. These reference values are assumed to be comparable with using directly pilot input or healthy surface deflections as in the WVU YF-22 implementation.

The self was obtained as described in Chapter 3 from the data generated for the implementation presented in [72]. The data is normalized and clustered using the presented algorithms and 4999 clusters are created using hyper-spheres and Euclidean distance. Tests to assess the benefit of changing the distance definition concluded that Euclidean distance presented the best performance. This performance was evaluated as shown for the case of the WVU YF-22 using Positive Detection.

The ENSA-RV algorithm was used to create 500 detectors. Different sizes of detector sets were tried, but no significant improvement was experienced. The Positive Selection Enhancement is applied to this 500 detector set using the training flights in the following order:

1. A010: Stabilator failed at 5 degrees
2. A011: Stabilator failed at 5 degrees
3. A015: Aileron failed at 5 degrees
4. A019: Aileron failed at 5 degrees
5. S063: Stabilator failed at 8 degrees
6. S064: Stabilator failed at 8 degrees
7. A073: Aileron failed at 5 degrees
8. S073: Aileron failed at 5 degrees
9. A001: Nominal flight
10. S001: Nominal flight

The two nominal flights are left for the end as these will probably remove many detectors that can cause false activations. The time window and threshold used are the same as presented for the WVU YF-22 implementation.

It is important to note that as the flight consists of many different stages, it is possible that the variables may return to the self if the set of identifiers is not complete and/or the coverage is not adequate. The consequence is that not all the data that are considered failed will be in fact found outside the self, thus the detection rates are considerably lower than the ones presented for the case of the WVU YF-22.

Flight ID	Failure	Injection Point	DR_{PP} [%]	FA_{PP} [%]	Det. [s]	Id. [s]	Result
A001	No failure	N/A	0.00	0.00	N/A	N/A	N/A
S001	No failure	N/A	0.00	0.00	N/A	N/A	N/A
A010	L. Stab. 5 deg.	Aileron doublet	33.38	0.00	0.24	0.24	Actuator/Stabilator
A011	L. Stab. 5 deg.	After Aileron doublet	24.78	0.00	0.38	0.38	Actuator/Stabilator
A015	L. Ail. 5 deg.	Left turn	16.65	0.00	0.74	0.74	Actuator/Aileron
A019	L. Ail. 5 deg.	Aileron doublet	21.68	0.00	0.74	0.74	Actuator/Aileron
S063	L. Stab. 8 deg.	After Stabilator doublet	30.01	0.00	0.32	0.32	Actuator/Stabilator
S064	L. Stab. 8 deg.	Aileron doublet	41.58	0.00	0.34	0.34	Actuator/Stabilator
A073	L. Ail. 8 deg.	Aileron doublet	11.94	0.00	0.88	0.88	Actuator/Aileron
S073	L. Ail. 8 deg.	Aileron doublet	24.47	0.00	0.26	0.26	Actuator/Aileron
A002	No failure	N/A	0.00	0.06	N/A	N/A	N/A
A003	No failure	N/A	0.00	0.04	N/A	N/A	N/A
A005	No failure	N/A	0.00	0.12	FALSE	N/A	Unknown Failure
A006	L. Stab. 5 deg.	Left turn	7.87	0.00	0.46	0.46	Actuator/Stabilator
A007	L. Stab. 5 deg.	After left turn	8.99	0.03	0.40	0.40	Actuator/Stabilator
A018	L. Ail. 5 deg.	After Stabilator doublet	1.99	0.30	11.82	11.82	Actuator/Aileron
A021	L. Ail. 5 deg.	Rudder doublet	2.88	0.43	12.52	12.52	Actuator/Aileron
A064	L. Stab. 8 deg.	Ail. Doublet	4.99	0.36	0.40	0.40	Actuator/Stabilator
S008	L. Stab. 5 deg.	Stabilator Doublet	10.56	0.00	0.46	0.46	Actuator/Stabilator
S015	L. Ail. 5 deg.	Left turn	1.65	0.00	75.78	75.78	Actuator/Aileron
S021	L. Ail. 5 deg.	Right turn	3.09	0.52	FALSE	FALSE	Actuator/Aileron

Table 26: FDI results for training and testing flights using the NS approach.

As for the FL + NN scheme, the first 10 flights shown correspond to the training flights and as such have the maximum attainable DR_{PP} with zero FA_{PP}. All the training flights are of course detected and identified correctly. For the testing flights (below the darker horizontal line), two flights out of the eleven present unsuccessful detection and/or identification results. Flight A005, a nominal flight, presents a false alarm (something not encountered for the FL + NN scheme); however, the scheme does not identify the false alarm thus categorizing the failure as Unknown. Flight S021 presents also a false alarm as the threshold is met before the failure hits. In this case, as the failure eventually is present, the identification is correct.

Due in part to the rather low DR_{PP} presented by the testing flight, the detection times are enlarged from those found for the training flights. Flights A007 and A018 present a delay of more than 10 seconds to detect the failure while the worst case is flight S015 which presents a delay of more than one minute. The experienced delays can partially be attributed to lack of proper coverage of the non-self. Other cause can be the fact that this simulation contains an adaptive controller that reduces the effects of failures due to adaptation. This adaptation makes necessary that, to have a comprehensive set of identifiers, it would be required to include variables that contain information about the behavior of the controller.

Training for this scheme is rather easy albeit time consuming. This is not a burden as the scheme is trained offline. As for the case of the FL + NN scheme, this one would also get benefits of using more training data, however, the main problems encountered in this implementation are related to the selected identifiers. The identifiers used for the WVU YF-22 FDI scheme, were selected out of the limited amount of variables available and it was shown previously that most of the difficulties in attaining good detection capabilities were produced by the lack of a comprehensive set of identifiers. In spite of that, the present identifiers were chosen to resemble as close as possible that used for the WVU YF-22 to allow an analysis of the effect of using not so strong identifiers in a more demanding flying scenario. The results suggest that it will be very difficult if no impossible to obtain a successful FDI scheme out of using just angular rates and reference angular rates. The NN derived parameters used for the FL + NN scheme have been shown to capture actuator failure signatures in [26], so it would be expected that a self using such parameters would present better overall results as those shown in [72].

As a summary, the main drawbacks of this approach are the need for a better selection of identifiers and the time consumed for training it. On the other hand, its main benefits include that it possess the potential of detecting untrained failures, and the possibility of an easy and automated training.

5.6.3. Analysis of pilot situational awareness possible increase thanks to FDI scheme

The following table presents the pilot detection results for the training and test sets. The detection is recorded by the pilot pressing a button in the joystick whenever he senses the presence of an abnormal condition. The results of identification are obtained at the end of each flight and registered manually.

Flight ID	**Failure**	**Injection Point**	**FL + NN Det. [s]**	**NS Det. [s]**	**Pilot Det. I [s]**	**Pilot Det. II [s]**	**Pilot Id.**
A001	No failure	N/A	N/A	N/A	N/A	N/A	No Failures
S001	No failure	N/A	N/A	N/A	FALSE	N/A	No Failures
A010	L. Stab. 5 deg.	Aileron doublet	0.24	0.24	1.42	N/A	Actuator
A011	L. Stab. 5 deg.	After Aileron doublet	0.26	0.38	1.00	N/A	Yaw sensor
A015	L. Ail. 5 deg.	Left turn	0.48	0.74	2.14	N/A	Actuator/Aileron
A019	L. Ail. 5 deg.	Aileron doublet	0.16	0.74	N/A	N/A	No Failures
S063	L. Stab. 8 deg.	After Stabilator doublet	0.16	0.32	1.28	N/A	Actuator/Stabilator /Right Stab. Up
S064	L. Stab. 8 deg.	Aileron doublet	0.14	0.34	FALSE	1.32	Sensor
A073	L. Ail. 8 deg.	Aileron doublet	0.14	0.88	1.34	N/A	Actuator/Aileron
S073	L. Ail. 8 deg.	Aileron doublet	0.16	0.26	1.08	N/A	Failure/No Id.
A002	No failure	N/A	N/A	N/A	N/A	N/A	No Failures
A003	No failure	N/A	N/A	N/A	N/A	N/A	No Failures
A005	No failure	N/A	N/A	FALSE	FALSE	N/A	Failure/No Id.
A006	L. Stab. 5 deg.	Left turn	0.24	0.46	0.90	N/A	Actuator/Aileron
A007	L. Stab. 5 deg.	After left turn	0.28	0.40	0.90	N/A	Failure/No Id.
A018	L. Ail. 5 deg.	After Stabilator doublet	0.42	11.82	1.06	N/A	Failure/No Id.
A021	L. Ail. 5 deg.	Rudder doublet	0.88	12.52	1.12	N/A	Actuator/Aileron
A064	L. Stab. 8 deg.	Ail. Doublet	0.22	0.40	1.00	N/A	Actuator/Stabilator
S008	L. Stab. 5 deg.	Stabilator Doublet	0.18	0.46	FALSE	1.98	Actuator/Stabilator /Locked left stab.
S015	L. Ail. 5 deg.	Left turn	0.18	75.78	1.50	N/A	Actuator/Rudder or Stabilator
S021	L. Ail. 5 deg.	Right turn	0.22	FALSE	FALSE	1.76	Actuator/Aileron

Table 27: Detection delays for the two FDIs and for the pilot – Pilot identification.

The first thing that can be noticed from the preceding table is that the human pilots present more False Alarms than any of the FDI schemes presented. Moreover, it can be seen that the average time it takes for the pilots to detect the failures is around 1.37 seconds. It should be noted that these times include perception, processing, and reaction times and thus care should be taken in directly comparing the detection delays. The two columns for Pilot detection delay are presented because for some flights, after a false detection, the pilots were able to correctly detect the true failure.

These false alarms are attributed to the fact that, after several test sessions, the pilots start expecting the occurrence of failures and so become oversensitive in declaring. According to both pilots employed for this preliminary comparison, their false detections were mainly produced by their perception of extraneous movements of the motion base.

Regarding Identification skills, the pilots present some outstanding cases like flight S008 were the left side is correctly identified as failed. Nevertheless, the pilots present a couple of flights in which a failure is detected but they could not perform any identification like flights A007, A018 and A019. Finally, they also present confusion with completely different failures like flight A011 for which the pilot declared a Yaw Sensor failure.

Considering that these are just partial results only for a limited set of failures, it is believed that the biggest aid that the pilots will experience when exposed to interaction with an FDI scheme will be the availability of more precise information on identification of the particular failures. Fast and correct identification is assumed to facilitate the pilot's ability to perform correctly the flight scenario as introduced in [80]; however, comprehensive metrics for proving this concept need to be designed and tested with and without providing the pilot FDI information.

Chapter 6: CONCLUSIONS

Two Negative Selection algorithms were investigated and their performance evaluated. ENSA-RV algorithm was evidenced to produce limited coverage in the close proximity of the self, while Algorithm #2 was successful in solving this problem. However, ENSA-RV used more thorough development techniques that gave the resulting detector set a smaller size compared to that of the Algorithm #2.

A Positive Selection algorithm was proposed to enhance detector sets using training failed flights. This algorithm proved to be a very useful tool to obtain detectors in poorly explored areas of the non-self while also reducing false activations by removing detectors in poorly defined areas of the self.

The effects of changing the sensitivity parameters were shown for the known failures and these results can be useful if a more comprehensive set of faulty flights is used. This process is believed to be greatly simplified if the available identifiers are capable of exhibiting the failure signature's more clearly and preferably farther away from the self.

All in all, the Negative Selection approach to failure detection proved to be a valid technique that can cope with a variety of faulty situations. It was shown how with a low amount of training data a successful FDI scheme can be obtained. This was accomplished with a reduced list of identifiers and a limited optimization process. The possibility offered by this scheme of identifying failures as "unknown" is seen as one of the greatest benefits of this technique as compared to other FDI techniques.

The preliminary comparison between two different FDI schemes put the AIS based one in a slightly better position considering its ease of training. The FL + NN based FDI scheme presents a valid approach; however, in the absence of a better training method than trial and error, its application to a wide variety of failures is largely limited. The AIS based scheme has as another interesting and desirable benefit, the possibility of defining a failure as unknown which would be the case for a failure not used in the training process. This advantage is of great importance when considering a comprehensive FDI scheme.

Chapter 7: RECOMMENDATIONS AND FUTURE WORK

Some of the possible paths that can aid to improve the Failure Detection Scheme for the WVU YF-22 UAV based on the Negative Selection principle include, but are not limited to:

1. A specific set of flight test should be designed and performed in order to obtain a broader set of nominal data that includes all three control surfaces excited in a variety of flight conditions to allow for the creation of a more complete and accurate self.
2. Configure the On-board instrumentation to acquire more parameters that can serve as better identifiers. Some of these desired variables are outputs of reference models and direct measure of pilot inputs.
3. Perform failed flights with a wider variety of failures including surface of different sides and also contemplate the presence of failures for longer periods of time during the flight. However it should be perfectly clear that security must always be the first concern when designing these flights.
4. Test other detectors shapes such as hyper-rectangles or hyper-ellipsoids in combination with different definitions of distance to assess the suitability of each of these for the hyper-space defined by the selected identifiers.
5. If needed, Algorithm #2 should be improved to reduce overlapping.

Point 2 may allow for the creation of a SELF in which the failures can be found farther away from the self than for the ones implemented in this thesis. This may allow the use of the ENSA-RV algorithm without even needing PSE algorithm. If this is the case, phase 2 of the algorithm presented in [73] could also be used to optimize the detector set.

If points 1 and 2 allow the creation of an FDI scheme that can perform in real time with the available computing capabilities of the onboard processor, the performance of the scheme could be tested in flight to assess its strengths and weaknesses when exposed to real conditions. If this online implementation is available, techniques for AIS online learning can be tested.

Regarding the comparison of the AIS approach to other FDI schemes, the proposed scenario is assumed to provide enough information. However, a more thorough definition of how many flights and of which type will be use to train/tune the FDI schemes. For the case of the FL approach using NNs outputs, it is also important to create an automated way to train it because tuning it for two actuator failures took a considerable effort via trial and error. The complexity of the tuning increases as more failures are added because more signatures need to be interpreted and successfully included within the detection and identification logic.

For the case of the AIS approach, an extended set of identifiers should be used in order to include information of control system activity. This should not be a burden as the simulation used provides a vast amount of information.

During the comparison, the results of the FDIs were contrasted with human pilot detection capabilities. The next step would be to obtain measures of pilot performance in the absence of an FDI scheme and using the FDI to generate visual messages on the cockpit that will inform the pilot of the particular failed element. Metrics for assessing pilot's performance need to be developed.

Chapter 8: BIBLIOGRAPHY

[1]. **FAA.** History. *Federal Aviation Adiministration.* [Online] Updated: 10:23 am ET March 3, 2005, FAA. [Cited: March 02, 2009.] http://www.faa.gov/about/history/brief_history/.

[2]. **Kebabjian, R.** Statistics. *PlaneCrashInfo.com.* [Online] [Cited: March 04, 2009.] http://www.planecrashinfo.com/cause.htm.

[3]. **National Transportation Safety Board (NTSB).** National Transportation Safety Board. *Publications.* [Online] [Cited: March 23, 2009.] http://www.ntsb.gov/Publictn/publictn.htm.

[4]. **White, J.** Aeronautics Research Mission Directorate. *Referenced Materials: Presentations.* [Online] January 12, 2006. [Cited: February 27, 2009.] http://www.aeronautics.nasa.gov/reno_presentations/avsp_reno_011206.pdf.

[5]. *Intelligent Control Approaches for Aircraft Applications.* **Gundy-Burlet, K., et al.** Destin, FL, USA : NASA, 2001. JANNAF Interagency Propulsion Committee Meeting. Doc. ID: 20020054263.

[6]. *Intelligent Systems For Aerospace Engineering--An Overview.* **KrishnaKumar, K.** Belgium : National Technical Information Service (NTIS), 2002. von Karman Lecture series on Intelligent Systems for Aeronautics. Prod. code: N20020065377.

[7]. **Wilsky, A. S.** *Failure detection in dynamic systems.* Neuilly-sur-Seine, France, 1980. Agard Rep. LS-109.

[8]. *Observer-based strategies for actuator fault detection, isolation and estimation for certain class of uncertain nonlinear systems.* **Chen, W. and Saif, M.**, Institution of Engineering and Technology, Six Hills Way, Stevenage, UK, 2007, IET Control Theory and Applications, Vol. 1, Iss. 6, pp. 1672-1680.

[9]. *Adaptive observer based actuator fault diagnosis for ASV.* **Wang, J., et al.** Dalian, Liaoning, China : Inst. of Elec. and Elec. Eng. Computer Society, Piscataway, NJ, USA, 2008. 2008 3rd International Conference on Innovative Computing Information and Control. ISBN: 9780769531618.

[10]. *Kalman filters and neural-network schemes for sensor validation in flight control systems.* **Napolitano, M. R., et al.**, IEEE, Piscataway, NJ, USA, September 1998, IEEE Transactions on Control Systems Technology, Vol. 6, Iss. 5, pp. 596-611. ISSN: 10636536.

[11]. *A multiple model predictive scheme for fault-tolerant flight control design.* **Gopinathan, M., et al.**, IEEE, 1998. Proceedings of the 37th IEEE Conference on Decision and Control. Vol. 2, pp. 1376-81. ISBN-10: 0780343948.

[12]. *Evaluation of a multiple-model failure detection system for the F-16 in a full-scale nonlinear simulation.* **Eide, P. and Maybeck, P.** Dayton, OH, USA : IEEE, 1995. Aerospace and Electronics Conference, 1995. NAECON 1995., Proceedings of the IEEE 1995 National. Vol. 1, pp. 531 - 536. ISBN-10: 0780326660.

[13]. *Analytical redundancy and the design of robust failure detection systems.* **Chow, E. and Willsky, A.**, IEEE, July 1984, IEEE Transactions on Automatic Control, Vols. AC-29, Iss. 7, pp. 603-614. ISSN: 00189286.

[14]. *Parity relation approach to fault diagnosis in manipulation robots.* **Filaretov, V. F., Vukobratovic, M. K. and Zhirabok, A. N.**, Elsevier Science, Oxford, ROYAUME-UNI, 2003, Mechatronics, Vol. 13, Iss. 2, pp. 141-152. ISSN: 0957-4158.

[15]. *Fault detection and isolation in a greenhouse using parity relations.* **Kabbaj, N., et al.** Lisbon, Portugal : IEEE, 2003. Proceedings of 2003 IEEE Conference on Emerging Technologies and Factory Automation (EFTA). Vol. 2, pp. 747 - 752. ISBN: 0780379373.

[16]. *Sensor failure detection using generalized parity relations for flexible structures.* **Mercadal, M.**, AIAA, 1989, Journal of Guidance, Control, and Dynamics, Vol. 12, Iss. 1, pp. 125-127. ISSN: 07315090.

[17]. *Automatic control system failure detection via parameter identification techniques.* **Broussard, K. J. and Trahan, R. E., Jr.** Williamsburg, VA, USA : IEEE, 1991. IEEE Proceedings of SOUTHEASTCON '91. Vol. 1, pp. 176-80. ISBN-10: 0780300335.

[18]. *Rocket Engine Failure Detection Using System Identification Techniques.* **Meyer, C. M. and Zakrajsek, J. F.** Orlando, FL, USA : AIAA, 1990. SAE, ASME, and ASEE, 26th Joint Propulsion Conference. pp. 16-18.

[19]. *Parameter identification for inflight detection and characterization of aircraft icing.* **Melody, J. W., et al.**, Elsevier Science Ltd., September 2000, Control Engineering Practice, Vol. 8, Iss. 9, pp. 985-1001. ISSN: 09670661.

[20]. *Application of neural networks to adaptive control.* **Elsley, R. K. and Lan, M.-S.** Pacific Grove, CA, USA : Maple Press, 1989. Conference Record. Twenty-Second Asilomar Conference on Signals, Systems and Computers. Vol. 2, pp. 517-22. ISSN: 1989-01195.

[21]. *Pattern-based fault diagnosis using neural networks.* **Dietz, W. E., Kiech, E. L. and Ali, M.** Tullahoma, Tn, USA : ACM, 1988. The First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE - 88. Vol. 2, pp. 13-23. ISBN-10: 0897912713.

[22]. *On-line learning neural architectures and cross-correlation analysis for actuator failure detection and identification.* **Napolitano, M. R., et al.**, Taylor & Francis Ltd, Basingstoke, UK, February 1996, International Journal of Control, Vol. 63, Iss. 3, pp. 433–455. ISSN: 00207179.

[23]. *A fault tolerant flight control system for sensor and actuator failures using neural networks.* **Napolitano, M. R., Younghawn, A. and Seanor, B.**, Elsevier Ltd., June 2000, Aircraft Design, Vol. 3, Iss. 2, pp. 103-128. ISSN: 13698869.

[24]. *Sensor validation using hardware-based on-line learning neural networks.* **Napolitano, M. R., et al.**, IEEE, April 1008, IEEE Transactions on Aerospace and Electronic Systems, Vol. 34, Iss. 2, pp. 456-68. ISSN: 0018-9251.

[25]. *Online learning neural architectures and cross-correlation analysis for actuator failure detection and identification.* **Napolitano, M. R., et al.**, Taylor & Francis, February 1996, International Journal of Control, Vol. 63, Iss. 3, pp. 433-55. ISSN: 0020-7179.

[26]. *An adaptive threshold approach for the design of an actuator failure detection and identification scheme.* **Perhinschi, M.G., et al.**, IEEE, May 2006, IEEE Transactions on Control Systems Technology, Vol. 14, Iss.3, pp. 519-25. ISSN: 10636536.

[27]. *Integration of Fault Tolerant System for Sensor and Actuator Failures within the WVU NASA F-15 Simulator.* **Perhinschi, M., et al.** Austin, TX, USA : AIAA, 2003. AIAA Guidance Navigation and Control Conference. AIAA-2003-5644.

[28]. *Adaptive neural network inverse controller for general aviation safety.* **Pesonen, U. J., et al.**, AIAA, May-June 2004, Journal of Guidance, Control, and Dynamics, Vol. 27, Iss. 3, pp. 434-43. ISSN: 0731-5090.

[29]. *Fault diagnosis based on improved Elman neural network for a hydraulic servo system.* **H., Liu, S., Wang and P., Ouyang.** Bangkok, Thailand : IEEE, 2006. 2006 IEEE Conference on Robotics, Automation and Mechatronics. ISBN-10: 1424400244.

[30]. *Tool failure diagnosis in milling using a neural network.* **Tarng, Y.S., Hwang, S.T. and Hseih, Y.W.**, Elsevier B.V., January 1994, Mechanical Systems and Signal Processing, Vol. 8, Iss. 1, pp. 21-9. ISSN: 0888-3270.

[31]. *Robust fault detection using robust l1 estimation and fuzzy logic.* **Curry, T., Collins, E.G., Jr. and Selekwa, M.** Arlington, VA, USA : IEEE, 2001. Proceedings of the 2001 American Control Conference. Vol. 2, pp. 1753-1758. ISBN: 0780364953.

[32]. *A fuzzy controlled neural network for sensor fusion with adaptability to sensor failure.* **Chen, J., et al.** San Diego, CA, USA : SPIE-Int. Soc. Opt. Eng, 1997. Proceedings of the SPIE - The International Society for Optical Engineering. Vol. 3165, pp. 283-91. ISSN: 0277-786X.

[33]. **Dasgupta, D., [ed.].** *Artificial Immune Systems and Their Applications.* s.l. : Springer, 1998. ISBN: 3540643907.

[34]. *Self-nonself discrimination in a computer.* **Forrest, S., et al.** Oakland, CA, USA : IEEE, 1994. Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy. pp. 202-12. ISBN-10: 0818656751.

[35]. **Dasgupta, D. and Forrest, S.** An Anomaly Detection Algorithm Inspired by the Immune System. [book auth.] D. Dasgupta. [ed.] D. Dasgupta. *Artificial Immune Systems and Their Application.* s.l. : Springer-Verlag Berlin Heidelberg New York, 1998, 14, pp. 262-277.

[36]. *Negative Selection Algorithm for Aircraft Fault Detection.* **Dasgupta, D., KrishnaKumar, K., Wong, D. and Berry, M.** [ed.] G. Nicosia, et al. Catania, Sicily, Italy : Springer Berlin / Heidelberg, 2004. LNCS: Artificial Immune Systems. Third International Conference, ICARIS 2004. Vol. 3239, pp. 1-13. ISBN-13: 978-3-540-23097-7.

[37]. *Towards a Conceptual Framework for Artificial Immune Systems.* **Stepney, S., et al.** [ed.] G. Nicosia, et al. Catania, Sicily, Italy : Springer Berlin / Heidelberg, 2004. LNCS: Artificial Immune Systems. Third International Conference, ICARIS 2004. Vol. 3239, pp. 53-64. ISBN-13: 978-3-540-23097-7.

[38]. *Theoretical Basis of Novelty Detection in Time Series Using Negative Selection Algorithms.* **Pasek, R.** [ed.] H. Bersini and J. Carneiro. Oeiras, Portugal : Springer Berlin / Heidelberg, 2006. LNCS: Artificial Immune Systems. 5th International Conference, ICARIS 2006. Vol. 4163, pp. 376-389. ISBN-13: 978-3-540-37749-8.

[39]. *MILA — Multilevel Immune Learning Algorithm.* **Dasgupta, D., S., Yu and Majumdar, N. S.** [ed.] E. Cantú-Paz, et al. Chicago, IL, USA : Springer Berlin / Heidelberg, 2003. LNCS: Genetic and Evolutionary Computation — GECCO 2003, Part I. Vol. 2723, pp. 183-194. ISBN-13: 978-3-540-40602-0.

[40]. *A Novel Fuzzy Anomaly Detection Method Based on Clonal Selection Clustering Algorithm.* **Fenghua, L., Jian, L. and Y., Yixian.** [ed.] D. S. Yeung, et al. Guangzhou, China : Springer Berlin / Heidelberg, 2006. LNCS: Advances in Machine Learning and Cybernetics. 4th International Conference, ICMLC 2005. Vol. 3930/2006, pp. 642-651. ISBN-13: 978-3-540-33584-9.

[41]. *Real-Valued Negative Selection Algorithm with Variable-Sized Detectors.* **Ji, Z. and Dasgupta, D.** [ed.] K. Deb, et al. Seattle, WA, USA : Springer Berlin / Heidelberg, 2004. LNCS: Genetic and Evolutionary Computation – GECCO 2004. Vol. 3102, pp. 287-298. ISBN-13: 978-3-540-22344-3.

[42]. **Dasgupta, D. and Forrest, S.** *Tool breakage detection in milling operations using Negative Selection algorithm.* University of New Mexico. Albuquerque, NM, USA : s.n., 1995. Technical Report No.CS95-5.

[43]. *Fault Detection Algorithm for Telephone Systems Based on the Danger Theory.* **Pinto, J. C. L. and Von Zuben, F. J.** [ed.] C. Jacob, et al. Banff, Alberta, Canada : Springer Berlin / Heidelberg,

2005. LNCS: Artificial Immune Systems. 4th International Conference, ICARIS 2005. Vol. 3627/2005, pp. 418-431. ISBN-13: 978-3-540-28175-7.

[44]. *An investigation of the negative selection algorithm for fault detection in refrigeration systems.* **Taylor, D. W. and Corner, D. W.** [ed.] J. Timmis, P. Bentley and E. Hart. Edinburgh, UK : Springer Berlin / Heidelberg, 2003. LNCS: Artificial Immune Systems. Second International Conference, ICARIS 2003. Vol. 2787, pp. 34-45. ISBN-13: 978-3-540-40766-9.

[45]. *Immunising Automated Teller Machines.* **Ayara, M., et al.** [ed.] C. Jacob, et al. Banff, Alberta, Canada : Springer Berlin / Heidelberg, 2005. LNCS: Artificial Immune Systems. 4th International Conference, ICARIS 2005. Vol. 3627/2005, pp. 404-417. ISBN-13: 978-3-540-28175-7.

[46]. *Immunity-Based Aircraft Fault Detection System.* **Dasgupta, D., et al.** Chicago, IL, USA : AIAA, 2004. AIAA 1st Intelligent Systems Technical Conference. pp. 20-22. AIAA-2004-6277.

[47]. *Aircraft fault detection and classification using multi-level immune learning detection.* **Wong, D. and Krishnakumar, K.** Arlington, VA : AIAA, 2005. Infotech@Aerospace. AIAA-2005-6998 .

[48]. **de Castro, L. .N and Timmis, J.** *Artificial Immune Systems: A New Computational Intelligence Approach.* s.l. : Springer, 2002. ISBN: 1852335947.

[49]. **Dasgupta, D.** Advances in Artificial Immune Systems. *IEEE Computational Intelligence Magazine.* November 2006, Vol. 1, 4, pp. 40-49.

[50]. **Abbas, A. K. and Lichtman, A. H.** *Cellular and Molecular Immunology.* 4th. s.l. : W.B. Saunders Company, 2000. ISSB: 0721682332.

[51]. **National Cancer Institute, USA.** Antigens and Antibodies. *web-books.* [Online] [Cited: April 15, 2009.] http://www.web-books.com/eLibrary/Medicine/Physiology/Immune/Antigen.htm.

[52]. *Clonal Selection in a Lymphocytic Network.* **Jerne, N. K.** [ed.] G. M. Edelman. N. Y. : Raven Press, 1974, Cellular Selection and the Regulation in the Immune Response, p. 39.

[53]. *Artificial Immune System Approaches for Aerospace Applications.* **Krishnakumar, K.** Reno, NV : AIAA, 2003. 41st Aerospace Sciences Meeting and Exhibit. AIAA 2003-0457 .

[54]. *Architecture for an artificial immune system.* **Hofmeyr, S. A. and Forrest, S.** 4, s.l. : MIT Press, winter 2000, Evolutionary Computation, Vol. 8, pp. 443-73. ISSN: 1063-6560.

[55]. *An Artificial Immune Network Approach for Pattern Recognition.* **Jiuying, D., Yongsheng, J. and M., Zongyuan.** Haikou, China : IEEE, 2007. 3rd International Conference on Natural Computation, ICNC 2007. pp. 610-15. ISBN: 0769528759.

[56]. *Applying artificial immune system and ant algorithm in air-conditioner market segmentation.* **Chiu, Chui-Yu, Kuo, I-Ting and Lin, Chia-Hao.**, Elsevier Ltd., April 2009, Expert Systems with Applications, Vol. 36, Iss. 3 Part 1, pp. 4437-4442. ISSN: 09574174.

[57]. *An artificial immune system for fuzzy-rule induction in data mining.* **Alves, R.T., et al.** Birmingham, UK : Springer-Verlag, 2004. LNCS: Parallel Problem Solving from Nature. 8th International Conference, PPSN 2004. Vol. 3242, pp. 1011-20. ISBN: 3-540-23092-0.

[58]. *Use of an Artificial Immune System for Job Shop Scheduling.* **Coello Coello, C. A., Cortés Rivera, D. and C., Nareli.** [ed.] J. Timmis, P. Bentley and E. Hart. Edinburgh, UK : Springer Berlin / Heidelberg, 2003. LNCS: Artificial Immune Systems. Second International Conference, ICARIS 2003. Vol. 2787, pp. 1-10. ISBN-13: 978-3-540-40766-9.

[59]. *Learning and optimization using the clonal selection principle.* **de Castro, L. N. and Von Zuben, F. J.**, IEEE, June 2002, IEEE Transactions on Evolutionary Computation, Vol. 6, Iss. 3, pp. 239-251. ISSN: 1089778X.

[60]. *A Novel Fuzzy Anomaly Detection Method Based on Clonal Selection Clustering Algorithm.* **Lang, F., Li, J. and Yang, Y.** [ed.] D.S. Yeung, et al. Guangzhou, China : Springer Berlin / Heidelberg, 2005. Advances in Machine Learning and Cybernetics: 4th International Conference, ICMLC 2005, Revised Selected Papers. Vol. 3930, pp. 642-651. ISBN-13: 978-3-540-33584-9.

[61]. *On The Use Of Negative Selection In An Artificial Immune System.* **Ebner, M and Breunig, H., Albert, J.** New York, NY, USA : Morgan Kaufmann Publishers Inc., 2002. Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002. pp. 957-964. ISBN: 1558608788.

[62]. *Is negative selection appropriate for anomaly detection?* **Stibor, T., Mohr, P. and Timmis, J.** Washington, D.C., USA : Association for Computing Machinery, 2005. Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005. pp. 321-328. ISBN: 1595930108.

[63]. *A comparative study of real-valued negative selection to statistical anomaly detection techniques.* **Stibor, T., Timmis, J. and Eckert, C.** [ed.] C. Jacob, et al. Banff, Alberta, Canada : Springer Verlag, 2005. LNCS: Artificial Immune Systems. 4th International Conference, ICARIS 2005. Vol. 3627, pp. 262-275. ISBN-13: 978-3-540-28175-7.

[64]. *Applicability issues of the real-valued negative selection algorithms.* **Ji, Z. and Dasgupta, D.** Seattle, WA, USA : Association for Computing Machinery, 2006. Genetic and Evolutionary Computation Conference, GECCO 2006. pp. 111-118. ISBN: 1595931864.

[65]. *An artificial immune system architecture for computer security applications.* **Harmer, P. K., et al.**, IEEE, June 2002, IEEE Transactions on Evolutionary Computation, Vol. 6, Iss. 3, pp. 252-280. Iss. 1089778X.

[66]. *Revisiting negative selection algorithms.* **Ji, Z. and Dasgupta, D.** 2, s.l. : MIT Press, Summer 2007, Evolutionary Computation, Vol. 15, pp. 223-251. 1063-6560 .

[67]. *On the Use of Hyperspheres in Artificial Immune Systems as Antibody Recognition Regions.* **Stibor, T., Timmis, J. and Eckert, C.** s.l. : Springer Berlin / Heidelberg, September 2006, LNCS: Artificial Immune System. 5th International Conference, ICARIS 2006, Vol. 4163, pp. 215-228. ISBN: 3-540-37749-2.

[68]. *Integrated Framework for Aircraft Sub-System Failure Detection, Identification, and Evaluation Based on the Artificial Immune System Paradigm.* **Perhinschi, M., Moncayo, H. and Davis, J.** Chicago, IL, USA : AIAA, 2009. Guidance, Navigation and Control Conference. To be published. AIAA-2009-6261.

[69]. **E., Bellman R.** *Adaptive Control Processes: A Guided Tour.* 1st Ed. s.l. : Princeton University Press, 1961.

[70]. *On-line Negative Databases.* **Esponda, O., Ackley, E. S. and Forrest, S., Helman, P.** [ed.] G. Nicosia, et al. Catania, Sicily, Italy : Springer Berlin / Heidelberg, 2004. LNCS: Artificial Immune Systems. Third International Conference, ICARIS 2004. pp. 175–188. ISBN-13: 978-3-540-23097-7.

[71]. *Using the Triangle Inequality to Accelerate k-Means.* **Elkan, C.** San Diego, CA, USA : American Association for Artificial Intelligence (AAAI), 2003. Proceedings, Twentieth International Conference on Machine Learning. Vol. 1, pp. 147-153. ISBN-10: 1577351894.

[72]. *Immunity - Based Aircraft Failure Detection and Identification Using an Integrated Hierarchical Multi-Self Strategy.* **Moncayo, H., Perhinschi, M. and Davis, J.** Chicago, IL, USA : AIAA, 2009. Guidance, Navigation and Controls Conference.

[73]. *Evolutionary Algorithm for Artificial Immune System-Based Failure Detector Generation and Optimization.* **Davis, J., Perhinschi, M. and Moncayo, H.** Chicago, IL, USA : AIAA, 2009. Guidance, Navigation and Controls Conference. To be published. AIAA-2009-5891.

[74]. *Real-valued negative selection algorithm with variable-sized detectors.* **Ji, Z. and Dasgupta, D.** Seattle, WA, USA : Springer-Verlag, 2004. Proceedings of Genetic and Evolutionary Computation Conference. Part I. pp. 287-98. ISBN-10: 3-540-22344-4.

[75]. *A feedback negative selection algorithm to anomaly detection.* **Zeng, J., et al.** Piscataway, NJ, USA : IEEE, 2007. International Conference on Natural Computation. pp. 579-83. ISBN-13: 978-0-7695-2875-5.

[76]. *Autonomous Formation Flight: Hardware Development.* **Gu, Y., et al.** Ancona, Italy : IEEE, 2006. 14th Mediterranean Conference on Control and Automation. pp. 1-6. 10.1109/MED.2006.328709.

[77]. *Parameter Identification for Application within a Fault Tolerant Flight Control System.* **Phillips, K., et al.** Chicago, IL, USA : AIAA, 2009. Guidance, Navigation, and Control Conference,. To be published. AIAA-2009-5723.

[78]. *Aircraft Model for the AIAA Controls Design Challenge.* **W., Brumbaugh R.**, AIAA, 1994, Journal of guidance, control, and dynamics, Vol. 17, Iss. 4, pp. 747-752. ISSN: 0731-5090.

[79]. *A simulation environment for testing and research of neurally augmented fault tolerant control laws based on non-linear dynamic inversion.* **Perhinschi, M. G., et al.** Providence, RI, United states : AIAA, 2004. Collection of Technical Papers - AIAA Modeling and Simulation Technologies Conference. Vol. 1, pp. 147-157. AIAA 2004-4913.

[80]. **Sagoo, G.** *Pilot in Loop Assessment of Fault Tolerant Flight Control Schemes in a Motion Flight Simulator.* Morgantown : WVU Libraries, 2008. Thesis.

[81]. *Pilot-in-the-Loop Assessment of Neurally Augmented Dynamic Inversion Based Fault Tolerant Control Laws in a Motion-Based Flight Simulator.* **Sagoo, G., et al.** Honolulu, Hawaii, USA : AIAA, 2008. AIAA Guidance, Navigation and Control Conference and Exhibit. AIAA-2008-6843.

[82]. **Supnik, Ben.** X-Plane plugin SDK Documentation. *The House of X-PLane.* [Online] [Cited: May 29, 2008.] http://www.xsquawkbox.net/xpsdk/phpwiki/index.php?Documentation.

Appendix A: Effect of λ-Distance Definition in Hyper-Body

The following figure shows how higher power definitions of distance cover at least the previous power definition.

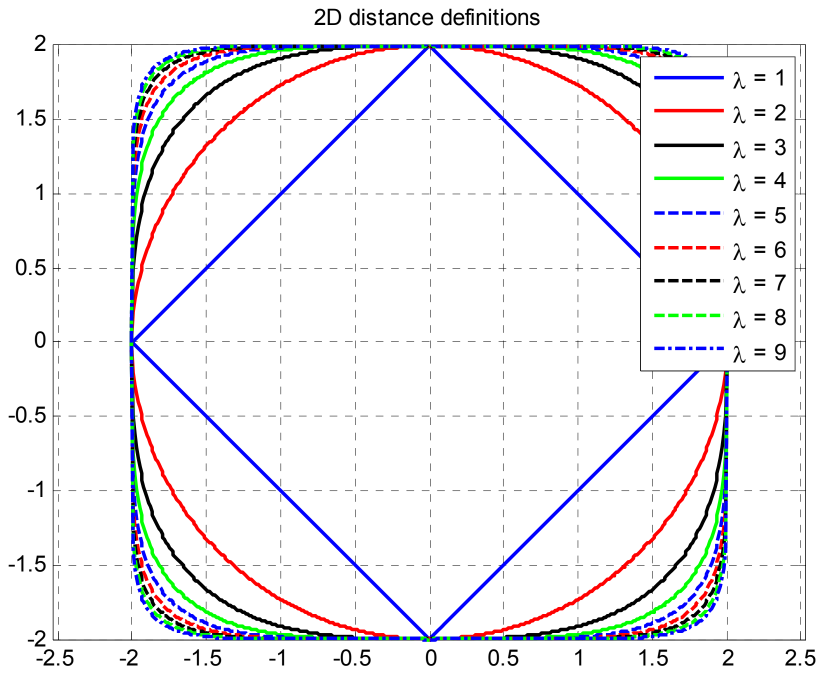


Figure 39: 2D circles of radius 2 using different definitions of distance.

This does not necessarily mean that it is true for any dimension, or that it continues to be true even at higher values of λ. The following is a more formal mathematical proof.

The formula for the λ-distance (2.4) expressed as a function is:

$$d_\lambda\left(\vec{x},\vec{y}\right)=\left(\sum_{i=1}^{n}\left|x_i-y_i\right|^\lambda\right)^{1/\lambda} \tag{A.1}$$

If the limit as λ goes to infinity is computed, the result is:

$$\lim_{\lambda\to\infty} d_\lambda\left(\vec{x},\vec{y}\right)=\lim_{\lambda\to\infty}\left(\sum_{i=1}^{n}\left|x_i-y_i\right|^\lambda\right)^{1/\lambda}=\lim_{\lambda\to\infty}\left(\sum_{i=1}^{n}\left|Q_i\right|^\lambda\right)^{1/\lambda}=\lim_{\lambda\to\infty}\left(K^\lambda\cdot\sum_{i=1}^{n}\frac{\left|Q_i\right|^\lambda}{K^\lambda}\right)^{\frac{1}{\lambda}} \tag{A.2}$$

Where $Q_i = x_i - y_i$ and K is the max of the absolute values of Q_i:

$$K = \max\left(\left|Q_1\right|, \left|Q_2\right|, \cdots, \left|Q_n\right|\right) \;\Rightarrow\; K = \left|Q_j\right| \geq 0 \tag{A.3}$$

It is also important to assume that there is not only one max value, thus the max value $\left|Q_j\right|$ is repeated in h dimensions. (A.2) then becomes:

$$\lim_{\lambda\to\infty} d_\lambda\left(\vec{x},\vec{y}\right) = \lim_{\lambda\to\infty}\left(K\cdot\left(h+\sum_{i=1}^{n}\left|\frac{Q_i}{K}\right|^{\lambda}\right)^{\frac{1}{\lambda}}\right) \tag{A.4}$$

The h terms that contain the maximum values of $\left|Q_i\right|$ have been removed from the summation expression, thus, all remaining Q_i are less than Q_j what implies that:

$$\left|\frac{Q_i}{K}\right| < 1 \tag{A.5}$$

This means that all the terms in the summation converge to zero as λ goes to infinity, then:

$$\lim_{\lambda\to\infty} d_\lambda\left(\vec{x},\vec{y}\right) = \lim_{\lambda\to\infty}\left(K\cdot h^{1/\lambda}\right) = K = \left|Q_j\right| \tag{A.6}$$

$$d_\infty\left(\vec{x},\vec{y}\right) = \max\left(\left|Q_1\right|, \left|Q_2\right|, \cdots, \left|Q_n\right|\right) \tag{A.7}$$

This distance definition, as well as all n-distance definitions, comply with the four requirements for a distance to be valid:

1. Non-negativity: $d_\lambda\left(\vec{x},\vec{y}\right) \geq 0$
2. Reflexivity: $d_\lambda\left(\vec{x},\vec{y}\right) = 0$ iff $C = P$
3. Symmetry: $d_\lambda\left(\vec{x},\vec{y}\right) = d_\lambda\left(\vec{y},\vec{x}\right)$
4. Triangle inequality: $d_\lambda\left(\vec{x},\vec{y}\right) + d_\lambda\left(\vec{y},\vec{z}\right) \geq d_\lambda\left(\vec{x},\vec{z}\right)$

Now, let's examine the previous definition for a constant $d_\infty = r$. (A.7) is then always equal to the maximum absolute value of the components, which means $\left|Q_j\right|$ and as only the boundary is being considered, it means $\left|Q_j\right| = d_\infty = r$ thus the boundary is a hypercube of side r. This hypercube coincides with the hypersphere defined by $d_2 = r$ in only n points (one for each dimension); the rest of the points of the hypersphere are completely enclosed in the boundaries defined by the hypercube.

So far it has only been shown that the hyper-body defined by a constant value of $d_\infty = r$ encloses the hyper-body defined by $d_2 = r$. The next step consists in checking the behavior in values in between. For this purpose, the distance definition with respect to λ is derived as:

$$d_\lambda(\vec{x},\vec{y}) = \left(\sum_{i=1}^{\lambda} \left|Q_i\right|^\lambda\right)^{\frac{1}{\lambda}} \tag{A.8}$$

$$d_\lambda^{'}(\vec{x},\vec{y}) = \frac{1}{\lambda}\cdot\left(\sum_{i=1}^{n} \left|Q_i\right|^\lambda\right)^{\frac{1}{\lambda}-1}\cdot\lambda\cdot\sum_{i=1}^{n}\left|Q_i\right|^{\lambda-1} \tag{A.9}$$

$$d_\lambda^{'}(\vec{x},\vec{y}) = \left(\sum_{i=1}^{n} \left|Q_i\right|^\lambda\right)^{\frac{1}{\lambda}-1}\cdot\sum_{i=1}^{n}\left|Q_i\right|^{\lambda-1} = 0 \tag{A.10}$$

The last expression represents the condition necessary to have either a maximum or a minimum. This condition is true for the cases when any of the two summations is zero. The following expressions are obtained by evaluating these conditions:

$$\sum_{i=1}^{n}\left|QX_i\right|^{\lambda-1} = 0 \quad\Rightarrow\quad Q_i = 0 \;\forall\; i = 1,2,\cdots,n \tag{A.11}$$

$$\left(\sum_{i=1}^{n} \left|Q_i\right|^\lambda\right)^{\frac{1}{\lambda}-1} = \sum_{i=1}^{n}\left|Q_i\right|^\lambda = 0 \quad\Rightarrow\quad Q_i = 0 \;\forall\; i = 1,2,\cdots,n \tag{A.12}$$

This result implies that the function is monotonic and as was shown before, for λ that tends to infinity the hyper-shape encloses the Euclidean definition thus it can be stated that: A hyper-sphere defined using a distance λ_k encloses at least all the points contained in hyper-spheres defined using a distance λ_j as long as $j \le k$.