

2014

Immunity-Based Accommodation of Aircraft Subsystem Failures

Adil Togayev

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

Recommended Citation

Togayev, Adil, "Immunity-Based Accommodation of Aircraft Subsystem Failures" (2014). *Graduate Theses, Dissertations, and Problem Reports*. 6814.

<https://researchrepository.wvu.edu/etd/6814>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Immunity-Based Accommodation of Aircraft Subsystem Failures

Adil Togayev

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

Master of Science
in
Mechanical Engineering

Powsiri Klinkhachorn, Ph.D.
Hever Moncayo, Ph.D.
Mario Perhinschi, Ph.D., Chair

Department of Mechanical and Aerospace Engineering
Morgantown, West Virginia
2014

Copyright ©2014 Adil Togayev
All Rights Reserved

ABSTRACT

Immunity-Based Accommodation of Aircraft Subsystem Failures

This thesis presents the design, development, and flight-simulation testing of an artificial immune system (AIS) based approach for accommodation of different aircraft subsystem failures.

Failure accommodation is considered as part of a complex integrated AIS scheme that contains four major components: failure detection, identification, evaluation, and accommodation. The accommodation part consists of providing compensatory commands to the aircraft under specific abnormal conditions based on previous experience. In this research effort, the possibility of building an AIS allowing the extraction of pilot commands is investigated.

The proposed approach is based on structuring the self (nominal conditions) and the non-self (abnormal conditions) within the AIS paradigm, as sets of artificial memory cells (mimicking behavior of T-cells, B-cells, and antibodies) consisting of measurement strings, over pre-defined time windows. Each string is a set of features values at each sample time of the flight including pilot inputs, system states, and other variables. The accommodation algorithm relies on identifying the memory cell that is the most similar to the in-coming measurements. Once the best match is found, control commands corresponding to this match will be extracted from the memory and used for control purposes.

The proposed methodology is illustrated through simulation of simple maneuvers at nominal flight conditions, different actuators, and sensor failure conditions. Data for development and demonstration have been collected from West Virginia University 6-degrees-of-freedom motion-based flight simulator. The aircraft model used for this research represents a supersonic fighter which includes model following adaptive control laws based on non-linear dynamic inversion and artificial neural network augmentation.

The simulation results demonstrate the possibility of extracting pilot compensatory commands from the self/non-self structure and the capability of the AIS paradigm to address the problem of accommodating actuator and sensor malfunctions as a part of a comprehensive and integrated framework along with abnormal condition detection, identification, and evaluation.

DEDICATION

*To my mother Gulnara Yezhebayeva
and my grandparents,
Kabduahit Yezhebayev and Gulshat Manakova*

ACKNOWLEDGEMENTS

During the time spent in graduate school, I had the pleasure of interacting with intelligent and supportive people. Without these people, I would not have been able to accomplish this achievement. I would like to extend my appreciation to each and all of them.

First and foremost, I would like to express my special appreciation to my advisor, Dr. Mario Perhinschi, who has been a tremendous mentor to me. I would like to thank him for giving me the opportunity to pursue my Master's degree within this research field, his continuous support, encouragement, patience, and immense knowledge.

I am deeply thankful to my Committee members, Dr. Powsiri Klinkhachorn and Dr. Hever Moncayo, for providing me with helpful suggestions and meaningful comments. They have contributed a lot with their time reviewing this thesis and giving me advice for improvement.

Great thanks to the volunteer pilots Steven Hard and Caraline Griffith for performing tests in the flight simulator. They have dedicated their time in order to help me to acquire the necessary data, without which it would have been impossible to get the results.

I would like to thank Dia Al Azzawi, Israel Moguel, and Andres Perez for sharing their experience and knowledge with me while working on the project, and for their insightful comments on our papers.

Finally, I would like to thank my family and friends who supported and encouraged me throughout the entire time of pursuing my degree.

CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
NOMENCLATURE	xii
Chapter 1 Introduction	1
Chapter 2 Literature Review	7
2.1 Fault-Tolerant Control for Flight Systems	7
2.2 Immunity-based Systems	9
2.3 Artificial Immune System for Failure Accommodation	11
Chapter 3 The Aircraft Subsystem Failure Detection, Identification, Evaluation, and Accommodation (FDIEA) Problem	14
3.1 FDIEA Problem Formulation.....	14
3.2 FDIEA Problem Components	15
3.2.1 Detection.....	15
3.2.2 Identification	15
3.2.3 Evaluation	15
3.2.4 Accommodation.....	16
3.3 FDIEA Problem Overview.....	17
Chapter 4 FDIEA Problem within the AIS Paradigm	19
4.1 Natural Immune System.....	19
4.2 AIS Paradigm	20
Chapter 5 Biologically Inspired Failure Accommodation	23
5.1 AIS Accommodation Problem Outline	23
5.1.1 Defining a Mathematical Model	24
5.1.2 Generating the Artificial Memory Cells	25
5.1.3 Matching Algorithm Design and Development	26
5.1.4 Preparation of the Simulation Environment.....	27
5.1.5 Practical Implementation	27
5.1.6 Post-processing Analysis	27

5.2 Mathematical Model	29
5.3 AIS Memory Cells Representation	30
5.3.1 Antigens	31
5.3.2 T-cells and B-cells	31
5.3.3 Antibodies	32
5.4 Matching Algorithm	33
5.4.1 Instant Matching Algorithm	33
5.4.2 Artificial Memory Cell Matching	34
5.4.3 Normalization Process	35
5.4.4 Weighting the Features	37
Chapter 6 Simulation Environment	38
6.1 WVU Flight Simulator	38
6.1.1 Motion-Based 6-DOF Flight Simulator	38
6.1.2 Desktop PC Simulator	41
6.1.3 Aircraft Model	43
6.1.4 Aircraft Subsystems Failure Modeling	44
6.1.4.1 Actuator Failure	44
6.1.4.2 Sensor Failure	44
6.2 Simulation Scenarios	44
6.2.1 Symmetric Climb/Descent	45
6.2.2 Coordinated Half Turn	46
Chapter 7 Implementation of Failure Accommodation System	47
7.1 Data Acquisition for AIS Development	47
7.1.1 Symmetric Climb/Descent	47
7.1.2 Coordinated Half Turn	48
7.2 Accommodation Simulink Model	50
Chapter 8 Results and Performance Assessment	53
8.1 Symmetric Climb	54
8.1.1 Nominal Conditions	54
8.1.2 Actuator Failure	57
8.1.2.1 Aileron Failure	58
8.1.2.2 Stabilator Failure	62

8.1.2.3 Rudder Failure.....	66
8.1.3 Sensor Failure	70
8.1.4 Analysis and Evaluation.....	73
8.2 Coordinated Half Turn	76
8.2.1 Nominal Conditions	76
8.2.2 Actuator Failure	80
8.2.2.1 Aileron Failure	80
8.2.2.2 Stabilator Failure.....	84
8.2.2.3 Rudder Failure.....	87
8.2.3 Sensor Failure	91
8.2.4 Analysis and Evaluation.....	95
8.3 Symmetric Descent	97
8.4 Comprehensive Analysis and Assessment	101
Chapter 9 Conclusions and Future Work	103
9.1 Research Summary.....	103
9.2 Future Work	105
Bibliography	107
APPENDIX.....	115
MATLAB code for Matching Algorithm.....	115

LIST OF FIGURES

Figure 2.1: General Structure of Fault-Tolerant Control Systems	8
Figure 3.1: FDIEA General Process Diagram	14
Figure 3.2: Two Types of Failure Accommodation.....	17
Figure 3.3: FDIEA Problem Block Diagram	18
Figure 4.1: Block Diagram of the Immune System Feedback Response.....	20
Figure 4.2: AIS-based FDIEA	21
Figure 5.1: Accommodation Problem Main Components	24
Figure 5.2: Simplified Artificial Memory Cells Generating Process.....	25
Figure 5.3: Matching Algorithm Process	26
Figure 5.4: AIS-Based Failure Accommodation Process	28
Figure 5.5: AIS Memory Structure	33
Figure 5.6: Matching Algorithm with Floating Time Window.....	35
Figure 5.7: a) Normalization Process without Additional Margins. b) Normalization Process with Additional Margins	36
Figure 5.8: Matching Algorithm with Increased Weight of Features	37
Figure 6.1: The WVU 6-DOF Motion-based Flight Simulator.....	39
Figure 6.2: Flight Simulator Cabin and Motion System.....	39
Figure 6.3: Interface of the WVU Flight Simulator with External Models	40
Figure 6.4: WVU Desktop PC Flight Simulator	41
Figure 6.5: WVU Simulation Environment Simulink Top Level Diagram	42
Figure 6.6: FlightGear Visualization of the Supersonic Fighter	43
Figure 6.7: a) Symmetric Uniform Descent; b) Symmetric Uniform Climb	45
Figure 6.8: Coordinated Half Turn	46
Figure 7.1: Accommodation System Components.....	50
Figure 7.2: Aircraft Control System Block.....	51
Figure 7.3: Detailed Structure of the Control Commands Generating Block	52
Figure 8.1: Aircraft Altitude - Nominal Conditions Symmetric Climb	54
Figure 8.2: Aircraft Velocity - Nominal Conditions Symmetric Climb	55
Figure 8.3: Longitudinal Channel Commands - Nominal Conditions Symmetric Climb.....	55
Figure 8.4: Throttle Displacement - Nominal Conditions Symmetric Climb	56
Figure 8.5: Pitch Rate - Nominal Conditions Symmetric Climb	56
Figure 8.6: Pitch Angle - Nominal Conditions Symmetric Climb.....	57
Figure 8.7: Aircraft Altitude - Aileron Failure Symmetric Climb	58
Figure 8.8: Aircraft Velocity - Aileron Failure Symmetric Climb	58
Figure 8.9: Longitudinal Channel Commands - Aileron Failure Symmetric Climb.....	59
Figure 8.10: Pitch Angle - Aileron Failure Symmetric Climb.....	59
Figure 8.11: Aircraft Bank Angle - Aileron Failure Symmetric Climb	60
Figure 8.12: Lateral Channel Commands - Aileron Failure Symmetric Climb	60
Figure 8.13: Aircraft Pitch Rate - Aileron Failure Symmetric Climb.....	61
Figure 8.14: Aircraft Roll Rate - Aileron Failure Symmetric Climb	61

Figure 8.15: Y Coordinate - Aileron Failure Symmetric Climb	62
Figure 8.16: Aircraft Altitude - Stabilator Failure Symmetric Climb.....	63
Figure 8.17: Aircraft Velocity - Stabilator Failure Symmetric Climb	63
Figure 8.18: Aircraft Pitch Angle - Stabilator Failure Symmetric Climb.....	63
Figure 8.19: Longitudinal Channel Commands - Stabilator Failure Symmetric Climb	64
Figure 8.20: Lateral Channel Commands - Stabilator Failure Symmetric Climb.....	64
Figure 8.21: Y Coordinate - Stabilator Failure Symmetric Climb.....	65
Figure 8.22: Aircraft bank angle - Stabilator Failure Symmetric Climb	65
Figure 8.23: Aircraft Altitude - Rudder Failure Symmetric Climb.....	66
Figure 8.24: Aircraft Velocity - Rudder Failure Symmetric Climb.....	66
Figure 8.25: Aircraft Yaw angle - Rudder Failure Symmetric Climb	66
Figure 8.26: Directional Channel Commands - Rudder Failure Symmetric Climb.....	67
Figure 8.27: Aircraft Pitch Angle - Rudder Failure Symmetric Climb.....	68
Figure 8.28: Longitudinal Channel Commands - Rudder Failure Symmetric Climb	68
Figure 8.29: Aircraft Bank Angle - Rudder Failure Symmetric Climb	69
Figure 8.30: Lateral Channel Commands - Rudder Failure Symmetric Climb	69
Figure 8.31: Y Coordinate - Rudder Failure Symmetric Climb.....	69
Figure 8.32: Aircraft Altitude - Sensor Failure Symmetric Climb	70
Figure 8.33: Aircraft Velocity - Sensor Failure Symmetric Climb.....	71
Figure 8.34: Throttle Command - Sensor Failure Symmetric Climb.....	71
Figure 8.35: Longitudinal Channel Commands - Sensor Failure Symmetric Climb	71
Figure 8.36: Aircraft Pitch Angle - Sensor Failure Symmetric Climb.....	71
Figure 8.37: Lateral Channel Commands - Sensor Failure Symmetric Climb	72
Figure 8.38: Y Coordinate - Sensor Failure Symmetric Climb	72
Figure 8.39: Aircraft Bank Angle - Sensor Failure Symmetric Climb	73
Figure 8.40: Flight Trajectory - Nominal Conditions Coordinated Half Turn.....	76
Figure 8.41: Lateral Channel Commands - Nominal Conditions Coordinated Half Turn.....	77
Figure 8.42: Aircraft Bank Angle - Nominal Conditions Coordinated Half Turn.....	77
Figure 8.43: Aircraft Roll Rate - Nominal Conditions Coordinated Half Turn.....	78
Figure 8.44: Longitudinal Channel Commands - Nominal Conditions Coordinated Half Turn.....	78
Figure 8.45: Aircraft Altitude - Nominal Conditions Coordinated Half Turn	79
Figure 8.46: Aircraft Pitch Rate – Nominal Conditions Coordinated Half Turn.....	79
Figure 8.47: Aircraft Velocity – Nominal Conditions Coordinated Half Turn.....	79
Figure 8.48: Throttle Displacement - Nominal Conditions Coordinated Half Turn.....	79
Figure 8.49: Flight Trajectory - Aileron Failure Coordinated Half Turn.....	80
Figure 8.50: Lateral Channel Commands - Aileron Failure Coordinated Half Turn.....	81
Figure 8.51: Aircraft Bank Angle - Aileron Failure Coordinated Half Turn.....	81
Figure 8.52: Aircraft Roll Rate - Aileron Failure Coordinated Half Turn.....	82
Figure 8.53: Longitudinal Channel Commands - Aileron Failure Coordinated Half Turn.....	82
Figure 8.54: Aircraft Altitude - Aileron Failure Coordinated Half Turn	83
Figure 8.55: Aircraft Pitch Angle - Aileron Failure Coordinated Half Turn	83
Figure 8.56: Aircraft Velocity - Aileron Failure Coordinated Half Turn	83
Figure 8.57: Throttle Displacement - Aileron Failure Coordinated Half Turn.....	83
Figure 8.58: Flight Trajectory - Stabilator Failure Coordinated Half Turn	84

Figure 8.59: Longitudinal Channel Commands - Stabilator Failure Coordinated Half Turn	84
Figure 8.60: Aircraft Pitch Angle - Stabilator Failure Coordinated Half Turn.....	85
Figure 8.61: Aircraft Pitch Rate - Stabilator Failure Coordinated Half Turn	85
Figure 8.62: Lateral Channel Commands - Stabilator Failure Coordinated Half Turn.....	85
Figure 8.63: Aircraft Bank Angle - Stabilator Failure Coordinated Half Turn.....	86
Figure 8.64: Aircraft Roll Rate - Stabilator Failure Coordinated Half Turn.....	86
Figure 8.65: Aircraft Velocity - Stabilator Failure Coordinated Half Turn	87
Figure 8.66: Throttle Displacement - Stabilator Failure Coordinated Half Turn.....	87
Figure 8.67: Flight Trajectory - Rudder Failure Coordinated Half Turn	87
Figure 8.68: Longitudinal Channel Commands - Rudder Failure Coordinated Half Turn	88
Figure 8.69: Aircraft Altitude - Rudder Failure Coordinated Half Turn	88
Figure 8.70: Aircraft Pitch Angle - Rudder Failure Coordinated Half Turn.....	88
Figure 8.71: Lateral Channel Commands - Rudder Failure Coordinated Half Turn	89
Figure 8.72: Aircraft Bank Angle - Rudder Failure Coordinated Half Turn	89
Figure 8.73: Directional Channel Commands- Rudder Failure Coordinated Half Turn.....	90
Figure 8.74: Aircraft Yaw Angle - Rudder Failure Coordinated Half Turn	90
Figure 8.75: Aircraft Velocity – Rudder Failure Coordinated Half Turn	91
Figure 8.76: Throttle Displacement – Rudder Failure Coordinated Half Turn.....	91
Figure 8.77: Flight Trajectory - Sensor Failure Coordinated Half Turn	91
Figure 8.78: Aircraft Altitude - Sensor Failure Coordinated Half Turn	92
Figure 8.79: Lateral Channel Commands - Sensor Failure Coordinated Half Turn	93
Figure 8.80: Aircraft Bank Angle - Sensor Failure Coordinated Half Turn	93
Figure 8.81: Longitudinal Channel Commands - Sensor Failure Coordinated Half Turn	94
Figure 8.82: Aircraft Pitch Angle - Sensor Failure Coordinated Half Turn	94
Figure 8.83: Aircraft Roll Rate - Sensor Failure Coordinated Half Turn	94
Figure 8.84: Aircraft Velocity - Sensor Failure Coordinated Half Turn.....	95
Figure 8.85: Aircraft Altitude - Nominal Conditions Symmetric Descent	97
Figure 8.86: Longitudinal Channel Commands - Nominal Conditions Symmetric Descent.....	98
Figure 8.87: Aircraft Pitch Rate - Nominal Conditions Symmetric Descent.....	99
Figure 8.88: Throttle Command - Nominal Conditons Symmetric Descent.....	100
Figure 8.89: Aircraft Velocity - Nominal Conditions Symmetric Descent.....	100

LIST OF TABLES

Table 7.1: Symmetric Climb - Maximum and Minimum Values for Normalization	48
Table 7.2: Coordinated Turn - Maximum and Minimum Values for Normalization	49
Table 8.1: Flight Simulation Performance Assessment - Symmetric Climb	74
Table 8.2: Pilot Workload - Symmetric Climb	75
Table 8.3: Flight Simulation Performance Assessment - Coordinated Half Turn	96
Table 8.4: Pilot Workload - Coordinated Half Turn	96

NOMENCLATURE

English

AB_v/AB	Vector/Matrix of Antibodies
AG_v	Antigens Vector
FF	Set of Flight Features
H	Aircraft Altitude
INC_v/INC	Vector/Matrix of In-coming Values
PC_v/PC	Vector/Matrix of Pilot Control Commands
PF_v/PF	Vector/Matrix of Pilot Features Values
REF_v/REF	Vector/Matrix of Desired Values
TB_v/TB	Vector/Matrix of T-cells and B-cells
V	Aircraft Velocity
de, da, dr	Stick Displacement on Longitudinal, Lateral, and Yaw Channels
dt	Throttle Displacement
p, q, r	Aircraft Roll, Pitch, and Yaw Rates

Greek

φ	Bank Angle
θ	Pitch Angle
ψ	Yaw Angle

Acronyms

AC	Abnormal Conditions
AIS	Artificial Immune System
ANN	Artificial Neural Networks
DC	Dendritic Cell
FDI	Failure Detection and Identification
FDIE	Failure Detection, Identification, and Evaluation
FDEIA	Failure Detection, Identification, Evaluation, and Accommodation
GA	Genetic Algorithms
NN	Neural Network
PID	Proportional-Integral-Derivative
UAV	Unmanned Aerial Vehicle

Chapter 1 Introduction

Flight safety has always been a number one concern for aircraft operation. Vehicle subsystem failures and structural damages have been proven to be the major accident causes for both civilian and military aircraft [1-4]. Pilot has to detect a problem in a timely manner and produce fast compensation to handle the abnormal situation. In order for this to happen, substantial amount of information has to be provided to the pilot about the failure in a short period of time and the pilot must have specific skills to handle the abnormal condition. There are several factors which may negatively affect the successful outcome. The complexity of modern machines, redundancy of information, and huge variety of all types of devices inside the cockpit may prevent pilot from reacting properly and generating appropriate response in an extreme situation. Over the entire history of aviation, a lot of effort was focused on minimizing pilot workload. A variety of control system design methodologies have been developed, starting with simple stability augmentation systems and ending up with sophisticated adaptive control laws [5-7]. Significant recent research efforts were aimed at increasing aircraft operation safety through the development of integrated fault tolerant control systems capable of high performance real time failure detection, identification, evaluation, and accommodation (FDIEA) [8-10].

The aircraft subsystem FDIEA problem consists of four components. The first component refers to acknowledging the presence of an abnormal condition (AC). The second component, failure identification, consists of specifying which subsystem and which of its elements has failed. The third component is evaluation, and it refers to the assessment of failure type, failure magnitude/severity, and the prediction of its impact on the reduction of the flight

envelope [11]. Finally, the last component is accommodation and it is responsible for providing an adaptive control law that will help mitigating the consequences of the failure.

Significant research efforts towards solving the FDIEA problem have been performed in recent years [12, 13]; however, these efforts were mostly focused on the individual classes of failures within limited regions of the flight envelope. For this reason, a strong need for a comprehensive integrated solution to the given problem for aircraft subsystems has been widely acknowledged [14-16].

Finding a comprehensive solution to the FDIEA problem is an extremely complicated, multidimensional task that requires appropriate tools, high-level accuracy and most importantly - extensive robustness. Such a complexity of the problem has made engineers think outside of the conventional frame of control systems and seek ideas from somewhere else. Biological world inspired methodologies have become very popular among researchers lately. Various control systems based on these methodologies have been successfully developed and implemented, such as artificial neural network [17], evolutionary algorithms [18], DNA computation [19], and artificial immune systems [8, 20, 21].

A biologically inspired framework that integrates all major aspects of fault tolerant control capabilities - detection, identification, evaluation, and accommodation - is currently under development at West Virginia University (WVU) and Embry-Riddle Aeronautical University (ERAU) [8, 10]. This research effort is aimed at providing a novel robust adaptive system that can deal with the enormous variety of known and unknown disturbances and failures

by obtaining a comprehensive solution to the FDIEA problem on the basis of biologically inspired techniques, such as the immune system paradigm [22]

Techniques based on artificial immune systems (AIS) are extremely promising candidates for solving the FDIEA problem because of the capabilities of their biological sources. The natural immune system responds to every requirement that FDIEA problem is restrained to: capability of detecting the harmful entities, defining which exact part of the organism is under attack, generating antibodies which can fight the harmful intruders and being able to remember the disease. Therefore, when the harmful antigens invade the body in the future, the immune response becomes faster and more effective. The biological immune system functionality is based on the self/non-self discrimination principle, when it distinguishes between the entities that belong to the organism and the entities that do not. For AIS-based schemes, this principle is transposed to a real aircraft: all the flight data for aircraft collected under nominal conditions represent the self, and those that were collected under abnormal conditions or simply outside self represent the non-self.

The immunity-based AC accommodation problem can be approached based on two different concepts. The first concept is the biological feedback that establishes a balance between the activation and suppression of the antibodies generation. This mechanism can be converted into an adaptive component augmenting a baseline controller [23]. The idea is to model the biological system by providing governing equations and then to correlate those laws with the real aircraft features. The second conceptual approach under the AIS paradigm for control purposes is based on the assumption that the classification capabilities of the AIS can be extended and used

not only to detect, identify, and evaluate, but also to provide some solution that would minimize or exclude the AC effects [10, 24]. For this approach, both control and controlled variables have to be a part of the feature set and the flight performance should be presented under specific conditions and constraints.

This research effort is focused on the second conceptual approach for solving the AC accommodation problem. It represents a preliminary critical phase with the main objective of investigating the possibility of extracting compensatory commands from the AIS self/non-self structure. The method to complete the task in this thesis relies on generating artificial memory cells, which represent the self (nominal conditions) and the non-self (abnormal conditions) within the artificial immune system paradigm. Both self and non-self are structured as memory cells consisting of measurement strings over pre-defined time windows. Each string is a set of features values at each sample time of the flight including pilot inputs, system states, and other variables. These flight features are selected such that they capture the dynamic fingerprint of the aircraft operation at normal and abnormal conditions and the required pilot action. The accommodation process works as follows: collections of strings over several time samples of current flight are compared to the collections of strings in the memory. Once the best match is found, control commands corresponding to this match will be extracted from the memory and used for control purposes.

This thesis includes the description, development, and assessment of all the steps that were mentioned above. A brief review of fault tolerant control systems, and biologically inspired methodologies used for designing these systems is provided in Chapter 2. A description of the

integrated FDIEA problem is presented in Chapter 3. FDIEA problem within the AIS paradigm is described in Chapter 4. AIS-based aircraft subsystem failure accommodation schemes are covered in Chapter 5 followed by the description of the simulation environment in Chapter 6. The developed algorithms implementation is depicted in Chapter 7. Test results, analysis, and the research evaluation are discussed in Chapter 8. Finally, some conclusions and future work are proposed in Chapter 9.

The main contributions of this research effort are:

- 1) Investigating the use of the AIS paradigm for control purposes using a novel methodology
- 2) Investigating the possibility of extracting compensatory control commands from properly built self/non-self structures within the AIS paradigm
- 3) Demonstrating the effectiveness of the solution using a motion-based flight simulator for several actuator and sensor failures

The research effort presented in this thesis has resulted in the following publications and submissions:

- 1) A. Togayev, M. G. Perhinschi, D. A. Azzawi, H. Moncayo, I. Moguel, and A. Perez, "Immunity-Based Abnormal Condition Accommodation of Aircraft Subsystem Failures", accepted in the *American Society of Mechanical Engineers Annual Dynamic Systems and Control Conference*, San Antonio, TX, October 2014
- 2) Moguel I., Moncayo H., Perhinschi M.G., Perez A., Azzawi D. A., Togayev A., "Structured Non-Self Approach for Aircraft Failure Identification within an Immunity-based Fault Tolerance Architecture", submitted to the *IEEE TAES*, Jan. 2014

- 3) Moguel I., Moncayo H., Perhinschi M. G., Perez A. E., Al Azzawi D., Togayev A., "Bio-Inspired Approach for Aircraft Health Assessment and Flight Envelope Estimation", submitted to *ASME Annual Dynamic Systems and Control Conference*, San Antonio, Texas, Oct. 2014
- 4) Perez A. E., Moncayo H., Moguel I., Perhinschi M. G., Al Azzawi D., Togayev A., "Development of Immunity-based Adaptive Control Laws for Aircraft Fault Tolerance", submitted to *ASME Annual Dynamic Systems and Control Conference*, San Antonio, Texas, Oct. 2014
- 5) Al Azzawi D., Perhinschi M. G., Togayev A., Moncayo H., Moguel I., Perez A. E., "Evaluating Aircraft Abnormal Conditions Using an Artificial Dendritic Cell Mechanism", abstract submitted to *International Conference and Exhibition on Mechanical and Aerospace Engineering*, Philadelphia, Pennsylvania, Sept. 2014
- 6) Moguel I., Moncayo H., Perhinschi M. G., Al Azzawi D., Perez A. E., Togayev A., "Biologically-Inspired Approach for Aircraft Management under Upset Conditions", abstract submitted to *International Conference and Exhibition on Mechanical and Aerospace Engineering*, Philadelphia, Pennsylvania, Sept. 2014
- 7) Perez A. E., Moncayo H., Togayev A., Perhinschi M. G., Al Azzawi D., "Pilot-in-the-Loop Evaluation of a Bio-Inspired Adaptive Fault Tolerant Control System in a Motion Based Flight Simulator", submitted to *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, Jan. 2015

Chapter 2 Literature Review

2.1 Fault-Tolerant Control for Flight Systems

Fault tolerance is a property that enables a system to continue operating properly in the event of the failure of one or more of the system components. Fault-tolerant control system design enables the system to continue functioning, possibly at a reduced level of performance, rather than failing completely [25].

Fault-tolerant control (FTC) for aircraft systems has become extremely popular for control engineers over the past decades [26-31]. Aircraft safety has always been a number one concern in aviation. Therefore, designing a control system that can mitigate the effects of the failure and maintain the desirable stable state after the occurrence of the failure is the first and foremost objective. A *failure* is defined as a malfunction of any physical component of aircraft subsystem that results in operating differently from the desired pre-designed manner. The following are types of failures that might be encountered during aircraft operation:

- Structural Failure/Damage
- Engine Failure
- Control Surface Failure
- Sensor Failure

After the failure occurs, both pilot and autopilot (control system) must be informed in a timely manner. Time issue might play a crucial role in preventing catastrophic consequences.

Once the malfunction is detected, an appropriate action must be taken by pilot and/or autopilot. Typical fault-tolerant control system architecture is illustrated in Fig. 2.1 [32]. Note that structural and engine failures are not included in this system.

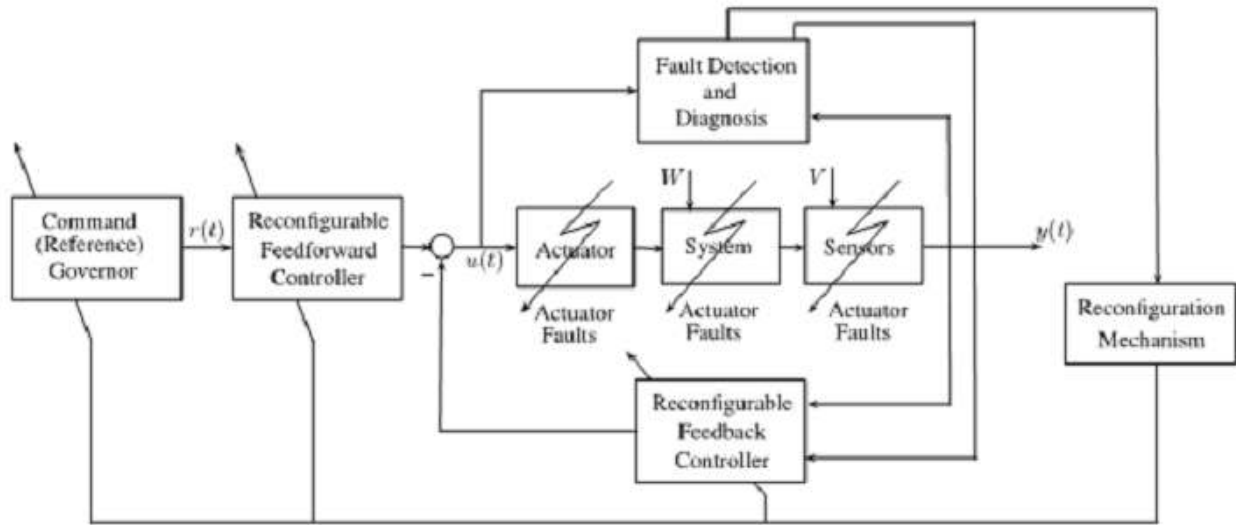


Figure2.1: General Structure of Fault-Tolerant Control Systems

System consists of two major blocks: fault/failure detection and diagnosis block, and reconfiguration mechanism block. Each of the two components has received a lot of attention from researchers throughout the history of the FTC development, customization, and improvement [27, 33].

Fault-tolerant control systems have been developed for different types of aircraft units: civil and military airplanes [34, 35], helicopters [36], UAVs [37], and near space vehicles [38]. However, most of the research efforts in the field of FTC were focused on specific limited problems: electro-mechanical actuator failures [39], loss of actuator efficiency [40], partial loss

of control [41], aircraft structural damage [34], etc. The need for a holistic approach that is capable of providing a comprehensive and integrated solution in this area has been widely acknowledged [9, 42, 43].

Undoubtedly, the ultimate mission is to design and implement such a fault-tolerant control system that would be able to cover various types of failures over wide regions of the flight envelope. In other words, the designed system must possess robustness. Several models of robust control system that could detect, identify, evaluate, and accommodate the failure have been proposed in this research field – [9, 35, 44]. Yet these efforts do not cover desirably broad areas of flight envelope and failure varieties. The complexity of modern aircraft models makes designing a comprehensive adaptive system extremely problematic. Hence, the control engineers had to broaden their minds and look for new ideas from different sources.

2.2 Immunity-based Systems

Utilizing biologically inspired methodologies as a basis for adaptive control systems has become a widely acknowledged practice in recent years. Among these systems are artificial neural networks, genetic algorithms, artificial immune systems [8, 17-21]. The more recent and promising artificial immune system approach is the least investigated and implemented. However, the biological origin - natural immune system - possesses such a complexity, robustness, and extensive set of capabilities that it could be an excellent source of inspiration for the design of comprehensive integrated fault-tolerant control system. There is a great number of different texts describing the fundamentals of natural immune system in all details; one such is by [45].

The immune system consists of a large network of cells, tissues, and organs that operate together to protect the organism. Immune system is a subject of great interest for researchers because of its capability to perform a vast variety of complicated tasks including pattern recognition, noise tolerance, generalization, optimization, etc. [24]. However, the three main features that make the immunity-based-approach very promising are [45]:

- 1) *Discrimination* - capability of distinguishing between the substances that belong to organism and the ones that do not
- 2) *Specificity* - the immune system can recognize an extremely large number of different antigens and induce a specific lymphocyte for each of them
- 3) *Memory* - if the invader that had attacked in the past is encountered, the secondary response of the immune system will be more rapid and aggressive

There are several widely used techniques inspired from the natural immune system features:

- *clone selection algorithm*- most commonly applied to optimization and pattern recognition problems [46],
- *negative selection algorithm* (inspired from the selection processes that occur during the generation of T-cells in thymus) - typically used for classification and pattern recognition[47],
- *immune network algorithms* - mostly used for clustering, data visualization, optimization, and control [48],
- *dendritic cell algorithms*- algorithms inspired by natural dendritic cells (DCs) behavior and mostly focused on multi-scale processing and pattern recognition [22, 49].

What is AIS? “*Artificial Immune Systems are adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving*”[50]. The AIS was first mentioned in the articles on neural networks published in the mid-1980s by Farmer, Packard and Perelson [51], and Bersini and Varela [52]. However, as a new computational paradigm on its own right, the AIS emerged in the mid-1990s. The concept has proven itself as an inviting ground for research in various applications such as anomaly or breakage detection [53-56], data mining [54, 57], computer safety [47, 58-62], pattern recognition [63, 64], and adaptive control [24, 51, 65].

In recent years, the AIS-based methodology has shown promising results in flight failure detection, identification and evaluation (FDIE) [8, 10, 66]. The key idea in these research efforts was based on the distinguishing capability of biological immune system: self-non-self discrimination. When an aircraft operates under nominal conditions (none of the subsystems are failed), a certain configuration of flight features forms "self". After the failure of any subsystem occurs, a configuration of the features which does not match the "self" occurs - "non-self". Developing control systems within the AIS paradigm has been a major focus of the research groups from WVU and ERAU in recent years [8, 11, 22, 66-78].

2.3 Artificial Immune System for Failure Accommodation

The ultimate step in creating an integrated adaptive system which can ensure aircraft safety is *failure accommodation*. Once the failure is detected, identified, and evaluated, a certain necessary action must be taken to mitigate the malfunction consequences. If for FDIE problem the fundamental inspiring feature of the immune system was ability to distinguish between self and non-self, for accommodation problem there will be two different key features. The first is

ability to fight the intruders by producing special substances - antibodies, and the second – capability of remembering the past experience to react more aggressively if the same intruder comes back again. Some researchers even describe immune system as a "second brain" for the ability to remember past experiences [79].

Certain success has been achieved lately in developing AIS-based adaptive control systems. A very good example of implementing AIS to create an adaptive control system is presented in an article by Karr [24]. Here, a control system operation is divided in two parts. The first part is reaction to the failures that already had taken place before (immune system memory). The second part is to produce new control commands to new failures using genetic algorithms (adaptive immune system mechanism).

Another case of using human memory/learning system as an example is presented by Weng [80] for micro aerial vehicle flapping motion control. In this work, assimilation between human memory system and neural-memory network is taken as a ground for creating an adaptive control system.

AIS-based failure accommodation can also be approached by imitating a biological feedback between activation and suppression of the antibodies generation. This mechanism can be converted into an adaptive component augmenting a baseline controller [23]. Described methodology was developed and successfully implemented to a supersonic fighter model by Perez [81]. A significant improvement in performance has been achieved by implementing the AIS based controller as compared to a regular PID controller.

AIS has proven itself as a promising ground for building integrated adaptive control systems, although no systematical theoretical background has been provided to support this methodology so far. However, one has to remember that the AIS concept is still relatively young and the most part of research and work is to be done in the future.

Chapter 3 The Aircraft Subsystem Failure Detection, Identification, Evaluation, and Accommodation (FDIEA) Problem

3.1 FDIEA Problem Formulation

The aircraft subsystem FDIEA problem has to be defined in details: the subsystems addressed, the factors and levels of the failures considered, and the mission, must be established for each stage of the problem. The four processes grouped under the acronym FDIEA must be performed in subsequent phases in order to avoid unrecoverable post-failure fight conditions, regain equilibrium, and continue the mission. The general simplified methodology [82] of solving the FDIEA problem is presented in Fig. 3.1.

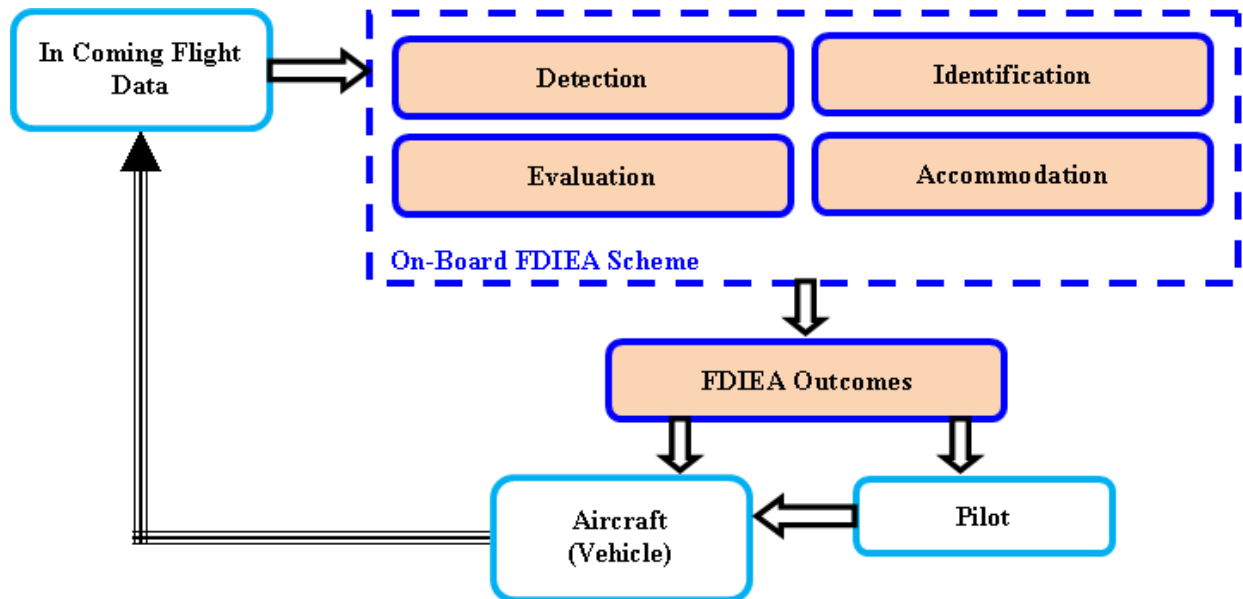


Figure 3.1: FDIEA General Process Diagram

3.2 FDIEA Problem Components

3.2.1 Detection

The detection represents the process of announcing that a generic malfunction of any subsystem has occurred. Any one or several subsystems may fail. These subsystems can be actuators, sensors, structural or propulsion systems, etc. In addition to the named ones, items such as software, human pilot, and the environment can be considered in the process. This would allow broadening the abnormal conditions set, such that situations like pilot fatigue, non-standard weather or turbulence would be considered as subsystem abnormal conditions as well within a generalized framework [82].

3.2.2 Identification

The identification or isolation process determines which subsystem has failed. The identification process can have different phases depending on the complexity of the addressed system. The first phase would be declaring which exact subsystem group has failed (e.g. sensor, actuator). The second phase would be specifying the element within the subsystem group. For example, if the first phase declared actuator failure, second phase should determine whether it was aileron, stabilator, or rudder. The last phase will define whether right or left aircraft surface failed (for the actuator example) [82].

3.2.3 Evaluation

Once the failure has been detected and correctly identified, the evaluation process has three aspects to address. One is of a qualitative nature and involves determining the type of failure. For example, the qualitative evaluation is expected to determine if an actuator failure consists of a locked actuator, or a freely moving control surface, or a reduction of control efficiency. The other two aspects are of a quantitative nature and can be defined as direct and

indirect. Direct evaluation has to provide a certain value - magnitude of failure e.g. severity. For example, right rudder locked at +4 degrees, or pitch rate sensor bias of +5 deg/s etc. The indirect evaluation is responsible for reassessment of flight envelope at post-failure condition with taking into consideration the entire set of limitations and constraints on the performance produced by the presence of the failure [82].

3.2.4 Accommodation

The ultimate step of the FDIEA problem is accommodation. It represents the actual reaction of pilot-aircraft system - compensating the failure by generating control commands. Within the second conceptual approach for solving the accommodation problem, there are two scenarios to be considered. The first one involves an unknown failure that requires a specific new compensation. The second scenario, assumes that substantial information about the failure and its compensation is available and stored in the AIS. This can be achieved by memorizing the dynamic fingerprint during first scenario situations, by generating and recording pilot compensation during actual or simulated failures, or by properly composing bits of information within the self and non-self. A simplified block diagram illustrating the process is shown in Fig 3.2.

In order to accomplish the last step of FDIEA problem successfully, each of the three components prior to the accommodation must complete their prescribed tasks accurately. In other words, the information about which subsystem failed, what exact part of subsystem failed, what is the magnitude of the failure must be provided. This is needed for the fault-tolerant control system to define whether this failure had been faced before or not. If positive, then appropriate commands would be retrieved from prior pilot performance, and if not, then an original algorithm has to be adopted to overcome the malfunction. Within this thesis only the extraction of the control commands based on previously encountered failure will be addressed.

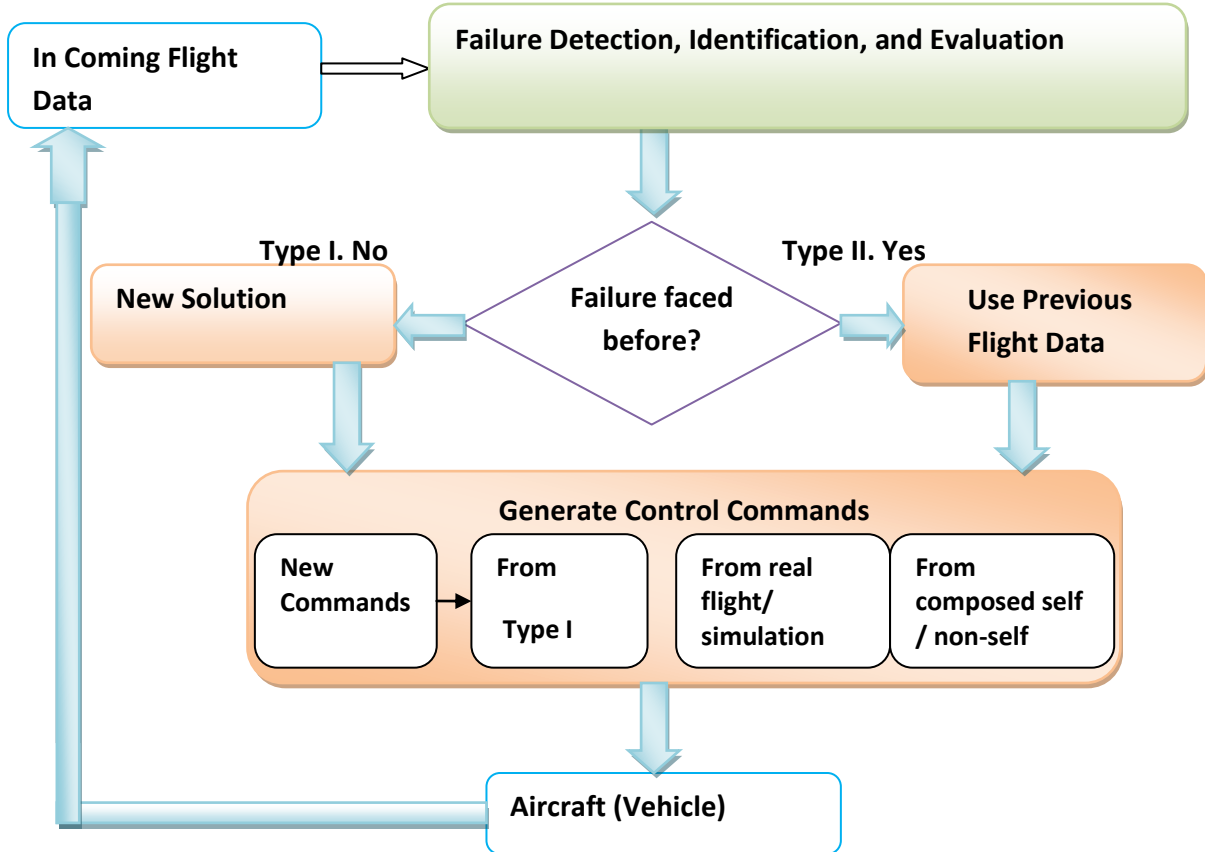


Figure 3.2: Two Types of Failure Accommodation

3.3 FDIEA Problem Overview

Figure 3.3 represents general aspects of the FDIEA process and summarizes its ultimate objectives:

- 1) Detect malfunction
- 2) Identify the subsystem and component subject to the failure
- 3) Provide qualitative and quantitative information about the failure. Evaluate and assess the possible flight envelope reduction due to the presence of the failure. Provide information

to the pilot and control system for decision making regarding modification of control and navigation strategies

- 4) Generate control commands and have the aircraft subjected to those in order to avoid/minimize the undesired consequences of the failure

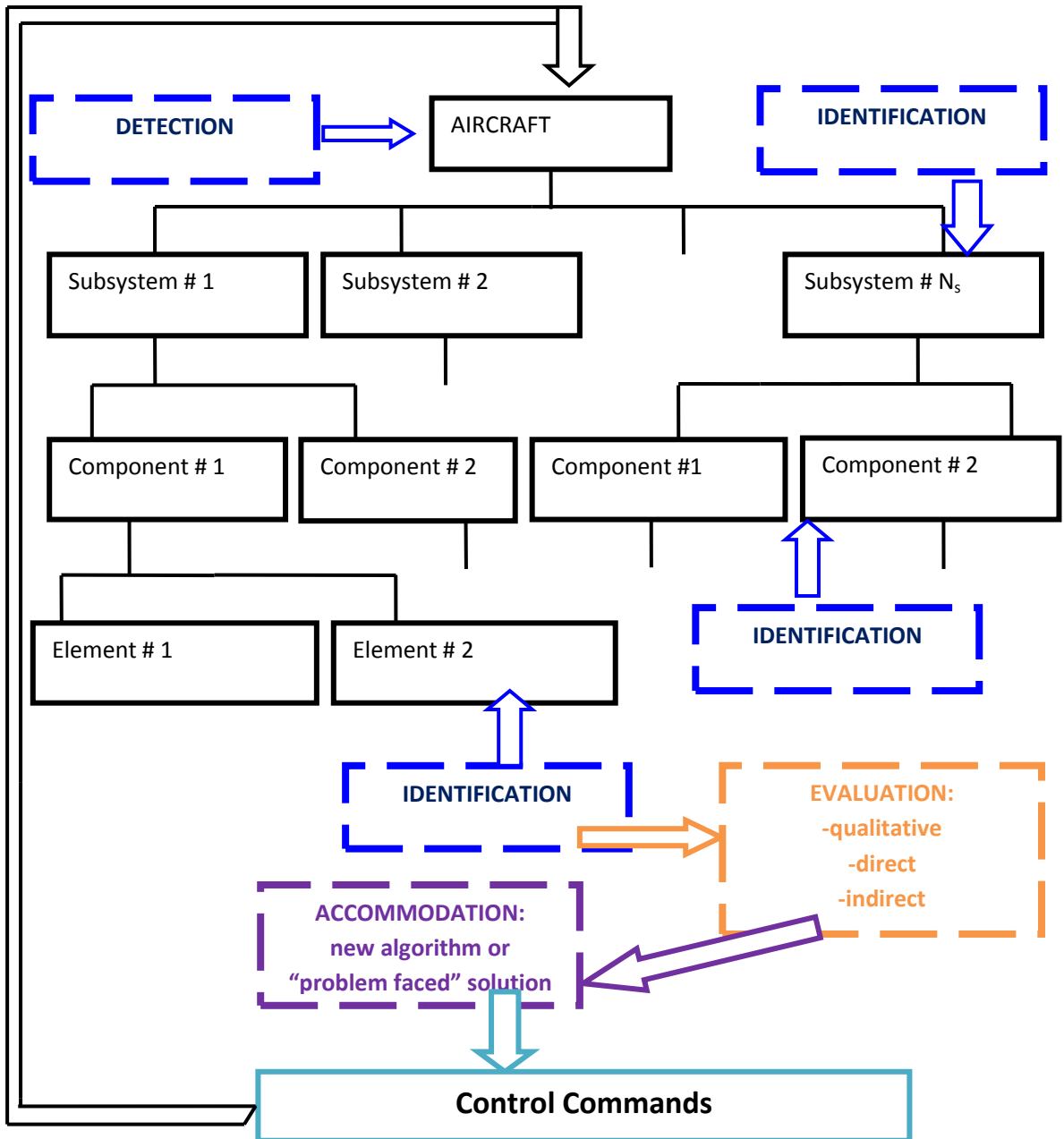


Figure 3.3: FDIEA Problem Block Diagram

Chapter 4 FDIEA Problem within the AIS Paradigm

4.1 Natural Immune System

The immune system is designed to defend the body against foreign or dangerous invaders. Such invaders include bacteria, viruses, parasites, cancer cells, and even transplanted organs and tissues. The biological immune system consists of two components: innate system and adaptive system [83].

Innate system does not require a previous encounter with a microorganism or other invader to work effectively. It responds to invaders immediately, without any need to learn to recognize them. Mostly, the following cells are involved in the process performing specific roles [84]:

- Phagocytes - they ingest invaders. Phagocytes include macrophages, neutrophils, monocytes, and dendritic cells.
- Natural killer cells - are formed ready to recognize and kill cancer cells and cells that are infected with certain viruses.
- Antigen-presenting cells - recognize invaders. They consist of DCs, macrophages, and B-cells. The DCs use antigen biochemical markers associated with danger- and safe-signals to regulate the production of T-cells.

Adaptive system is built through previous exposure to invading antigens. Here, lymphocytes (B-cells and T-cells) encounter an invader, learn how to attack it, and remember the

specific invader so they can attack it even more efficiently the next time they encounter it. T-cells are the component of adaptive system with the most important role in the defending process. These cells are first generated in bone marrow and proliferate in thymus through a pseudo-random genetic rearrangement mechanism. The DCs make sure that those T-cells, whose markers match the dangerous antigen, are produced in a larger quantity. On the contrary, if the environment is safe, this process is suppressed by suppressor T-cells (Ts-cells). Special helpers T-cells (Th-cells) activate the generation of cytotoxic T-cells (Tc-cells) and B-cells that produce antibodies specific to the antigen. The block diagram of the immune system feedback mechanism is presented in Figure 4.1 [79]:

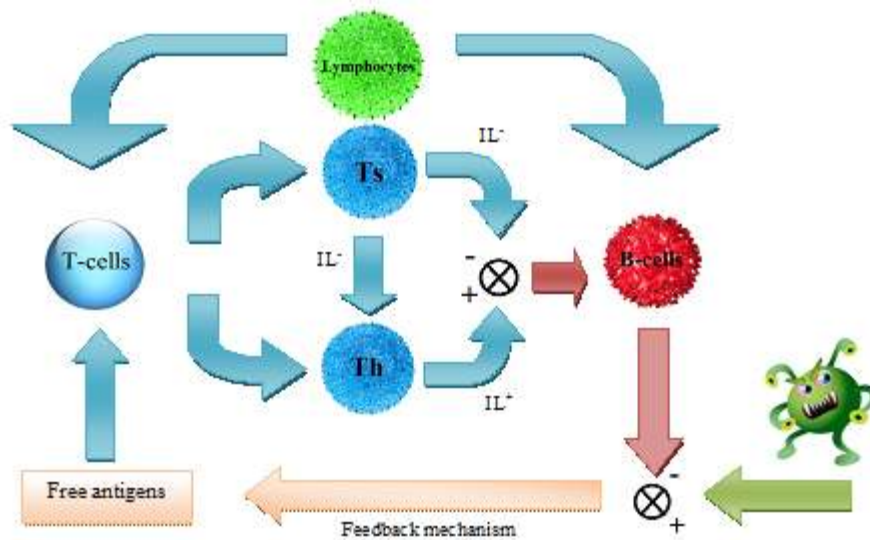


Figure 4.1: Block Diagram of the Immune System Feedback Response

4.2 AIS Paradigm

When developing an AIS, it is important to remember that the final objective is not to duplicate the actions of the immune system, but to borrow the ideas from natural bio-world and

create a certain computational tool to solve the given problem using those ideas. There are four crucial features of biological immune system that should be transposed to the AIS:

- 1) Ability to accurately distinguish between what belongs to the body (self) and what does not (non-self)
- 2) Capability of fighting and destroying the dangerous invaders
- 3) Potential for facing and handling a great variety of known and unknown harmful substances
- 4) Ability to remember the information about past encounters and facilitate more rapid and aggressive response (generate corresponding T-cells and B-cells) if the same antigen invades the organism again.

AIS-based FDIEA will consist of three main components functionally connected in a closed loop as shown in Fig. 4.2 [82]:

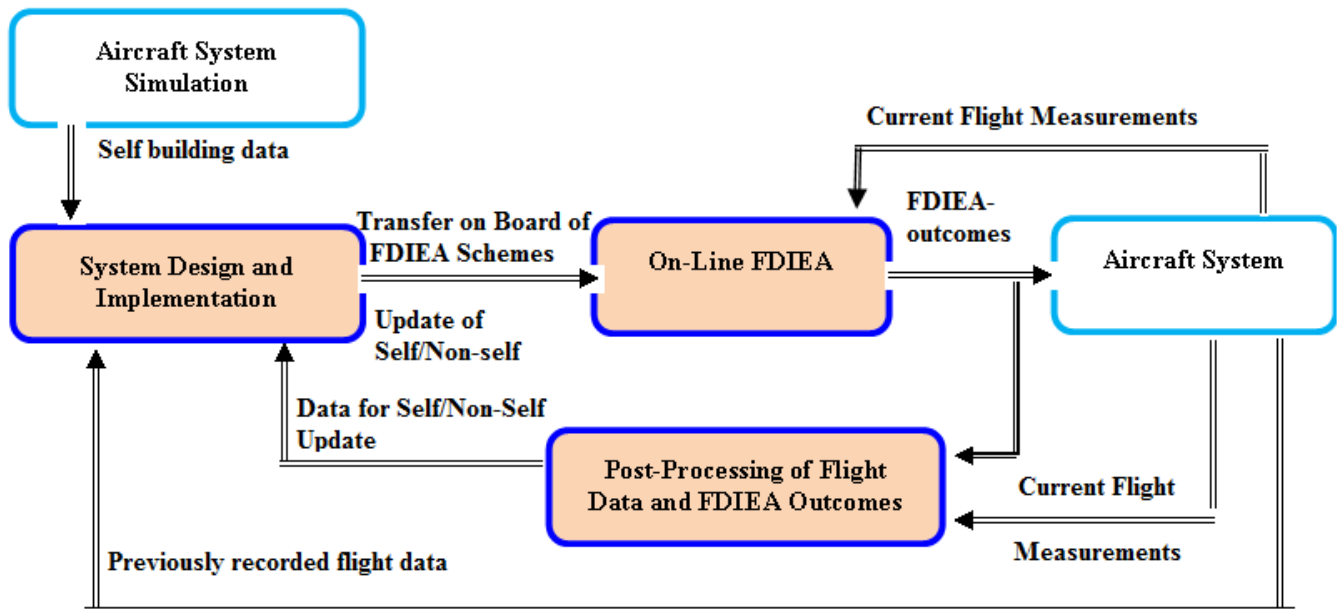


Figure 4.2: AIS-based FDIEA

- Pre-processing of information and flight data
- On-line AC detection, identification, evaluation, and accommodation
- Post-flight analysis of data and FDIEA outcomes

The *pre-processing* stage represents the development of aircraft subsystem FDIEA comprehensive and integrated scheme. This includes clear definition of the scheme components such as aircraft subsystems, abnormal conditions types (known and unknown), failure severity scales, flight envelope features, etc. Only after every item of the system is clearly defined, a robust model of self and non-self can be built.

The *on-line FDIEA* process implies the real time operation of the FDIEA scheme. Current flight measurements at a certain sampling rate are compared against the detectors, identifiers, evaluators, and compensators. This process is followed by generating the FDIEA outcomes. These outcomes are transferred to the pilot and/or aircraft and to the post-processing block.

The *post-processing* block is responsible for analyzing and evaluating the on-line scheme outcomes. Outcomes such as failed detection, false alarms, incorrect evaluations, wrong control commands need to be assessed. Using this assessment the appropriate changes must be added to the FDIEA scheme in order to improve the overall flight performance in the future.

Chapter 5 Biologically Inspired Failure Accommodation

5.1 AIS Accommodation Problem Outline

The main objective of this research effort is to investigate the possibility of using AIS to extract compensatory commands from pilot performance followed by implementing those commands to the same aircraft. Within this research effort, this process will be referred to as “accommodation problem”. However, note that the actual accommodation problem is more complicated and firmly connected to the other parts of FDIEA problem.

It is important to clarify that we assume successful execution of all the steps prior to the accommodation part in the FDIEA process. This means that a certain failure was detected, identified, and evaluated within the framework as it was formulated in Chapter 4. The focus in this thesis is on the accommodation part only; therefore, data used for building the self/non-self is limited and not necessarily sufficient to address the other components of the FDIEA process. The pilot will perform ad-hoc designed flight simulation tests under nominal and failure conditions. Afterwards, the data from these tests will be used to build the specific self/non-self and serve as base ground for building the accommodation algorithm. To validate the proposed algorithm, failures, previously handled by the pilot, will be injected during the simulation of the same maneuvers.

In order to solve the accommodation problem, a set of certain steps must be accomplished. There are six main components of the problem connected with each other. Figure

5.1 presents the block diagram of the failure accommodation problem main components. Every component of this block diagram has to be accurately defined within the AIS paradigm.

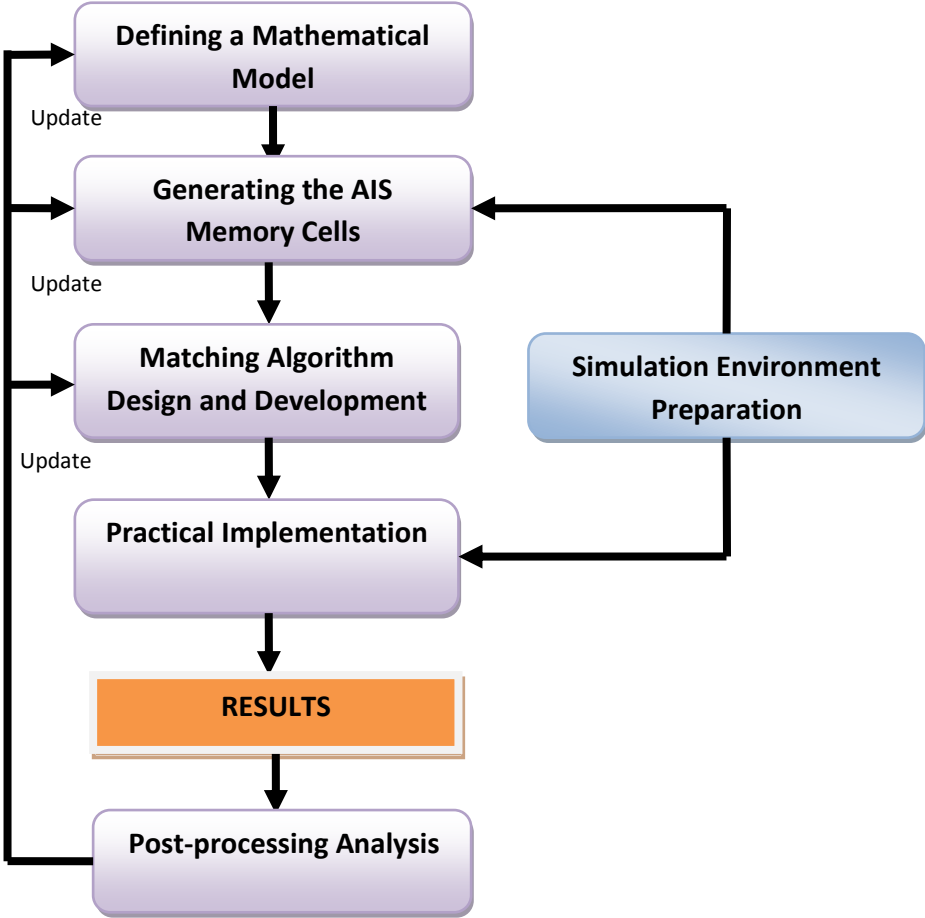


Figure 5.1: Accommodation Problem Main Components

5.1.1 Defining a Mathematical Model

An appropriate mathematical model with components that are maximally pertinent to the AIS terminology must be proposed. Defining the right set of variables, the form of their representation and interaction is crucial, because the entire following algorithm, and the AIS components definition will be based on the chosen model.

5.1.2 Generating the Artificial Memory Cells

There is a certain flight path that the aircraft should follow. There are also specific flight conditions which the aircraft should maintain during the flight (e.g. constant altitude, constant velocity). These parameters define the task and are considered as the input to the *pilot+aircraft* system. After being informed about this flight path and about the specific flight conditions, an experienced pilot intends to execute the task as close as possible to command. An aircraft following the desirable flight path under the commanded flight conditions will be assimilated to a healthy organism. Every value of in-coming flight variable that alters from the desirable value during the pilot performance will be defined as invading entity/antigen. The control commands provided by the pilot that are trying to bring the aircraft to the desirable state will be defined as immune system antibodies. This process is an analogy to the immune system being affected by a disease for the first time. The adaptive immune system fights back and generates antibodies, which eventually eliminate the disease. The recorded and saved flight data will define a set of created artificial memory cells (B-cells and T-cells, and antibodies) and antigens.

Artificial memory cells generating process is illustrated in Fig. 5.2:

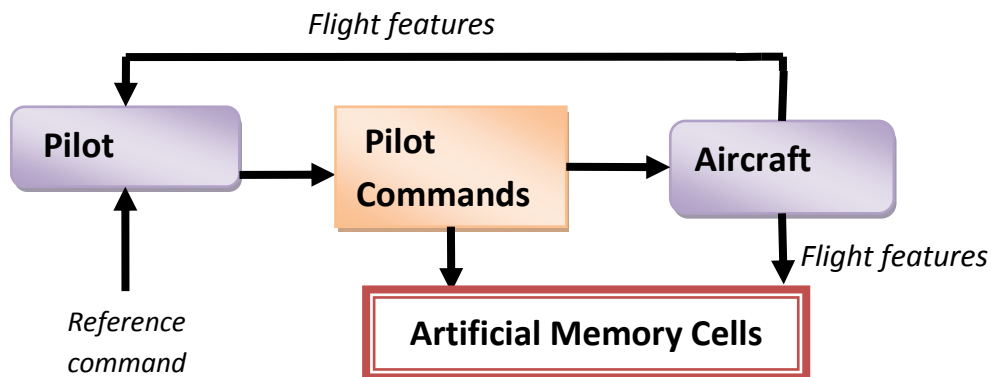


Figure 5.2: Simplified Artificial Memory Cells Generating Process

5.1.3 Matching Algorithm Design and Development

The secondary response of the immune system to the same invaders is more rapid and aggressive. This happens mainly because the memory has information about the invaders (antigens) and the cells that are needed to suppress those invaders (T-cells and B-cells, antibodies). Unlike the natural immune system, our "aircraft model system" cannot automatically match the incoming antigens with the corresponding antibodies, and T-cells and B-cells. That is why at this stage, an algorithm of finding the best corresponding match should be designed and implemented.

The simplified version of the entire process can be described as follows. An in-coming set of antigens is compared to the sets of antigens that had been previously encountered by the system. Once the most similar set is found, a set of antibodies that had been used for suppressing the antigens is extracted. Note that for each set of antigens there is a set of antibodies stored in the AIS memory. The overview of the matching algorithm is presented in Fig. 5.3:

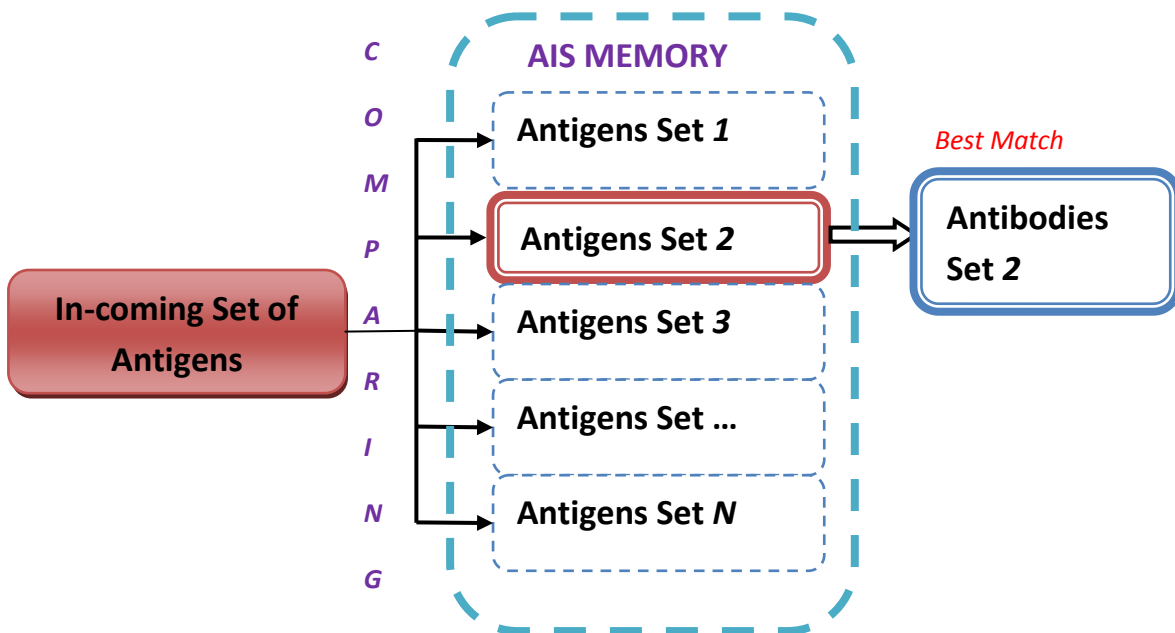


Figure 5.3: Matching Algorithm Process

5.1.4 Preparation of the Simulation Environment

In order to develop, design, and demonstrate the operation of the accommodation algorithm, the simulation environment has to be prepared very carefully. One needs to define properly all the flight conditions for the pilot so that he/she could perform the task accurately. Quality of the data acquired from pilot's performance plays an important role in obtaining good results.

5.1.5 Practical Implementation

After successful generation of the artificial memory and algorithm design, the entire mechanism can be implemented within the simulation environment. This process can be described as a second encounter of disease invasion. The immune system recognizes the antigens, because the information is already in the memory, and fast response will be provided which will result in the destruction of antigens. For our model, all in-coming flight features to the aircraft will be considered as previously identified antigens which can be handled by the control command from the memory.

5.1.6 Post-processing Analysis

At this stage, the control commands produced by the designed mechanism during the flight simulation are analyzed and assessed. Based on the results of the assessment, conclusions about the eligibility of the entire scheme are made. Afterwards, the necessary corrections are added to the scheme in order to improve the future performance.

General aspects of AIS-based failure accommodation problem are illustrated in Fig. 5.4. It describes the entire failure accommodation process. The data from the reference input block and from the current flight measurements are compared first. The difference is considered

as an antigen to the system. In-coming antigen is compared to the previously suppressed antigens in the matching algorithm block. An outcome of this block is a set of control commands extracted from the AIS memory corresponding to the best match that has been found. This set of control commands are the input to the aircraft and determine its future behavior. Afterwards, the entire process is repeated. The data from all the blocks are collected to the post-processing analysis block. This is needed for the following analysis of the results, which is expected to provide the opportunity to improve the performance level of the entire algorithm.

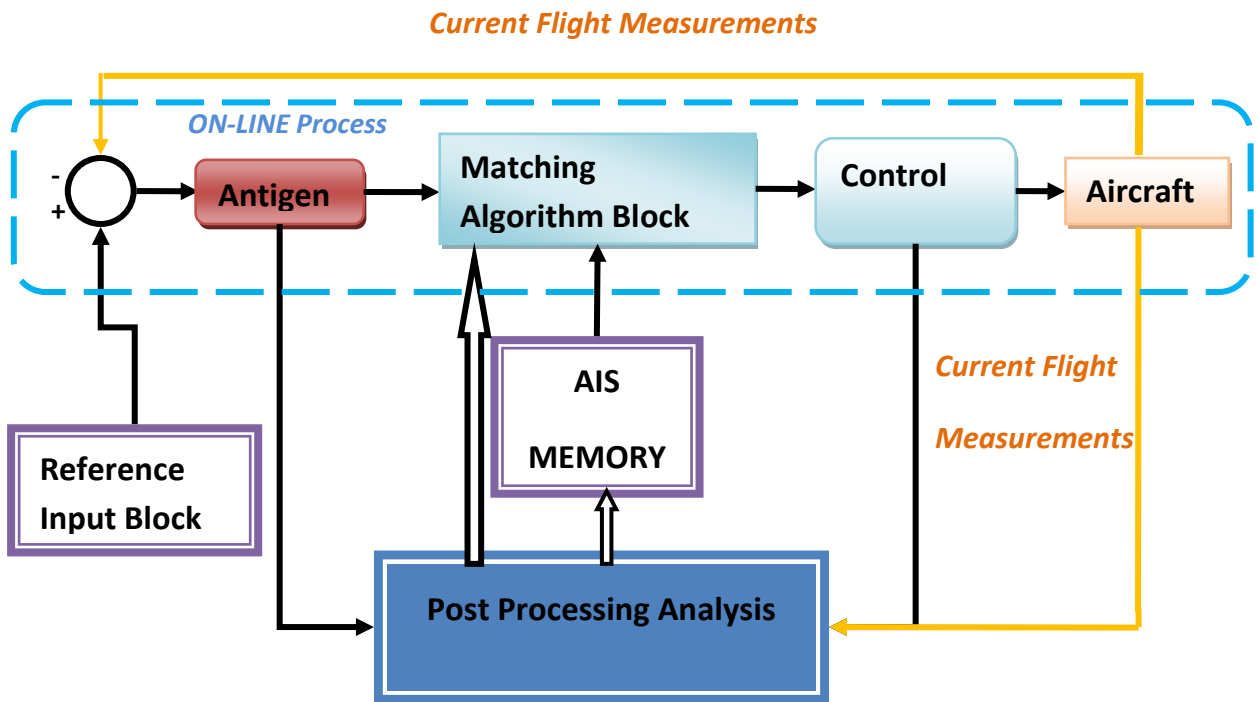


Figure 5.4: AIS-Based Failure Accommodation Process

It should be emphasized that considering in-coming values of flight parameters under both nominal and abnormal conditions as antigens is essential to the proposed approach. In the case of the failure, all the information about it is available, as an outcome from the detection, identification, and evaluation phases of the FDIEA framework.

5.2 Mathematical Model

A set of *flight features* (FF) must be selected such that they capture the dynamic fingerprint of the aircraft operation at normal and abnormal conditions. FF can be defined as a set of N_f components:

$$FF = \{ feat_1, feat_2, \dots, feat_{N_f} \} \quad (5.1)$$

where $feat_j$ ($j=1,2, \dots, N_f$) is the feature (e.g. roll rate, velocity) that was selected for the algorithm.

An input (or mission objectives) for a pilot can be defined by matrix REF . Every row in this matrix is represented by a vector REF_v of length N_f .

$$REF_v = [df_1, df_2, \dots, df_{N_f}] \quad (5.2)$$

Each element df_j of the vector REF_v represents a desirable value of the corresponding feature in set FF . Duration of the flight is defined by number of time samples ts . Consequently, matrix REF consists of ts rows.

An outcome of pilot's performance is a matrix PF . Every row in this matrix is represented by a vector PF_v of length N_f .

$$PF_v = [pf_1, pf_2, \dots, pf_{N_f}] \quad (5.3)$$

Each element pf_j of the vector PF_v represents a value of the corresponding feature of set FF obtained from pilot performance. Matrix PF consists of ts rows as well.

Pilot generated control commands are defined by matrix PC . Every row in this matrix is represented by a vector PC_v of a length N_{cc} , where N_{cc} is the number of control commands.

$$PC_v = [ctrl_1, ctrl_2, \dots, ctrl_{N_{cc}}] \quad (5.4)$$

Each element $ctrl_j$ of the vector PC_v represents a control command provided by pilot at each time sample of the flight. Matrix PC consists of ts rows.

Vector of in-coming values of features at every time sample is defined as INC_v . The length of the vector is N_f .

$$INC_v = [if_1, if_2, \dots, if_{N_f}] \quad (5.5)$$

Each element if_j of the vector INC_v represents an in-coming value of the corresponding feature of set FF .

5.3 AIS Memory Cells Representation

The detailed configuration and representation of antigens, B-cells and T-cells, and antibodies must be defined. It is important to note that correspondence of biological immune system components to elements of the AIS is conceptual and must be understood in a generic way, in some instances lacking a perfect similarity.

5.3.1 Antigens

Set of antigens AG_v will be presented as a difference between in-coming data and reference input (pilot mission). This difference is defined by subtraction of corresponding elements of vector INC_v from corresponding elements of vector REF_v at specific time sample tsc .

$$AG_v = REF_v - INC_v \quad (5.6)$$

$$AG_v = [df_1 - if_1, df_2 - if_2, \dots, df_{N_f} - if_{N_f}] \quad (5.7)$$

As a result of the subtraction, vector AG of length N_f is obtained.

Depending on flight scenarios, the content of the *antigen* cell might be extended and include not only values of differences but actual values of incoming features as well:

$$AG_v = [df_1 - if_1, \dots, df_{N_f} - if_{N_f}, if_1, \dots, if_{N_f}] \quad (5.8)$$

These additional features are selected according to a specific task and prescribed maneuver.

Collection of AG_v vectors over the entire flight time defines the matrix AG . The overall number of time samples during the flight is ts , consequently matrix AG has ts rows.

5.3.2 T-cells and B-cells

The immune system memory has information about T-cells and B-cells produced during the prior infection encounters, and antibodies that were used to suppress these antigens.

Within the AIS paradigm, T-cells together with B-cells will be assimilated to a difference between pilot features data and reference input data. Set of ***T-cells and B-cells*** for a specific time sample tsc is a vector TB_v , defined by subtraction of elements of vector PF_v from corresponding elements of vector REF_v .

$$TB_v = REF_v - PF_v \quad (5.9)$$

$$TB_v = [df_1 - pf_1, df_2 - pf_2, \dots, df_{N_f} - pf_{N_f}] \quad (5.10)$$

Similarly to AG_v vector, a TB_v vector might need an extension and include some actual values of features obtained from pilot performance:

$$TB_v = [df_1 - pf_1, \dots, df_{N_f} - pf_{N_f}, pf_1, \dots, pf_{N_f}] \quad (5.11)$$

The additional features for TB_v should be the same as for AG_v . Collection of TB_v vectors over the entire flight time defines the matrix TB . The overall number of time samples during the flight is ts consequently matrix TB has ts rows.

5.3.3 Antibodies

The set of ***antibodies*** for a specific time sample tsc is a vector AB_v defined by corresponding vector from matrix PC .

$$AB_v = PC_v \quad (5.12)$$

The collection of AB_v vectors over the entire flight time defines matrix AB . Structure of AIS memory is presented in Fig. 5.5:

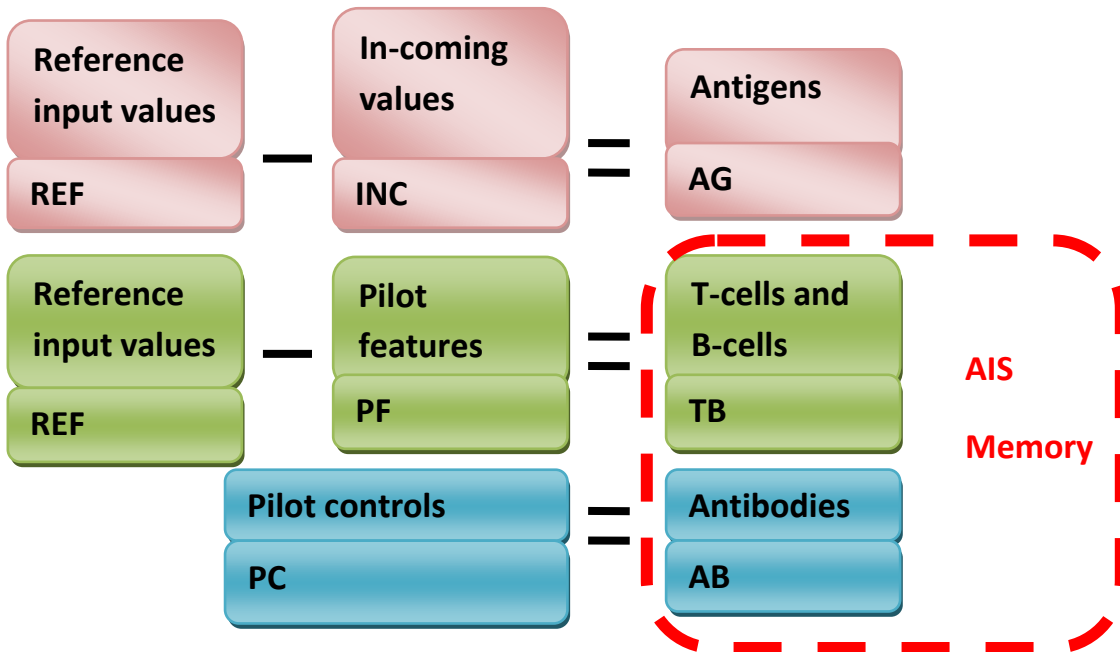


Figure 5.5: AIS Memory Structure

5.4 Matching Algorithm

During the entire time period dedicated to this research effort, the matching algorithm has been adjusted several times. Each adjustment makes the algorithm more sophisticated, and, as expected, improves the performance. In this section, each stage of designing the matching algorithm will be presented.

5.4.1 Instant Matching Algorithm

At each time sample an in-coming antigen AG_v is compared to the sets of T-cells and B-cells from matrix TB . Once the “best match” is found, a corresponding set of antibodies AB_v is retrieved from matrix AB . As a result, we have control commands dictating the behavior of the aircraft. The “best match” is a row vector TB_v (from matrix TB) which is most similar to an in-coming antigen vector AG_v .

Let vector ERR be difference between AG_v and TB_v .

$$ERR = AG_v - TB_v \quad (5.13)$$

Then sum S can be defined as follows:

$$S = \sum_{i=1}^{N_f} abs(ERR(i)) \quad (5.14)$$

where $ERR(i)$ is an element of vector ERR . Therefore, a row vector TB_v that leads to the lowest value of S is defined as the “best match”. It should be noted that more sophisticated matching algorithm may be used to improve performance.

5.4.2 Artificial Memory Cell Matching

The matching algorithm presented in 5.4.1 relies on comparing in-coming antigen with AIS memory components at a single time sample, which typically does not ensure adequate robustness if feature derivatives are not included. In other words typically for higher performance, one should take into consideration the current state and the state in which aircraft had been before (i.e. several time samples before) or the trend. Also, considering time history of variables enhances the "memory" concept. It is envisioned that similar effects can be achieved by either including the feature derivatives or using a floating time window, or both. Including these elements is expected to increase the performance of the accommodation process; however, it will also require more computational power. It is expected that using a floating window of a certain width should be sufficient enough in terms of robustness. Undoubtedly, the larger the size of the window is, the more computational time will be necessary.

The visual representation of the described matching process is illustrated in Fig. 5.6:

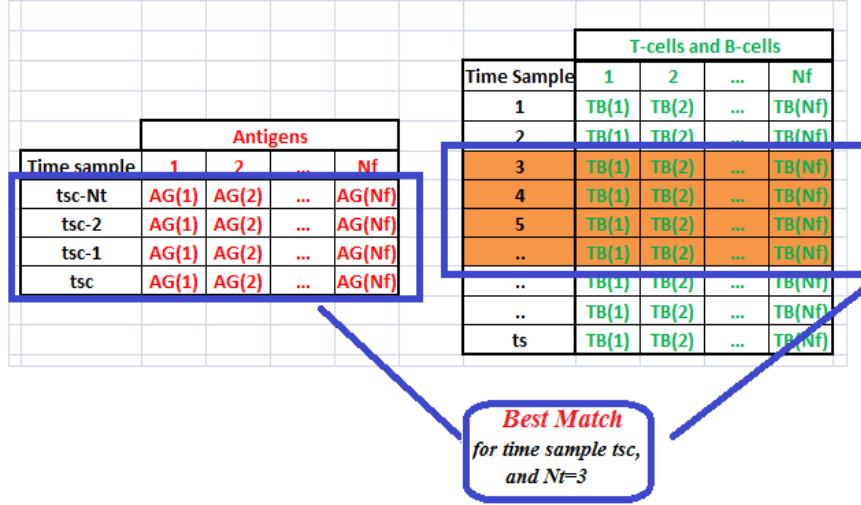


Figure 5.6: Matching Algorithm with Floating Time Window

Let us denote N_t the size of a floating window that is taken into account during the matching process. Then vector $(ERR_v)_j$ will be redefined as follows:

$$(ERR_v)_j = (AG_v)_j - (TB_v)_j, j = tsc - N_t : tsc \quad (5.15)$$

where tsc is a current time sample. Consequently, the sum S will be redefined as well:

$$S = \sum_{j=tsc-N_t}^{tsc} \sum_{i=1}^{N_f} abs(ERR(i))_j \quad (5.16)$$

5.4.3 Normalization Process

Vector ERR is a result of subtraction of vector TB_v from vector AG_v . Elements of vectors AG_v and TB_v are different values of flight features that were defined previously. Since different flight features are involved in the algorithm, their values might vary in different ranges. In order to be consistent during the matching algorithm one must perform normalization. A linear interpolation is used to normalize all the values:

$$X_{norm} = \frac{X - MinVal}{MaxVal - MinVal} * (Max - Min) \quad (5.17)$$

where

- X_{norm} - is a current difference value in a new normalized system
- X - is a current difference value that needs to be normalized,
- $MinVal$ and $MaxVal$ - minimum and maximum difference values of the considered feature within the certain flight scenario respectively,
- Min and Max - minimum and maximum values in a new normalized system respectively.

For ensuring a better coverage of the possible range within a certain framework, a margin M can be added to both minimum and maximum values of the considered features:

$$MaxVal = MaxVal + M \quad (5.18)$$

$$MinVal = MinVal - M \quad (5.19)$$

Example of normalization process is illustrated in Fig. 5.7:

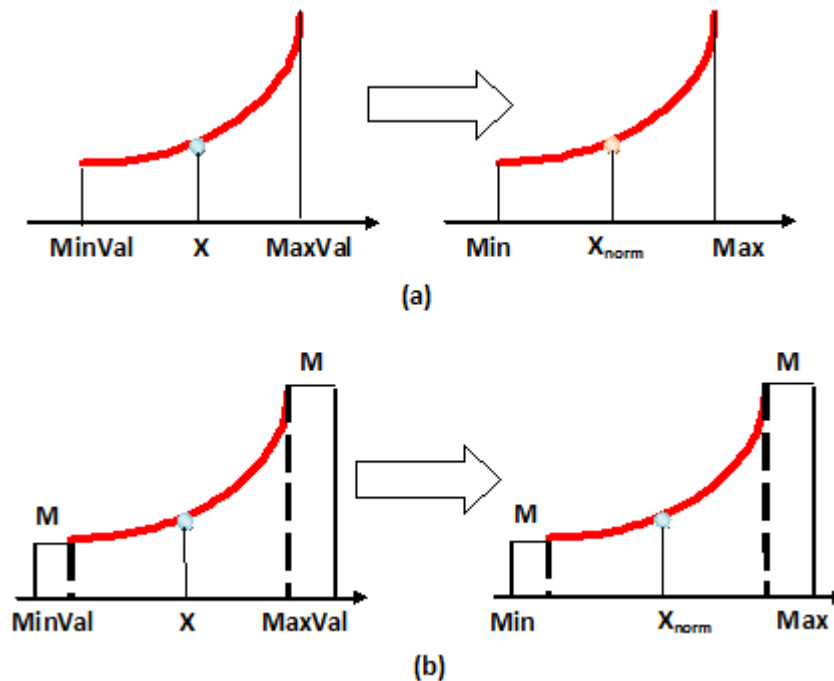


Figure 5.7: a) Normalization Process without Additional Margins. b) Normalization Process with Additional Margins

5.4.4 Weighting the Features

Due to the large variety of the flight features and the complexity of the entire aircraft system including structural and aerodynamic characteristics, the values of different features typically have different numerical ranges. Some of the failures have more relevance in defining the dynamics of the flight than others. During the algorithm development and implementation it is essential to know to which flight feature one must pay more attention. For example, during the flight with an aileron failure, the error of the roll rate variable significantly affects the behavior of the aircraft. Therefore, to ensure that this error is tracked and matched accurately, it may be beneficial to increase the weight of this feature within the matching algorithm.

Example of increasing weight of a certain flight feature is illustrated in Fig. 5.8. As shown, the weight of flight feature number 2 is increased by W . This means that every feature value in the second column is multiplied by the number W for both antigens and T-cells and B-cells matrices. After the weighting is completed, matching process is initiated. Note that more than one flight feature can have its weight altered.

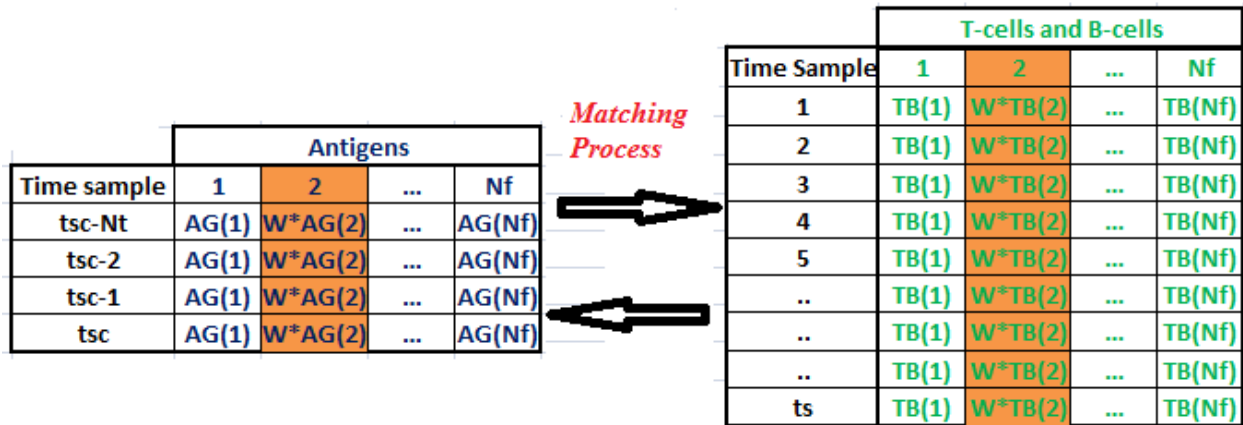


Figure 5.8: Matching Algorithm with Increased Weight of Features

Chapter 6 Simulation Environment

6.1 WVU Flight Simulator

The experimental data acquisition, practical implementation of developed algorithms, validation, and results assessment were all executed within the WVU simulation environment, which consists of the 6-DOF motion-based flight simulator and PC desktop simulator.

6.1.1 Motion-Based 6-DOF Flight Simulator

The WVU Motus 600 Flight Simulator (see Fig. 6.1) manufactured by Fidelity Flight Simulation, Inc., Pittsburgh, PA includes the following components [85]:

- 6-DOF motion platform driven by electrical induction motors
- Laminar Research X-Plane flight simulation software
- LCD mosaic wall four-monitor external visual display
- Instructors operating station
- Computer and control cabinet.

The motion platform provides adequate six-degrees-of-freedom translational and rotational motion cues (see Fig. 6.2). Electrical motors are used to drive the motion base, which represents a very versatile and inexpensive solution to this type of application. Motion drive algorithms convert the motion of the aircraft as resulting from the dynamic model into motion of the platform such that the perception of the pilot is optimized within the physical limitations of the ground based simulator [86].



Figure 6.1: The WVU 6-DOF Motion-based Flight Simulator

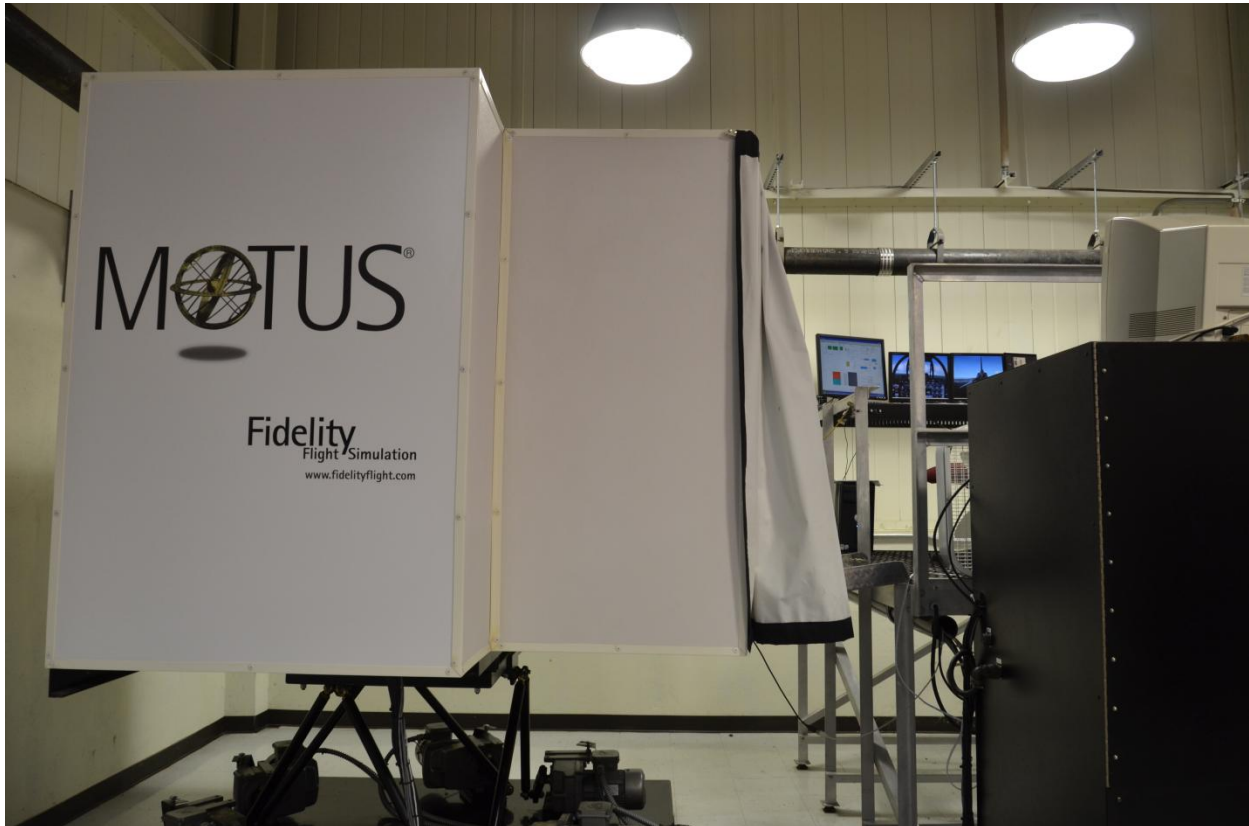


Figure 6.2: Flight Simulator Cabin and Motion System

The WVU Flight Simulator has been interfaced with an external computer so that the chosen aircraft model can be run within Matlab/Simulink environment and drive the motion platform [87] (see Fig. 6.3 [85]). For this purpose, Aircraft Health Management environment using Matlab/Simulink has been developed in WVU [88]. The entire system operation can be described as follows. Pilot input signals are transmitted from the cockpit into Matlab/Simulink model. In turn, Matlab/Simulink model is connected to X-Plane [89] software. Outputs of the model are transferred to X-Plane to control all the simulator subsystems and to generate the visual cues as well. However, to drive the motion base, the control signals must be sent from external computer to the motion computer directly (not through X-Plane software). This system, developed within the WVU Aircraft Health Management environment, allows us to use various Simulink aircraft models to drive the simulator [90].

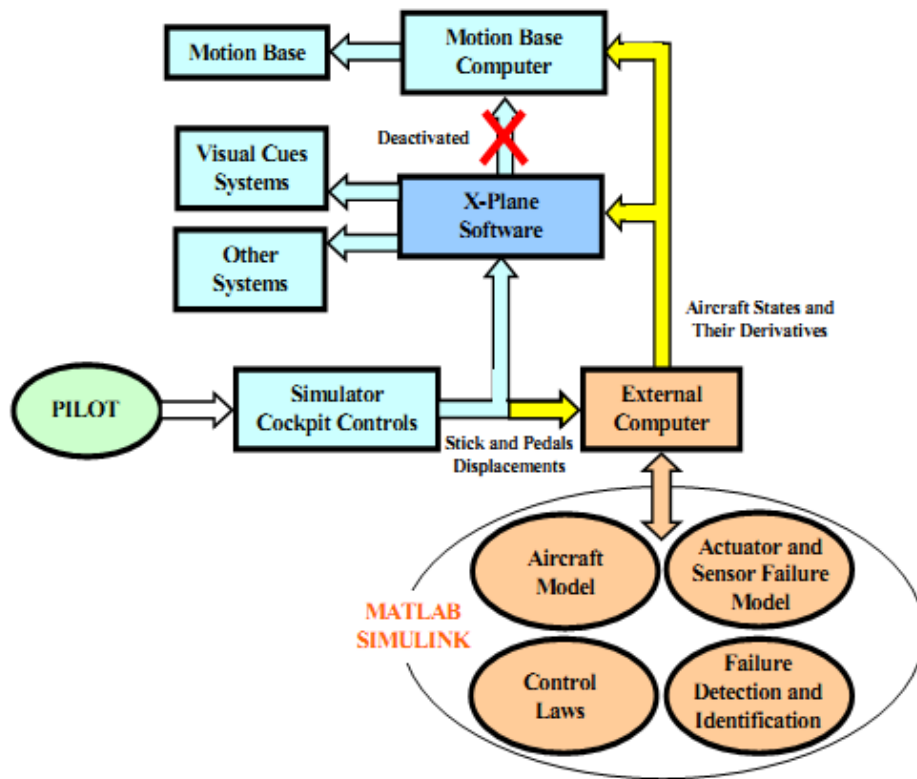


Figure 6.3: Interface of the WVU Flight Simulator with External Models

6.1.2 Desktop PC Simulator

Besides the motion-based flight simulator, a desktop PC flight simulator is available in the WVU simulation environment featuring identical mathematical models (see Fig. 6.4).

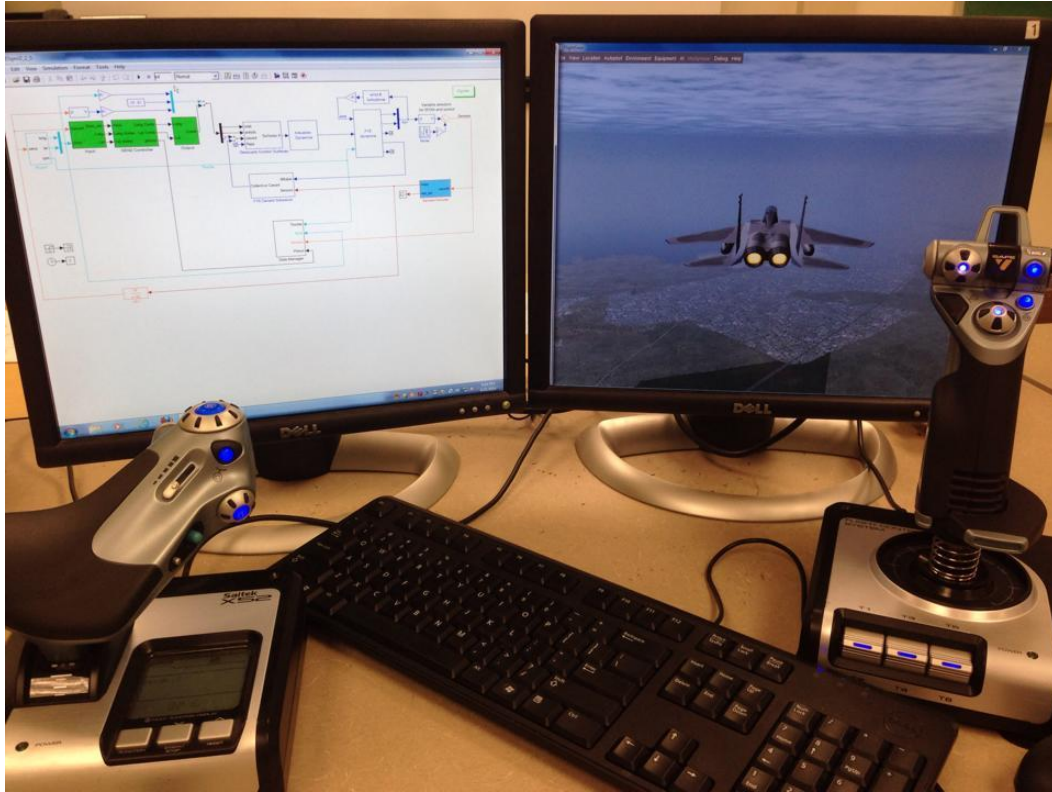


Figure 6.4: WVU Desktop PC Flight Simulator

The Matlab/Simulink model used for desktop computer flight simulator is the same as the one used for the motion-based simulator. The interaction between the pilot and the model is accomplished by an external joystick. The pilot input signals are transferred to the Matlab/Simulink model from the joystick attached to the desktop computer. There are two joysticks attached to the computer (see Fig. 6.4). The first joystick (on the right) provides control commands on longitudinal, lateral and directional channels. The second joystick (on the left) is a throttle and regulates the engine thrust level. The outputs of the model are sent to FlightGear to control all the subsystems of the aircraft and for visualization purposes.

FlightGear - software [91] used in desktop PC simulator - is an open-source flight simulator. It supports a variety of popular platforms (Windows, Mac, Linux, etc.) and is developed by skilled volunteers from around the world. Source code for the entire project is available and licensed under the GNU General Public License [92].

Figure 6.5 illustrates the top level Simulink diagram interfaced with WVU Flight simulator. The developed Simulink system includes non-linear dynamics of the aircraft and models failures/damages of actuators, sensors, wings, and engine. The large blocks at the bottom of the diagram contain computational variables that are needed to transfer visual and aural cues to the simulator. These variables are needed to drive the motion of the simulator platform as well.

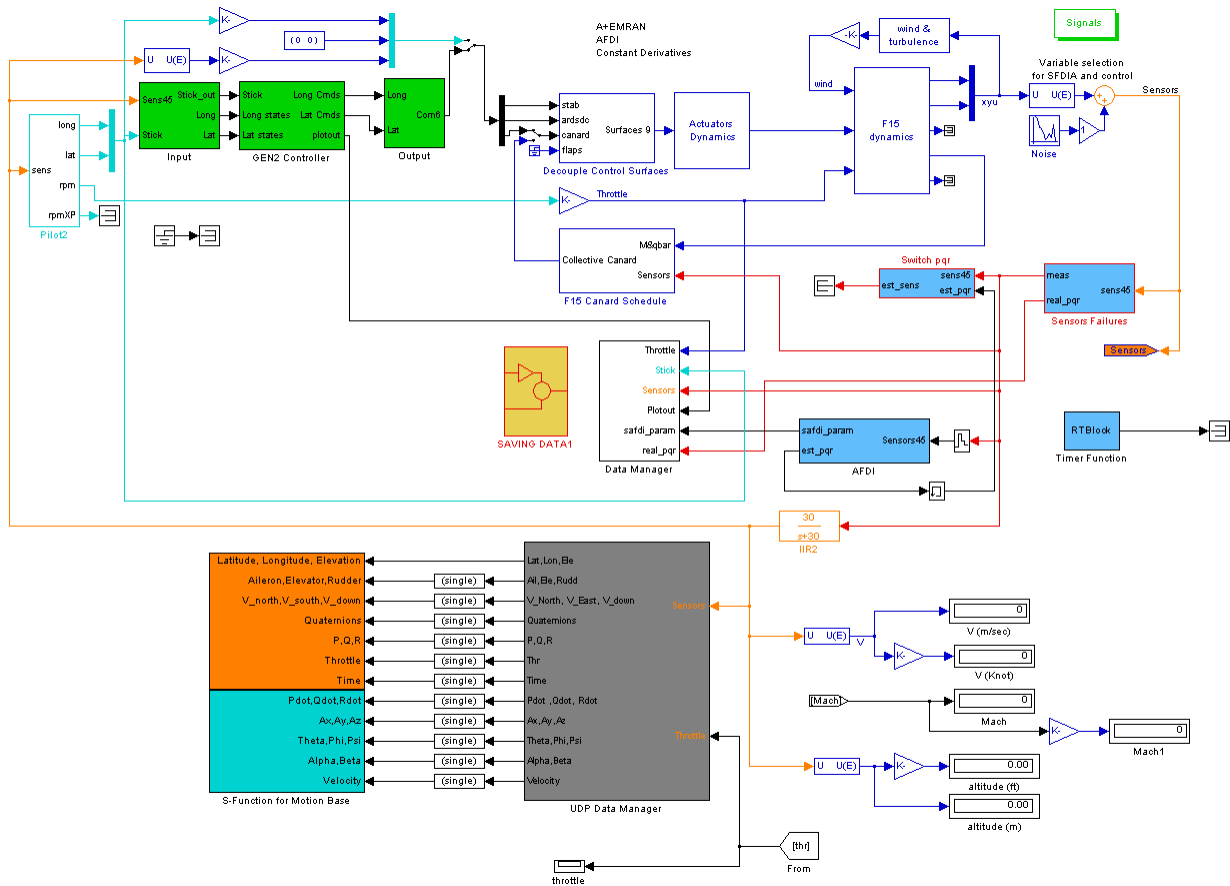


Figure 6.5: WVU Simulation Environment Simulink Top Level Diagram

6.1.3 Aircraft Model

The aircraft aerodynamic model was derived from a non-linear model of a high performance military supersonic fighter [93]. This generic aircraft model was customized through some additional modeling of canard surfaces [88]. The aerodynamic and thrust characteristics are provided through 42 look-up tables, that is 16 tables for the longitudinal dynamics as functions of Mach number, angle of attack and stabilator deflection; 20 tables for the lateral-directional dynamics as functions of Mach number, angle of attack, sideslip angle and rudder; 2 tables for engine thrust and fuel flow as functions of Mach number and altitude. The visualization interface of the described aircraft provided by FlightGear is shown in Fig. 6.6:



Figure 6.6: FlightGear Visualization of the Supersonic Fighter

Additional look-up tables have been added for the modeling of the canards. The aircraft also includes model following adaptive control laws based on non-linear dynamic inversion and artificial neural networks.

6.1.4 Aircraft Subsystems Failure Modeling

To support the development and testing of the AIS-based failure accommodation process the flights under nominal and failure conditions were considered. Flight under nominal conditions implies that during the entire flight time none of the aircraft subsystems experienced any malfunctions/damages. Two types of failures were investigated within this research effort: actuator and sensor failures.

6.1.4.1 Actuator Failure

Failures of individual left actuators (aileron, stabilator, or rudder) were considered. The failure implies that the aerodynamic control surface is stuck. At the failure occurrence, the actuator gets stuck at the current position or moves to a pre-defined position and stays fixed there. The blockage of the control surface at a fixed deflection does not alter the aerodynamic characteristics of the surface. However, it might change the balance between the moments and forces involved in the process.

6.1.4.2 Sensor Failure

Failure of the gyro on the lateral channel was considered within this research effort. The simulation of the sensor failure implies a false sensor output. The transition to the biased sensor output is executed instantaneously (step bias). Note that gyro outputs on all three channels are used by the automatic feedback control augmentation system.

6.2 Simulation Scenarios

In order to validate the proposed algorithm for solving the AIS-based failure accommodation problem one has to design specific flight scenarios. Since this research effort is a preliminary step to a more comprehensive problem and being held within certain restraints, the flight scenarios should not be extremely complicated. However, the flight envelope should be covered over a wide area to capture the dynamic fingerprint of the particular scenario (nominal

or failure conditions). The flight scenarios, lasting between 2 and 4 minutes, were designed to include steady state flight conditions and moderate maneuvers. These flight scenarios were first simulated under nominal flight conditions. They were repeated under various failure conditions for both design/development and validation purposes. Only one failure at a time is injected.

Two simplified maneuvers were considered, one in the vertical plane of symmetry and one on the lateral-directional channel with longitudinal coupling. The sampling rate of the simulation is 50Hz .

6.2.1 Symmetric Climb/Descent

The following are detailed steps of the symmetric climb/descent maneuver (Fig. 6.7):

- steady state symmetric flight at 6050 m and Mach 0.75 for 30 seconds/1 minute
- uniform climb to 6900 m (or descent to 5400 m) at Mach 0.75
- steady state symmetric horizontal flight for 1 minute/30 seconds at the destination altitude (6900 m or 5400 m) and Mach 0.75

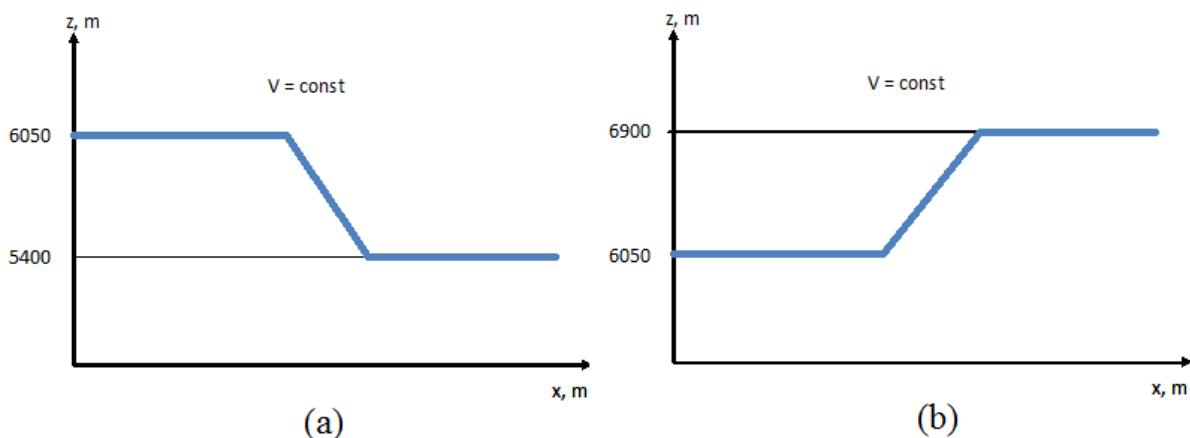


Figure 6.7: a) Symmetric Uniform Descent; b) Symmetric Uniform Climb

6.2.2 Coordinated Half Turn

The following are detailed steps of the coordinated half turn maneuver (Fig. 6.8):

- steady state symmetric flight at 6050 m and Mach 0.75 for 30 seconds
- coordinated half turn at constant bank angle while maintaining altitude and velocity

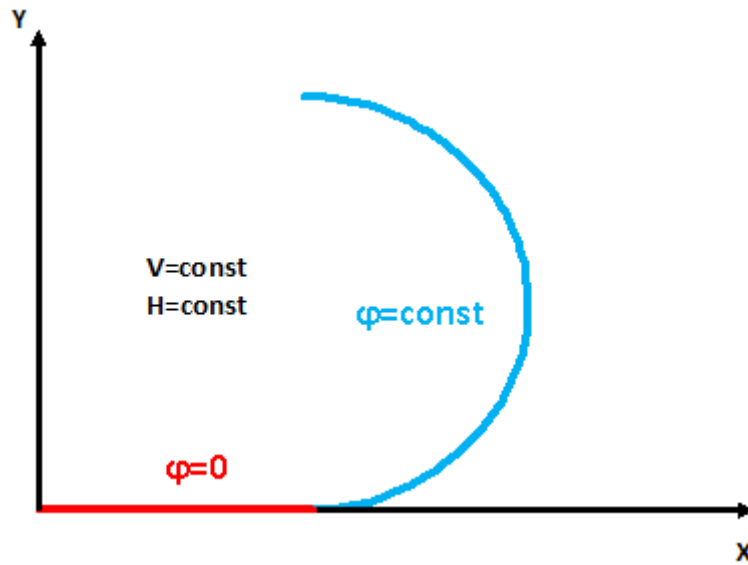


Figure 6.8: Coordinated Half Turn

Chapter 7 Implementation of Failure Accommodation System

7.1 Data Acquisition for AIS Development

The pilot performs the flight tests according to the simulation scenarios that are described in Chapter 6. The tests are performed under nominal and failure conditions. After all the data from sensors, stick displacements, and throttle are collected, one can proceed to the design and implementation of the accommodation algorithm.

7.1.1 Symmetric Climb/Descent

For the climbing flight scenario the following set of flight features FF was selected: aircraft velocity V , roll rate p , pitch rate q , yaw rate r , altitude H , and pitch angle θ .

The total number of selected features is $N_f = 6$. Therefore the selected set of flight features FF is:

$$FF = \{V, p, q, r, H, \theta\} \quad (7.1)$$

The pilot generated commands are the following: longitudinal channel stick displacement de , lateral channel stick displacement da , directional channel pedals displacement dr , and throttle displacement dt .

The total number of pilot control commands is $N_{cc}=4$. The set of pilot control commands PC can be expressed as:

$$PC = \{de, da, dr, dt\} \quad (7.2)$$

The antigen vector AG_v will consist of elements that represent the differences between the desired and actual values of corresponding feature. However, antigen vector has to be extended by including additional elements that represent the actual values of certain features. The additional features for the extension of antigen vector are: bank angle ϕ , pitch angle θ , yaw angle ψ , pitch rate q , altitude H .

Finally, the antigen vector AG_v can be represented as follows:

$$AG_v = \{\Delta V, \Delta p, \Delta q, \Delta r, \Delta H, \Delta \theta, \phi, \theta, \psi, q, H\} \quad (7.3)$$

The flight duration time is $T=130s$, consequently the overall number of time samples is $ts=6500$. The number of time samples used prior to the current moment in the matching algorithm is selected to be $N_t=4$. After analyzing the data from pilot performance, the minimum and maximum values of corresponding elements of the AG matrix have been identified. These values are used for the normalization process and are presented in Table 7.1:

Table 7.1: Symmetric Climb - Maximum and Minimum Values for Normalization

Flight feature	$\Delta V, m/s$	$\Delta p, rad/s$	$\Delta q, rad/s$	$\Delta r, rad/s$	$\Delta H, m$	$\Delta \theta, rad$	ϕ, rad	ψ, rad
Maximum	50	0.35	0.15	0.07	500	0.2	0.07	0.07
Minimum	-50	-0.35	-0.15	-0.07	-500	-0.2	-0.07	-0.07

7.1.2 Coordinated Half Turn

For the coordinated half turn maneuver the following set of flight features FF was selected: aircraft velocity V , roll rate p , pitch rate q , yaw rate r , altitude H , and bank angle ϕ .

The total number of selected features is $N_f = 6$. The selected set of flight features FF can be expressed as:

$$FF = \{V, p, q, r, H, \phi\} \quad (7.4)$$

The pilot generated commands are the same as in the climb maneuver. The total number of pilot control commands is $N_{cc} = 4$. Therefore, the set of pilot control commands PC can be presented as follows:

$$PC = \{de, da, dr, dt\} \quad (7.5)$$

The additional features for the extension of antigen vector are: bank angle ϕ , pitch angle θ , yaw angle ψ , roll rate p .

Finally, the antigen vector AG_v can be represented as:

$$AG_v = \{\Delta V, \Delta p, \Delta q, \Delta r, \Delta H, \Delta \phi, \theta, \psi, p\} \quad (7.6)$$

The flight duration time is $T=215s$, consequently the overall number of time samples is $ts=10750$. The number of time samples used prior to the current moment in the matching algorithm is selected to be $N_t=4$. After analyzing the data from pilot performance, the minimum and maximum values of corresponding elements of the AG matrix have been identified. These values are used for the normalization process and presented in Table 7.2:

Table 7.2: Coordinated Turn - Maximum and Minimum Values for Normalization

Flight feature	$\Delta V, m/s$	$\Delta p, rad/s$	$\Delta q, rad/s$	$\Delta r, rad/s$	$\Delta H, m$	$\Delta \phi, rad$	θ, rad	ψ, rad
Maximum	50	0.35	0.15	0.07	500	0.7	0.1	4
Minimum	-50	-0.35	-0.15	-0.07	-500	-0.7	-0.1	-4

7.2 Accommodation Simulink Model

The purpose of this section is to briefly describe the implementation of the developed system using Matlab/Simulink environment.

A special Simulink block was built for implementing the failure accommodation system (Fig.7.1) and integrated within the WVU simulation environment. This block consists of two main components: the commands generating block and the algorithm outcomes. The inputs to the generating block are the signals from aircraft sensors. The outcomes of the algorithm block are the control commands on longitudinal, lateral and directional channels, and a thrust command. These commands are sent back to the control system and dictate the behavior of the aircraft.

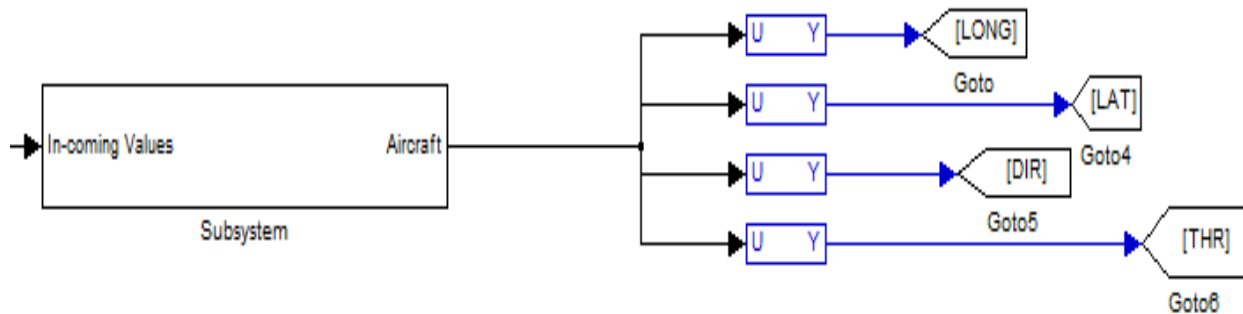


Figure 7.1: Accommodation System Components

Figure 7.2 presents the aircraft control system block of the model described in previous chapter with implemented failure accommodation system block in the bottom.

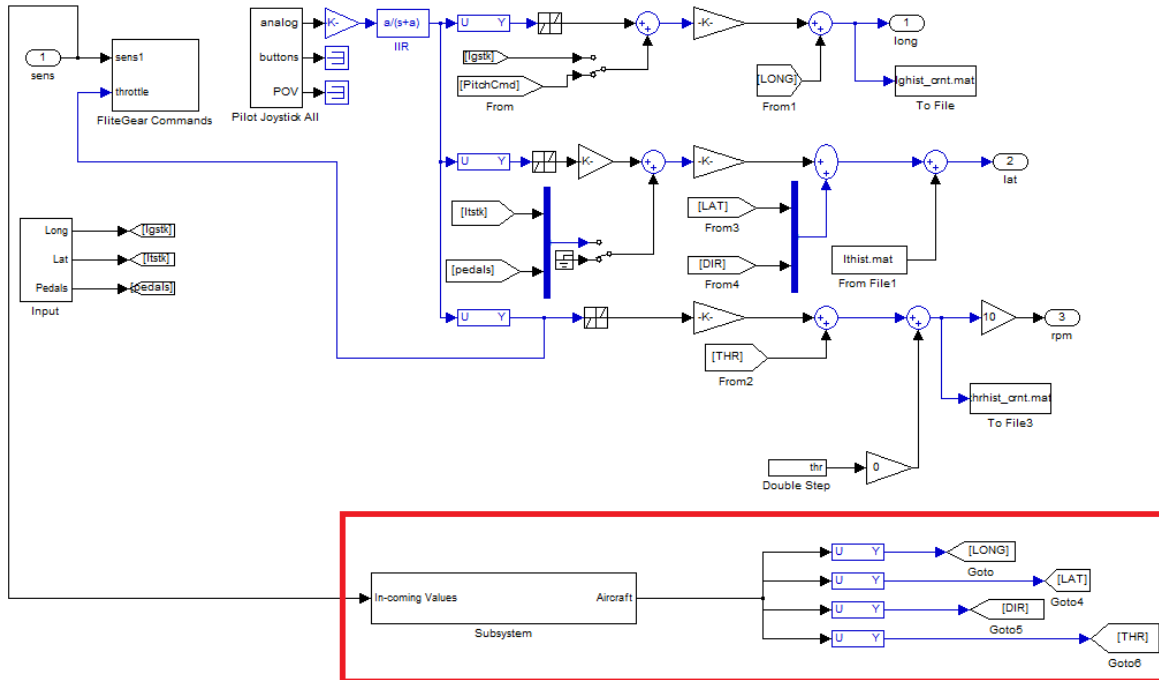


Figure 7.2: Aircraft Control System Block

The detailed structure of the commands generating block is illustrated in Fig. 7.3. It consists of four main components: reference input block, antigens, AIS memory block, and control laws generation block. The algorithm functions as follows:

- 1) The signals coming from the aircraft sensors are compared to the ones in the reference input block. The outcome signals form the Antigens block. The process illustrates a healthy organism being attacked by invaders.
- 2) T-cells and B-cells block and Antibodies block altogether constitute the AIS memory.
- 3) All the information from AIS memory block proceeds to control laws generation block, where the matching algorithm is performed (a MATLAB code which executes the matching algorithm is presented in thesis Appendix).
- 4) Outcome of the matching algorithm is a set of generated control commands that are sent to the control system to command the motion of the aircraft.

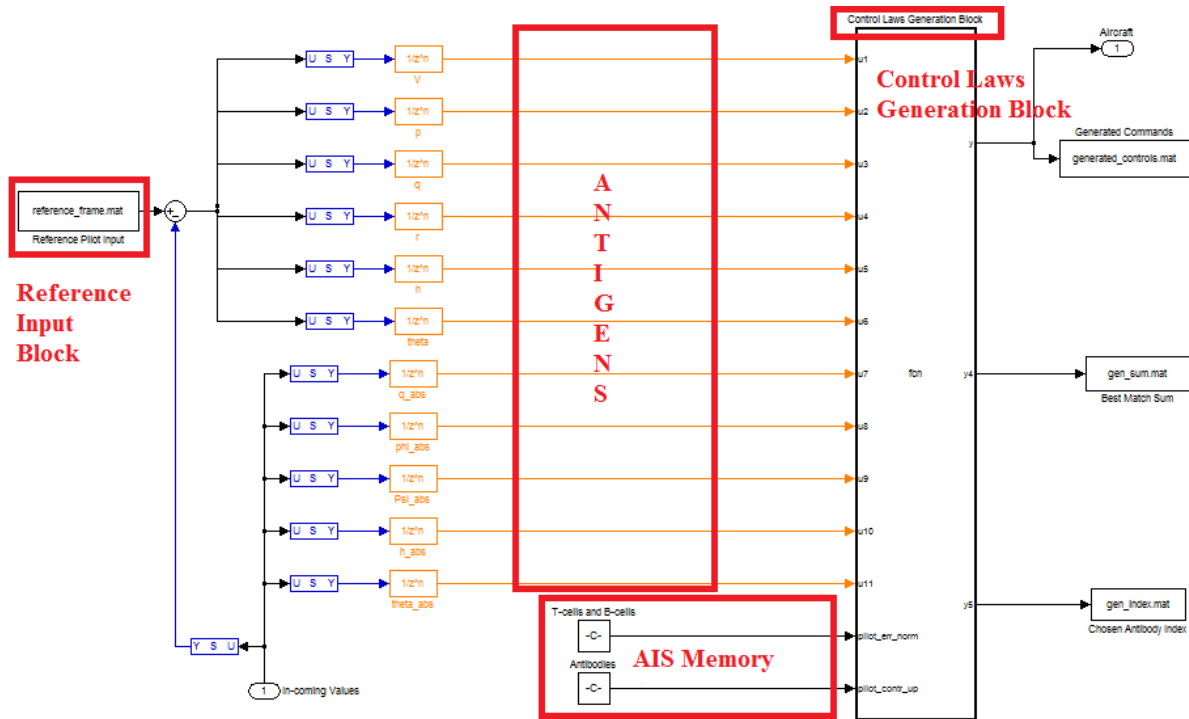


Figure 7.3: Detailed Structure of the Control Commands Generating Block

The delay chain blocks allow accumulating values of in-coming feature over several time samples. This would represent the matching algorithm with floating window described in Chapter 5.

Note, that AIS memory block, which consists of T-cells, B-cells, and antibodies, is formed after all the data from the pilot performance were acquired. The artificial memory cells are built based on data from the flight under nominal conditions, aileron, stabilator, and rudder failures flight, and roll rate sensor failure flight.

The minimum total sum *ERR* together with its index in AIS memory are saved at each time sample for post-processing and analysis purposes.

Chapter 8 Results and Performance Assessment

After all the necessary data were acquired, the AIS-based failure accommodation algorithm was designed, developed, and implemented within the WVU simulation environment. This chapter will illustrate the results of implementation of the proposed methodology to extract the control commands from the pilot performance under different flight conditions. All the figures that will be presented in this chapter will illustrate the time histories of the aircraft features and the commands. Every "feature" figure will contain three curves:

- reference - the desirable value of the feature over the entire flight time that was prescribed during the experimental design stage
- pilot - the value of the feature that was obtained during flight simulation when the aircraft was controlled by a pilot
- generated - the value of the feature that was obtained during the flight simulation when the aircraft was controlled by the commands generated from the implemented failure accommodation system

Every "command" figure will contain two curves:

- pilot - value of stick displacement (which defines the control command) that was produced by the pilot during the flight simulation
- generated - value of stick displacement that was obtained as an outcome of the failure accommodation system

For each type of maneuver and each type of failure, a certain set of results (figures) will be illustrated. Not all of the features involved into the entire accommodation mechanism will be

presented, because they do not affect the behavior of the aircraft or do not contribute significantly to the dynamic fingerprint of a particular flight condition.

8.1 Symmetric Climb

The features that have the most significant dynamic impact during the symmetric climb maneuver are: pitch rate q , pitch angle θ , altitude H , and velocity V .

8.1.1 Nominal Conditions

Figure 8.1 presents the variation of aircraft altitude during the symmetric flight under nominal conditions:

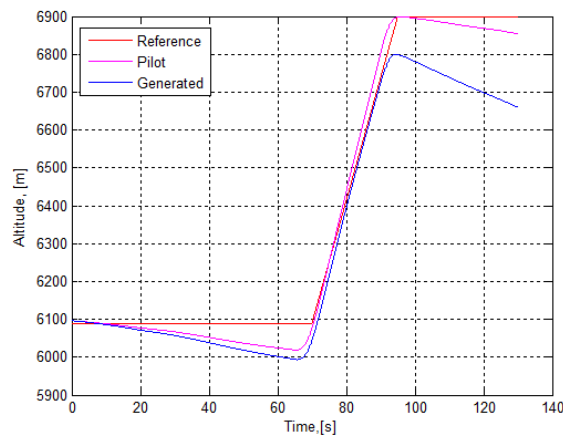


Figure 8.1: Aircraft Altitude - Nominal Conditions Symmetric Climb

The generated altitude was very accurately reproduced and is very close to the altitude of pilot performance. There is a little deviation at the end of the flight; however, the trend - drifting down - is preserved.

Figure 8.2 presents the velocity of the aircraft. A noticeable drop of velocity value can be observed from seconds 70 to 90. This happens due to the fact that the aircraft climbs during this period of time. The increased drop in velocity during automatic flight is due to the inaccuracy of throttle command extraction during the final interval of the maneuver. Also, the

discontinuity of the extracted throttle command curve might have affected such a difference in velocity. However, it should be noticed that the maximum velocity tracking error is approximately 7% of commanded velocity.

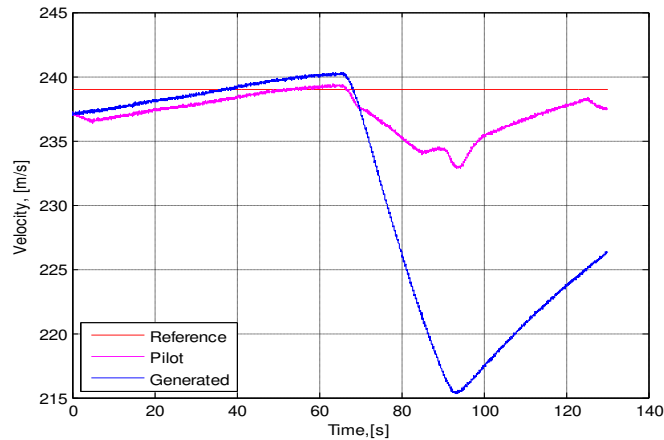


Figure 8.2: Aircraft Velocity - Nominal Conditions Symmetric Climb

Figure 8.3 presents the longitudinal channel control commands. The outcomes of the failure accommodation system almost coincide with the pilot performance. Small insignificant inaccuracies might be noticed, but they can appear due to randomness of sensor noise and turbulence.

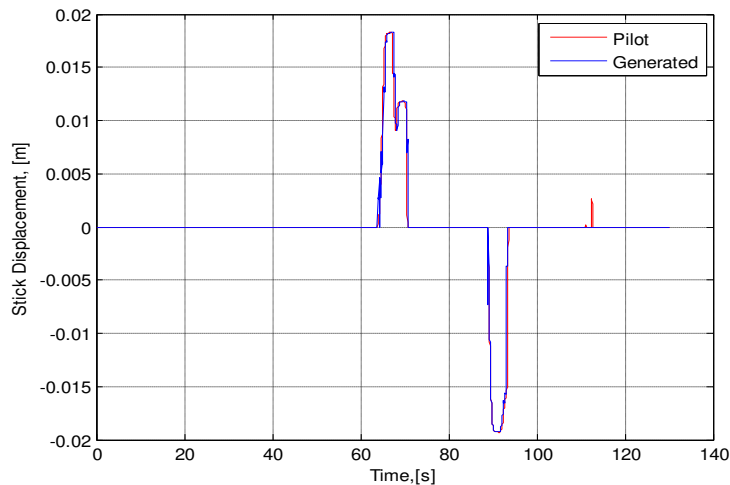


Figure 8.3: Longitudinal Channel Commands - Nominal Conditions Symmetric Climb

Figure 8.4 presents the throttle displacement. For most of the time, the extraction of the thrust command from the AIS is adequate. However, there are some inaccuracies at the beginning and at the end of the flight. Due to these inaccuracies the deviation of the generated aircraft velocity from the desired value is slightly more significant than the one obtained under pilot's control.

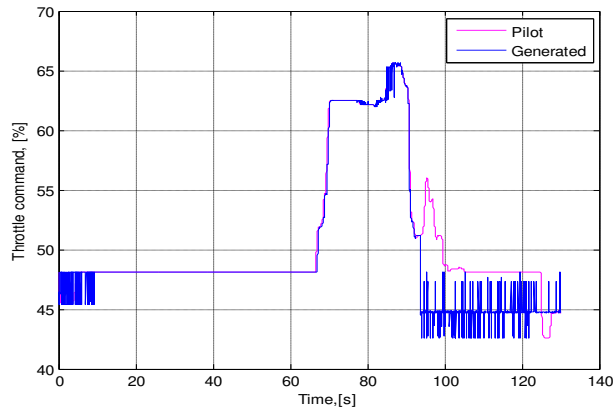


Figure 8.4: Throttle Displacement - Nominal Conditions Symmetric Climb

Figure 8.5 presents the pitch rate of the aircraft. Pitch rate variation is mostly dictated by the longitudinal command. Therefore, since the generated longitudinal commands almost coincide with the pilot outcome, the pitch rate behaves similarly.

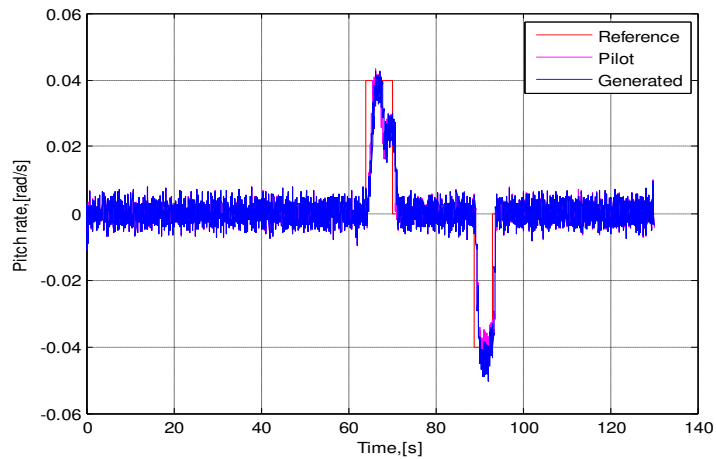


Figure 8.5: Aircraft Pitch Rate - Nominal Conditions Symmetric Climb

Figure 8.6 presents the pitch angle of the aircraft. Notice that the trim pitch angle is a slightly above zero. The pilot executed the climb with a constant θ value, and the same trend can be observed for the generated flight simulation.

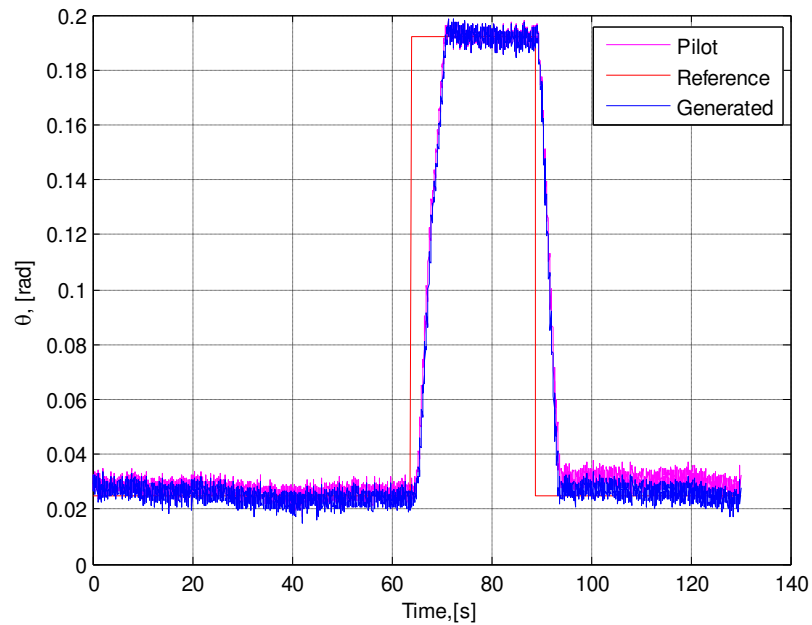


Figure 8.6: Aircraft Pitch Angle - Nominal Conditions Symmetric Climb

Based on these results, it can be concluded that the command extraction under nominal conditions for the symmetric climb maneuver was successful.

8.1.2 Actuator Failure

Three types of the actuators were subjected to a failure (blockage at the deflection of +4 degrees): left aileron, left stabilator, and left rudder. Failure injection time in all of the tests is

$$T_f=15s$$

8.1.2.1 Aileron Failure

Figure 8.7 presents the variation of the flight altitude. Similarly to the nominal conditions flight, the generated altitude slightly deviates from pilot performance at the end of the flight by approximately 1.5%.

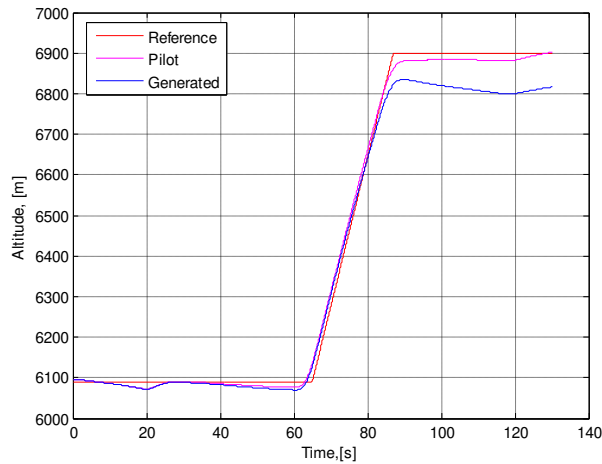


Figure 8.7: Aircraft Altitude - Aileron Failure Symmetric Climb

Figure 8.8 presents the aircraft velocity. The velocity drops down during the climbing phase of the flight. The increased drop in velocity during automatic flight is due to the inaccuracy of throttle command extraction during the final interval of the maneuver. However, notice that the similarity is higher as compared to the flight under nominal conditions.

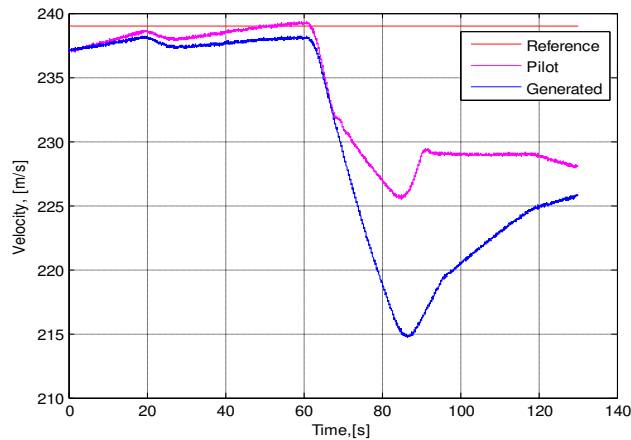


Figure 8.8: Aircraft Velocity - Aileron Failure Symmetric Climb

Figure 8.9 presents the longitudinal channel commands. Although the failure mostly affects the roll channel, it had some influence on the pitch channel as well. The pilot had to provide some compensation after the failure to maintain the steady state flight. The generated commands duplicate pilot input extremely well.

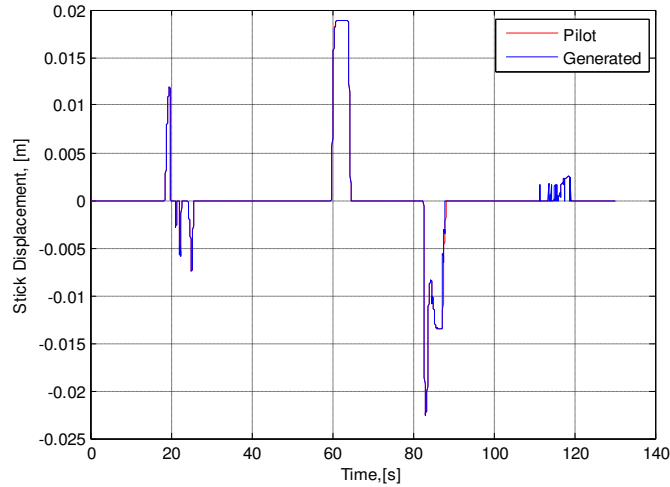


Figure 8.9: Longitudinal Channel Commands - Aileron Failure Symmetric Climb

Figure 8.10 illustrates aircraft pitch angle. A little bump can be noticed right after the failure injection. The pilot executed the climb maneuver at a constant θ value.

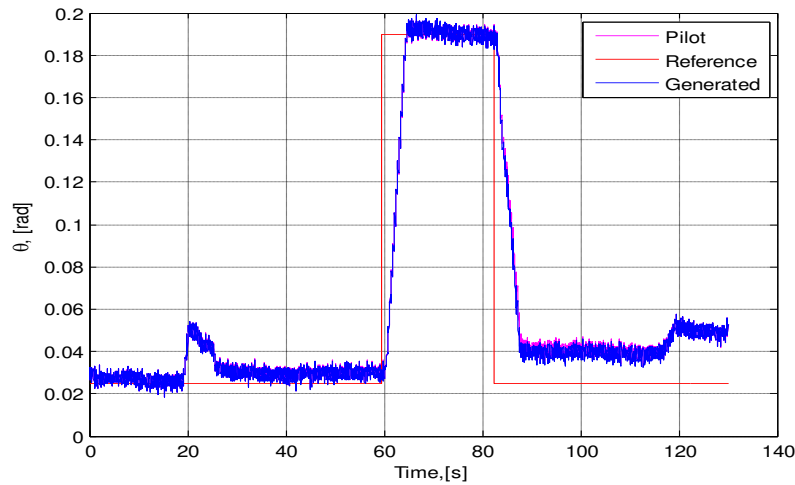


Figure 8.10: Aircraft Pitch Angle - Aileron Failure Symmetric Climb

Figure 8.11 presents the bank angle ϕ . A significant change of bank angle after the aileron failure can be observed. However, the value was brought back due to pilot's input on the lateral channel.

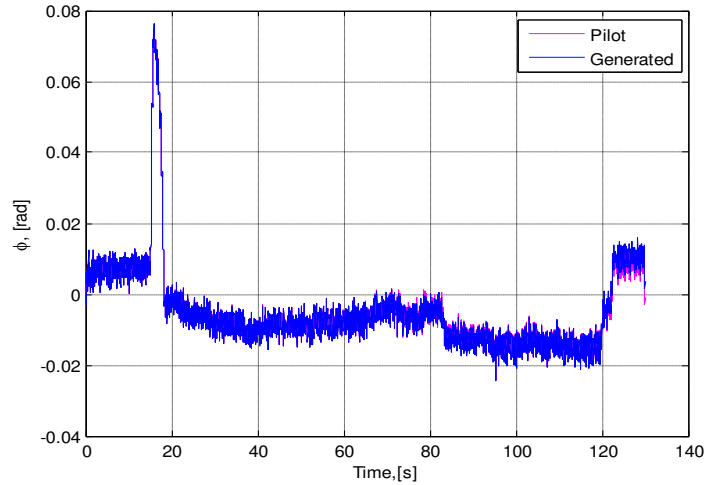


Figure 8.11: Aircraft Bank Angle - Aileron Failure Symmetric Climb

Figure 8.12 presents the lateral commands. Pilot provided significant input on the lateral channel after the failure was injected to maintain prescribed steady state flight. The generated commands repeated this action accurately.

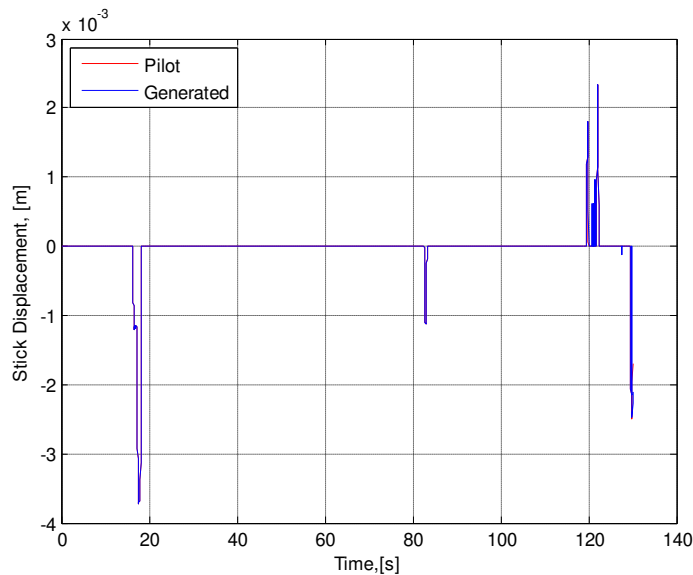


Figure 8.12: Lateral Channel Commands - Aileron Failure Symmetric Climb

Figures 8.13 and 8.14 present aircraft pitch and roll rates, respectively. Behavior of these features is mostly governed by longitudinal and lateral commands; therefore, the results are quite precise too. A significant peak of the roll rate can be noticed right after the failure occurrence.

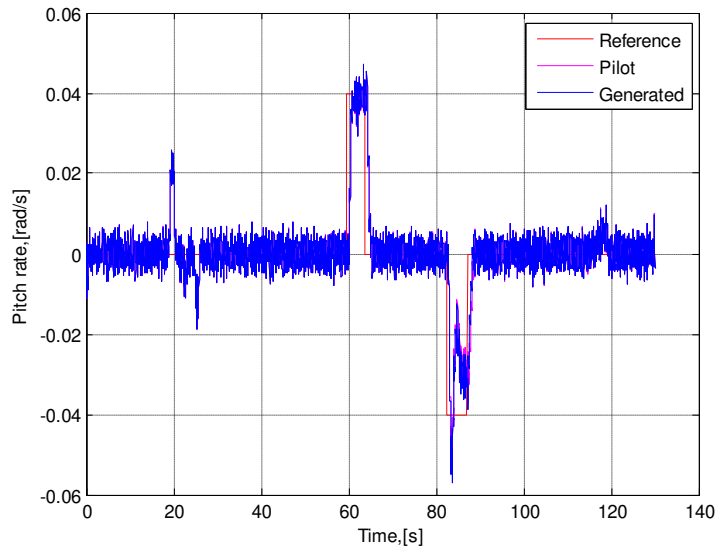


Figure 8.13: Aircraft Pitch Rate - Aileron Failure Symmetric Climb

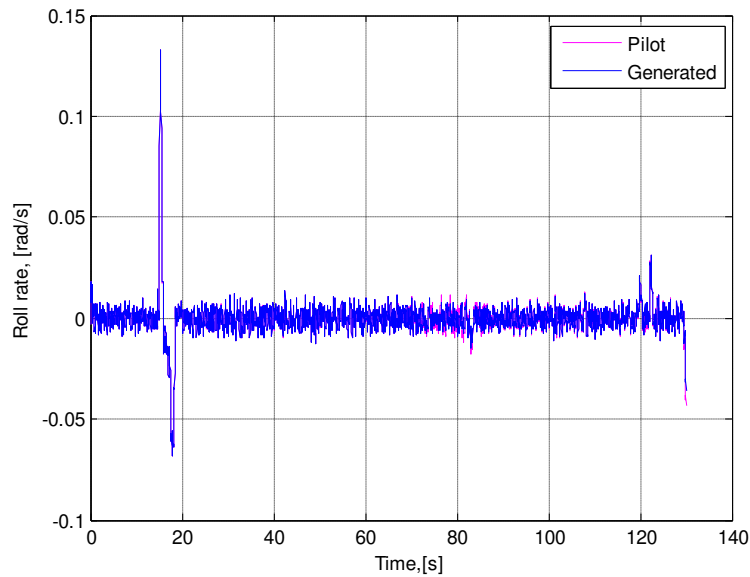


Figure 8.14: Aircraft Roll Rate - Aileron Failure Symmetric Climb

Figure 8.15 illustrates the variation of the Y coordinate. After the failure occurrence, the aircraft intends to deviate from the straight flight to the right. At the time, pilot intends to maintain a straight forward trajectory. As a result, we can observe that the aircraft returned back to the initial Y coordinate position (close to zero).

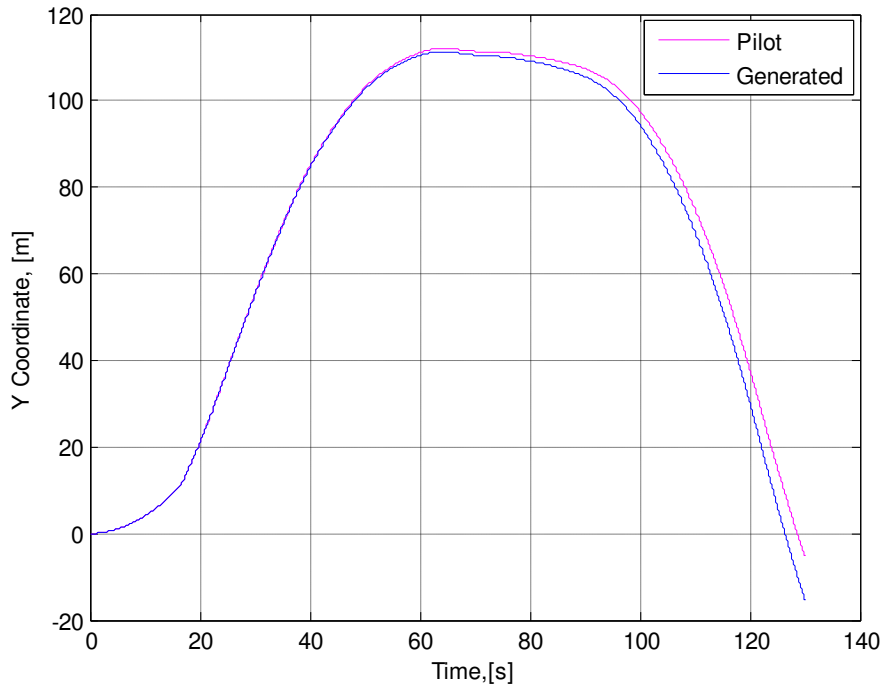


Figure 8.15: Y Coordinate - Aileron Failure Symmetric Climb

8.1.2.2 Stabilator Failure

Figure 8.16 and 8.17 present the aircraft altitude and velocity, respectively. A small decrease of altitude can be noticed at around $T=20s$ due to the stabilator failure. Aircraft velocity variation during automatic flight is very similar to the one during piloted flight and does not deviate significantly as in previous tests.

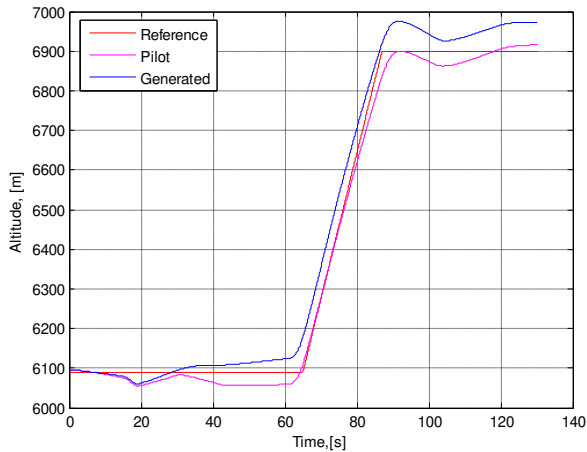


Figure 8.16: Aircraft Altitude - Stabilator Failure Symmetric Climb

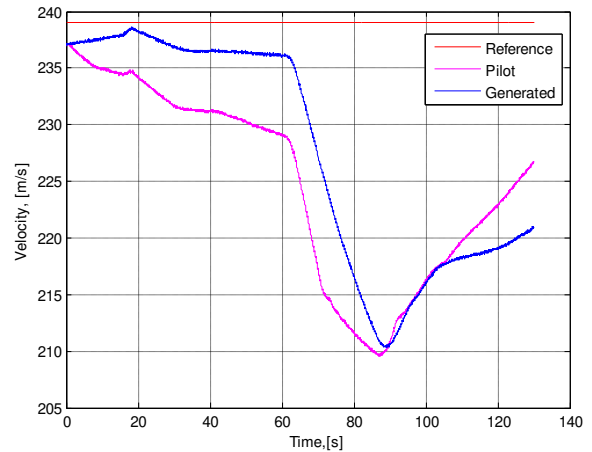


Figure 8.17: Aircraft Velocity - Stabilator Failure Symmetric Climb

Figure 8.18 illustrates variation of the aircraft pitch angle θ . A significant change of pitch angle is observed due to stabilator failure.

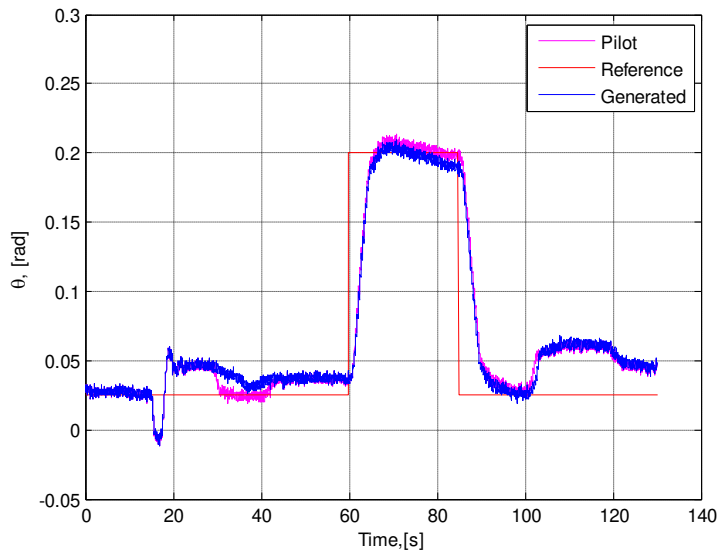


Figure 8.18: Aircraft Pitch Angle - Stabilator Failure Symmetric Climb

Figure 8.19 illustrates the longitudinal channel commands. Notice that the pilot workload on the longitudinal channel has significantly increased, in comparison with previously illustrated results. Small insignificant inaccuracies can be observed.

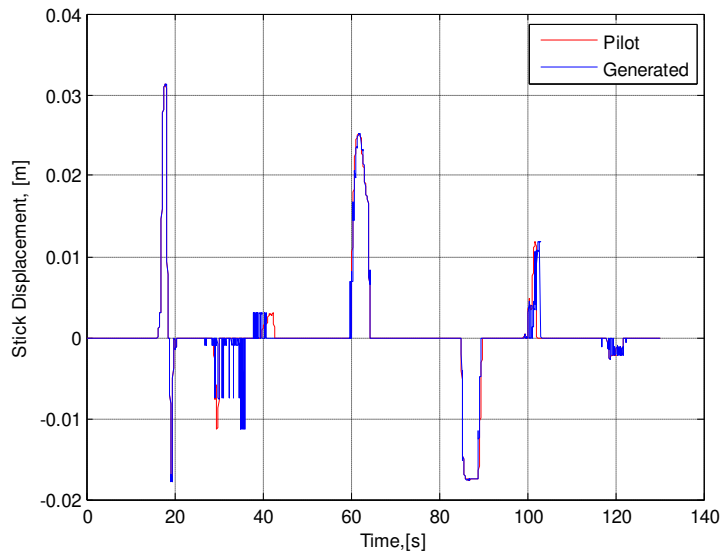


Figure 8.19: Longitudinal Channel Commands - Stabilator Failure Symmetric Climb

Figure 8.20 illustrates the lateral channel commands. Although the failure was mostly on the longitudinal channel, pilot workload on the lateral channel is larger. The generated commands almost coincide with the pilot input.

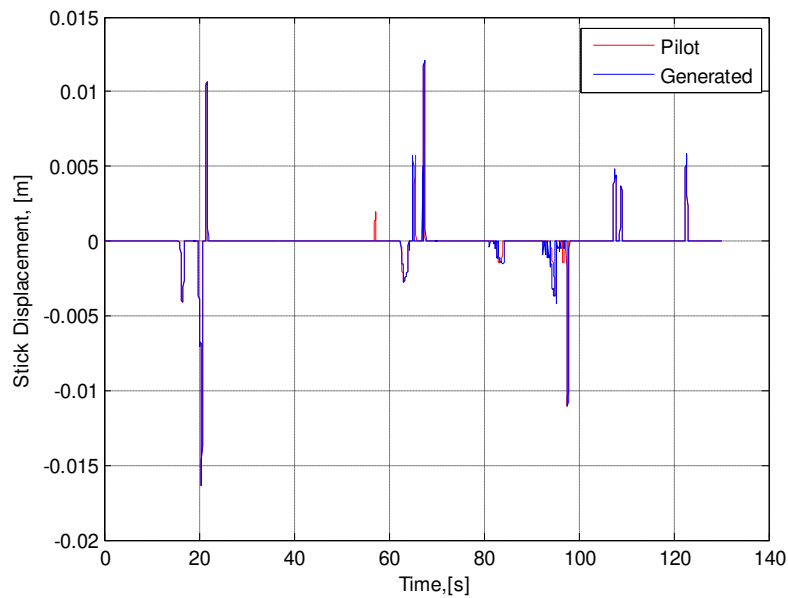


Figure 8.20: Lateral Channel Commands - Stabilator Failure Symmetric Climb

Figure 8.21 illustrates Y coordinate variation. Although the stabilator failure mostly affects the longitudinal channel, a coupling with lateral channel takes place. Therefore, because of the bank angle ϕ alteration, the aircraft intends to deviate from the straight forwards trajectory (see Fig. 8.22).

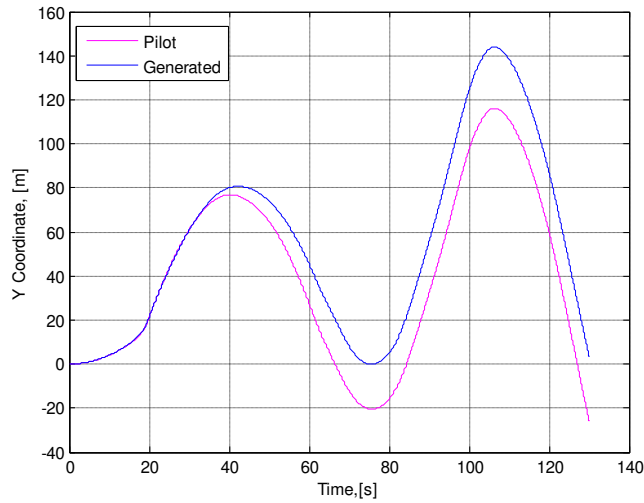


Figure 8.21: Y Coordinate - Stabilator Failure Symmetric Climb

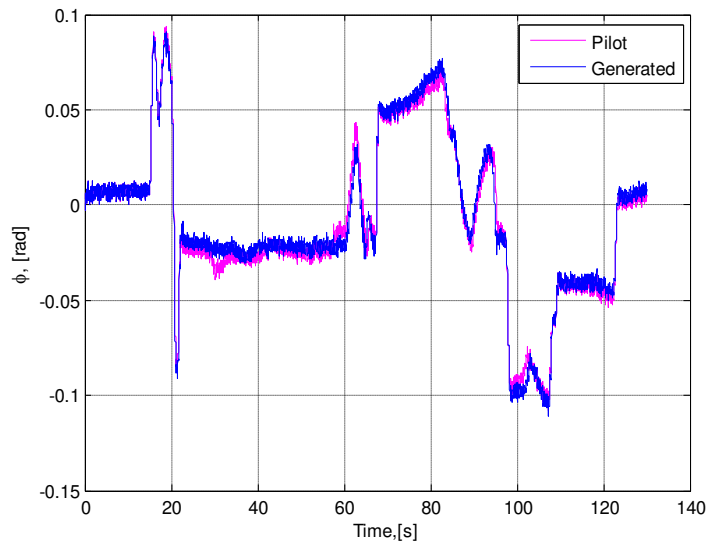


Figure 8.22: Aircraft Bank Angle - Stabilator Failure Symmetric Climb

8.1.2.3 Rudder Failure

Figures 8.23 and 8.24 illustrate the aircraft altitude and velocity respectively. The altitude of the generated flight almost coincides with the pilot performance outcome. Velocity has been reproduced quite accurately as well.

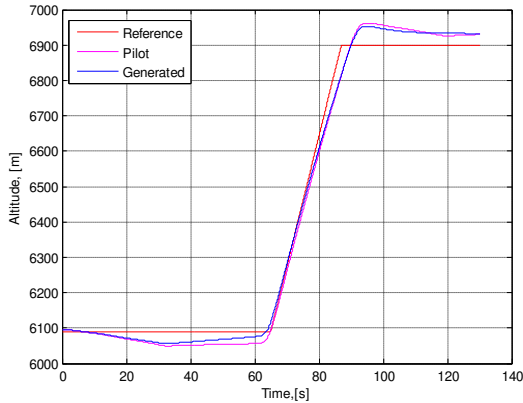


Figure 8.23: Aircraft Altitude - Rudder Failure Symmetric Climb

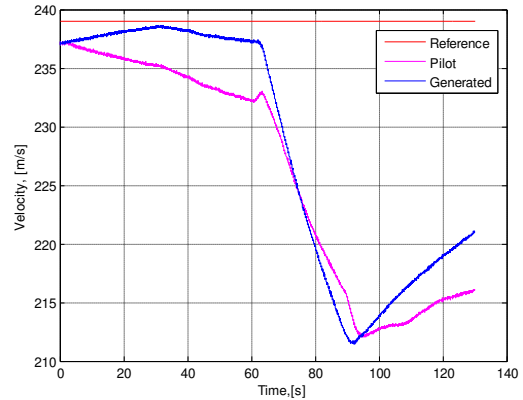


Figure 8.24: Aircraft Velocity - Rudder Failure Symmetric Climb

Figure 8.25 illustrates the variation of the yaw angle ψ . Yaw angle was directly affected by the rudder failure. However, one can notice that the change of the feature value after the failure is not so significant (as compared to pitch angle change after stabilator failure or bank angle change after aileron failure).

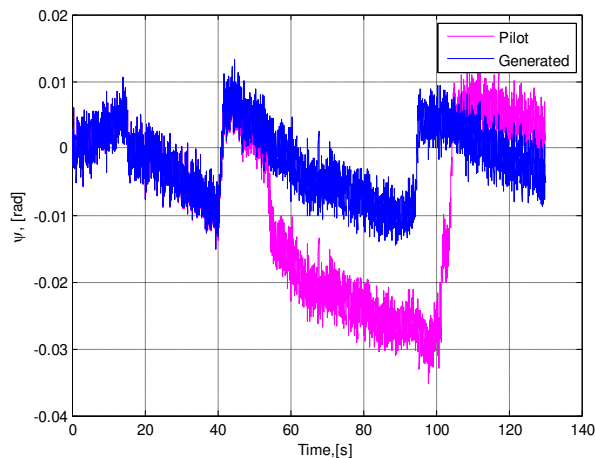


Figure 8.25: Aircraft Yaw Angle - Rudder Failure Symmetric Climb

Figure 8.26 illustrates directional channel commands. Directional channel commands were extracted successfully. However at the end of the simulation, some of the commands were not accurately extracted. This happened because of the following: small inaccuracies in generated directional commands at $t=40-50s$ have led to deviations of the yaw angle. This deviation, afterwards, has influenced the matching algorithm in the ultimate interval of the flight.

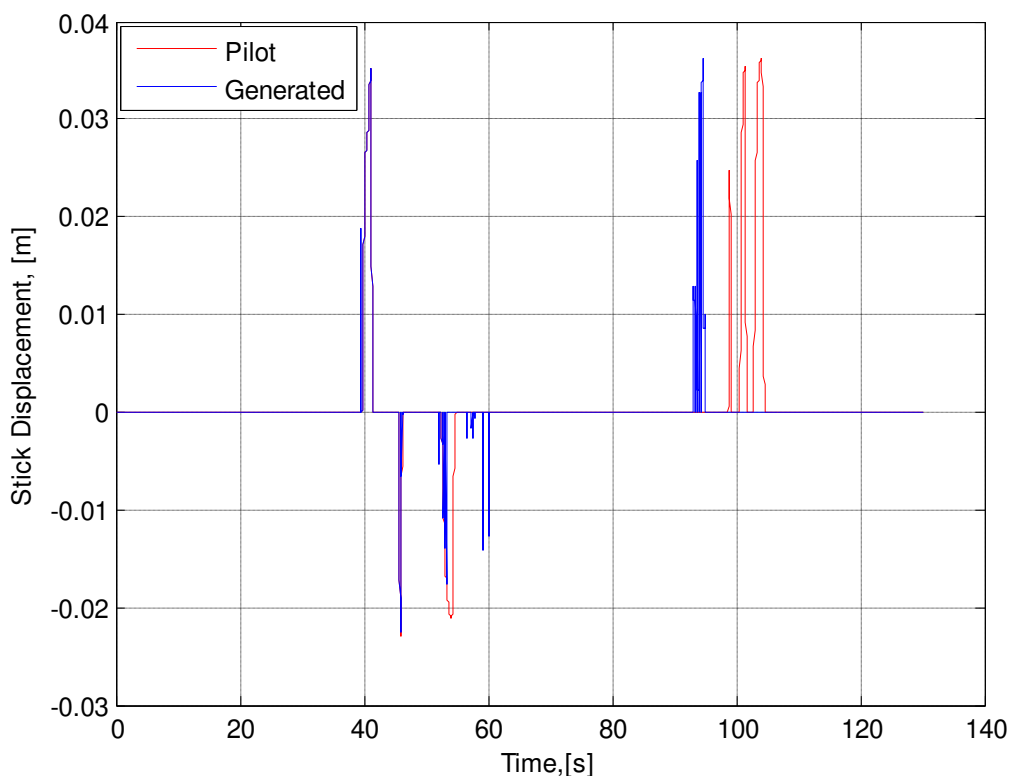


Figure 8.26: Directional Channel Commands - Rudder Failure Symmetric Climb

Figure 8.27 illustrates variation of the pitch angle θ and shows that the climbing maneuver was performed at a constant θ value.

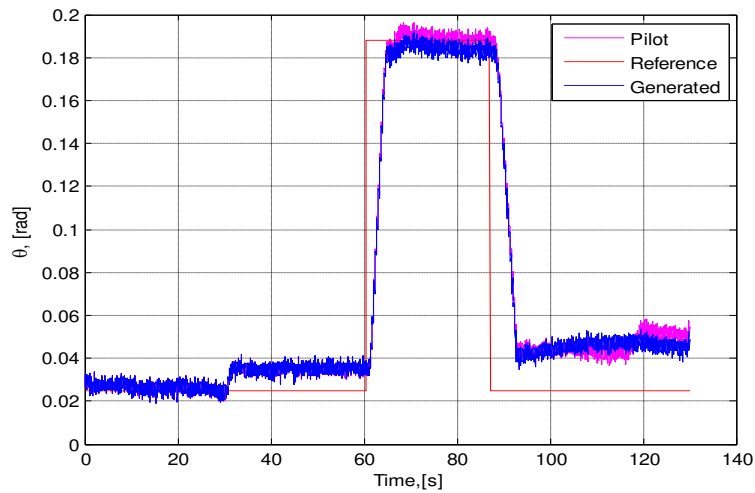


Figure 8.27: Aircraft Pitch Angle - Rudder Failure Symmetric Climb

Longitudinal channel commands have been extracted from the AIS memory successfully, which is proven by almost coinciding curves in Fig. 8.28

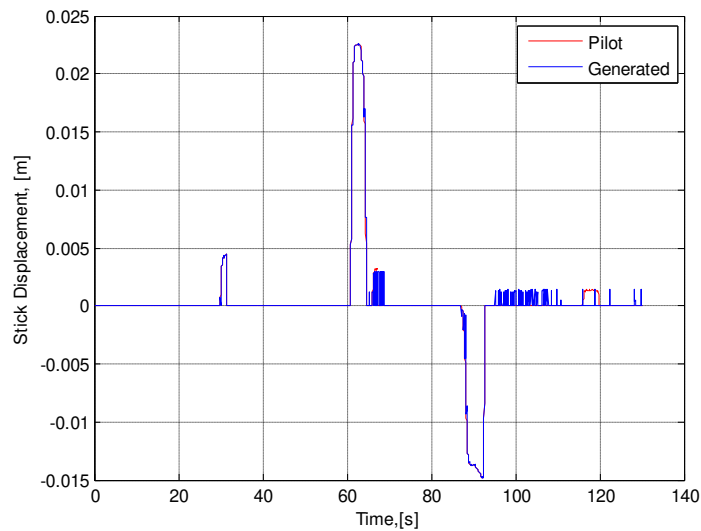


Figure 8.28: Longitudinal Channel Commands - Rudder Failure Symmetric Climb

Figure 8.29 presents the variation of the bank angle φ . A significant change of the value can be noticed after the failure injection. Only one lateral channel impulse input was needed to fix the consequences of the failure (see Fig. 8.30).

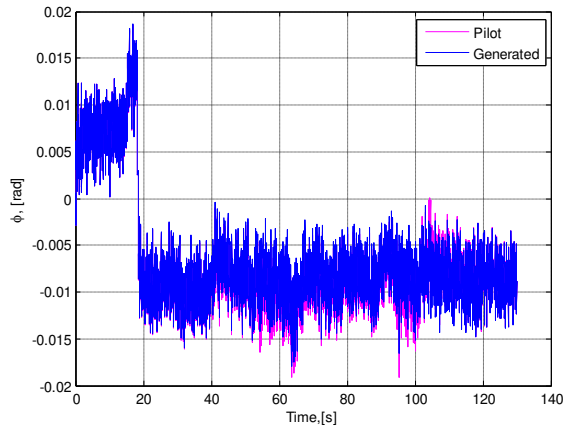


Figure 8.29: Aircraft Bank Angle - Rudder Failure Symmetric Climb

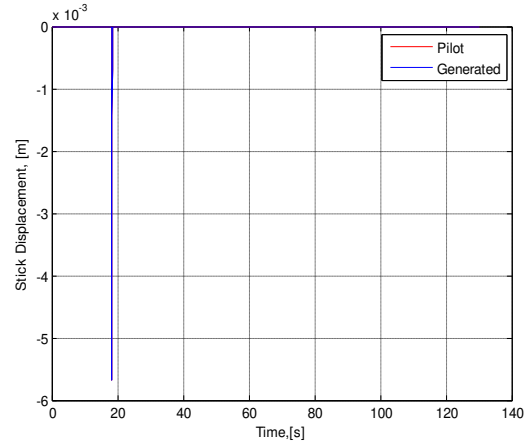


Figure 8.30: Lateral Channel Commands - Rudder Failure Symmetric Climb

Figure 8.31 illustrates the variation of the Y coordinate. Because of some inaccuracies of the generated commands on directional channel, the Y coordinates depart after $T=50s$. However, it should be noted that given the forward velocity of the aircraft and the 2 minute duration of the test, the approximately 200m final offset in lateral position is small.

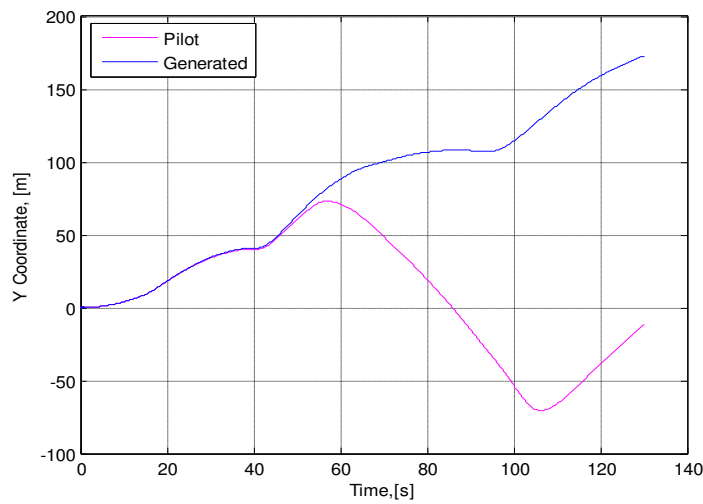


Figure 8.31: Y Coordinate - Rudder Failure Symmetric Climb

8.1.3 Sensor Failure

Roll rate sensor failure was investigated within this research effort. A sensors bias failure was considered, which implies shifting the value of the measured roll rate by $+3 \text{ deg/s}$ from the true value. Failure injection time is $T_f=15s$.

Figure 8.32 illustrates the trajectory of symmetric climb. Trajectory has been accurately simulated and followed the pilot's outcome. A small delay in climbing process can be noticed. Also, the altitude deviates from the desired value in the final stage of the flight.

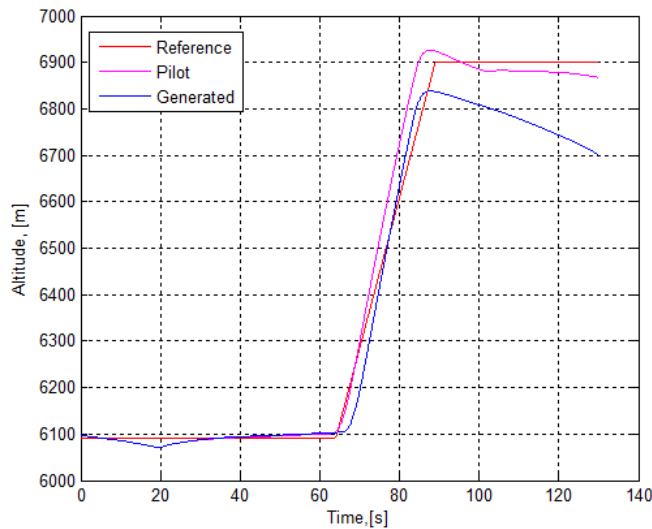


Figure 8.32: Aircraft Altitude - Sensor Failure Symmetric Climb

Figures 8.33 and 8.34 present the variation of aircraft velocity and throttle commands, respectively. Throttle command has been accurately generated until the moment of approximately $T=90s$, where it dropped down, while pilot was still intending to accelerate for about 25-30 seconds. This error in throttle command resulted in a minor velocity difference at the final stage of the flight.

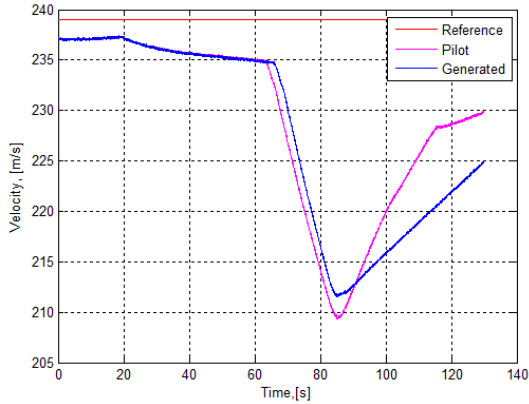


Figure 8.33: Aircraft Velocity - Sensor Failure Symmetric Climb

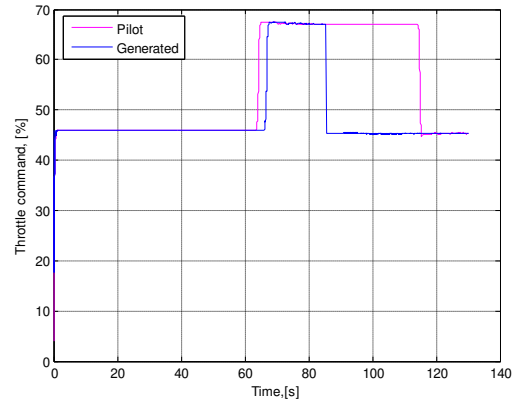


Figure 8.34: Throttle Command - Sensor Failure Symmetric Climb

Longitudinal channel commands have been generated accurately enough during the sensor failure flight and are presented in Fig. 8.35. The generated commands followed the pilot trend, including the beginning of the flight when the failure was injected. Several inaccuracies can be observed during the climb maneuver; however, they did not have an essential effect on the aircraft pitch angle (see Fig.8.36). Looking at the variation of pitch angle and aircraft altitude it can be concluded that longitudinal commands have been extracted successfully.

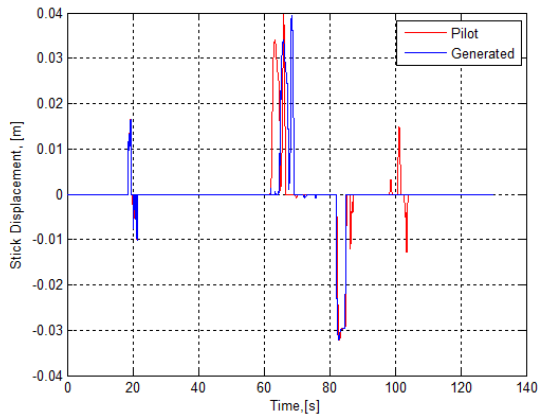


Figure 8.35: Longitudinal Channel Commands - Sensor Failure Symmetric Climb

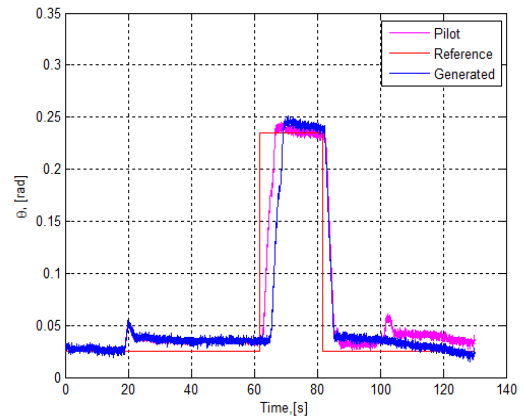


Figure 8.36: Aircraft Pitch Angle - Sensor Failure Symmetric Climb

Lateral channel commands extraction was not as successful as the one for the longitudinal channel – Fig. 8.37. Generally, the plot does not look very neat with significant inaccuracies in the extraction. However, it can be noted that at the initial stage of the flight, the commands that were dedicated to overcome the injected failure have been generated very accurately. The outcome of these inaccuracies can be observed in Fig. 8.38. The aircraft deviates from the straight flight due to incorrect lateral channel commands. The deviation of the aircraft from the straight flight can also be explained by Fig. 8.39, which illustrates the variation of the bank angle during the flight. It can be seen that the bank angle was accurately generated at the beginning (failure injection time) and in the middle of the flight time. However, in between and after these periods, we can observe that the impulse commands generated on the lateral channel were not large enough to bring the bank angle back to zero. The aircraft experienced a slight turn, which resulted in diverging from the desired direction

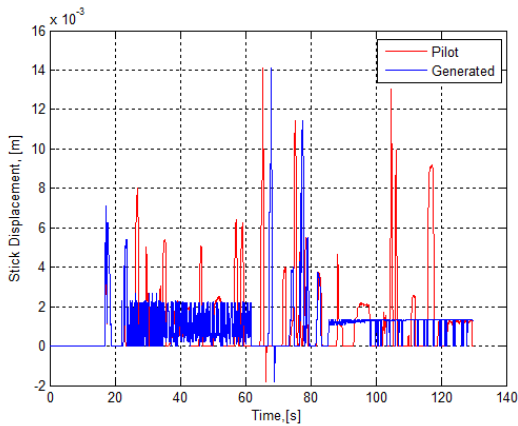


Figure 8.37: Lateral Channel Commands -
Sensor Failure Symmetric Climb

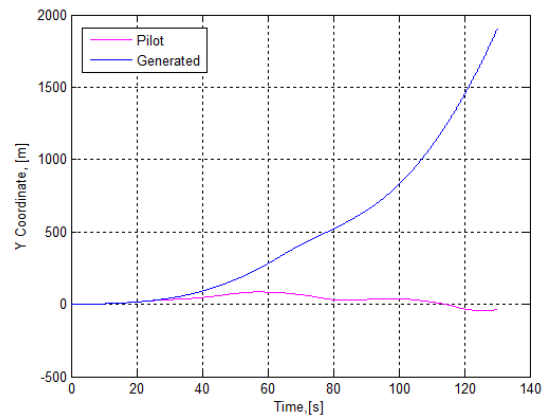


Figure 8.38: Y Coordinate - Sensor Failure
Symmetric Climb

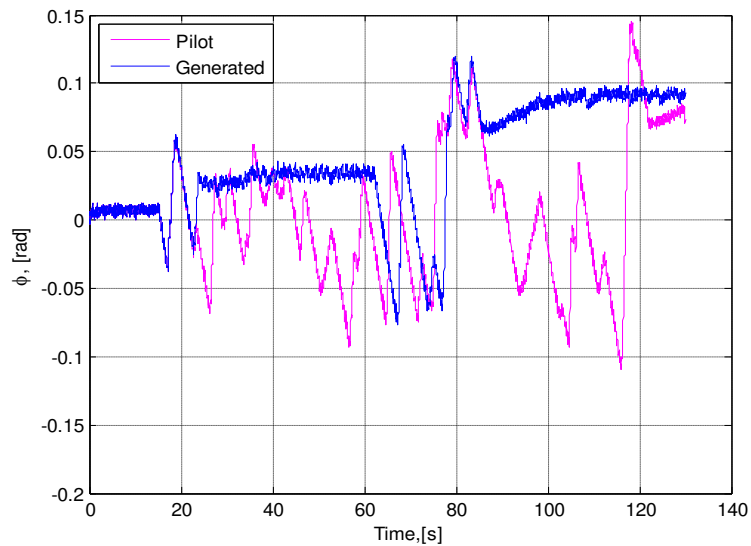


Figure 8.39: Aircraft Bank Angle - Sensor Failure Symmetric Climb

With these results, it can be concluded that the flight mission under sensor failure has been partially accomplished. The roll rate bias has been overcome on the longitudinal channel - the aircraft did not lose the altitude and performed the desired maneuver. However, the compensation on the lateral channel was insufficient, which resulted in a deviation from the straight forward flight. More investigation should be performed in this direction in future research.

8.1.4 Analysis and Evaluation

In addition to graphical results of extracting pilot control commands, some numerical computations have been executed to confirm and validate the outcomes. Since the flight maneuvers have been mostly defined by altitude and velocity, analysis with regard to these flight features has been performed. Three metrics have been used for the evaluation: *Pilot Flight Error* shows how accurately pilot performed the prescribed mission. *Generated Flight Error* shows how accurately the generated simulation performed the prescribed mission. *Relative Error* is an error of the generated simulation with respect to pilot performance.

$$PilotFlightError = \left| \frac{DesV - PilotV}{DesV} \right| * 100\% \quad (8.1)$$

$$GeneratedFlightError = \left| \frac{DesV - GenV}{DesV} \right| * 100\% \quad (8.2)$$

$$RelativeError = \left| \frac{PilotV - GenV}{PilotV} \right| * 100\% \quad (8.3)$$

where *DesV* is desired value of the feature at each moment of the flight, *PilotV* is a value of the feature obtained from pilot performance, *GenV* is a value of the feature obtained from the generated simulation.

Table 8.1: Flight Simulation Performance Assessment - Symmetric Climb

Flight Conditions	Nominal Conditions		Aileron Failure		Stabilator Failure		Rudder Failure		Sensor Failure	
	H	V	H	V	H	V	H	V	H	V
Pilot Flight Error,%	0.46	0.81	0.21	2.28	0.30	5.96	0.51	5.40	0.35	4.18
Generated Flight Error,%	1.18	3.36	0.58	3.75	0.59	4.77	0.41	4.49	0.79	4.64
Relative Error,%	0.90	2.78	0.42	1.51	0.78	1.85	0.13	1.19	0.79	0.90

As can be seen from Table 8.1, none of the altitude errors exceed mark of 1.5%, and none of the velocity errors exceed mark of 6%. Notice, that despite having significant errors for lateral channel commands under the sensor failure flight, the results for altitude and velocity are still very accurate. This is mostly due to the fact that lateral and longitudinal channels are decoupled.

Table 8.2 illustrates the amount of pilot workload on all four channels under different flight conditions. The first figure in the table relates to pilot workload, the second figure presents the integrated workload value of generated commands. The following metrics have been used: W_{long} , W_{lat} , W_{dir} , and W_{thr} - representing the integral of the workload on longitudinal, lateral, directional, and throttle channels over the entire flight time, respectively.

$$W = \int_0^{tf} \frac{|stick|}{tf} dt \quad (8.4)$$

Where W is a workload on a corresponding channel, and $stick$ is a corresponding stick displacement; entire time of flight is tf .

The workload generally increases with the presence of the failure. Interestingly, comprehensive input on lateral channel under stabilator failure flight is larger than under aileron failure flight. Notice that the workload on the lateral channel is excessive under the roll rate sensor failure. Injected failures did not affect throttle channel significantly (except for the throttle channel)

Table 8.2: Pilot Workload - Symmetric Climb

Channel	Nominal Conditions	Aileron Failure	Stabilator Failure	Rudder Failure	Sensor Failure
Longitudinal	6.03/5.92	6.56/6.46	9.10/8.79	5.61/5.46	8.34/7.83
Lateral	0/0	0.26/0.26	1.49/1.54	0.04/0.04	5.29/5.33
Directional	0/0	0.23/0.23	0/0	5.99/3.25	0.00/0.00
Throttle	15.63/17.87	16.94/16.99	16.95/16.84	16.94/15.76	44.23/24.29

8.2 Coordinated Half Turn

The symmetric climb maneuver was designed mostly to involve longitudinal channel commands, although coupling occurred due to the failures. Coordinated half turn is more focused on involving lateral channel commands. However, restriction about preserving the constant altitude makes pilot to work on the longitudinal channel as well. The flight features that should be highlighted during the analysis of this set of flight scenarios are: roll rate p , bank angle φ , altitude H , velocity V , and pitch angle θ .

8.2.1 Nominal Conditions

Figure 8.40 presents the trajectory of the coordinated half turn in the XY horizontal plane. The entire maneuver was generated successfully without any significant differences from the pilot performance outcome.

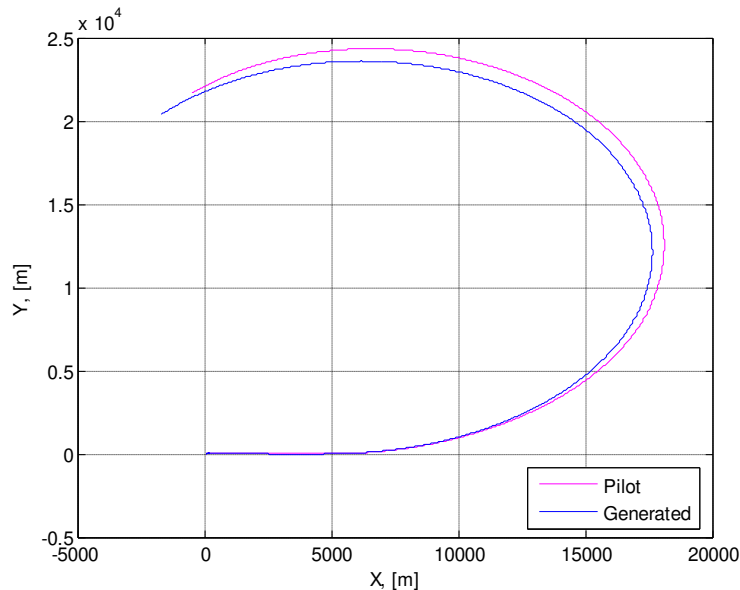
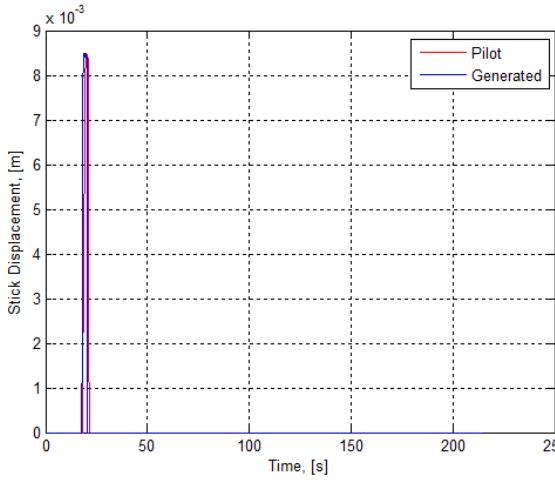
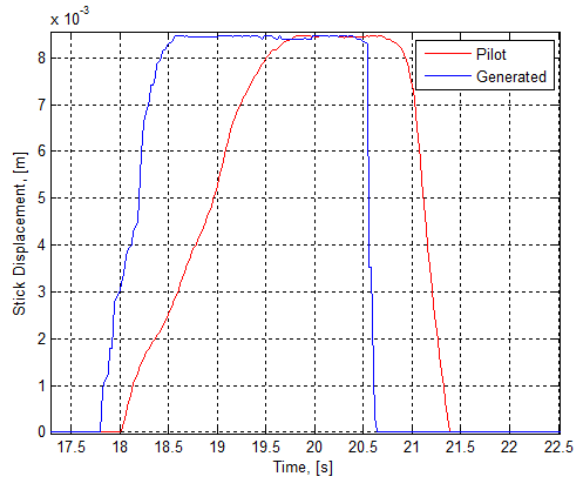


Figure 8.40: Flight Trajectory - Nominal Conditions Coordinated Half Turn

Figure 8.41 presents the lateral channel commands. A short command at the beginning of the flight was produced to reach the desirable bank angle φ (see Fig. 8.42). Since it is a nominal conditions flight, the rest of the flight time no commands on this channel were needed.



a) Entire Flight Time



b) Failure Injection Time

Figure 8.41: Lateral Channel Commands - Nominal Conditions Coordinated Half Turn

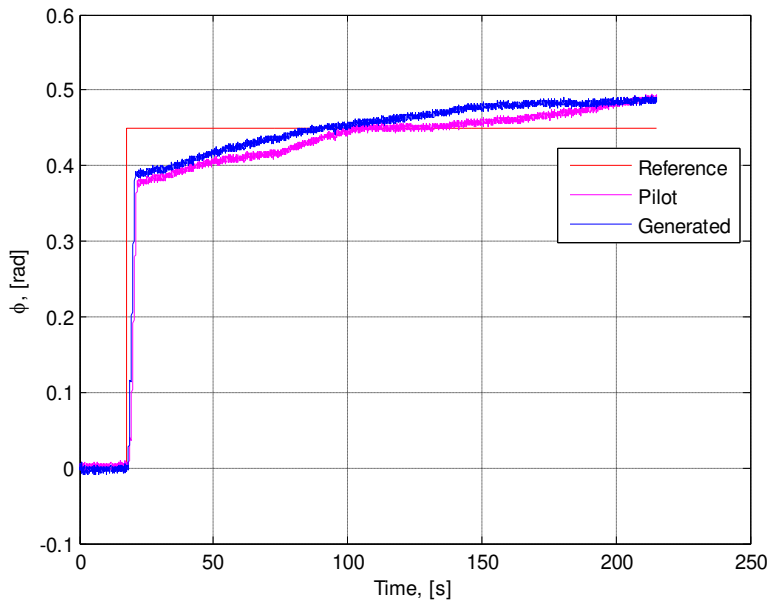


Figure 8.42: Aircraft Bank Angle - Nominal Conditions Coordinated Half Turn

Aircraft roll rate p follows the trend of the lateral stick input (see Fig. 8.43). Therefore, it has a short impulse at the beginning of the flight as well.

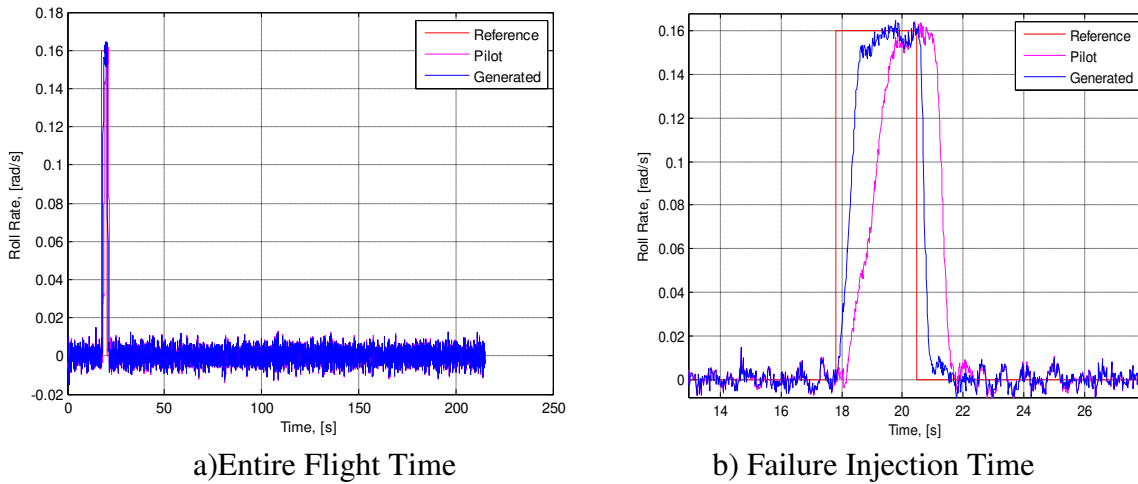


Figure 8.43: Aircraft Roll Rate - Nominal Conditions Coordinated Half Turn

Figure 8.44 presents the longitudinal channel commands. When the aircraft is banked at a constant angle it tends to lose altitude; that is why the pilot must provide certain longitudinal commands to maintain the height. This task is not trivial because constant lateral input has to be preserved as well. Longitudinal command reproduction is not as neat and accurate; however it follows the pilot trend.

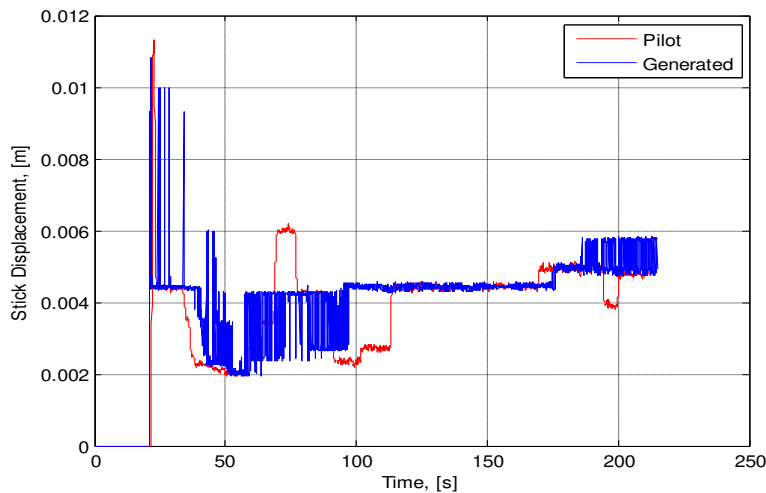


Figure 8.44: Longitudinal Channel Commands - Nominal Conditions Coordinated Half Turn

Figures 8.45 and 8.46 illustrate the aircraft altitude H and pitch rate q variation, respectively. These figures prove that, although the generated longitudinal commands were slightly inaccurate, they served the purpose and the aircraft altitude was well maintained and does not differ a lot from the pilot performance outcome.

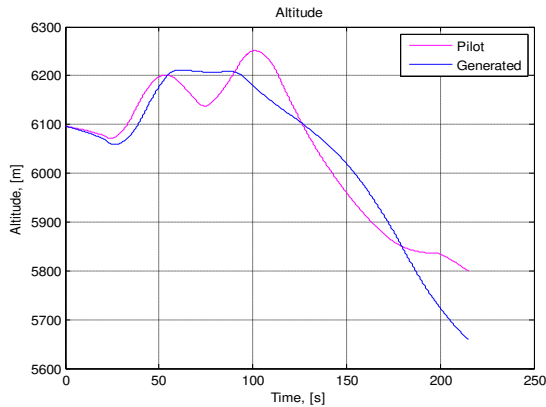


Figure 8.45: Aircraft Altitude - Nominal Conditions Coordinated Half Turn

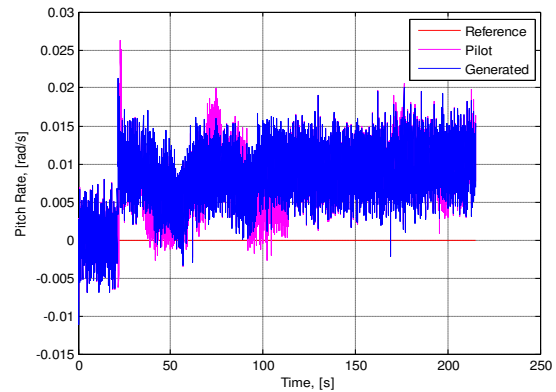


Figure 8.46: Aircraft Pitch Rate – Nominal Conditions Coordinated Half Turn

Figure 8.47 illustrates the aircraft velocity variation. Notice that the velocity did not change its value significantly during flight. This happened because pilot did not provide excessive throttle input (see Fig. 8.48). Velocity alteration was mainly dictated by the aircraft nose direction: up - decelerating, down - accelerating.

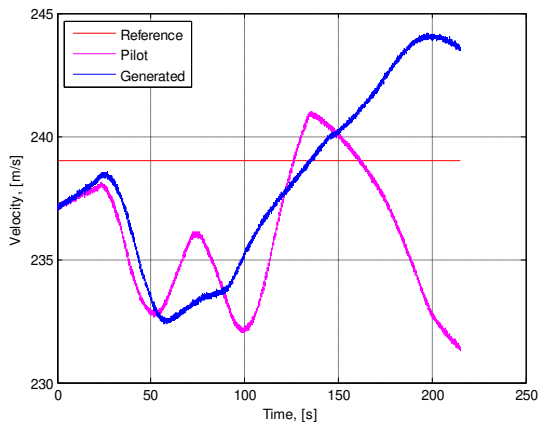


Figure 8.47: Aircraft Velocity – Nominal Conditions Coordinated Half Turn

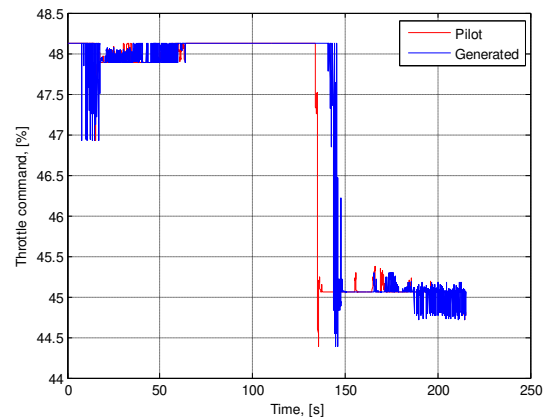


Figure 8.48: Throttle Displacement - Nominal Conditions Coordinated Half Turn

8.2.2 Actuator Failure

The same types of actuator failures as for the climb maneuver have been implemented for coordinated half turn: left aileron, left stabilator, and left rudder locked at +4 degrees. Failure injection time for all the cases is at $T_f=15s$.

8.2.2.1 Aileron Failure

Figure 8.49 presents the flight trajectory in XY plane. The generated trajectory almost coincides with the one from pilot performance.

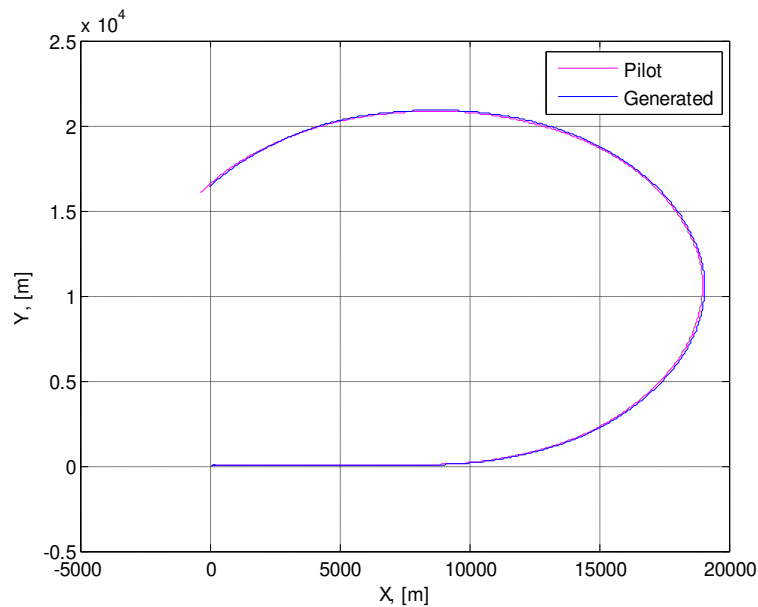


Figure 8.49: Flight Trajectory - Aileron Failure Coordinated Half Turn

Figure 8.50 illustrates the lateral channel commands. A short lateral input is noticed right after the failure occurrence. It was provided to mitigate the consequences of the failure. Over the entire period of time the lateral channel commands have been extracted successfully with some minor inaccuracies at the end.

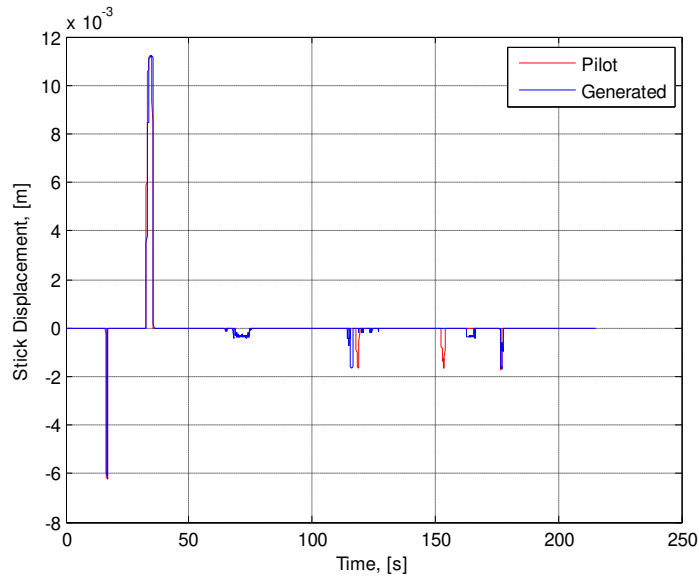


Figure 8.50: Lateral Channel Commands - Aileron Failure Coordinated Half Turn

Bank angle ϕ experienced a significant change at the beginning due to the failure; however, it has been brought back to zero value by pilot efforts (see Fig. 8.51).

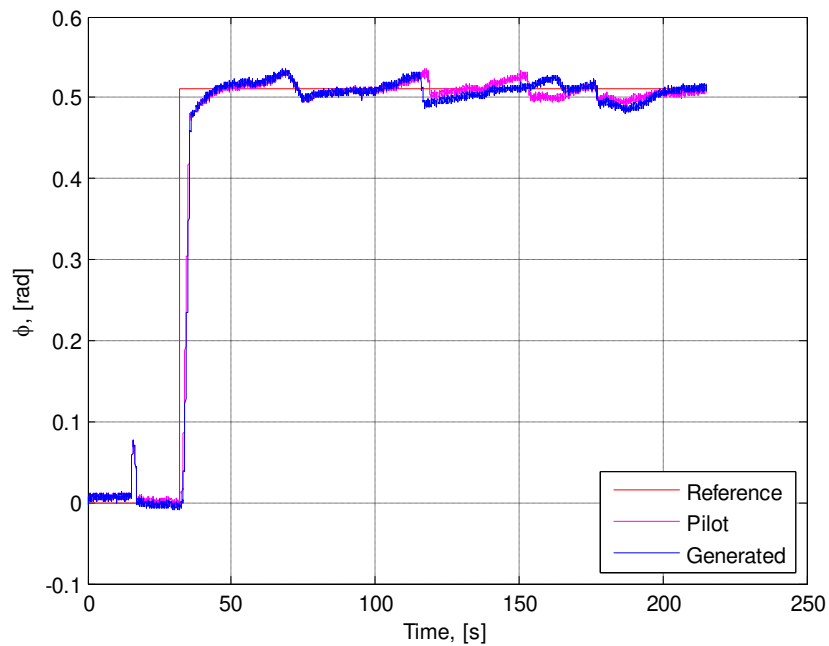


Figure 8.51: Aircraft Bank Angle - Aileron Failure Coordinated Half Turn

Aircraft roll rate p variation is governed mostly by lateral channel input; therefore, it follows its trend. (see Fig.8.52):

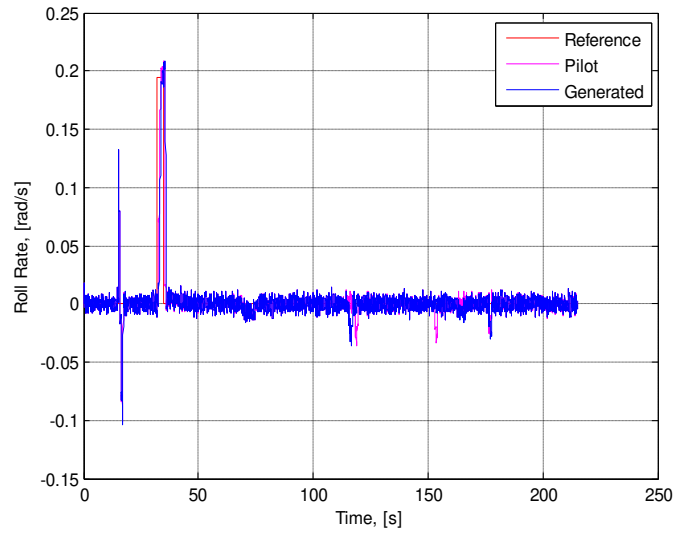


Figure 8.52: Aircraft Roll Rate - Aileron Failure Coordinated Half Turn

Figure 8.53 illustrates longitudinal channel commands. Note that pilot workload on this channel is very large. The pilot input has been very accurately extracted and reproduced. The lines almost coincide at the beginning, with some minor deviations at the end.

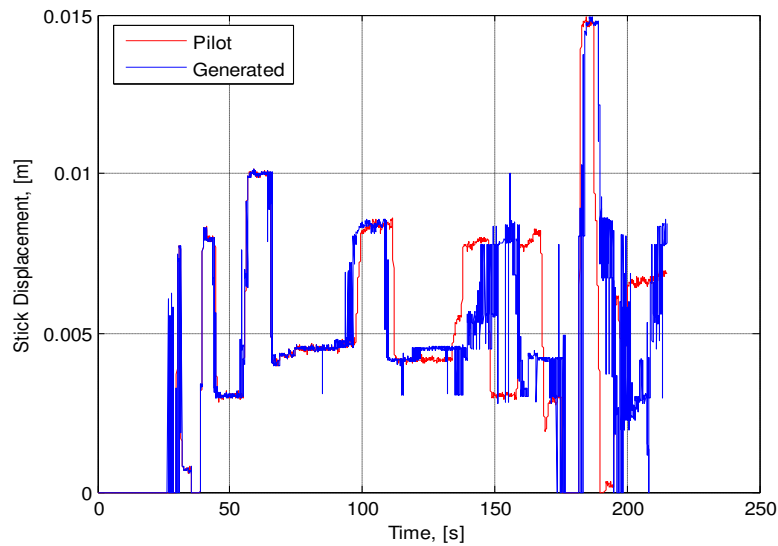


Figure 8.53: Longitudinal Channel Commands - Aileron Failure Coordinated Half Turn

The result of generated longitudinal stick input can be seen in Fig.8.54 and Fig. 8.55, which represent aircraft altitude H and pitch angle θ , respectively. An overall decreasing tendency can be observed due to persistence of constant bank angle φ .

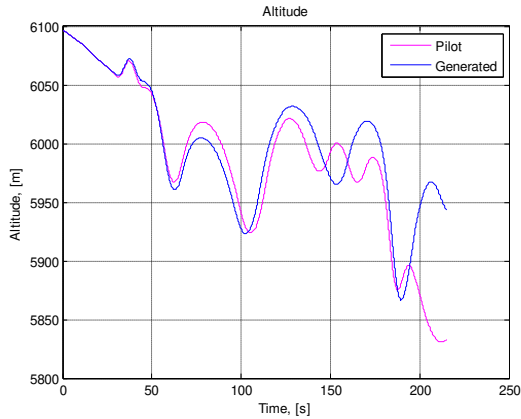


Figure 8.54: Aircraft Altitude - Aileron Failure Coordinated Half Turn

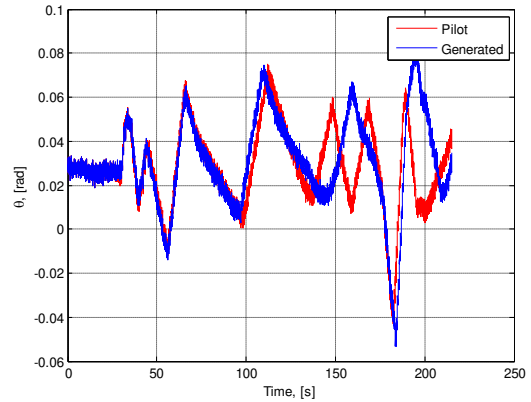


Figure 8.55: Aircraft Pitch Angle - Aileron Failure Coordinated Half Turn

Figures 8.56 and 8.57 illustrate aircraft velocity and throttle displacement, respectively. Pilot workload on changing the thrust level is not very significant. The velocity variation was mainly affected by aircraft pitch (upwards or downwards).

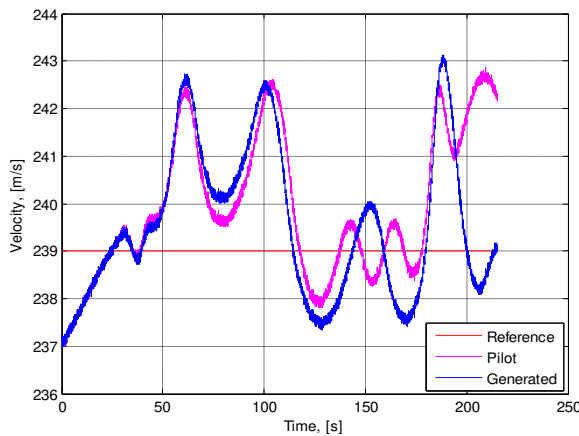


Figure 8.56: Aircraft Velocity - Aileron Failure Coordinated Half Turn

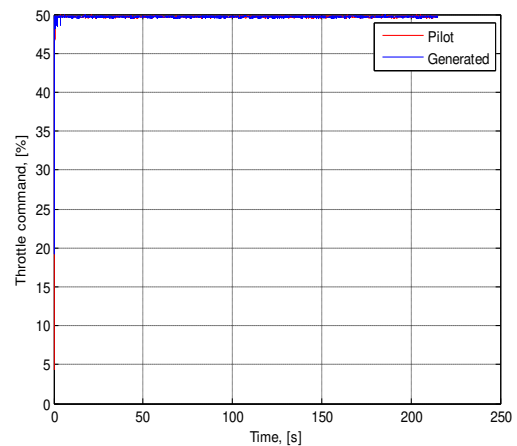


Figure 8.57: Throttle Displacement - Aileron Failure Coordinated Half Turn

8.2.2.2 Stabilator Failure

Figure 8.58 presents the flight trajectory in XY plane. An excellent duplication of pilot performance can be observed. The lines almost coincide.

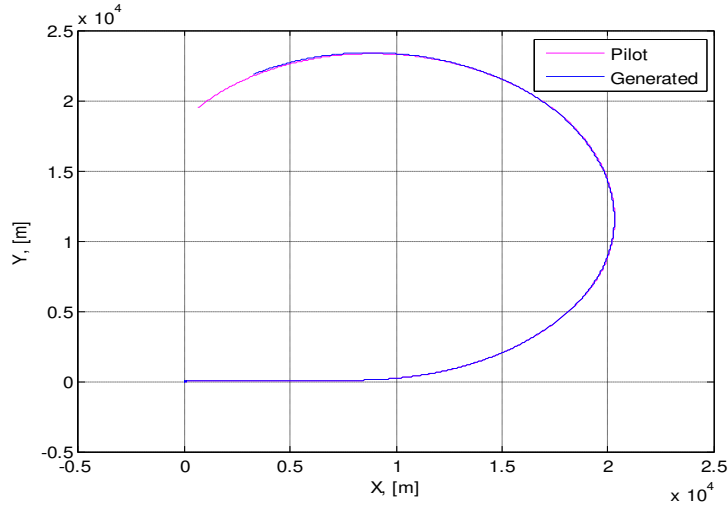


Figure 8.58: Flight Trajectory - Stabilator Failure Coordinated Half Turn

Longitudinal channel commands can be seen in Fig. 8.59. There is a considerable input signal at the beginning of the flight that has been provided to mitigate the failure. The generated simulation provided that signal as well. During the rest of the time, the commands are extracted very accurately, with several inaccuracies at the end.

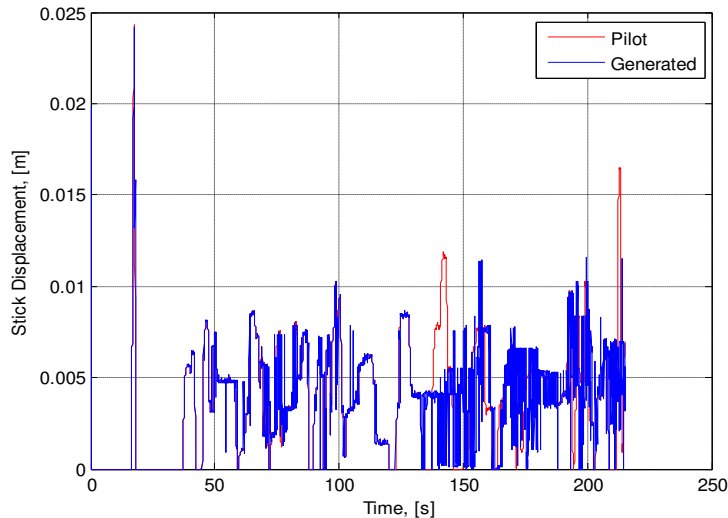


Figure 8.59: Longitudinal Channel Commands - Stabilator Failure Coordinated Half Turn

Longitudinal channel commands dictate the behavior of the pitch angle θ and pitch rate q during the flight. Similarity of the trends can be seen in Fig. 8.60 and Fig. 8.61. Due to the inaccuracies of the longitudinal stick displacement at the end, pitch angle θ slightly deviates from the feature obtained from pilot performance.

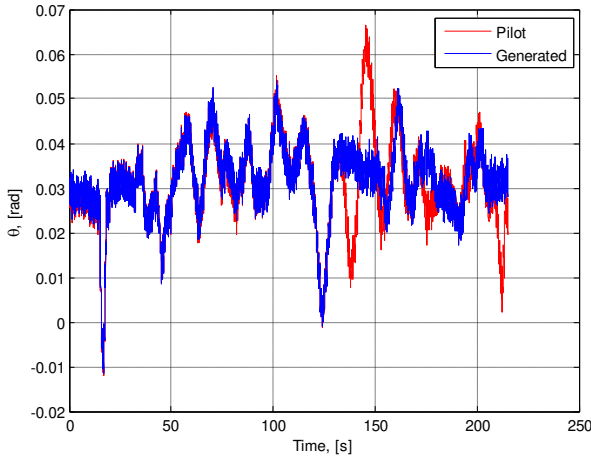


Figure 8.60: Aircraft Pitch Angle - Stabilator Failure Coordinated Half Turn

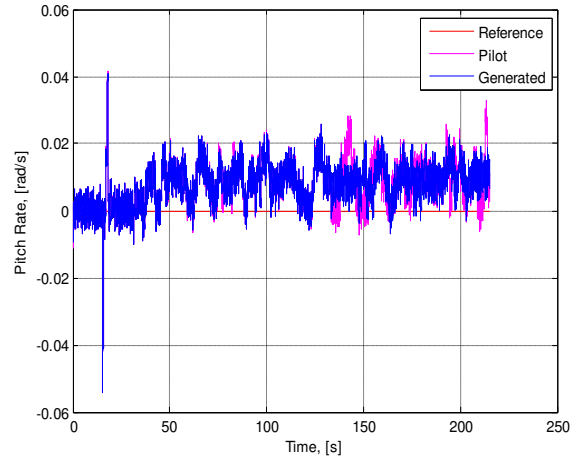


Figure 8.61: Aircraft Pitch Rate - Stabilator Failure Coordinated Half Turn

Lateral channel control commands are presented in Fig. 8.62:

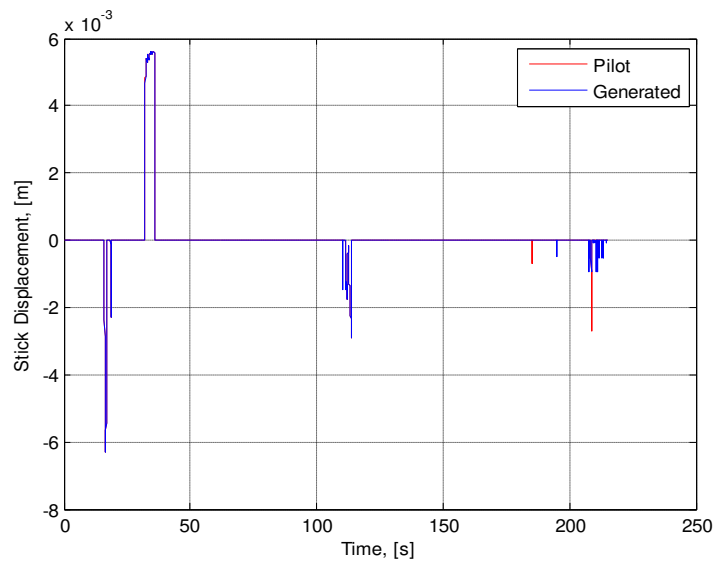


Figure 8.62: Lateral Channel Commands - Stabilator Failure Coordinated Half Turn

Because of the coupling, the pilot had to provide some commands on the lateral channel as well right after the failure was injected. Over the entire flight, the commands have been extracted precisely: the lines almost coincide.

Generated bank angle ϕ dictated by the lateral commands also follows the pilot performance outcome accurately (Fig. 8.63). A little peak is observed at the beginning of the flight due to the failure; however, the value has been brought back close to zero by pilot compensation. Aircraft roll rate p is presented in Fig. 8.64.

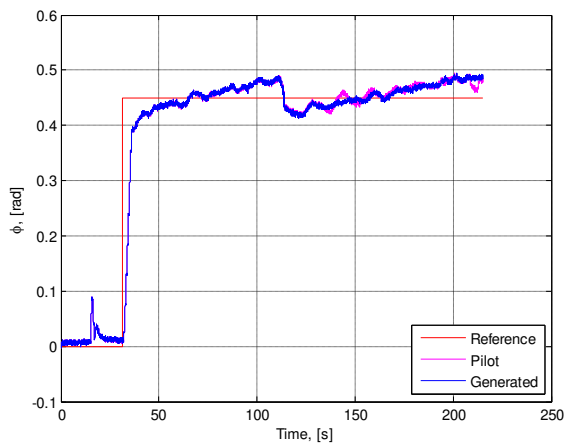


Figure 8.63: Aircraft Bank Angle - Stabilator Failure Coordinated Half Turn

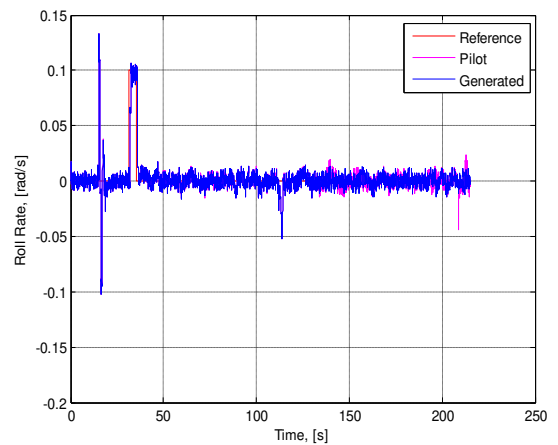


Figure 8.64: Aircraft Roll Rate - Stabilator Failure Coordinated Half Turn

Aircraft velocity and throttle displacement are illustrated in Fig. 8.65 and Fig. 8.66, respectively. No significant activity can be observed on throttle. Generated velocity follows the pilot trend very accurately, with some deviations close to the end of the flight.

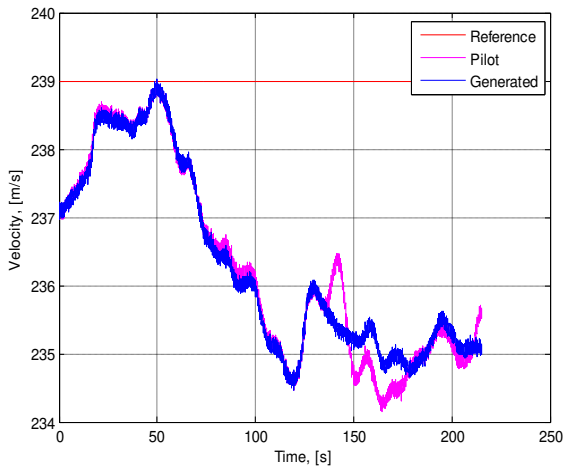


Figure 8.65: Aircraft Velocity - Stabilator Failure Coordinated Half Turn

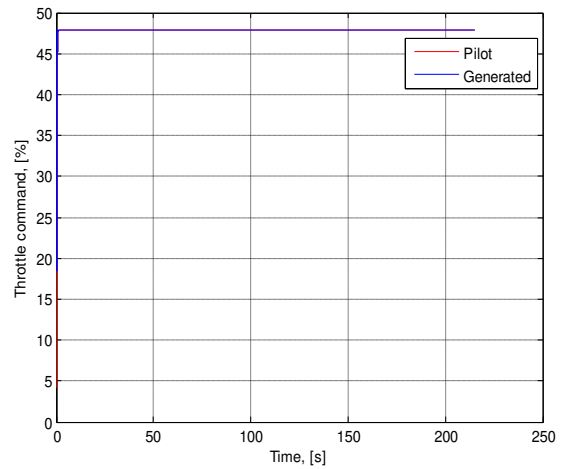


Figure 8.66: Throttle Displacement - Stabilator Failure Coordinated Half Turn

8.2.2.3 Rudder Failure

Figure 8.67 presents the flight trajectory in the XY horizontal plane. The generated trajectory is not as accurate as under aileron or stabilator failure, however, still remains valid and follows the pilot's trend.

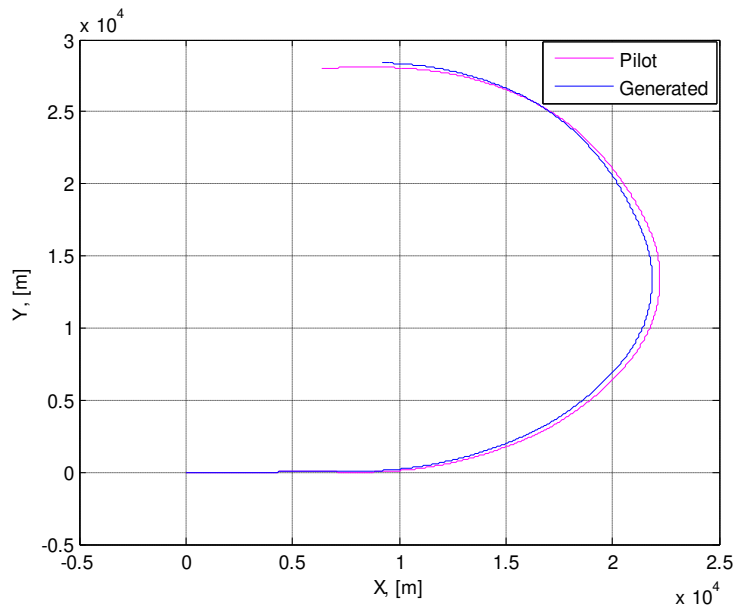


Figure 8.67: Flight Trajectory - Rudder Failure Coordinated Half Turn

Longitudinal channel commands are presented in Fig. 8.68. Generated controls follow the pilot's inputs quite accurately. However, the overall performance is not very accurate as compared to the previous tests:

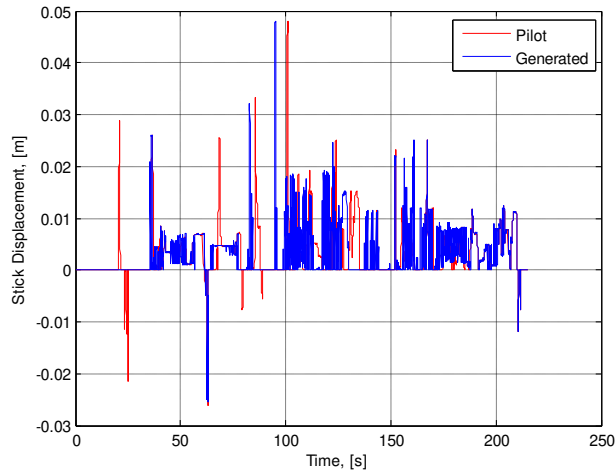


Figure 8.68: Longitudinal Channel Commands - Rudder Failure Coordinated Half Turn

Inaccuracies in generated longitudinal commands affected the behavior of pitch angle θ , which resulted in some deviations as well (see Fig.8.69). However, the generated altitude alteration was not significantly different from the one from pilot performance (Fig.8.70).

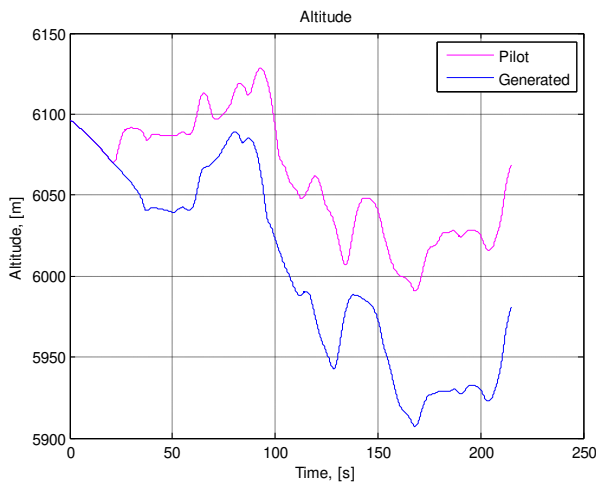


Figure 8.69: Aircraft Altitude - Rudder Failure Coordinated Half Turn

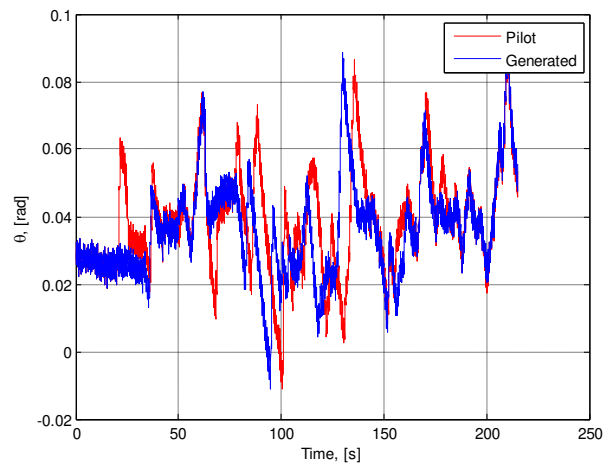


Figure 8.70: Aircraft Pitch Angle - Rudder Failure Coordinated Half Turn

Figure 8.71 illustrates lateral channel commands. A substantial pilot input signal can be observed at the beginning of the flight. The generated signal was extracted adequately; however the magnitude of the signal was not as high. The effect of this inaccuracy can be seen in Fig. 8.72 at approximately 20-30 seconds. During the rest of the time, the commands were extracted quite accurately.

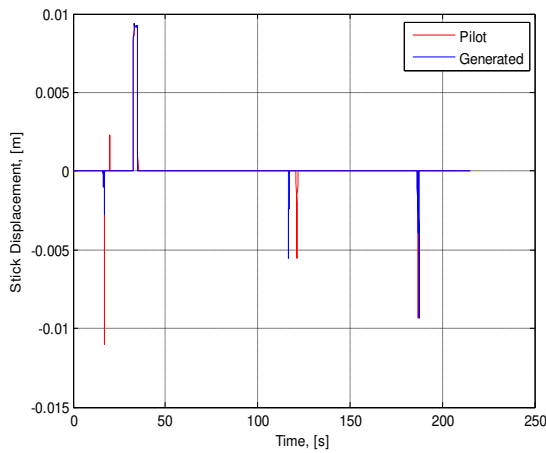


Figure 8.71: Lateral Channel Commands - Rudder Failure Coordinated Half Turn

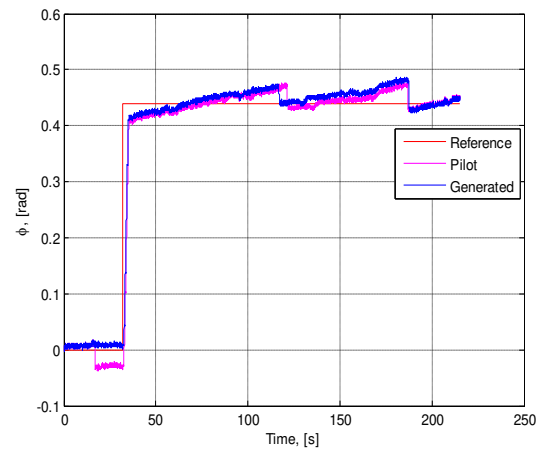


Figure 8.72: Aircraft Bank Angle - Rudder Failure Coordinated Half Turn

Pilot workload on directional channel appeared to be large (Fig. 8.73). Although generally the figure does not look neat, generated commands follow the pilot trend and result in quite accurate duplication of yaw angle ψ (Fig. 8.74).

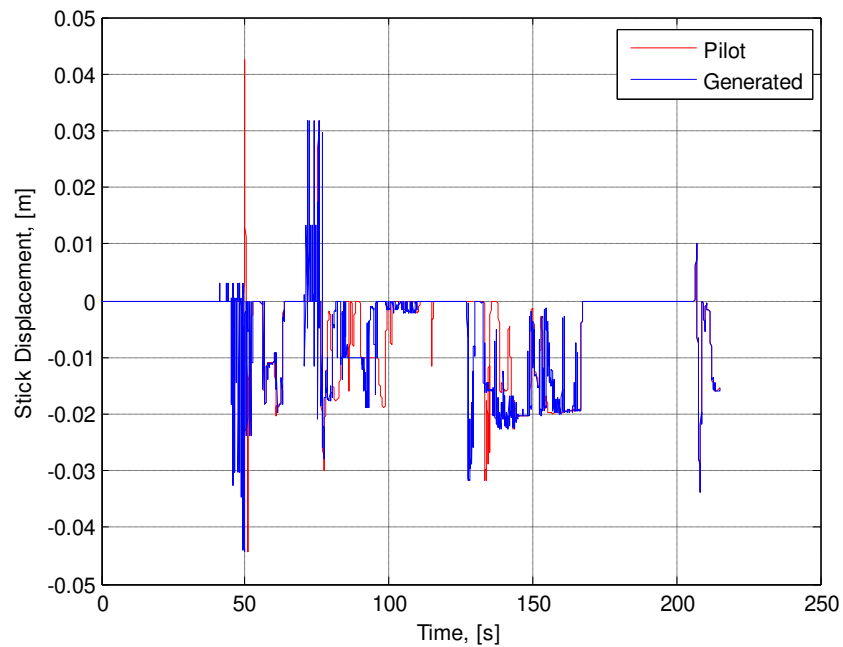
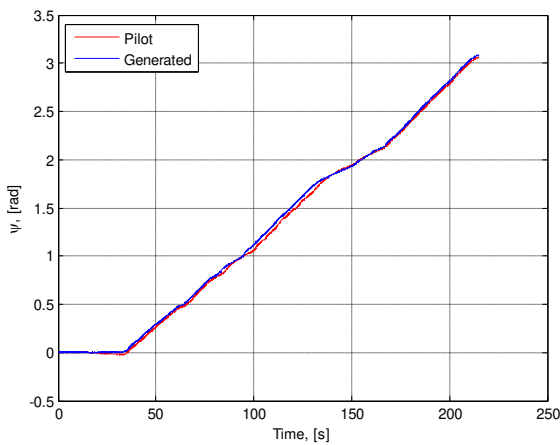
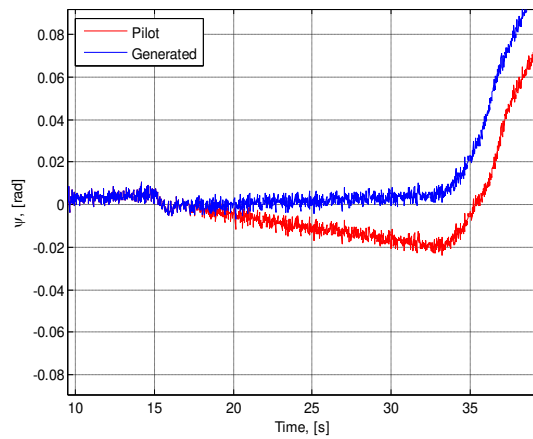


Figure 8.73: Directional Channel Commands- Rudder Failure Coordinated Half Turn



a) Entire Flight Time



b) Failure Injection Time

Figure 8.74: Aircraft Yaw Angle - Rudder Failure Coordinated Half Turn

Aircraft velocity and throttle commands are illustrated in Fig. 8.75 and Fig. 8.76, respectively:

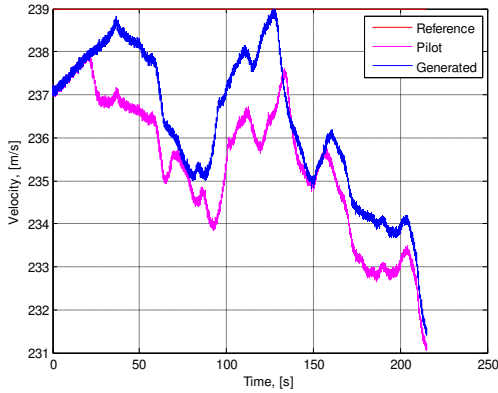


Figure 8.75: Aircraft Velocity – Rudder Failure Coordinated Half Turn

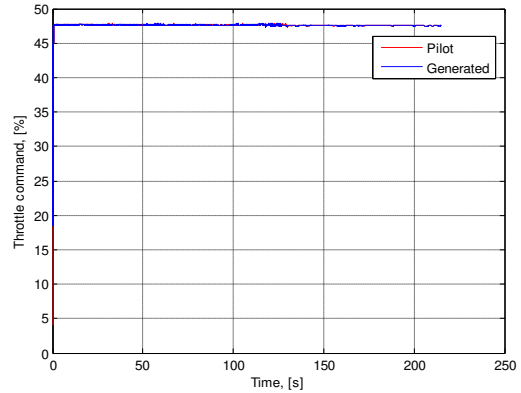


Figure 8.76: Throttle Displacement – Rudder Failure Coordinated Half Turn

8.2.3 Sensor Failure

The same sensor failure as for the symmetric climb has been investigated - roll rate sensor step bias. Failure injection time is $T_f=15s$.

Figure 8.77 illustrates the flight trajectory in the XY horizontal plane. Generated trajectory follows the pilot trend; however, the generated curve has a smaller radius of turn as compared to the one from the pilot performance outcome. This can be explained by several inaccuracies in reproducing lateral channel commands.

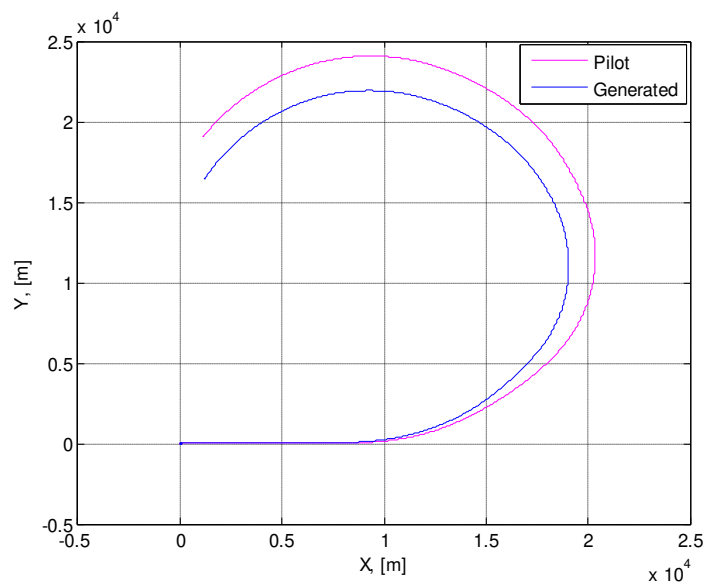


Figure 8.77: Flight Trajectory - Sensor Failure Coordinated Half Turn

Figure 8.78 presents the variation of the altitude during the flight simulation. The generated outcome follows the pilot trend during the entire flight time, except for the last 20 seconds. In the end a drop in altitude (200m) can be observed. Note that the drop can be observed during the pilot performance as well; however, its magnitude is less significant.

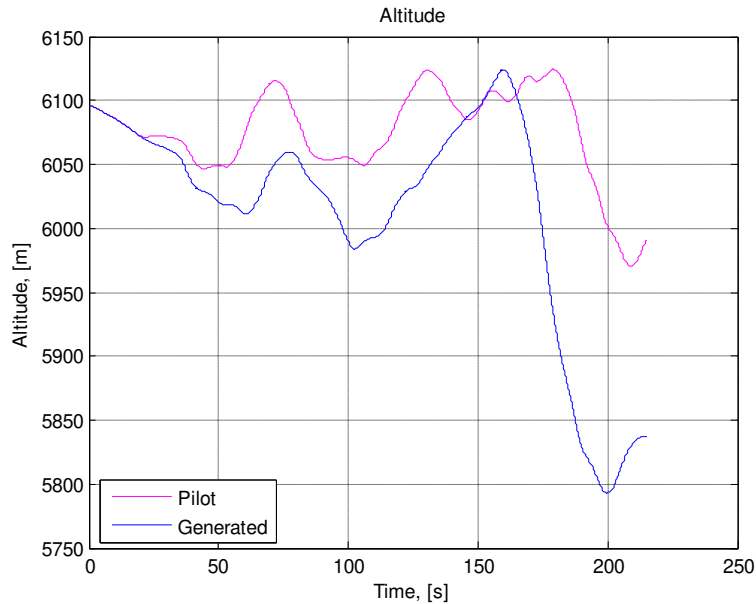


Figure 8.78: Aircraft Altitude - Sensor Failure Coordinated Half Turn

Figure 8.79 presents the lateral channel commands provided during the flight simulation. The commands have not been extracted as accurately as for actuator failures; however, they behave similarly to pilot generated commands with a small delay and several inaccuracies of small magnitude. The delay and these small inaccuracies did not have a significant effect on the behavior of bank angle (see Fig. 8.80). Note that there is no crucial need in extracting the pilot commands extremely accurately. The main idea is to extract the commands in such way, such that the corresponding maneuver key features behave similarly to the ones from pilot performance outcome (e.g. bank angle, altitude). It can be seen that the bank angle was reproduced accurately with several imperfections at the end of the flight simulation.

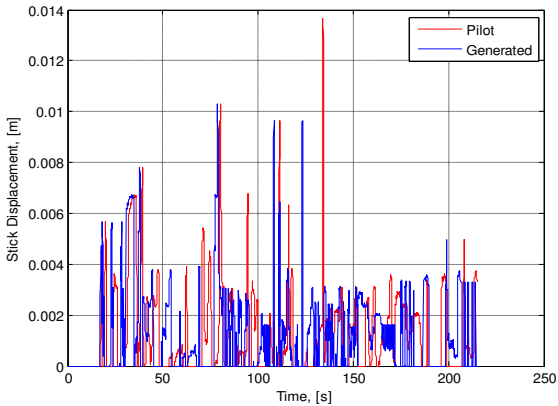


Figure 8.79: Lateral Channel Commands - Sensor Failure Coordinated Half Turn

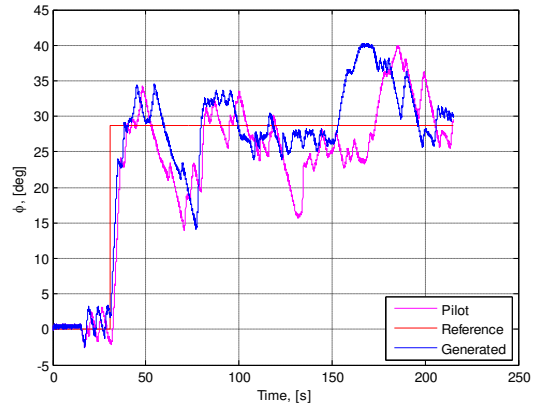


Figure 8.80: Aircraft Bank Angle - Sensor Failure Coordinated Half Turn

Figure 8.81 illustrates the commands on the longitudinal channel. Similarly to the lateral channel, the command extraction is not as accurate as for actuator failure; however, the similar trend can be observed in the behavior. These inaccuracies on longitudinal and lateral channels can be explained by excessive amount of the pilot workload and rapid change of stick displacement during the flight. For the flight with a large pilot workload and very intensive stick commands, a more advanced matching algorithm might be needed for obtaining higher performance results. Also, there is a delay between pilot input and aircraft response which becomes more critical when abrupt maneuvers are involved. Under such conditions, this delay should be better taken into account. One possible solution could be to increase the time window.

Figure 8.82 presents the variation of pitch attitude angle θ and, as expected, it resembles the aircraft altitude variation. The feature has been generated accurately during the entire time of the flight with a small delay. However, a larger difference can be observed during the final period of the flight ($T=170-180$ seconds).

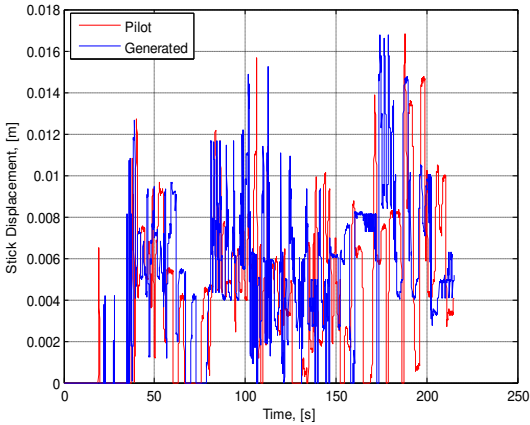


Figure 8.81: Longitudinal Channel
Commands - Sensor Failure Coordinated
Half Turn

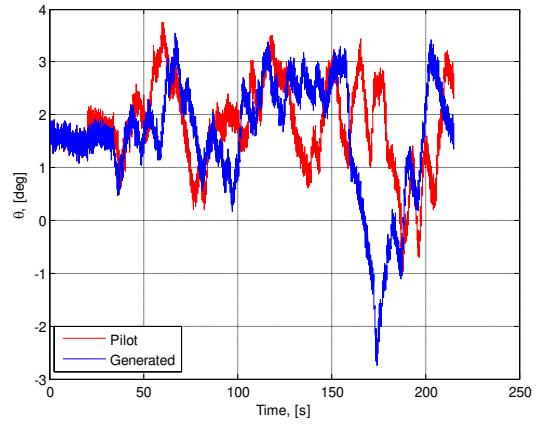
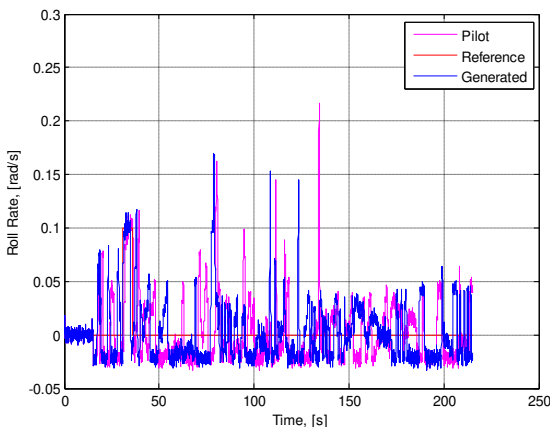
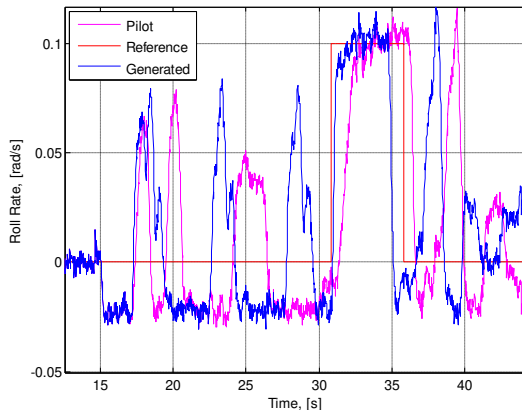


Figure 8.82: Aircraft Pitch Angle - Sensor
Failure Coordinated Half Turn

Aircraft roll rate has been generated precisely enough throughout the entire flight time, and was particularly accurate at the failure injection time (see Fig 8.83). Interestingly, at the moment of the turn (30sec), the generated signal reacts faster than pilot to the reference input signal. Overall performance is satisfactory, except for one peak approximately at 130 sec, when the generated signal did not follow the pilot command.



a) Full Time



b) Failure Injection Period

Figure 8.83: Aircraft Roll Rate - Sensor Failure Coordinated Half Turn

Aircraft velocity during the generated simulation is very similar to the result of pilot performance with an insignificant difference of approximately 1-2 m/s during almost the entire flight (Fig. 8.84). A slightly more visible deviation can be observed during the last period of flight - generated velocity has higher magnitude here. This can be explained by the fact that the aircraft was diving during this period of time.

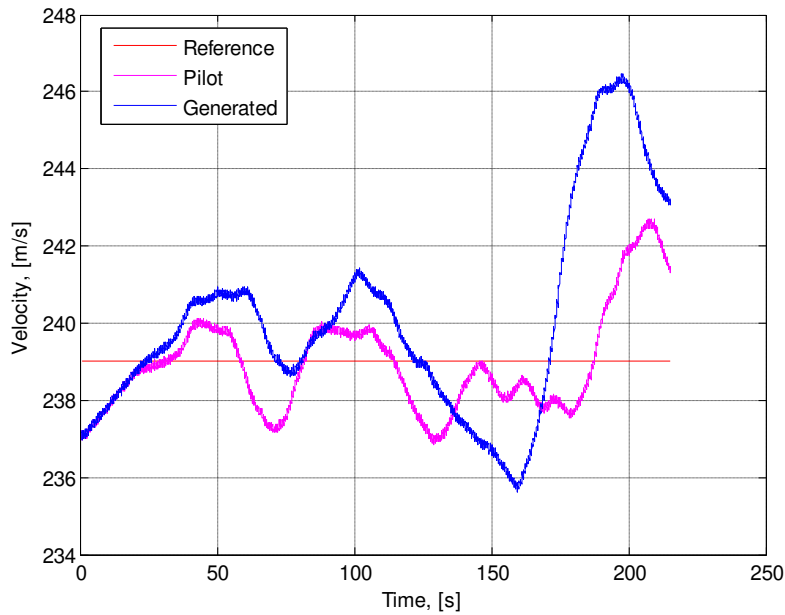


Figure 8.84: Aircraft Velocity - Sensor Failure Coordinated Half Turn

8.2.4 Analysis and Evaluation

The same metrics, as for symmetric climb maneuver, have been used in this section.

Table 8.3 presents the coordinated half turn simulation assessment. As the table illustrates, the results are very accurate: none of the errors exceed the mark of 2%. Table 8.4 illustrates the amount of pilot workload on all four channels under different flight conditions. Pilot workload generally increases with the presence of the failure, as expected. Flights under aileron and sensor failures require the maximum overall pilot workload. Notice that in general, the figures for the coordinated half turn are much higher than for the symmetric climb. This

means that the coordinated half turn maneuver is more complicated, and requires more pilot activity. Only throttle activity is higher for the symmetric climb maneuver, as expected. In order to maintain constant velocity, pilot has to provide a lot of throttle input during the process of climbing.

Table 8.3: Flight Simulation Performance Assessment - Coordinated Half Turn

Flight Conditions	Nominal Conditions		Aileron Failure		Stabilator Failure		Rudder Failure	
	H	V	H	V	H	V	H	V
Pilot Flight Error, %	1.92	1.41	1.68	0.57	0.36	1.13	0.36	1.13
Generated Flight Error, %	1.94	1.34	1.48	0.55	0.31	1.13	1.40	1.12
Relative Error, %	0.72	1.24	0.34	0.31	0.07	0.06	1.24	0.46

Table 8.4: Pilot Workload - Coordinated Half Turn

Channel	Nominal Conditions	Aileron Failure	Stabilator Failure	Rudder Failure	Sensor Failure
Longitudinal	18.31/19.31	22.90/22.67	18.44/18.49	16.40/15.84	20.55/24.33
Lateral	0.46/0.48	0.85/0.88	0.71/0.69	0.71/0.62	5.90/5.91
Directional	0/0	0/0	0/0	23.56/25.00	0/0
Throttle	5.90/5.24	8.07/8.00	1.15/1.08	2.73/2.65	9.11/9.08

8.3 Symmetric Descent

The features that have the most significant dynamic impact during the symmetric descent maneuver are the same as for the symmetric climb maneuver: pitch rate q , pitch angle θ , altitude H , and velocity V . Symmetric descent maneuver has not been investigated in such detail as symmetric climb. The results have been obtained for the nominal conditions flight.

Figure 8.85 presents the aircraft altitude variation during the symmetric descent flight. Altitude response has been generated very accurately, although there is a small deviation from pilot performance outcome at the final stage of the flight. Approximate tracking error does not exceed 60 meters. This type of behavior of the aircraft altitude curve, namely very accurate result for the initial period of the flight, and less accurate result for the final period of the flight, can be explained by the longitudinal channel commands extraction.

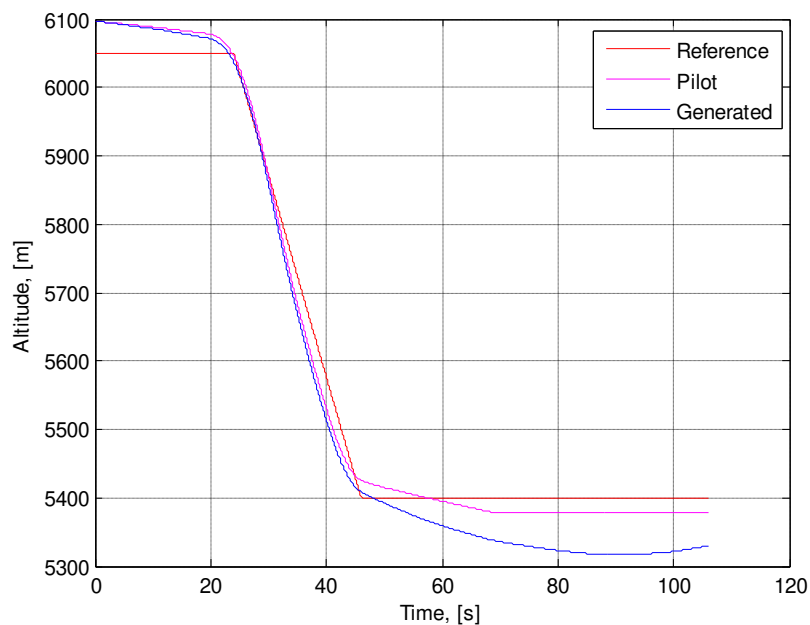


Figure 8.85: Aircraft Altitude - Nominal Conditions Symmetric Descent

Figure 8.86 illustrates the longitudinal channel commands. A perfect command extraction can be observed up until the time $T=30$ seconds. This explains the almost coinciding curves of the altitude plot during the initial period of the flight. Afterwards, several minor inaccuracies can be noticed in commands extraction. This time period of the maneuver corresponds to approaching the desired altitude and maintaining a steady state flight. During the final period of the flight, a constant generated input of a small magnitude on the longitudinal channel can be observed. This extraction imperfection can explain the deviation from the desired altitude at the end of the maneuver.

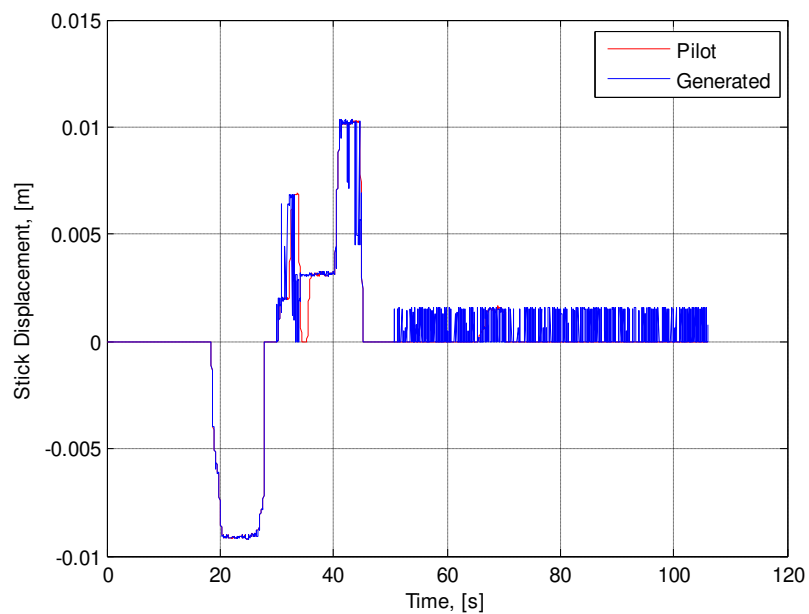


Figure 8.86: Longitudinal Channel Commands - Nominal Conditions Symmetric Descent

Aircraft pitch rate variation is mostly dictated by the longitudinal channel commands. Therefore, as expected, it behaves very similarly, and the same trend can be noticed in Fig. 8.87. Interestingly, the constant small inaccuracies on the longitudinal channel did not affect significantly the aircraft pitch rate – the generated signal is almost the same as pilot output.

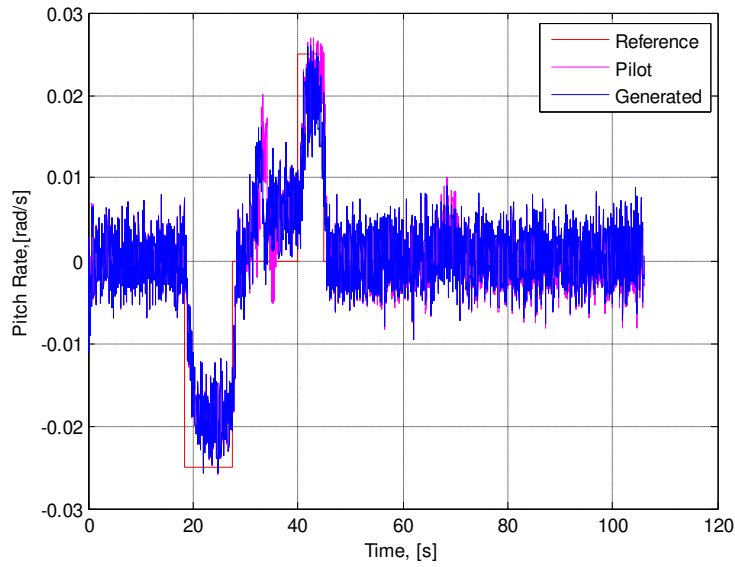


Figure 8.87: Aircraft Pitch Rate - Nominal Conditions Symmetric Descent

Figures 8.88 and 8.89 illustrate aircraft velocity and throttle commands, respectively. Significant amount of inaccurate command extraction can be observed during the first 15-20 seconds of the flight; however, as we can see from Fig. 8.89, it did not affect substantially the behavior of aircraft velocity, and the feature has been generated quite accurately. Although during the rest of the flight time the throttle commands were extracted precisely, a significant tracking error of the aircraft velocity can be noticed (the error is approximately 10-11 m/s, which does not exceed 5% error margin). This can be explained by the nature of the maneuver – the aircraft was diving, and the provided throttle commands were not enough to maintain the constant velocity.

Based on these results, we can conclude that the extraction of the control commands for the symmetric descent maneuver under nominal conditions has been performed successfully. Implementing the same algorithm for different types of maneuvers (climb, descent, and turn) and getting good results proves the robustness of the proposed methodology. However, another set of

tests with different subsystem failures should be investigated for symmetric descent maneuver in order to completely confirm the credibility of the designed failure accommodation model.

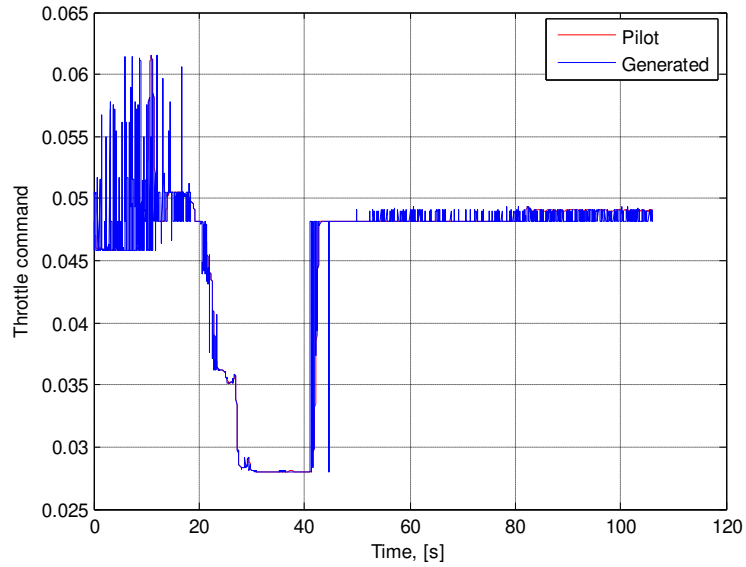


Figure 8.88: Throttle Command - Nominal Conditions Symmetric Descent

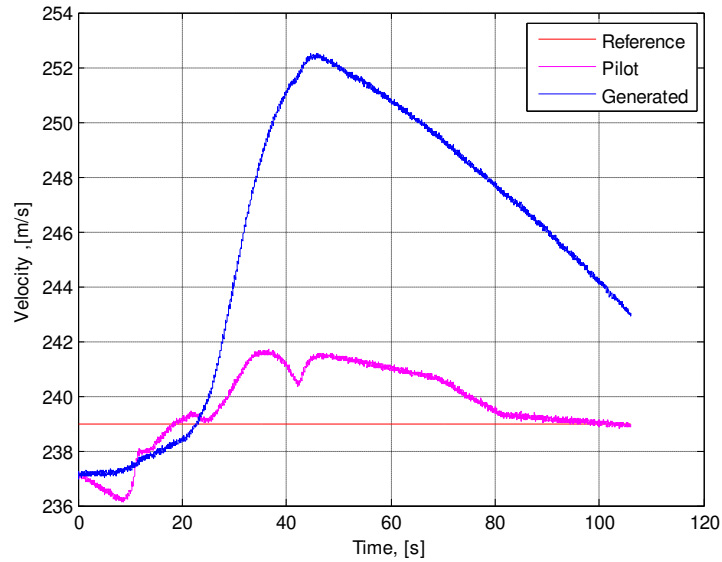


Figure 8.89: Aircraft Velocity - Nominal Conditions Symmetric Descent

8.4 Comprehensive Analysis and Assessment

Based on the results presented, it can be concluded that the extraction of compensatory commands for all the maneuvers that were considered has been performed successfully. High performance has been achieved for both nominal and failure conditions. For all the maneuvers, aileron and stabilator failures have been adequately handled by the algorithm with very high accuracy of extraction. Rudder failure has not been handled as successfully as other failures; however, the results are still very good and illustrate the capability of the proposed methodology of extracting commands.

During the analysis of the flight performance under pilot control and under accommodation system control one might notice that for several flight conditions the designed system provides better and more accurate results than the pilot. The accommodation system proposed in this research effort accumulates all the data from all the flights and selects the set of control commands that corresponds to the set of features that best match the current measured state. This matching set of features may occur at different times or even in a different test than the one involved in the comparison. It should also be noticed that the extracted pilot commands are applied dynamically and may result in a different response compared to the pilot performance outcome.

Analysis has shown that generally, as expected, an injected failure significantly increases pilot workload. The failure that has the most significant impact on the dynamic of the flight is locked stabilator. This happened due to coupling of longitudinal and lateral channels combined with the desirable condition of maintaining the altitude.

A special attention has to be paid to the roll rate sensor failure. The results for this failure could be considered partially successful. Flight features that have a significant impact on the maneuver such as velocity and altitude have been generated very accurately. However,

significant errors on the lateral channel have resulted in the deviation of the aircraft from the straight line flight during the climb maneuver, and smaller radius of the trajectory during the coordinated turn maneuver. Nevertheless, these results remain very promising with regard to compensatory commands extraction within the AIS paradigm, and require more extensive in-detailed investigation in future.

During the process of investigation, development and validation of the proposed methodology of the flight failure accommodation, an approximate total of 50 flight tests have been executed by the pilot, and approximate total of 165 flight simulations have been performed.

Chapter 9 Conclusions and Future Work

9.1 Research Summary

The purpose of this research was to introduce a novel methodology for aircraft failure accommodation as a part of a comprehensive and integrated FDIEA process within the AIS paradigm. To achieve the purpose, the following steps have been accomplished:

- An overview of the comprehensive and integrated FDIEA problem has been provided. Different solutions and alternatives for solving the FDIEA problem have been investigated in order to find a methodology that would meet all the problem requirements.
- AIS paradigm has been chosen as a ground for solving the FDIEA problem in general and the accommodation part in particular. To support the choice of the methodology AIS-scheme main advantages have been identified: complexity, specificity, and robustness.
- Specific algorithms for failure accommodation within the AIS paradigm were designed, developed, and implemented within the WVU simulation environment.
- Simulation results have been presented and analyzed to demonstrate the validity and promising capabilities of the proposed methodology.

The major effort during this research was the accommodation algorithm design and development. The following are the steps that have been accomplished to achieve this:

- A new mathematical model has been proposed with elements representing the main components of the natural immune system. Behavior of each element has been assimilated to the corresponding element of aircraft control system.
- A specific matching algorithm that imitates antigen recognition by the immune system has been introduced. Different ways of improving the performance of the algorithm have been described as well.
- WVU simulation environment has been utilized for data acquisition and algorithm implementation purposes. For the accommodation problem, two different types of maneuvers have been designed: symmetric climb/descent and coordinated half turn. The maneuvers were designed such that the dynamic fingerprint of the aircraft behavior under certain conditions could be captured.
- WVU simulation environment aircraft health management analysis based on Matlab/Simulink software has been utilized for the proposed algorithm implementation. A special additional Simulink model has been built for the implementation of the accommodation algorithm. This additional model that consists of 2 major blocks has been successfully integrated within the existing WVU simulation environment.

The main contributions of this research effort are:

- Possibility of extracting compensatory commands from previous pilot experience for failure accommodation purposes within the AIS paradigm has been demonstrated.
- Capability of the developed algorithm to operate as an automatic control system and to follow the mission maneuvers has been confirmed.

- Effectiveness of the proposed solution on the unmanned aircraft model using the motion-based flight simulator for a variety of actuator failures (and one sensor failure) has been demonstrated.
- High performance in extracting compensatory commands has been achieved. The percentage errors in the key flight features - velocity and altitude - do not exceed the mark of 6% and remain below 1.5% for the large majority of the tests.

Implementation of the proposed AIS-scheme for failure accommodation can potentially have a significant impact on the safety of aircraft safety operation. However, extracting compensatory commands for control purposes within the AIS paradigm was only a first step in solving the more complicated and comprehensive general accommodation problem. In real life situations, the most important feature of the adaptive control system is to be able to deal with new, previously un-encountered types of failures.

9.2 Future Work

Considering future research studies to improve and extend the proposed AIS-based methodology for integrated FDIEA problem in general and accommodation problem in particular, several recommendations are proposed as following:

- Extend the area of possible flight failures. Different levels and severities of actuator and sensors failures have to be investigated. Structural and engine failures have to be included in the list of interest.

- Investigate the possibilities of improving the matching algorithm. This is expected to achieve higher performance in command extraction.
- Build the AIS using comprehensive flight data. This involves the design of a set of more complicated coupled maneuvers over extended areas of the flight envelope. That would close the gap to real-life situations and allow investigating the robustness of the developed accommodation algorithm.
- Integrate the proposed algorithm for failure accommodation with existing scheme for flight failure detection, identification, and evaluation. That would allow combining all components of the comprehensive FDIEA problem, integrating them into one single mechanism and implementing to some aircraft model to validate its functionality.
- Investigate additional processing of the AIS extracted commands such as scaling and filtering.
- Increase robustness of the proposed system by exposing different pilots for the data acquisition process

Bibliography

- [1] Anonymous. Annual reviews of aircraft accident data. Technical report, National Transportation Safety Board, 2011.
- [2] Anonymous. General aviation accident trends and factors for 1997 - 2006. Technical report, Aircraft Owners and Pilots Association, 2006.
- [3] Anonymous. Aviation safety database 1943 - 2008. Technical report, Flight Safety Foundation, 2008.
- [4] Anonymous. Accident-incident data system. Technical report, Federal Aviation Administration, 2007.
- [5] Medsker L.R. Hybrid intelligent systems. In Kluwer Academic Publishers, Boston, 1995.
- [6] Zhang X. and Yamane Y. Optimal design of model following control with genetic algorithm. *Differential Equations and Control Processes (Electronic Journal)*, 4, 1998.
- [7] Zhang, Y. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229-252, 2008.
- [8] Perhinschi, M. G., Moncayo, H., and Davis, J. Integrated framework for artificial immunity-based aircraft failure detection, identification, and evaluation. *AIAA Journal of Aircraft*, 47(6):1847-1859, Nov-Dec 2010.
- [9] Belcastro, C. M. and Jacobson, S. R. Future integrated systems concept for preventing aircraft loss-of-control accidents. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, Aug 2010.
- [10] Perhinschi, M. G, Moncayo, H., and Al Azzawi, D. Integrated immunity-based framework for aircraft abnormal conditions management. *Submitted to AIAA Journal*. April 2013.
- [11] Moncayo H., Perhinschi M. G., and Davis J. Artificial-immune-system-based aircraft failure evaluation over extended flight envelope. *Journal of Guidance, Control and Dynamics*, 34(4), July 2011.
- [12] Nguyen, N., Krishna Kumar, K., Kaneshige, J., and Nespeca, P. Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006.
- [13] Tessler. A. Structural analysis methods for structural health management of future aerospace vehicles. In *NASATM-2007-214871*, Hampton, VA, August 2007. Langley Research Center.

- [14] Totah, J., Krishna Kumar, K. and Viken, S. Stability, maneuverability, and safe landing in the presence of adverse conditions. Integrated Resilient Aircraft Control Technical Plan, 2010.
- [15] Srivastava, A. N., Mah, R.W., and Meyer, C. Automated detection, diagnosis, prognosis to enable mitigation of adverse events during flight. Integrated Vehicle Health Management Technical Plan, 2010.
- [16] Young, S. D. and Quon, L. Integrated intelligent flight deck. Integrated Intelligent Flight Deck Technical Plan Aviation Safety Program, 2010.
- [17] Fiesler, E. and Beale, R. Handbook of Neural Computation. Oxford University Press, New York, NY, 1997.
- [18] Back, T., Fogel, D.B., and Michalewicz, Z. Handbook of Evolutionary Computation. Oxford University Press, New York, NY, 1997.
- [19] Jungheui, C. and Deaton, R. Introduction to DNA computing. In Genetic and Evolutionary Computation Conference Tutorial Program, pages 480-492, Francisco, CA, 2001.
- [20] Hofmeyr, S.A and Forrest, S. Immunity by design: An artificial immune system. In Genetic and Evolutionary Computing Conference, pages 1289-1296, Orlando, FL, 1999.
- [21] S.A Hofmeyr. An Interpretative Introduction to The Immune System. Oxford University Press. New York, NY, 2010.
- [22] Al Azzawi D., Perhinschi M. G., and Moncayo H. Dendritic cell mechanism for aircraft immunity-based failure detection and identification. In AIAA Guidance, Navigation, and Control Conference, Boston, MA, August 2013.
- [23] Takahashi, K. and Yamada, T. Application of an immune feedback mechanism to control systems. The Japan Society of Mechanical Engineers, 41(2): Series C., 1998.
- [24] Karr C. L., Nishita K., Graham K. S. Adaptive aircraft flight control simulation based on an artificial immune system. Applied Intelligence, pages 295-308, 2005.
- [25] B. W. Johnson. Fault-tolerant microprocessor-based systems. Fault-Tolerant Microprocessor-Based Systems, 4(6):6- 21, 1985.
- [26] Blanke, M., Kinnaert, M. J., Lunze, M., and Staroswiecki, M. Diagnosis and fault-tolerant control. In Proc. of Springer Verlag, Heidelberg, 2006.
- [27] R. J. Patton. Fault tolerant control systems: The 1997 situation. In Proc. IFAC Safe process, pages 1033-1055, Hull, United Kingdom, 1997.
- [28] M. M. Polycarpou. Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach. IEEE Trans. Aut. Contr., 46(01):736 -742, 2001.

- [29] R. J. Patton. Robustness issues in fault tolerant control. In Proc. of IEE Coll. on Condition Monitoring and Fault Tolerance, pages 1/1-1/20, 1990.
- [30] Robust Model Based Fault Diagnosis for Dynamic Systems, Boston, 1999. Kluwer Academic Publishers.
- [31] Zhou, J. and Ren, Z. A new controller architecture for high performance robust, and fault-tolerant control. IEEE Transactions on Automatic Control, 46(10):1613-1618, 1995.
- [32] Zhang, Y. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. Annual. Rev. Control, 32:229-252, 2008.
- [33] Zhang, Y. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. In Proc. Of the IFAC Symposium - Safe Process, pages 265-276, WA, USA, 2003.
- [34] Zhao, J., Jiang, B., Shi, P., and He, Z. Fault tolerant control for damaged aircraft based on sliding mode control scheme. International Journal of Innovative Computing, Information and Control, 10(1):293-302, 2014.
- [35] A Fekih. A robust fault tolerant control strategy for aircraft systems. In 18th IEEE International Conference on Control Applications Part, Saint Petersburg, Russia, July 2009.
- [36] Zhang, Y.M., Chamseddine, A., Rabbatha, C. A, Gordona, B. W., Sua, C.-Y., Rakhejaa, S., Fulfordb, C., Apkarianb, J., and Gosselinc P. Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter tested. Journal of the Franklin Institute, 2013.
- [37] Park, S., Bae, J., Kim, Y., and Kim, S. Fault tolerant flight control system for the tilt-rotor UAV. Journal of the Franklin Institute, 2013.
- [38] Xua, D., Jianga , B., Liub, H., and Shi, P. Decentralized asymptotic fault tolerant control of near space vehicle with high order actuator dynamics. Journal of the Franklin Institute, 2013.
- [39] Bennouna, O. and Langlois, N. Development of a fault tolerant control for aircraft electromechanical actuators, In Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS), IEEE conference, pages 1-5, Bologna, Italy, Oct 2012.
- [40] Fekih, A. and Pilla, P. A new fault tolerant control strategy for aircraft systems under adverse flying conditions. J. Automation and Systems Engineering, 2009.
- [41] Nayebpanaha, N., Rodrigues, L., and Zhanga, Y. Fault tolerant control for partial loss of control authority in aircraft using piecewise affine slab models. Journal of the Franklin Institute, 2013.

- [42] Belcastro C. M., Jacobson S. R., “Future Integrated Systems Concept for Preventing Aircraft Loss-of-Control Accidents”, Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, Aug. 2010.
- [43] Tang L., Roemer M., Bharadwaj S., Belcastro C., “An Integrated Aircraft Health Assessment and Fault Contingency Management System”, Proc. of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, Aug 2008
- [44] Li, Q., Ren, Z., Dai, S., and Wang, W. A new robust fault-tolerant controller for self-repairing flight control system. *Journal of the Franklin Institute*, 2013.
- [45] Roitt, I., Brostoff, J., and Male, D. *Immunology* (second edition). J.B. Lippincott Co, London, 1989.
- [46] de Castro, L. N. and Von Zuben, F. J. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239-251, 2002.
- [47] Forrest, S., Perelson, A.S., Allen, L., and Cherukuri, R. Self-nonsel discrimination in a computer. In 1994 IEEE Symposium on Research in Security and Privacy, pages 202 - 212, Los Alamitos, CA, 1994.
- [48] Timmis, J., Neal, M., and Hunt, J. An artificial immune system for data analysis. *Bio-Systems*, 55(1):143 - 150, 2000.
- [49] Greensmith, J. and Aickelin, U. Artificial dendritic cells: Multi-faceted perspectives. *Human-Centric Information Processing Through Granular Modeling*, pages 375 - 395, 2009.
- [50] de Castro, L. and Timmis, J. Artificial immune systems: A new computational intelligence approach. Springer, pages 57 - 58, 2002.
- [51] Farmer, J.D., Packard, N., and Perelson, A. The immune system, adaptation and machine learning. *Physica D*, 2:187 - 204, 1986.
- [52] Bersini, H. and Varela, F.J. Hints for adaptive problem solving gleaned from immune networks. *Parallel Problem Solving from Nature, First Workshop PPSW 1*, October 1990.
- [53] Forrest, S., Perelson, A.S., Allen, L., and Cherukuri R. A change-detection algorithm inspired by the immune system. *IEEE Transactions on Software Engineering*, 1995.
- [54] Dasgupta, D. and Forrest, S. Novelty detection in time series data using ideas from immunology. In *Proceedings of the International Conference on Intelligent Systems*, Reno, Nevada, 1996.

- [55] Zhi-tang, L., Yao, L., and Li, W. Novel fuzzy anomaly detection algorithm based on artificial immune system. In Proceedings of the Eighth International Conference on High-Performance Computing in Asia- Pacific Region, 2005.
- [56] Sanchez, S. P., Perhinschi, M. G., Moncayo H., Napolitano M. R., Davis J., and Fravolini M. L. In- Flight actuator failure detection and identification for a reduced size UAV using the artificial immune system approach. In AIAA, Chicago, IL, August 2009.
- [57] Dasgupta, D. and Majumdar N. Anomaly detection in multidimensional data using negative selection algorithm. In Congress on Evolutionary Computation CEC '02, volume 02, pages 1039-1044, August 2002.
- [58] de Castro, L. and Timmis, J. Artificial immune systems as a novel soft computing paradigm. *Soft Computing Journal*, 7, 2003.
- [59] Dasgupta, D. and Gonzalez, F. An immunity-based technique to characterize intrusions in computer networks. In Intelligent Security Systems Research Lab Computer Science Division. The University of Memphis, 2002.
- [60] Gonzalez, F., Dasgupta, D., and Kizma, R. An immuno-genetic technique to detect anomalies in network traffic. *GECCO*, pages 1081 - 1088, 2002.
- [61] Harmer, P., Williams, G., Gnusch, and Lamont, G. An artificial immune system architecture for computer security applications. *IEEE Transactions on Evolutionary Computation*, 6(3):252 - 280, June 2002.
- [62] Kim, J., Greensmith, J., Twycross, J., and Aickelin, U. Malicious code execution detection and response immune system inspired by the danger theory. *Adaptive and Resilient Computing Security Workshop*, 2005.
- [63] Dasgupta, D. and Nino, F. Comparison of negative and positive selection algorithms in novel pattern detection. In IEEE International Conference on Systems, Nashville, 2000. *Man and Cybernetics (SMC)*.
- [64] de Castro, L. and Timmis, J. Artificial immune systems: A novel paradigm to pattern recognition. *Artificial Neural Networks in Pattern Recognition*, pages 67 - 84, 2002.
- [65] Ko, A., Lau, H., and Lau, T. An immuno-control framework for decentralized mechatronic control. *ICARIS 2004*, pages 91 - 105, 2004.
- [66] Davis J., Perhinschi M. G., and Moncayo H. Evolutionary algorithm for artificial-immune-system-based failure-detector generation and optimization. *Journal of Guidance, Control and Dynamics*, 33(2), March 2010.

- [67] Perhinschi M. G., Moncayo H., Al Azzawi D., Moguel I., "Generation of Artificial Immune System Antibodies Using Raw Data and Cluster Set Union", *IC: International Journal of Immune Computation*, Vol. 2, No. 1, pp 1-15, March, 2014
- [68] Perhinschi M. G., Moncayo H., Wilburn B., Wilburn J., Karas O., Bartlett A. "Neurally Augmented Immunity-Based Detection and Identification of Aircraft Subsystem Failures", scheduled for publication in the *The Aeronautical Journal*, July, 2014
- [69] Perhinschi M. G., Porter J., Moncayo H., Davis J., Wayne W. S., "Artificial Immune System-Based Detection Scheme for Aircraft Engine Failures", *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5 pp 1423-1440, Sept.-Oct. 2011
- [70] Moncayo H., Perhinschi M. G., Davis J., "Aircraft Failure Detection and Identification Over an Extended Flight Envelope Using an Artificial Immune System", *The Aeronautical Journal*, Vol. 115, No. 1163, Jan. 2011
- [71] Moncayo H., Perhinschi M. G., Davis J., "Aircraft Failure Detection and Identification Using an Immunological Hierarchical Multi-Self Strategy", *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, pp. 302-320, Jul.-Aug. 2010
- [72] Moncayo H., Perhinschi M. G., Wilburn B., Wilburn J., Karas O., "UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity-based Mechanism", *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, August 2012
- [73] Lyons B., Moncayo H., Noriega A., Moguel I., Perhinschi M. G., "Hardware-in-the-Loop Simulation of an Extended Non-linear Dynamic Inversion Augmented with an Immunity-based Adaptive Control System", *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, August 2013
- [74] A. Togayev, M. G. Perhinschi, D. A. Azzawi, H. Moncayo, I. Moguel, and A. Perez, "Immunity-Based Abnormal Condition Accommodation of Aircraft Subsystem Failures", submitted to the American Society of Mechanical Engineers Annual Dynamic Systems and Control Conference, San Antonio, TX, October 2014
- [75] Moguel I., Moncayo H., Perhinschi M.G., Perez A., Azzawi D. A., Togayev A., "Structured Non-Self Approach for Aircraft Failure Identification within an Immunity-based Fault Tolerance Architecture", submitted to the IEEE TAES, Jan. 2014
- [76] Moguel I., Moncayo H., Perhinschi M. G., Perez A. E., Al Azzawi D., Togayev A., "Bio-Inspired Approach for Aircraft Health Assessment and Flight Envelope Estimation", submitted to ASME Annual Dynamic Systems and Control Conference, San Antonio, Texas, Oct. 2014

- [77] Al Azzawi D., Perhinschi M. G., Togayev A., Moncayo H., Moguel I., Perez A. E., "Evaluating Aircraft Abnormal Conditions Using an Artificial Dendritic Cell Mechanism", abstract submitted to International Conference and Exhibition on Mechanical and Aerospace Engineering, Philadelphia, Pennsylvania, Sept. 2014
- [78] Moguel I., Moncayo H., Perhinschi M. G., Al Azzawi D., Perez A. E., Togayev A., "Biologically-Inspired Approach for Aircraft Management under Upset Conditions", abstract submitted to International Conference and Exhibition on Mechanical and Aerospace Engineering, Philadelphia, Pennsylvania, Sept. 2014
- [79] Rowe, G.W. *The Theoretical Models in Biology*. Oxford University Press, Oxford, 1994.
- [80] Weng, L., Xia, M., Hu, K., and Wang, A. Micro aerial vehicle (MAV) flapping motion control using an immune network with different immune factors. *International Journal of Advanced Robotic Systems*, 2013.
- [81] Perez A. E., Moncayo H., Moguel I., Perhinschi M. G., Al Azzawi, D., and Togayev A. Development of immunity-based adaptive control laws for aircraft fault tolerance. submitted to ASME Annual Dynamic Systems and Control Conference, October 2014.
- [82] Perhinschi M. G., Al Azzawi, D., and Moncayo H. Development of immunity-based framework for aircraft abnormal conditions detection, identification, evaluation, and accommodation. *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, August 2013
- [83] Coico, R., Sunshine, G., and Benjamin, E. *Immunology: A Short Course*. Wiley-Blackwell; 6 edition, 2003.
- [84] Charles, J., Travers P., Walport, M., and Shlomchik M. *Immunobiology Fifth Edition*. New York and London: Garland Science, 2008.
- [85] Perhinschi M. G., Napolitano M. R., Campa G., and Fravolini M. L. Integration of fault tolerant system for sensor and actuator failures within the WVU NASA f-15 simulator. In *AIAA Guidance, Navigation, and Control Conference*, Austin, TX, August 2003.
- [86] Moncayo H. *Immunity-Based Detection, Identification, and Evaluation of Aircraft Subsystem Failures*. PhD thesis, West Virginia University, 2009.
- [87] Perhinschi, M. G., Napolitano, M. R., Campa, G., Fravolini M. L., Massotti, L., and Lando, M. Augmentation of a non linear dynamic inversion scheme within the NASA IFCS f-15 WVU simulator. *AIAA Guidance, Navigation, and Control*, 2003.
- [88] Perhinschi, M. G., Napolitano, M. R., and Campa, G. A simulation environment for testing and research of neurally augmented fault tolerant control laws based on non-linear dynamic

inversion. In AIAA Modeling and Simulation Technologies Conference and Exhibit, Providence, Rhode Island, 16-19 August 2004.

[89] Meyer, A. and Van Kampen, H. X-Plane On-Line Instruction Manual. Laminar Research Inc., Columbia SC, 8th edition, 2002.

[90] Perhinschi M. G., Beamer F., “Flight Simulation Environment for Undergraduate Education in Aircraft Health Management”, Computers in Education Journal, Vol. XXII, No. 3, pp. 50-62, July-Sept. 2012

[91] Barr, J. Flightgear takes off. Linux.com, 2006.

[92] <https://www.gnu.org/copyleft/gpl.html>, last access 6/2/2014.

[93] Antoniewicz, R.F., Duke, E.L., and Patterson, B.P. User's manual for interactive linear, a Fortran program to derive linear aircraft models.

APPENDIX

MATLAB code for Matching Algorithm

```
function [y,y4,y5]=
fcn(u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,pilot_err_norm,pilot_contr_up)

err_min=200; index=1;
Normalization of current error
for j=1:5
u1(j)=(u1(j))-min_features(1)/(max_features(1)-min_features(1));
u2(j)=(u2(j))-min_features(2)/(max_features(2)-min_features(2));
u3(j)=(u3(j))-min_features(3)/(max_features(3)-min_features(3));
u4(j)=(u4(j))-min_features(4)/(max_features(4)-min_features(4));
u5(j)=(u5(j))-min_features(5)/(max_features(5)-min_features(5));
u6(j)=(u6(j))-min_features(6)/(max_features(6)-min_features(6));
end

for k=5:6500
difference between current errors and pilot errors
err_diff=abs([(u1-pilot_err_norm(k-4:k,1)); ...
10*(u2-pilot_err_norm(k-4:k,2)); ...
15*(u3-pilot_err_norm(k-4:k,3)); ...
10*(u4-pilot_err_norm(k-4:k,4)); ...
(u5-pilot_err_norm(k-4:k,5)); ...
15*(u6-pilot_err_norm(k-4:k,6))] );

err_diff_q=abs(pilot_err_norm(k-4:k,7)-u7);
err_diff_phi=abs(pilot_err_norm(k-4:k,8)-u8);
err_diff_psi=abs(pilot_err_norm(k-4:k,9)-u9);
err_diff_h=abs(pilot_err_norm(k-4:k,10)-u10);
err_diff_theta=abs(pilot_err_norm(k-4:k,11)-u11);

for i=1:5
err_diff_q(i)=(err_diff_q(i)-min_features(3))/(max_features(3)-
min_features(3));
err_diff_phi(i)=(err_diff_phi(i)-min_features(7))/(max_features(7)-
min_features(7));
err_diff_psi(i)=(err_diff_psi(i)-min_features(8))/(max_features(8)-
min_features(8));
err_diff_h(i)=(err_diff_h(i)-min_features(5))/(max_features(5)-
min_features(5));
err_diff_theta(i)=(err_diff_theta(i)-min_features(6))/(max_features(6)-
min_features(6));

end

sum of errors
ERROR=sum(err_diff)+sum(err_diff_h)+sum(err_diff_q)+10*sum(err_diff_phi)+10*s
um(err_diff_psi)+15*sum(err_diff_theta);
finding minimum error; saving index number
if ERROR<err_min
err_min=ERROR;
index=k;
end
```

```
end
end
y=[pilot_contr_up(index,1) pilot_contr_up(index,2) pilot_contr_up(index,3)
pilot_contr_up(index,4)];
y4=err_min;
y5=index;
```