JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 7, February 2010

Evaluation of a Fault-Tolerant Scheme in a Six-Degree-of-Freedom Motion Flight Simulator

Girish K. Sagoo* , Srikanth Gururajan† , Brad Seanor‡ , Marcello R. Napolitano§ ,
Mario G. Perhinschi¶ , Yu Gu# , and Giampiero Campa**
West Virginia University, Morgantown, West Virginia 26506

DOI: 10.2514/1.42299

This paper discusses the evaluation of a neurally augmented fault-tolerant flight control scheme for a high-performance military aircraft featuring an adaptive actuator and sensor failure detection, isolation, and identification algorithm in a motion-based flight simulator. The design of the fault-tolerant control scheme is based on a nonlinear dynamic inversion scheme with a neural network-based augmentation for reducing the dynamic inversion errors associated with the occurrence of an actuator failure while a set of online learning neural observers is used for dealing with specific sensor failures. The failure detection, isolation, and identification scheme is based on an adaptive threshold technique for estimating failure bounds associated with both actuator and sensor failures. Also, an 'ad-hoc' parameter is proposed here for the novel task of evaluating the pilot workload in compensating for both actuator and sensor failures onboard the aircraft. A general outcome of the effort is a demonstration of the importance of realistic motion-based simulation environments for evaluation of this specific class of flight control laws. The study also demonstrated the importance of the neural augmentation for failure accommodation purposes and the effectiveness of the proposed adaptive threshold technique for failure identification purposes.

Nomenclature

b	bias in a floating limiter bound
n	number of samples for floating limiter calculation
p_{actual}	actual aircraft roll rate
p_{DNN}	roll rate estimate, from P-decentralized neural network
p_{MNN}	roll rate estimate, from main neural network
q_{actual}	actual aircraft pitch rate
q_{DNN}	pitch rate estimate, from Q-decentralized neural network

Received 20 November 2008; accepted for publication 20 October 2009. Copyright © 2009 by Girish K. Sagoo, Srikanth Gururajan, Brad Seanor, Marcello R. Napolitano, Mario G. Perhinschi, Yu Gu, and Giampiero Campa. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 1542-9423/09 $10.00 in correspondence with the CCC.

* Graduate Student, Mechanical and Aerospace Engineering, AIAA Student Member, girishaero@gmail.com
† Post Doctoral Fellow, Mechanical and Aerospace Engineering, AIAA Member, Srikanth.Gururajan@mail.wvu.edu
‡ Research Assistant Professor, Mechanical and Aerospace Engineering, AIAA Member, Brad.Seanor@mail.wvu.edu
§ Professor, Mechanical and Aerospace Engineering, AIAA Member, Marcello.Napolitano@mail.wvu.edu
¶ Assistant Professor, Mechanical and Aerospace Engineering, AIAA Senior Member, Mario.Perhinschi@mail.wvu.edu
# Research Assistant Professor, Mechanical and Aerospace Engineering, AIAA Member, Yu.Gu@mail.wvu.edu
** Research Assistant Professor, Mechanical and Aerospace Engineering, AIAA Member, Giampiero.Campa@mail.wvu.edu

q_{MNN}	pitch rate estimate, from main neural network
r_{actual}	actual aircraft yaw rate
r_{DNN}	Yaw rate estimate, from R-decentralized neural network
r_{MNN}	yaw rate estimate, from main neural network
R_{pq}	pitch roll cross-correlation coefficient
R_{rr}	autocorrelation coefficient in yaw channel
$\bar{X}$	average of signal computed over the time window

Greek

β	multiplicative factor for standard deviation
Δt	time following failure detection
Δt_1	time delay for failure isolation
Δt_2	time delay for failure identification
η	composite parameter
μ	scaling parameter for angular dominance
σ	standard deviation for number of samples evaluated for floating limiter
ω_{pq}	angular rate dominance parameter

Subscripts

FL	floating limiter bound (soft/hard) of the signal
HLB	hard lower bound
HUB	hard upper bound
SLB	soft lower bound
SUB	soft upper bound

I. Introduction

FAULT tolerance, failure detection, isolation, and identification (FDII) is a key research area in the flight control community and has received considerable attention in the last decade [1–3]. The need for a form of a FDII scheme gained visibility in the flight control community following commercial airliner accidents during the 1970s and 1980s, a classic example being the Eastern 1080 flight accident [4] where an L-1011 aircraft experienced an uncommanded elevator deflection to 19 deg. Thanks to 30 plus years of flight experience, the pilot was able to detect, isolate, and identify the failure and was able to take successful corrective actions. However, many other such incidents were not as fortunate. For example, it was concluded in a National Transportation and Safety Board report that the American Airlines DC10 aircraft accident [5] could have been avoided if some form of FDII logic would have assisted the pilot in his initial efforts to regain control of the aircraft. Similarly, the loss of control resulting from the movement of the rudder surface to its deflection limit resulted in the crash of the US Airways flight 427 in Pennsylvania [6]. While the need for FDII schemes to ensure safety of passengers and cargo is clearly evident on manned civilian and military aircraft, the implementation of this technology on unmanned aerial vehicles could also be extremely beneficial through an increase in performance, reliability, and, consequently, lower losses and higher mission success rates. Typically aircraft do not have redundant flight control surfaces and so, it becomes paramount for the pilot or the onboard flight controller to be able to reconfigure the existing healthy control surfaces to help regain equilibrium and to continue on with the mission. Several adaptive control approaches have been applied successfully toward this issue. Within this effort, fault tolerance or reconfiguration is achieved by means of a neurally augmented dynamic inversion-based control scheme [7].

Failures on sensors onboard an aircraft can also lead to catastrophic conditions — especially when the measurements are used in the feedback loop of the flight control laws. Classic examples are the crashes of the NASA X-31 aircraft [8] due to accumulation of ice on pitot static tube, providing false information to the flight control computers and the recent crash of a B-2 bomber [9]. Hardware redundancy is the traditional approach for sensor fault tolerance. However, this redundancy implies additional weight and volume, higher equipment and maintenance costs, and complexity. With higher premiums placed on cost, weight, and/or volume of the avionics payload on aircraft platforms, especially unmanned vehicles, hardware redundancy might not be an effective option. An alternative approach is given by leveraging analytical redundancy. Recent advances in control theory and mathematical modeling have lead to the development of FDII implementations based on analytical redundancy methods. Analytical redundancy approaches using model-based implementations [10,11], knowledge-based implementations [12,13], or a combination of the two [14] have been successful.

The most critical element of any scheme designed for online analysis of failure characteristics onboard an aircraft is the set of thresholds used for detecting, identifying, and isolating a failure. Using information represented by a system's mathematical model [15], sensor failures can be detected by setting thresholds on the residuals generated from the difference between actual measurements and estimates of the measurements from the mathematical model. Conventionally, fixed thresholds have been used for this purpose; however, it presents significant drawbacks since fixed thresholds might not adequately reflect changing flight conditions. The design of a threshold scheduling scheme could overcome this problem; however, this would entail a long and extensive statistical analysis of various flight regimes and all potential failure scenarios, making the approach fairly unrealistic, especially for dealing with sensor failures.

While fixed threshold approaches have been used for actuator [16] and integrated [17,18] sensor/actuator failures, Perhinschi et al. [19] have extended the actuator failure FDII with an adaptive threshold, resulting in a lower rate of false alarms and a relatively simpler logic. A similar effort for sensor failure [13] was performed using an adaptive threshold concept.

Within this effort, the performance of a fault-tolerant flight control scheme (FTFCS) with neural augmentation, based on the intelligent flight control system (IFCS) Gen-2 architecture is evaluated with an implementation of a floating limiter-based FDII scheme. The paper is organized as follows. Section II describes the architecture of the FTFCS, followed by a description of the adaptive FDII implementation. Section III describes the experimental setup and in Sec. IV, the results from the actuator and sensor failure experiments are discussed, followed by the conclusions in Sec. V.

II. FTFCS and FDII Algorithm for Actuator/Sensor Failures

The generic high-level architecture of the neurally augmented dynamic inversion-based control laws along with the modeling of the failure is shown in Fig. 1.

The aircraft dynamics is based on a Fortran code for the simulation of a high-performance military aircraft, distributed by NASA to academic institutions within the 1990 AIAA GNC Design Challenge [20]. The aerodynamic and thrust characteristics are provided through 42 look-up tables, with 16 tables for the longitudinal dynamics as functions of Mach number, angle of attack, and stabilator deflection; 20 tables for the lateral dynamics as functions

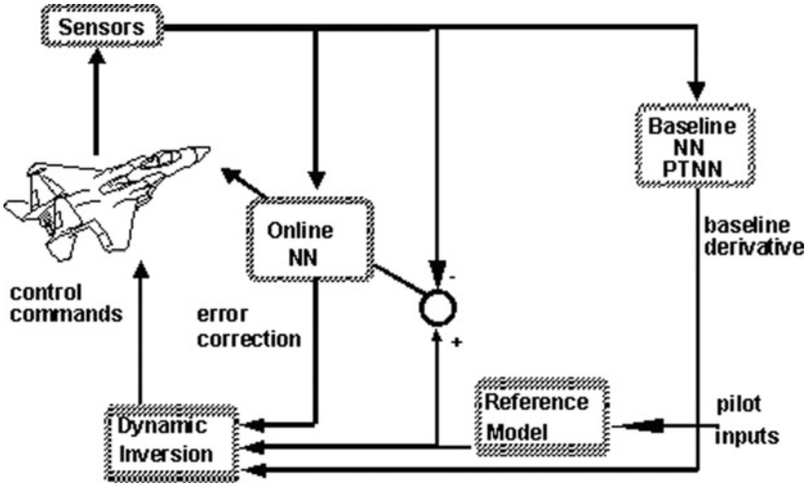


Fig. 1 IFCS Gen -2 architecture.

of Mach number, angle of attack, sideslip angle, and rudder deflection; 2 tables for engine thrust and fuel flow as functions of Mach number and altitude. The look-up tables have further been subdivided to isolate the contribution of individual control surfaces in order to be able to simulate control surface failures [21]. The scheme also features a set of pre-trained neural networks (PTNNs) to provide updated values of the aircraft aerodynamic coefficients once every 10 s, as it flies through different regions of its flight envelope. For all practical purposes, the PTNNs are equivalent to the look-up tables for the aerodynamic stability and control derivatives. The FDII scheme is exposed to the implicit change in the aerodynamic derivatives as the PTNNs are updating the derivatives used in the nonlinear dynamic inversion (NLDI). This is an additional feature/improvement of the control laws meant to reduce the "burden" on the neural networks (NN) that compensate for the inaccuracies of the NLDI. It is not expected that the behavior of the FDII is changed when the derivatives in the NLDI are updated by the PTNNs because compensation is provided anyway by the NN (working harder but still generating the required compensation). The only difference that occurs is in the outputs of the NNs and these are not used in the detection scheme.

The desired handling qualities are defined through the use of first- and second-order reference models of a high-performance fighter aircraft [22]. The reference model is then used to generate reference pitch, roll and yaw angular rate commands based on pilot stick inputs, followed by a NLDI to generate control surface commands (δ_a, δ_e, δ_r). Under nominal flight conditions, NLDI has inherent errors stemming from inaccurate modeling of the aircraft. The dynamic inversion errors associated with the NLDI scheme could be large due to modeling discrepancies. These discrepancies can be due to "conventional" modeling errors or be a result of modeling errors associated with failures and/or malfunctions of the control surfaces [23]. Regardless of the origin, the dynamic inversion error is minimized through the neural augmentation of the NLDI scheme.

On the basis of earlier experiments on '*Gen-2*' schemes, the schemes augmented with the EMRAN NN [16–18, 24, 25] demonstrated better results in comparison to schemes augmented with the Sigma-Pi and single hidden layer (SHL) neural networks and is selected to provide the neural augmentation for both the actuator failure and sensor failure scenarios in all the tests conducted for this effort. The extended minimal resource allocating neural network (EMRAN) algorithm is a more powerful version of the standard MRAN and its main architecture features growing and pruning mechanisms; moreover, the parameters are updated following a 'winner takes all' [26] strategy. The extended algorithm allows only the parameters of the most activated neurons to be updated, while all the others remain unchanged. In effect, the algorithm allocates resources (neurons) in order to decrease the estimation error in regions of the state space where the mapping accuracy is poor. This strategy results in a significant reduction of the number of parameters that need to be updated online, reducing the computational burden and thus making it particularly suitable for online applications.

An important task in the design of a FDI or FDII scheme is the selection of thresholds that trigger the logic. The resulting selection typically reflects a compromise between maximizing the detection capabilities while minimizing the false alarm rate. While a number of FDII schemes featuring "fixed thresholds" have shown capabilities in correctly detecting and identifying actuator and sensor failures [16,17] in desktop computer simulations, they tend to be very sensitive to false alarms and/or incorrect identifications in the presence of a pilot-in-the loop. West Virginia University (WVU) researchers [19] have introduced an adaptive threshold technique based on the concept of a "*Floating Limiter*" (FL); this scheme eliminates the need for scheduling the failure thresholds with varying flight conditions. This scheme was tested using the simulation code developed in house, as part of the WVU participation in NASA's IFCS program [11,18,19,23].

The FDII implementation in the overall scheme is based on two critical components, the main neural network (MNN) and a set of three decentralized neural networks (DNN) designed to learn/mimic the aircraft angular rates (p, q, r) from the aircraft states. For both the MNNs and the DNNs, a combination of an ADALINE [27] and an EMRAN network (A+EMRAN) working in parallel has been implemented and provides desirable performance in the presence of large nonlinearities without the computational burden on operation in areas without nonlinearities. The outputs of the MNN at any time instant "k" are estimates of the angular rates (p_{MNN}, q_{MNN}, r_{MNN}) using aircraft state data from time $(k - n)$ to $(k - 1)$, while each of the DNNs, designated as P, Q, and R-DNN, are designed to estimate the outputs of each of the three gyros, respectively. This approach is based on the inherent observability property of the aircraft dynamic system. An important distinction between the MNN and the DNNs is in the set of parameters that are used as inputs for learning. In fact, while the MNN uses p, q, r measurements from previous time steps to generate a current estimate of the angular rates, the DNNs (P, Q, and R-DNN) exclude from their input set, the measurements from the respective gyro they are designed to replicate. For example, the MNN inputs includes, among other parameters, the last "n" samples of sideslip angle (β), roll rate (p), yaw rate (r), aileron deflection (δ_A), and rudder deflection (δ_R) while the set of inputs to the P-DNN is the same, excluding the roll rate measurements, thus keeping the estimates from the P-DNN from being corrupted with the values from the failed sensor.

The estimates of the angular rates from the MNN are compared with the actual measurements (p_{actual}, q_{actual}, r_{actual}) at time instant "k" to define an error parameter; the main quadratic estimation error (MQEE) as follows:

$$\text{MQEE} = \tfrac{1}{2}[(p_{\text{actual}} - p_{\text{MNN}})^2 + (q_{\text{actual}} - q_{\text{MNN}})^2 + (r_{\text{actual}} - r_{\text{MNN}})^2] \tag{1}$$

Similarly, another error parameter, the output quadratic estimation error (OQEE) is obtained by comparing the angular rate estimates from the MNN and DNNs as follows:

$$\text{OQEE} = \tfrac{1}{2}[(p_{\text{DNN}} - p_{\text{MNN}})^2 + (q_{\text{DNN}} - q_{\text{MNN}})^2 + (r_{\text{DNN}} - r_{\text{MNN}})^2] \tag{2}$$

At nominal flight conditions, the outputs from the MNN and the DNNs closely follow the actual measurements from the P, Q, and R gyros; consequently, the error parameters MQEE and OQEE remain small. In the event of a failure on any of the three gyros (sensor failure), the faulty information continues to be used in the learning process of the MNN. This leads to a significant difference between the angular rate estimates from the MNN and the corresponding estimates from the DNNs as the learning of the MNN is 'contaminated' by faulty measurements from the gyros, while the DNNs are not. This facilitates the use of the parameters MQEE and OQEE during the failure detection phase. Another error parameter, the DNN quadratic estimation error ($\text{DQEE}_x(x = p, q, r)$) — defined as the difference between the sensor measurements and the angular rate estimate from the corresponding DNN — is used for sensor identification purposes.

$$\text{DQEE}_x = \tfrac{1}{2}[(X_{\text{actual}} - X_{\text{DNN}})^2] \tag{3}$$

In the event of actuator failures, especially failures on the elevator/stabilator or ailerons, a coupling between longitudinal and lateral directional dynamics is reflected in a cross-correlation between the roll and pitch rates of the aircraft, represented as R_{pq}. Additionally, an autocorrelation function of the yaw rate R_{rr} can be used for the specific task of isolating rudder failures Finally; an 'ad-hoc' defined angular dominance parameter ω_{pq} can be introduced and used for differentiating between stabilator and aileron failures. This parameter is defined as follows:

$$\omega_{pq} = |p| - \mu|q| \tag{4}$$

where "μ" is a scaling factor. Simulation studies [28] have consistently shown that the angular dominance parameter is significantly higher for an aileron failure than a stabilator failure as there is little change in pitch rate due to an aileron failure, while there is a significant change in roll rate due to a stabilator failure. After running a number of simulations, a value of the scaling factor μ in the range [1.5, 3.0] was found to be sensitive enough to distinguish between a stabilator failure and an aileron failure with failure magnitude of even less than 1 deg.

The FDII logic, shown in Fig. 2, is based on an adaptive autoregressive moving average (ARMA) filter, is termed as a FL, and is used to assign bounds to the monitored parameters (MQEE, OQEE, R_{pq}, R_{rr}). The bounds assigned

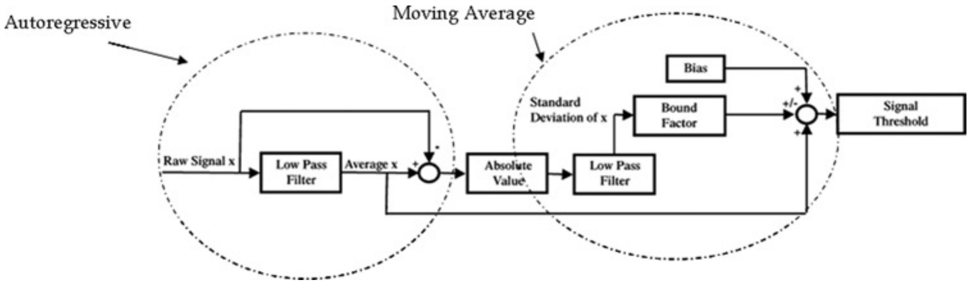


Fig. 2 ARMA filter used for FDII formulation.

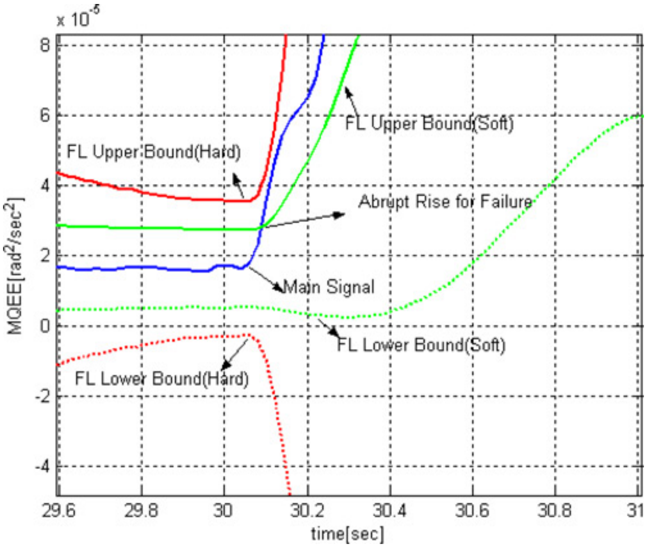


Fig. 3 Time history of the MQEE FL following a failure.

by the FL to each of the parameter drifts dynamically (floats over signal) at the rate of the signal but less than a pre-imposed limit, thus deriving its name. These bounds are a function of the autoregressive part that computes the mean of the signal $\bar{X}$ and the moving average part that computes standard deviation (σ) of the signal over a time window of n time steps. The bounds are further classified into "*Soft*" and "*Hard*" bounds as defined below, based on different values of the bias and the multiplicative factor, in order to distinguish between soft and hard failures on either the sensors or the actuators.

$$X_{\mathrm{HUB}} = \bar{X} + \beta_{\mathrm{HUB}} \cdot \sigma + b_{\mathrm{HUB}} \tag{5}$$

$$X_{\mathrm{HLB}} = \bar{X} - \beta_{\mathrm{HLB}} \cdot \sigma - b_{\mathrm{HLB}} \tag{6}$$

$$X_{\mathrm{SUB}} = \bar{X} + \beta_{\mathrm{SUB}} \cdot \sigma + b_{\mathrm{SUB}} \tag{7}$$

$$X_{\mathrm{SLB}} = \bar{X} - \beta_{\mathrm{SLB}} \cdot \sigma - b_{\mathrm{SLB}} \tag{8}$$

Under nominal conditions, a typical FL, as shown in Fig. 3, will move at the rate of the signal at the set bias "*b*". However, following a failure (either on actuators or sensors), a perturbation arises in the signal being monitored, which results in the signal exceeding the corresponding FL and, consequently, triggering a flag indicating a possible failure. Typically, for failures of lower magnitudes or a low rate, the signal crosses just the "soft" bounds and a persistence counter is then used to trigger subsequent detection or identification flags. On the other hand, typical failures of large magnitude or high rate breaches even the "hard" bound, triggering detection or identification flags.

The FDII logic is divided into three distinct phases, that is the detection phase, the isolation phase, and the identification phase, as explained below. A conceptual block diagram of the overall scheme is shown in Fig. 4.

A. Detection Phase

A failure on an actuator or a sensor leads to discrepancies between the measured angular rates and the corresponding NN estimates. Specifically, the failure will cause large values of MQEE and/or larger values of OQEE. Therefore, the condition for which a control surface failure is declared when the signal crosses the FL bounds is shown in Eq. (9)

$$\mathrm{MQEE} \geqslant \mathrm{MQEE}_{\mathrm{FL}} \quad \text{or} \quad \mathrm{OQEE} \geqslant \mathrm{OQEE}_{\mathrm{FL}} \tag{9}$$

where subscript "$_{\mathrm{FL}}$" signifies the associated soft or hard bound for the FL signal.

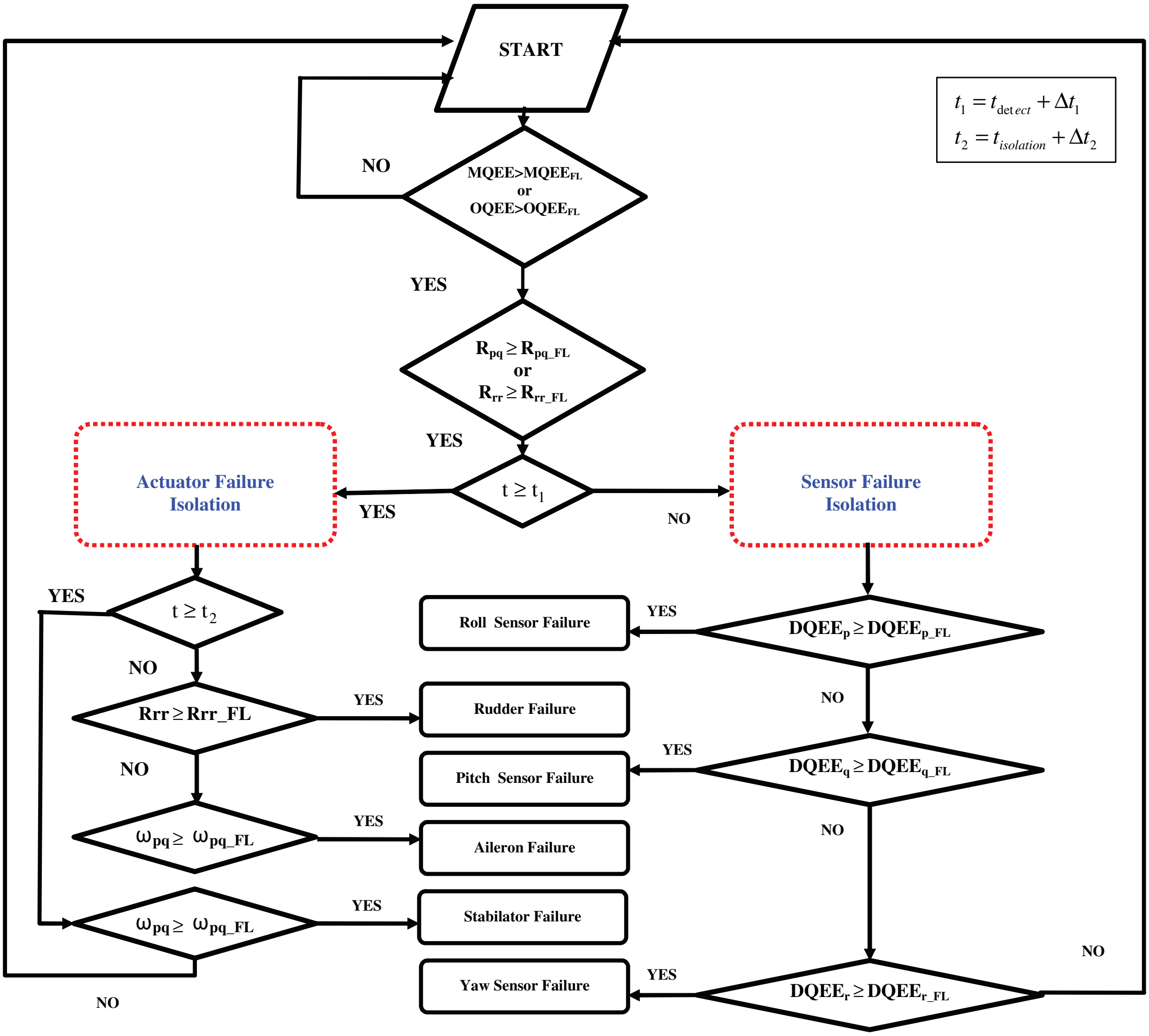

START
$t_1 = t_{detect} + \Delta t_1$
$t_2 = t_{isolation} + \Delta t_2$
NO
MQEE>MQEE$_{FL}$ or OQEE>OQEE$_{FL}$
YES
$R_{pq} \geq R_{pq\_FL}$ or $R_{rr} \geq R_{rr\_FL}$
YES
$t \geq t_1$
YES
NO
Actuator Failure Isolation
Sensor Failure Isolation
YES
$t \geq t_2$
NO
$Rrr \geq Rrr\_FL$
YES
NO
$\omega_{pq} \geq \omega_{pq\_FL}$
YES
$\omega_{pq} \geq \omega_{pq\_FL}$
YES
NO
Roll Sensor Failure
Rudder Failure
Pitch Sensor Failure
Aileron Failure
Stabilator Failure
Yaw Sensor Failure
YES
$DQEE_p \geq DQEE_{p\_FL}$
NO
YES
$DQEE_q \geq DQEE_{q\_FL}$
NO
NO
YES
$DQEE_r \geq DQEE_{r\_FL}$

Fig. 4 Logical diagram of adaptive FDII scheme.

B. Isolation Phase

Following successful failure detection, the next step is the classification of the failure as an actuator or a sensor failure. Since a substantial cross coupling of aircraft dynamics is associated with an actuator failure, the occurrence of an actuator failure can be differentiated or isolated from the sensor failure scenario through the analysis of the cross-correlation between the longitudinal and lateral-directional dynamics [16,17]. Thus, conditions for actuator/sensor failure isolation can be represented through the following Boolean relationships as follows:

$$R_{pq} \geqslant R_{pq\_\mathrm{FL}} \quad \text{or} \quad R_{rr} \geqslant R_{rr\_\mathrm{FL}} \text{ and } \Delta t \geqslant \Delta t_1 \Rightarrow \text{Actuator Failure} \tag{10}$$

$$R_{pq} \leqslant R_{pq\_\mathrm{FL}} \quad \text{or} \quad R_{rr} \leqslant R_{rr\_FL} \text{ and } \Delta t \geqslant \Delta t_1 \Rightarrow \text{Sensor Failure} \tag{11}$$

C. Identification for Actuator Failure Isolated

Following the isolation phase, the next step is to identify the failure as stabilator, aileron, or rudder failure—in the case of actuator failures—or a pitch, roll, or yaw gyro failure—in the case of sensor failures. Simulation analysis has consistently shown that an aileron failure induces a higher roll rate than pitch rate and leads to higher values of angular dominance ω_{pq} than stabilator failures. On the other hand, rudder actuator failures are marked by a noticeable change in the autocorrelation R_{rr} almost instantaneously and can therefore be used to identify a rudder failure from an aileron or stabilator failures. These are mathematically represented as follows:

$$\omega_{pq} \geqslant \omega_{pq\_\mathrm{FL}} \text{ and } \Delta t \geqslant \Delta t_1 \Rightarrow \text{Aileron Failure} \tag{12}$$

$$\omega_{pq} \geqslant \omega_{pq\_\mathrm{FL}} \text{ and } \Delta t \geqslant \Delta t_2 \Rightarrow \text{Stabilator Failure} \tag{13}$$

$$R_{rr} \geqslant R_{rr\_\mathrm{FL}} \Rightarrow \text{Rudder Failure} \tag{14}$$

D. Identification for Sensor Failure Isolated

In the event of sensor failures, the MNN continues its learning process, including the data from the faulty gyro; on the other hand, the DNNs still estimate angular rates based on the actual and true dynamic of the aircraft. This difference triggers a flag leading to the identification of the specific gyro with the failure. In case of a failure the DQEEs cross their respective FL bounds and issues an identification flag which are given below:

$$\mathrm{DQEE}_p \geqslant \mathrm{DQEE}_{p\_\mathrm{FL}} \Rightarrow \text{Roll Sensor Failure} \tag{15}$$

$$\mathrm{DQEE}_q \geqslant \mathrm{DQEE}_{q\_\mathrm{FL}} \Rightarrow \text{Pitch Sensor Failure} \tag{16}$$

$$\mathrm{DQEE}_r \geqslant \mathrm{DQEE}_{r\_\mathrm{FL}} \Rightarrow \text{Yaw Sensor Failure} \tag{17}$$

III. Experimental Setup

The MOTUS 622i Flight Simulator (MFS) at WVU is a 6-Degree-of-Freedom electromechanical actuator-based motion simulator, custom-developed and commercialized by fidelity flight simulation. The simulator, shown in Fig. 5, is extensively used to support flight simulation and control courses at the undergraduate and graduate levels [26], as well as a number of research activities.

The simulator includes the following main components:

1) 6-DoF motion platform and cockpit driven by electrical induction motors;
2) Laminar Research X-Plane flight simulation software;
3) LCD mosaic wall for external visual display.

Within the cockpit, visual information is provided to the pilot by means of 6 LCD panels, with two displays hosting the instrument clusters and four others serving as the windshield and windows of the airplane. The graphics is handled by '*X-Plane*', a commercially available flight simulation package. For this effort, the original cockpit

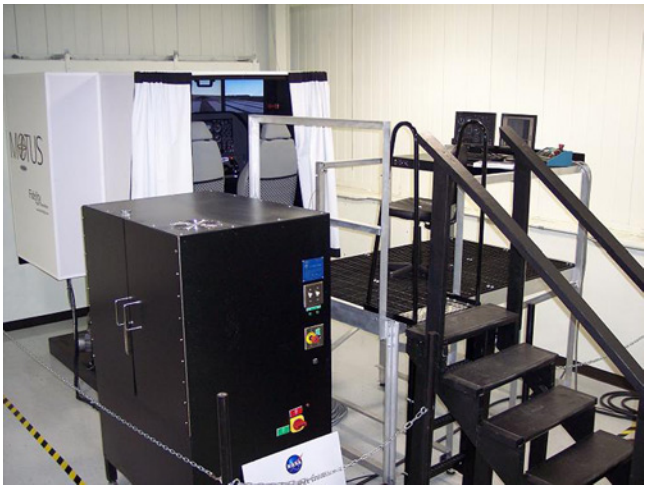


Fig. 5 WVU 6-DoF flight simulator system.

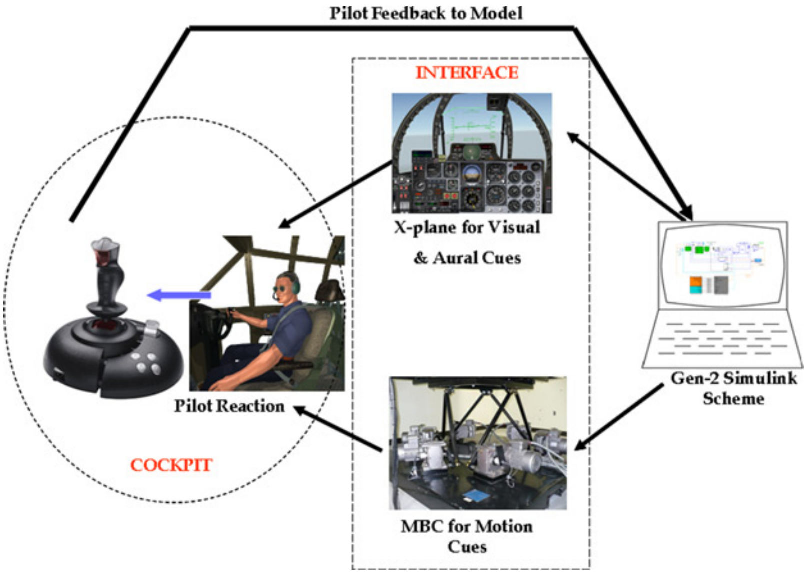


Fig. 6 Experimental setup of the Simulink Gen-2 interface with MFS.

configuration is replaced with the graphics-based cockpit of a fighter aircraft, along with a joystick. The overall schematic of the experimental setup is shown in Fig. 6.

In addition, the cockpit display is modified to provide some rudimentary FDII diagnostics in the form of brief text warnings to the pilot. The motion cues of the simulator are driven by the acceleration commands from the Matlab/Simulink-based flight control scheme, while the spatial cues were driven by the orientation and geographical coordinates of the aircraft in *X-Plane*. The Matlab/Simulink implementation is interfaced with the simulator using user datagram protocol (UDP) sockets. This provides a realistic simulation capability with visual and motion cues to the pilot in the cockpit, while running FTFCS and FDII algorithms.

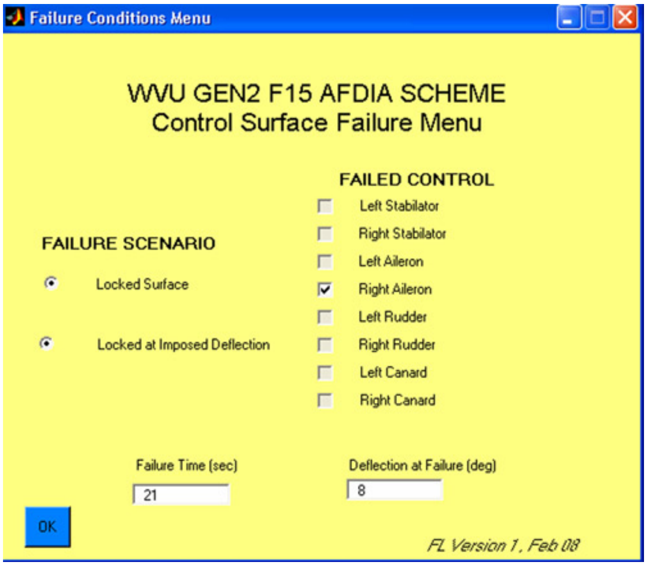
Within the scope of this effort, actuator failures are introduced in the form of stuck positions of the Left Stabilator and Left Aileron. Furthermore, such failures are characterized as "*Soft*", "*Medium*", and "*Hard*" failures, based on the control surface deflections at failure (2, 5, or 8 deg) representing typical actuator failures without any loss of generality. Rudder failures were not selected because of a lack of alternative control mechanisms over the directional dynamics. This is where we believe that fault-tolerant schemes coupled with thrust vectoring mechanisms will play a critical role in next generation high-performance aircraft. Sensor failures, on the other hand, are introduced on the angular rate gyros (pitch, roll, and yaw), assumed to be without hardware redundancy. However, this can be extended to cope with failures of any sensor whose measurements are used for feedback purposes. The sensor failures are simulated by inducing a drifting ramp bias categorized as '*Small*' (4 deg) and '*Large*' (8 deg) bias with different drifting speeds or slope of the ramp and categorized as '*Slow*' (8 s), '*Medium*' (5 s), and '*Fast*' (2 s). The specifications of the actuator and sensor failure scenarios are listed below in Table 1.

Prior to conducting the experiments, the pilot was given some stick time in the simulator to familiarize himself with the aircraft. During the experiments, sensor and actuator failures were introduced in a random fashion, in order to reduce any effects of pilot learning/familiarity with the failures. To ensure an unbiased response, the pilot was not briefed about the failure type (actuator/sensor) or its nature (hard/soft, small/large, or slow/medium/fast) prior to each simulation and was introduced randomly. The simulation scenarios including the total time of simulation, the time of failure as well as the nature and characteristics of the failure were set using custom developed Matlab/Simulink GUIs shown in Figs. 7 and 8.

Once the experiment was set up and the simulation initiated, the pilot was given a free reign to fly the aircraft until a failure occurred. Once a failure was detected, identified, and isolated, pilot was instructed to maintain a wings level flight within 200 ft of the original altitude as a mission profile. The results from the actuator failure and sensor failure experiments are discussed in the following section.

Table 1 Actuator and sensor failure scenarios

	Failure condition	Magnitude
L. Stab./R. Ail.	Hard actuator	+8 deg bias
L. Stab./R. Ail.	Medium actuator	+5 deg bias
L. Stab./R. Ail.	Soft actuator	+2 deg bias
p, q, or r gyro	Small, fast drifting bias (SF#1)	+4 deg bias (ramp in 2 s)
p, q, or r gyro	Large, fast drifting bias (SF#2)	+8 deg bias (ramp in 2 s)
p, q, or r gyro	Small, medium drifting bias (SF#3)	+4 deg bias (ramp in 5 s)
p, q, or r gyro	Large, medium drifting bias (SF#4)	+8 deg bias (ramp in 5 s)
p, q, or r gyro	Small, slow drifting bias (SF#5)	+4 deg bias (ramp in 8 s)
p, q, or r gyro	Large, slow drifting bias (SF#6)	+8 deg bias (ramp in 8 s)



Failure Conditions Menu

WVU GEN2 F15 AFDIA SCHEME
Control Surface Failure Menu

FAILURE SCENARIO

● Locked Surface

● Locked at Imposed Deflection

FAILED CONTROL

□ Left Stabilator
□ Right Stabilator
□ Left Aileron
☑ Right Aileron
□ Left Rudder
□ Right Rudder
□ Left Canard
□ Right Canard

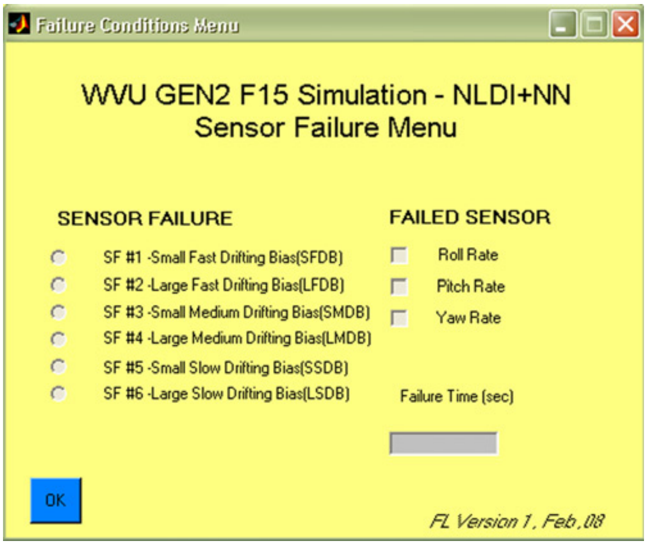
Failure Time (sec): 21

Deflection at Failure (deg): 8

OK

FL Version 1, Feb 08

Fig. 7 GUI for actuator failure scenario.



Failure Conditions Menu

WVU GEN2 F15 Simulation - NLDI+NN
Sensor Failure Menu

SENSOR FAILURE

○ SF #1 -Small Fast Drifting Bias(SFDB)
○ SF #2 -Large Fast Drifting Bias(LFDB)
○ SF #3 -Small Medium Drifting Bias(SMDB)
○ SF #4 -Large Medium Drifting Bias(LMDB)
○ SF #5 -Small Slow Drifting Bias(SSDB)
○ SF #6 -Large Slow Drifting Bias(LSDB)

FAILED SENSOR

□ Roll Rate
□ Pitch Rate
□ Yaw Rate

Failure Time (sec)

OK

FL Version 1, Feb,08

Fig. 8 GUI for sensor failure scenario.

IV. Results and Discussions

The objectives of the experiments performed in this effort are twofold. The first objective is the evaluation of the performance of a neurally augmented, dynamic inversion-based FTFCS with a pilot in the loop in a motion-based simulator The second objective is the evaluation of the effectiveness of an adaptive FDII implementation for the detection, identification, and isolation of sensor and actuator failures.

A parameter, the tracking error (TE), defined as the difference between the angular rates generated by the reference model and the actual aircraft angular rates is used for the purpose of evaluation of the performance of the controller.

$$TE_x = \mathrm{ref}_x - \mathrm{act}_x, \quad x = p, q, r \tag{18}$$

An actuator failure introduces a coupling between the lateral and longitudinal dynamics and a strong cross correlation between the aircraft angular rates. The coupling effects are also reflected as motion and visual cues to the pilot in the cockpit and thus significantly influence the pilot reaction/workload under postfailure conditions. The TE statistics along the pitch and roll channels, being dominant in the case of longitudinal and lateral failures, are analyzed over a 20-s time window around the instant of failure injection in the following sections.

For actuator failure scenarios, baseline results are established with the *Gen-2* FTFCS, without any NN augmentation to compensate for dynamic inversion errors. The same experiments are then repeated with augmentation using the EMRAN network and the resulting performance compared for various test conditions.

A. Analysis of Actuator Failures

Failures on actuators are limited to the left stabilator and right aileron, without any loss of generality and were generated by simulating "stuck positions" at user defined deflections and time instants. Each category of tests in Table 1 for the actuator failure are repeated five times for each surface with the pilot instructed to execute the same mission profile and the TE statistics averaged for performance analysis.

As can be seen from the mean and standard deviation of TE from five runs of left stabilator failure shown in Table 2, for a soft failure (2 deg), the mean of TE along pitch channel for EMRAN NN is 48% lower than for a no NN scenario. For the roll channel, it is 56% less than a no NN case. Along the yaw channel, there is however an increase of 22% in the TE compared with a no NN case, but this could be attributed to the relaxed constraints on maintaining the heading, following a failure. Similarly, in case of the medium stabilator failure (5 deg), the mean of TE shows a decrease of 52% along pitch and 61% along roll channel. The standard deviation (SD) from desired pitch rate is also lower in all test cases and thus an indication of better TE performance with NN augmentation.

The performance of the adaptive FDII scheme in detecting, identifying and isolating the actuator failure is shown in Fig. 9. For a soft failure on the left stabilator injected at $t = 30$ s, the MQEE parameter rises and crosses the soft upper bound at $t = 30.1$ s, triggering a failure detection flag. Once this flag is triggered, an actuator failure is identified when the parameter R_{pq} crosses its soft upper bound at $t = 30.15$ s. By monitoring the angular dominance parameter ω_{pq}, the failure can be isolated as either a stabilator failure or aileron failure. Since the angular dominance parameter in this case did not cross its bounds in the time window of consideration, the failure is isolated as a Stabilator Failure.

Similar to the stabilator failure scenario, a failure on the right aileron was introduced at $t = 30$ s into the simulation. The average of tracking error statistics for five runs of the right aileron failure scenarios is shown in Table 3.

For a soft failure, the mean roll channel TE with EMRAN augmentation is 64% lower than no NN case. For a medium failure, it is 51% lower, and for a hard failure it is 78% lower than without neural augmentation. It can also be observed that the mean TE along pitch channel is better for a soft failure, but increases with EMRAN for the medium and hard failures and could be attributed to the severity of the failure (medium and hard actuator failures), the flight condition and the pilot compensation (PC). The results of the adaptive FDII implementation for a soft failure on the right aileron are illustrated in Fig. 10.

For a Soft failure on the right aileron injected at $t = 30$ s, the MQEE parameter rises and crosses the soft upper bound at $t = 30.15$ s, triggering a failure detection flag. An actuator failure is declared when the cross-correlation parameter R_{pq} breaches its soft upper bound at $t = 30.2$ s. The angular dominance parameter ω_{pq}, clearly indicating an aileron failure when it crosses its bounds in the time window of consideration.

Table 2 Left stabilator failure tracking error statistics (averaged for 5 runs)

Tracking error	2 deg (soft)			5 deg (medium)			8 deg (hard)		
statistics	No NN	EMRAN	% Red	No NN	EMRAN	% Red	No NN	EMRAN	% Red
Mean roll	−1.8765E−01	−8.3327E−02	55.59	−4.4961E−01	−1.7658E−01	60.73	−4.4961E−01	−8.3327E−02	81.47
Mean pitch	7.7981E−02	4.0636E−02	47.89	1.8908E−01	9.2166E−02	51.26	3.3951E−01	4.0636E−02	88.03
Mean yaw	−1.4928E−02	−1.8260E−02	−22.32	−3.5871E−02	−3.7956E−02	−5.81	−6.5450E−02	−1.8260E−02	72.10
SD roll	7.6755E−01	7.8678E−01	−2.51	1.6278E+00	1.6485E+00	−1.27	3.0096E+00	7.8678E−01	73.86
SD pitch	3.5167E−01	3.2159E−01	8.56	7.0072E−01	6.6465E−01	5.15	1.2526E+00	3.2159E−01	74.33
SD yaw	1.6941E−01	1.6916E−01	0.15	2.0100E−01	2.0857E−01	−3.77	2.8094E−01	1.6916E−01	39.79

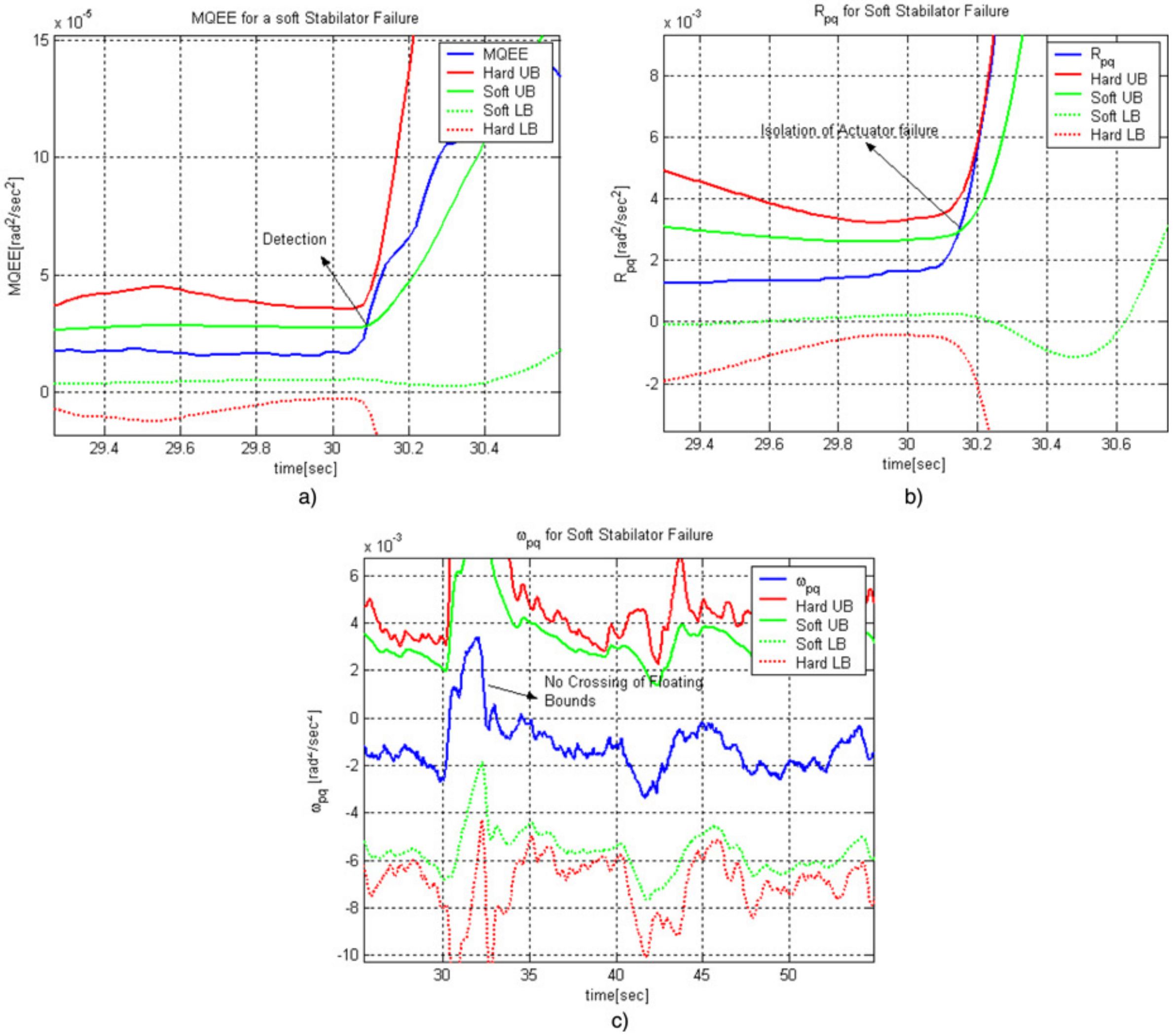


Fig. 9 FDII for stabilator failure: a) detection; b) isolation; and c) identification. UB and LB stand for Upper bound and Lower bound, respectively.

B. Analysis of Sensor Failures

Sensor failures are injected on the pitch, roll, and yaw rate gyros as listed in Table 1 and the experiments repeated five times for each scenario. A particular objective of sensor failure tests was to evaluate the effectiveness of the FDII implementation with a pilot in the loop. As with the actuator failure scenario, the scheme is tested with EMRAN augmentation, with failure injected at $t = 30$ s and the results for the pitch rate sensor failures in terms of average detection, isolation, and identification times from five runs are listed in Table 4 and the corresponding plots shown in Figs. 11–13.

In the case of the sensor failures, the parameter OQEE crosses its threshold, triggering a detection and is shown in Fig. 9. The plots are zoomed in near the failure time to show the time at which detection occurs.

The variation of the cross-correlation parameter R_{pq} for the pitch sensor failure cases is shown in Fig. 10 and it can be seen that FL bounds are not crossed for nearly 5 s after the failure is detected. The FDII algorithm subsequently triggers the flag for the sensor failure isolation and the logic switches to sensor failure identification by monitoring the DQEE. It can be seen from Fig. 11 that the parameter DQEE exceeds the FL bounds for the pitch channel,

Table 3 Right aileron failure tracking error statistics (averaged for five runs)

Tracking error	2 deg (soft)			5 deg (medium)			8 deg (hard)		
statistics	No NN	EMRAN	% Red	No NN	EMRAN	% Red	No NN	EMRAN	% Red
Mean roll	1.4506E−01	5.1887E−02	64.23	2.3700E−01	1.1722E−01	50.54	2.3700E−01	5.1887E−02	78.11
Mean pitch	−3.7471E−03	−2.8545E−03	23.82	−1.8024E−03	−2.1825E−03	−21.09	−2.5710E−03	−2.8545E−03	−11.02
Mean yaw	−5.0966E−04	−5.4564E−04	−7.06	−9.1948E−04	−1.6438E−03	−78.78	−1.4072E−03	−5.4564E−04	61.22
SD roll	7.8103E−01	7.6636E−01	1.88	1.2707E+00	1.4928E+00	−17.48	1.9352E+00	7.6636E−01	60.40
SD pitch	1.4634E−01	1.4626E−01	0.05	1.4678E−01	1.4632E−01	0.31	1.4852E−01	1.4626E−01	1.52
SD yaw	1.5544E−01	1.5460E−01	0.54	1.5529E−01	1.5454E−01	0.48	1.5638E−01	1.5460E−01	1.14

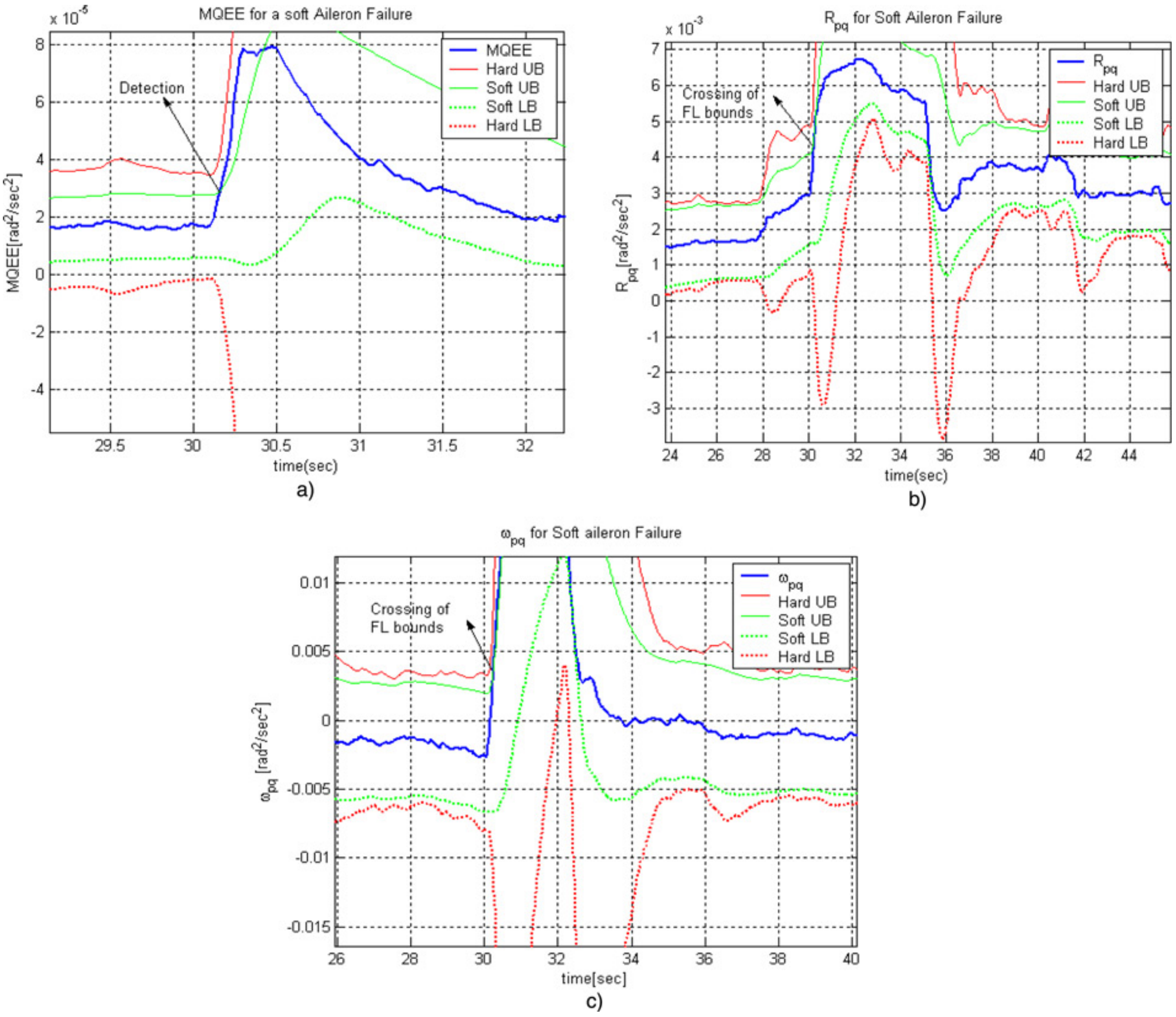


Fig. 10 FDII for aileron failure: a) detection; b) isolation; and c) identification.

Table 4 FDII results for pitch rate sensor failure

S.No.	Failure cases	Detection	Isolation	Identification
1	Pitch sensor SFDB	30.34	31.34	31.36
2	Pitch sensor LFDB	30.344	31.34	31.36
3	Pitch sensor SMDB	30.36	31.36	45.428
4	Pitch sensor LMDB	30.344	31.36	31.376
5	Pitch sensor SSDB	30.36	31.36	44.74
6	Pitch sensor LSDB	30.36	31.36	45.66

but does not exceed the bounds for both the roll channel and the yaw channel, thus providing the sensor failure identification.

The overall results of the FDII for both the actuator failure scenarios as well as the sensor failure scenarios are presented in Table 5. It can be observed that the adaptive FDII algorithm is very effective in detecting failures, with only one missed detection for actuator failure scenarios and three missed detections for sensor failure.

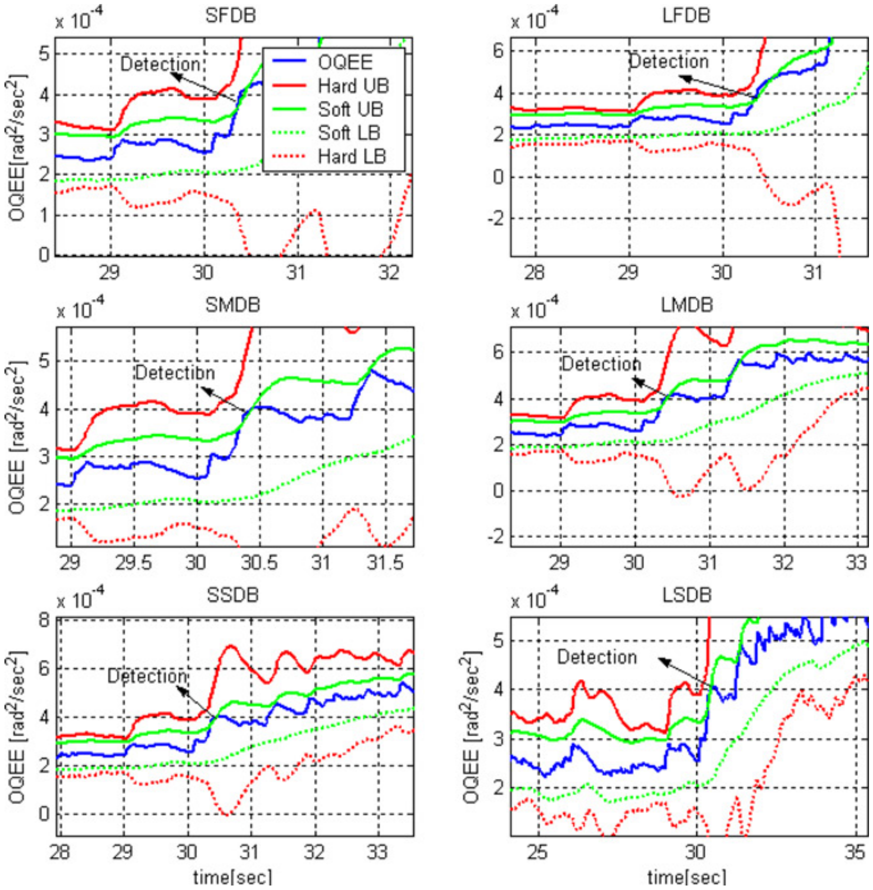


Fig. 11 OQEE for the pitch sensor failure (zoomed to show detections).

C. Pilot Workload

An additional result from the experiments was the evaluation of the amount of pilot compensation (PC), required to recover from actuator failure scenarios and maintain the predefined mission profile. Similar to the TE statistics, this is calculated over a 20-s time window (tw) around the instant of failure injection, along the pitch, roll, or yaw channel and is defined as the sum of the absolute values of total stick activity along each channel, similar to an approach by Gundy-Burlet et al. [29–31].

$$\mathrm{PC}_{\mathrm{Pitch,Roll,Yaw}} = \sum_{k=1}^{\mathrm{tw}} \left|\mathrm{Stick\ Activity}_{\mathrm{Pitch,Roll,Yaw}}\right| \tag{19}$$

The averaged value of five runs of the stabilator failure is shown in Table 6. For a soft failure on the stabilator, neural augmentation of the flight control scheme resulted in a 48% reduction in pilot activity along the longitudinal channel and 34% along the lateral channel as compared with a no NN case. In the case of a medium failure, neural augmentation EMRAN shows a 14% reduction along the longitudinal channel as compared with a no NN while the mean PC increases along lateral channel by 3%. Similarly, for a hard failure, there is a 45% lower PC along the longitudinal channel and 39% lower for lateral channel.

From the averaged value of five runs of the aileron failure shown in Table 7, it can be seen that there is a mixed trend with reduced PC; for a medium failure, the pilot activity is lower along the lateral and longitudinal channels, while it is higher for soft and hard failures in both lateral and longitudinal directions.

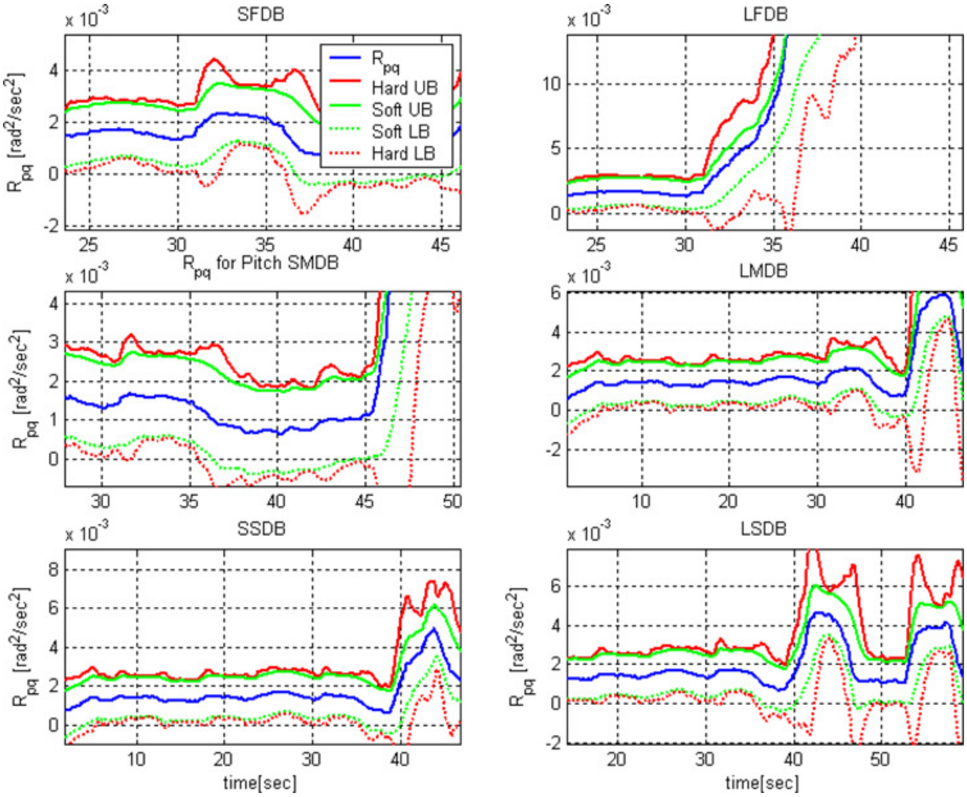


Fig. 12 R_{pq} **monitored to isolate the sensor failure.**

From the independent trends of the TE and pilot compensation (PC), it is challenging to draw definitive conclusions on the contribution of the neural augmentation in reducing pilot workload. For example, in the case of a hard aileron failure, the mean of TE is lower with neural augmentation (Table 3), but at a cost of additional pilot activity (Table 7).

Defining a composite parameter (CP) "η" as the absolute value of the product of TE and PC, could provide a useful performance index to quantify the performance of the adaptive controller as well as an indication of pilot workload under failure conditions.

$$\mathrm{CP} = \eta = \sum_{k=1}^{\mathrm{tw}} |\mathrm{TE}_k \times \mathrm{PC}_k| \tag{20}$$

The CP is not a metric for pilot workload but a metric for the performance of the fault-tolerant control laws. Under ideal conditions, there would be would be little errors with little pilot effort, but combinations of the two parameters at all possible levels are imaginable. If we obtain a reduction in the TEs at the expense of a large pilot workload, we cannot say that there is an improvement in the performance of the fault-tolerant control laws.

It can be seen from Tables 8 and 9 for the actuator failures, that the value of the CP "η" is lower with neural augmentation for all the actuator failure cases tested. For the stabilator failure scenario shown in Table 8, the average value of the CP is reduced by nearly 64, 59, and 87% in roll channel for soft, medium, and hard failure, respectively, and by 59, 54, and 91% in the corresponding values along the pitch channel.

In the case of the aileron failure scenario, the roll channel is predominant and much of the pilot activity is directed toward maintaining the mission profile along this channel. Considering the average value of the CP in Table 9, for the aileron failure, the CP η is reduced along roll the channel by 58, 55, and 77%, respectively, for a soft, medium, and hard failure, while there is a marginal increase along the pitch channel for soft , medium, and hard failures.

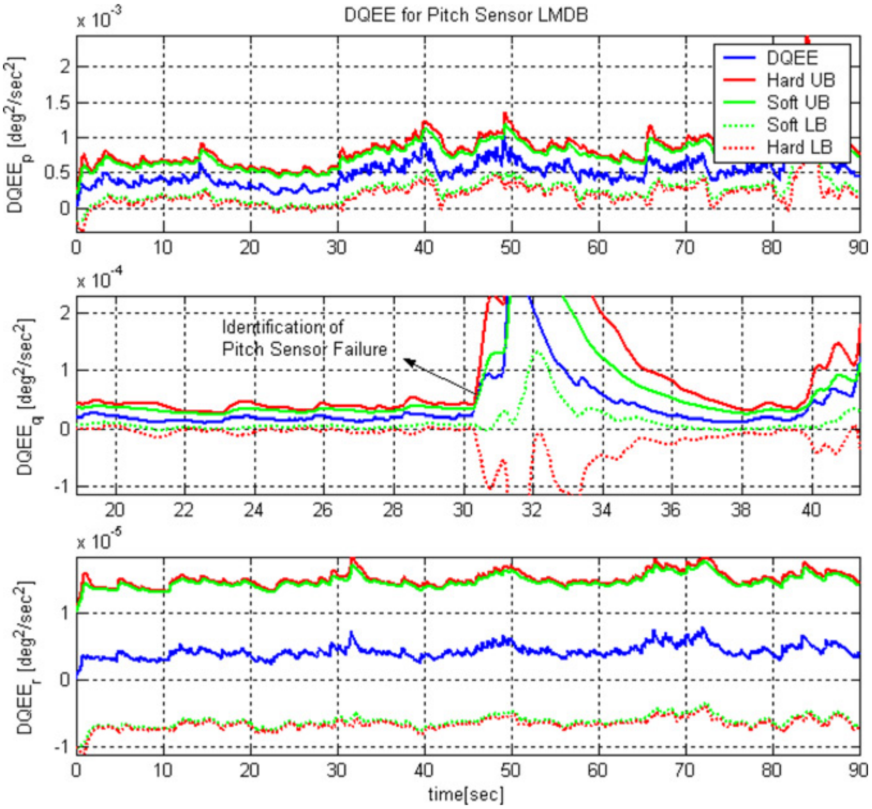


Fig. 13 Identification of the pitch sensor failure (SF#4).

Table 5 Summary of piloted tests with adaptive sensor FDII

Piloted Tests	Total failures tested	Correctly FDII	False detections	False isolations	False identifications	No detections
Stabilator failure	30	27	0	2	2	1
Aileron failure	30	28	0	2	2	0
Roll sensor failure	30	30	0	0	0	0
Pitch sensor failure	30	30	0	0	0	0
Yaw sensor failure	30	27	0	0	2	3

Table 6 Average pilot activity for stabilator failure

PC	2 deg (soft)		5 deg (medium)		8 deg (hard)	
Stick activity	No NN	EMRAN	No NN	EMRAN	No NN	EMRAN
Lateral	1.2617	0.8213	1.3003	1.3457	4.2966	2.6074
Longitudinal	0.8311	0.4265	1.1070	0.9452	2.2463	1.2302

Table 7 Average pilot activity for aileron failure cases

PC	2 deg (soft)		5 deg (medium)		8 deg (hard)	
Stick activity	No NN	EMRAN	No NN	EMRAN	No NN	EMRAN
Lateral	0.1336	0.3299	0.2931	0.1712	0.5979	0.6343
Longitudinal	0.2867	0.2977	0.5020	0.4295	0.4596	0.5955

Table 8 Average CP values for stabilator failure cases

CP	2 deg (soft)		5 deg (medium)		8 deg (hard)	
η	No NN	EMRAN	No NN	EMRAN	No NN	EMRAN
Roll	0.4244	0.1518	1.0342	0.4142	2.3814	0.3006
Pitch	0.1428	0.0580	0.3984	0.1793	1.1022	0.0906

Table 9 Average CP values for aileron failure cases

CP	2 deg (soft)		5 deg (medium)		8 deg (hard)	
η	No NN	EMRAN	No NN	EMRAN	No NN	EMRAN
Roll	0.1645	0.0690	0.3065	0.1373	0.1645	0.0690
Pitch	0.0005	0.0037	0.0003	0.0031	0.0005	0.0037

V. Conclusions

The paper presents the results of an effort evaluating the performance of a neurally augmented FTFCS, and an adaptive FDII scheme in a 6-DoF motion simulator. It was observed that the augmentation of FTFCS with a neural network reduced the TE under postfailure conditions for failures on the stabilator as well as aileron. The adaptive FDII scheme was found to be robust enough to take into account small perturbations from pilot activity and did not raise false flags or alarms. The FDII scheme performed the best for sensor failure cases, particularly along the roll and pitch channels with no cases of false FDII. However, a yaw sensor failure induced coupling along all three channels, and made the scheme more susceptible to false detections. The actuator failure FDII was found to be acceptable as it detected and identified the failures correctly for 90% of the test cases. However, due to the cross-coupling from pilot activity there were a small number of cases of false detections and isolations. The CP provided valuable insight into the contribution of the NN augmentation in reducing pilot workload while accommodating for failures. It was observed that the trend of the CP was lower under postfailure scenarios for both stabilator failure and aileron failure cases. The effectiveness of the FL-based approach to FDII could be expanded in the future to include elaborate mission profiles in various regions of the flight envelope as well as multiple failures (successive sensor or actuator failures) or reduced aerodynamic effectiveness (partial or complete loss of surfaces).

Acknowledgment

This research is supported by a NASA EPSCoR Grant Number # NNX07AT53A administered through the NASA WV Space Consortium.

References

[1] Himmelblau, D. M., *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*, Elsevier Press, Amsterdam.

[2] Kitamura, M., "Fault Detection in Nuclear Reactors with Aid of Parametric Modeling methods," *Fault Diagnosis in Dynamic Systems: Theory and Applications*, edited by P. Frank and Clark, Prentice-Hall, Upper Saddle River, NJ, Chap. 9.

[3] Duyar, A., and Merrill, W., "A Failure Diagnosis System Based on a Neural Network Classifier for the Space Shuttle Main Engine," *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, HI, 5–7 Dec. 1990, Vol. 4, pp. 2391–2400.

[4] McMahan, J., *Flight 1080*, Air Line Pilot, Vol. 47, No. 7.

[5] National Transportation Safety Board, "National Transportation Safety Board Accident Report of the American Airline DC10 Crash at Chicago O Hare' International Airport," NTAB-AAR-79-17, May 1979.

[6] National Transportation Safety Board, "Uncontrolled Descent and Collision with Terrain USAir Flight 427 Boeing 737-300," N513AU, NTSB No. AAR-99/01, Pennsylvania, 8 Sept. 1994.

[7] Perhinschi, M. G., Campa, G., Napolitano, M. R., Fravolini, M. L., Lando, M., and Massotti, L., "Performance Comparison of Fault Tolerant Control Laws within the NASA IFCS F-15 WVU Simulator," *Proceedings of the American Control Conference 2003*, Denver, CO, 4–6 June 2003, pp. 1661–1666.

[8] Anon., "NASA 584 X-31 Mishap Investigation Report, January 19, 1995," NASA Dryden Flight Research Center, 18, 18 Aug. 1995.

[9] Anon., "Air Force World: B-2 Crash Cause Identified," Air ForceE Magazine, Vol. 91, No. 7, July 2008, pp. 16–17.

[10] Mehra, R. K., and Peschon, J., "An Innovation Approach to Fault Detection and Diagnosis in Dynamic Systems," *Automatica*, Vol. 7, 1971, pp. 637–640.
doi: 10.1016/0005-1098(71)90028-8

[11] Mehra, R. K., Seereeram, S., Bayard, D., and Haydaegh, F., "Adaptive Kalman Filtering, Failure Detection and Identification for Spacecraft Attitude Estimation," *Proceedings of the 4th IEEE Conference on Control Applications*, Albany, NY, Sep. 1995, pp. 176–181.

[12] Napolitano, M. R., Younghawn, A., and Seanor, B., "A Fault Tolerant Flight Control System for Sensor and Actuator Failures using Neural Networks," *Aircraft Design*, Vol. 3, No. 2, 2000, pp. 103–128.
doi: 10.1016/S1369-8869(00)00009-4

[13] Qi, J., Zhao, X., Jiang, Z., and Han J., "An Adaptive Threshold Neural Network Scheme for Rotorcraft UAV Sensor Failure Diagnosis," *Advances in Neural Networks*, Vol. 4493, 2007.

[14] Wang, X., and Syrmos, V. L., "Design of Dynamic Fault-Tolerant Control System using IMM Estimation and RBF Neural Network," *Proceedings of the 14th Mediterranean Conference on Control and Automation*, June 2006, pp. 1–6.

[15] Willsky, A., "A Survey of Design Methods for Failure Detection," *Automatica*, Vol. 12, Nov. 1976, pp. 601–611.
doi: 10.1016/0005-1098(76)90041-8

[16] Perhinschi, M. G., Campa, G., Napolitano, M. R., Lando, M., Massotti, L., and Fravolini, M. L., "Modeling and Simulation of Failures for Primary Control Surfaces," *Proceedings of the 2002 AIAA Modeling and Simulation Conference*, Monterey, CA, Aug. 2002, AIAA Paper 2002-4786.

[17] Napolitano, M. R., Younghawn, A., and Seanor B., "A Fault Tolerant Flight Control System for Sensor and Actuator Failures Using Neural Networks," *Aircraft Design*, Vol. 3, No. 2, 2000, pp. 103–128.
doi: 10.1016/S1369-8869(00)00009-4

[18] 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," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug. 2003, AIAA Paper 2003-5643.

[19] Perhinschi, M., Napolitano, M. R., Seanor, B., Burken, J., and Larson, R., "An Adaptive Threshold Approach for Actuator Failure Detection and Identification," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 3, May 2006, pp. 519–525.
doi: 10.1109/TCST.2005.860522

[20] Brumbaugh, R. W. "An Aircraft Model for the AIAA Controls Design Challenge," NASA CR 186019, Dec. 1991.

[21] Perhinschi, M. G., Campa, G., Napolitano, M. R., Lando, M., Massotti L., and Fravolini M. L., "Modeling and Simulation of a Fault Tolerant Control System," *International Journal of Modeling and Simulation*, Vol. 26, No. 1, Jan. 2006, pp. 1–10.

[22] Anon., "Military Specification—Flying Qualities of Piloted Airplanes," U.S. Department of Defense, MIL F8785C, 1980.

[23] 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," *Proceedings of the 2003 American Control Conference*, Denver, CO, 4–6 June 2003, Vol. 2, pp. 1667–1672.

[24] Campa, G., Fravolini, M. L., and Napolitano, M. R., "A Library of Adaptive Neural Networks for Control Purposes," *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, Glasgow, Scotland, UK, 18–20 Sep. 2002, pp. 115–120.

[25] Sagoo, G. K., Gururajan, S., Napolitano, M. R., Perhinschi, M. G., Yu, G., and Seanor B., "Pilot-in-the-Loop Assessment of Neurally Augmented Dynamic Inversion Based Fault Tolerant Control Laws in a Motion-Based Flight Simulator," *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, HI, 18–21 Aug. 2008, AIAA Paper 2008-6843.

[26] Perhinschi, M. G., and Napolitano, M. R., "Integration of Computer Simulation for Flight Dynamics and Control Education," *Computers in Education Journal*, Vol. 18, No. 1, pp. 13–22.

[27] Polycarpou, M., "On-line Approximators for Nonlinear System Identification: A unified Approach," *Control and Dynamic Systems Series*, Vol. 7, Jan. 1998, pp. 191–230.

[28] Sagoo, G. K., "Pilot in Loop Assessment of Failure Detection Schemes in a Motion Flight Simulator," Master's Thesis, West Virginia University, 2008.

[29] Gundy-Burlet, K., Limes, G., and Bryant, D., "Control Reallocation Strategies for Damage Adaptation in Transport Class Aircraft," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX, 11–14 Aug. 2003, AIAA Paper 2003-5642.

[30] Gundy-Burlet, K., Krishnakumar, K., Limes, G., and Bryant, D., "Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft," *Journal of Aerospace Computing Information and Communication*, Vol. 1, No. 12, 2004, pp. 526–542.
doi: 10.2514/1.13246

[31] Kaneshige, J., and Gundy-Burlet, K., "Integrated Neural Flight and Propulsion Control System," *Proceedings of the AIAA Guidance, Navigation, and Controls Conference*, Montreal, 2001, AIAA Paper 2001-4386.

Ella Atkins
Associate Editor