JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 33, No. 4, July–August 2010

Aircraft Failure Detection and Identification Using an Immunological Hierarchical Multiself Strategy

Hever Moncayo,* Mario G. Perhinschi,† and Jennifer Davis*
West Virginia University, Morgantown, West Virginia 26508

DOI: 10.2514/1.47445

This paper presents the development and application of an integrated artificial-immune-system-based scheme for the detection and identification of a variety of aircraft sensor, actuator, propulsion, and structural failures/damages. The proposed approach is based on a hierarchical multiself strategy in which different self configurations are selected for detection and identification of specific abnormal conditions. Data collected using a motion-based flight simulator were used to define the self for a subregion of the flight envelope. The aircraft model represents a supersonic fighter, including model-following adaptive control laws based on nonlinear dynamic inversion and artificial neural network augmentation. The proposed detection scheme achieves low false alarm rates and high detection and identification rates for all four categories of failures considered.

Nomenclature

a_x	=	longitudinal acceleration, m/s^2
DQEE_x	=	decentralized quadratic estimation error
MQEE	=	main quadratic estimation error
OQEE	=	output quadratic estimation error
N_{clon}	=	number of clones around a detector
N_{MOV}	=	number of moved detector centers
N_{RD}	=	number of random detector centers
$\mathrm{NN}_{\mathrm{out}x}$	=	specific neural network output
$\mathrm{NN}_{w(x)}$	=	specific neural network weight
p, q, r	=	measured roll, pitch, and yaw rates, rad/s
$\hat{p}_{\mathrm{DNN}}$, $\hat{q}_{\mathrm{DNN}}$, $\hat{r}_{\mathrm{DNN}}$	=	neural estimates of angular rates that do not include the respective gyro
$\hat{p}_{\mathrm{MNN}}$, $\hat{q}_{\mathrm{MNN}}$, $\hat{r}_{\mathrm{MNN}}$	=	measurements of angular rates
r_m	=	minimum radius of hypersphere detectors
r_s	=	cluster radius
R_{pq}	=	roll-pitch cross-correlation coefficient
R_{rr}	=	yaw autocorrelation coefficient
Thr	=	relative detection threshold
thr	=	detection threshold
$\mathbf{x}$	=	state vector, input vector
x_{TE}	=	angular-rate tracking error, rad/s
w_i	=	overlapping measure of detector i
w_{thr}	=	overlapping threshold value
β	=	sideslip angle, rad or deg
ζ	=	detection parameter
μ_{rr}	=	scaling factor

Subscripts

i	=	index for number of points defining the detector from 1 to n
ref	=	reference channel
thr	=	threshold
x	=	angular-rate channels p, q, and r

Presented as Paper 5878 at the AIAA GNC, Chicago, IL, 10–13 August 2009; received 30 September 2009; revision received 20 January 2010; accepted for publication 25 January 2010.

*Graduate Student, Department of Mechanical and Aerospace Engineering.

†Assistant Professor, Department of Mechanical and Aerospace Engineering. AIAA Senior Member.

I. Introduction

THE development of fault-tolerant flight control systems has emerged in recent years as a key aspect to ensure increased safety and enhanced performance for both civilian and military aircraft [1]. This new research focus has produced a variety of techniques [2,3], many of which rely on high-performance real-time failure detection and identification (FDI) schemes. However, the research efforts in the field have focused on individual classes of failure at isolated and constraint flight conditions. A comprehensive integrated solution to the FDI problem for aircraft subsystems has not yet been proposed. The endeavor is extremely complex and multidimensional, requiring adequate tools. A promising candidate in this respect is the artificial immune system (AIS) concept [4]. The AIS-based fault detection [5–7] operates in a similar manner to the biological immune system (according to the principle of self/nonself discrimination) when it distinguishes between entities that belong to the organism and entities that do not. The basic idea is that an abnormal situation (i.e., failure of one of the aircraft subsystems) can be declared when a current configuration of features does not match with any configuration from a predetermined set known to correspond to normal situations.

The AIS emerged in recent years as a new computational paradigm in artificial intelligence. The concept has shown a very promising potential for a variety of applications such as anomaly detection [5,8], data mining [6], computer security [7,9], adaptive control [10–12], and pattern recognition [13]. Although new models and applications are currently being developed [4] and existing methods are improved continuously, the entire field of AIS (including negative-selection algorithms) is still relatively young and not well defined. Theoretical issues have been occasionally addressed in the attempt to assess and prove AIS applicability [14]; however, there is no systematic theoretical background yet to support the AIS.

In addition, the AIS concept has shown promising capabilities for fault detection of aerospace systems [11,15–17]. These applications focus primarily on aircraft actuator fault detection and identification and they have only considered isolated failures of high magnitudes for limited regions of the flight envelope. Therefore, the availability of failure detection and identification schemes with high rates of success, with comprehensive coverage, integrating all aircraft subsystems and operational modes is a critical objective of this paper.

In general, to make the AIS a practical fault/anomaly-detection technique, some specific aspects must be addressed: computational efficiency improvement of the algorithms, enhancement of the representation, and development of unified architectures that can integrate several AIS mechanisms.

An integrated set of methodologies for AIS-based detection, identification, and evaluation of a wide variety of aircraft sensor, actuator, propulsion, and structural failures/damages [18] is currently

under development at West Virginia University (WVU) within NASA's Aviation Safety Program. As part of this effort, the development of an integrated high-performance AIS-based FDI scheme using a hierarchical multiself strategy is presented in this paper. The scheme is capable of detecting and identifying several categories of subsystem abnormal conditions over an extended area of the flight envelope. The effectiveness of the approach in terms of high detection rate and low number of false alarms for the four categories of failures is tested using data from the WVU motion-based flight simulator. The aircraft model represents a supersonic fighter, including model-following adaptive control laws based on nonlinear dynamic inversion and artificial neural network augmentation [19].

A description of the proposed AIS-based FDI scheme is presented in Sec. II. The general framework for the AIS-based FDI scheme is presented in Sec. III, including aircraft subsystem failure modeling and motion-based flight simulator tests for self definition and detection testing. Section IV provides details on the design and implementation of the FDI scheme. Test results, analysis, and evaluation of the FDI scheme performance are presented in Sec. V. Finally, some conclusions are summarized in Sec. VI.

II. General Description of the AIS-Based FDI Scheme

The mechanisms and processes of the biological immune system are the inspiration for the AIS, as a new artificial intelligence technique for fault detection [10,20,21]. In living organisms, specialized cells (T-cells [22]) are generated such that they do not match (negative selection [15]) specific features of the organism cells coded as strings of proteins and polysaccharides. However, they can match intruding agents and mark them for destruction. Applying this paradigm to aircraft subsystem FDI requires that a set of adequate features be defined. These features can include various sensor outputs, states estimates, statistical parameters, or any other information expected to be relevant to the behavior of the system and able to capture the signature of abnormal situations. Extensive experimental data are necessary to determine the self or the hyperspace of normal conditions. Adequate numerical representations of the self/nonself must be used and the data processed such that they are manageable given the computational and storage limitations of the available hardware. The artificial antibodies (the detectors) must then be generated and optimized. This process may be repeated to produce several sets of detectors for different self configurations. At this point, the selves obtained can be organized and classified based on the capability of each one to detect and identify every type of failure. Finally, a detection logic must be designed for real-time operation with high detection rates and low numbers of false alarms. The block diagram of the general AIS design process for fault detection is presented in Fig. 1 [18].

A. Definition of Self-Characterizing Variables

A critical element for the success of the AIS-based FDI scheme is the selection of the appropriate parameters (features) to capture the dynamic signature of each and every type of failure. The candidate parameters for self/nonself definition can be grouped in the following five categories: aircraft state variables, pilot input variables, stability and control derivatives, variables generated within the control laws, and derived variables. Within this research effort, features in all categories have been considered, except stability and control derivatives, which are difficult to be determined online.

The aircraft state variables are a natural choice and measurements of aircraft angular rates have been used [15] in the literature for self/nonself definition and failure detection.

It is possible that the dynamic fingerprint of a failure be reproduced through intentional pilot input [18]. For example, it is well known that an elevator failure induces a coupling between the longitudinal and lateral channel. This characteristic may be used for detection. However, it is possible to achieve similar coupling under normal conditions through simultaneous pilot input on both channels. In such situations, information about the pilot input can help correct detection. Pilot input information is contained in several signals such as stick and pedals displacement, aerodynamic control-surface deflections, and reference state variables generated by model-following control laws.

The artificial neural network (NN) control augmentation implemented within the model provides useful signals with significant FDI capabilities. Since the adaptation activity increases after the occurrence of a failure, it is expected that these signals capture the increased adaptation activity and thus detect the failure. More details on the NN architecture, inputs, outputs, and training are presented in [23,24].

Finally, previous studies [25,26] have put into evidence promising parameters for FDI based on correlations between state variables (angular rates) and neural estimates of the angular rates.

After a preliminary analysis of the detection potential of a large number of candidate features, the following parameters were selected for further use within the FDI scheme:

1) The angular-rate tracking errors on all three channels is defined as

$$x_{\mathrm{TE}} = x - x_{\mathrm{ref}} \tag{1}$$

where $x = p$, q, or r (roll, pitch, or yaw rate) and x_{ref} is the reference rate determined within the model-following control laws directly from pilot input to meet first- or second-order responses as required for good handling qualities.

2) The NN output on all three channels (angular accelerations) is $\mathrm{NN}_{\mathrm{out}}$.

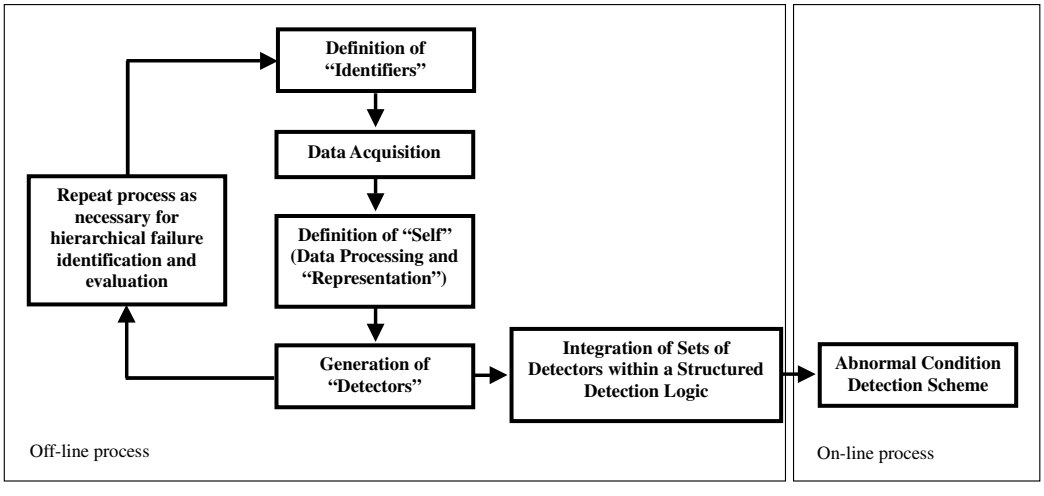


Fig. 1 Artificial-immune-system-based abnormal-condition detection.

3) Parameters based on NN estimates of angular rates are the main quadratic estimation error (MQEE), the output quadratic estimation error (OQEE), and the decentralized quadratic estimation error (DQEE). MQEE is defined as

$$\mathrm{MQEE}(k) = \tfrac{1}{2}[(p(k) - \hat{p}_{\mathrm{MNN}}(k))^2 + (q(k) - \hat{q}_{\mathrm{MNN}}(k))^2 + (r(k) - \hat{r}_{\mathrm{MNN}}(k))^2] \quad (2)$$

where $p(k)$, $q(k)$, and $r(k)$ are measurements of angular rates at sample k and $\hat{p}_{\mathrm{MNN}}(k)$, $\hat{q}_{\mathrm{MNN}}(k)$, and $\hat{r}_{\mathrm{MNN}}(k)$ are neural estimates of the angular rates based on sensor measurements, including the respective gyro, over a specified time window. OQEE is defined as

$$\mathrm{OQEE}(k) = \tfrac{1}{2}[(\hat{p}_{\mathrm{DNN}}(k) - \hat{p}_{\mathrm{MNN}}(k))^2 + (\hat{q}_{\mathrm{DNN}}(k) - \hat{q}_{\mathrm{MNN}}(k))^2 + (\hat{r}_{\mathrm{DNN}}(k) - \hat{r}_{\mathrm{MNN}}(k))^2] \quad (3)$$

where $\hat{p}_{\mathrm{DNN}}(k)$, $\hat{q}_{\mathrm{DNN}}(k)$, and $\hat{r}_{\mathrm{DNN}}(k)$ are neural estimates of the angular rates based on sensor measurements that do not include the respective gyro, over a specified time window. DQEE is defined as

$$\mathrm{DQEE}_x(k) = \tfrac{1}{2}(\hat{x}_{\mathrm{DNN}}(k) - x(k))^2, \qquad x = p, q, r \quad (4)$$

4) The angular-rate correlation parameters are $\bar{R}_{rr}$ and $\bar{R}_{pq}$:

$$\bar{R}_{rr}(k) = \mu_{rr} \cdot \mathrm{OQEE}(k) + \sum_{i=k-n}^{k} R_{rr}(i) \quad (5)$$

$$\bar{R}_{pq}(k) = \sum_{i=k-n}^{k} R_{pq}(i) \quad (6)$$

where μ_{rr} is a scaling factor and n defines the width of the time window over which the angular-rate correlation coefficients are summed.

Ten feature configurations were considered for self definition as presented in Table 1.

B. Representation of Self

A process that is of absolute importance for the AIS is the matching between the detectors and the explored data or candidates (data subject to the detection process). This is the equivalent of the biological matching between the antibodies and antigens, which is the basis for the recognition and selective elimination mechanism of intruding agents. In general, the matching rules rely on metrics for comparison and a logic to produce a binary output: match or not-match. They depend on the type of data representation. Data representation has an important impact on algorithm effectiveness and performance. It determines the possible matching rules, the detector generation mechanisms, and the detection process. In this paper, due to the nature of the variables involved, a real-valued vector representation is implemented. Within the real-valued vector representation, each data item is a vector of real numbers [9]. The matching rules and the measure of difference or similarity are based on the numeric elements of the vector, which are implicit in the Euclidian distance between the components tested [27]. Since the distance between the points is critical for the detection process, different scales of the identifiers is not desirable; therefore, the actual values of these variables were normalized between 0 and 1, covering the entire possible range under normal conditions. The normalization factor for each dimension is determined by the minimum and maximum values of the flight data plus a percentage margin. In this way, the solution space becomes a unit hypercube in which the self and nonself will be defined. Once the normalization process has been performed, the points are clustered to reduce the amount of information and computational requirements, as well as to eliminate possible holes of healthy data not represented in the original self database. Clustering is performed using an optimized version of the k-means algorithm [28]. These clustered data define the self space. In the case of hypersphere bodies, every cluster is defined by a radius r_s and a center c_s (point to which the sum of all distances from all points belonging that cluster is minimized).

Table 1 Feature configurations for self definition

Self number	Features	Solution-space dimension
Self 1	NN_{Wp}, $\mathrm{NN}_{\mathrm{out}x}$, MQEE, OQEE, and DQEE_x	9
Self 2	NN_{Wp} and $\mathrm{NN}_{\mathrm{out}x}$	4
Self 3	$\mathrm{NN}_{\mathrm{out}x}$	3
Self 4	DQEE_x	3
Self 5	$\mathrm{NN}_{\mathrm{out}x}$, MQEE, OQEE, and DQEE_x	8
Self 6	$\mathrm{NN}_{Wr(43)}$, $\mathrm{NN}_{\mathrm{out}r}$, β, and a_x	4
Self 7	$\mathrm{NN}_{Wr(183)}$, $\mathrm{NN}_{\mathrm{out}r}$, β, and a_x	4
Self 8	$\mathrm{NN}_{\mathrm{out}x}$ and x_{TE}	6
Self 9	MQEE, OQEE, $\bar{R}_{rr}$, and $\bar{R}_{pq}$	4
Self 10	$\mathrm{NN}_{\mathrm{out}x}$ and DQEE_x	6

During the clustering process, an important design aspect that must to be addressed is the amount of empty space that the clusters cover. Empty space is defined as the space enclosed by the clusters in which no self points are found. The presence of empty space has a significant impact on the detection performance. For example, when the number of clusters relative to the number of data points available is small, the radius of every cluster will likely be large and a certain amount of empty space (potentially nonself) is included. Since no detectors can be generated in that particular space, the detection rate will decrease. However, when generating a high number of clusters, the size of the clusters will not permit coverage of some space that could be part of the self, but will be covered by detectors instead. In consequence, the false alarms are likely to increase. The reduction of empty space can be achieved through an iterative clustering algorithm [29] in which the cluster radius is minimized progressively until the desired level of empty space is reached.

The geometric properties of the hyperbodies can potentially have an impact on the efficiency of the detector generation process and on the detection itself by adding geometric flexibility and diversity. They determine how well the nonself is covered, how many detectors are necessary, and how intensive the computational process is. For the purpose of this paper, only hypersphere shapes for the self/nonself are considered; however, the approach can be expanded to other hypershapes as hyperrectangle, hyperellipse, or even a combination of them.

C. Generation of Detectors

Existing algorithms have been evaluated, customized, and integrated within an evolutionary algorithm (EA)-based design tool for detector generation and optimization [30]. The two-phase EA achieves optimization of the detector set for good detection performance and computational effectiveness. An enhanced negative-selection algorithm for real-valued representation with variable detector radius (ENSA-RV) has been designed and implemented. The algorithm ensures that there is no overlapping with the self and that the nonself is covered to a desired predetermined level. It should be noted that the EA requires a large number of specific parameters that must be carefully selected and correlated. As is the case with any EA, the design process must properly balance exploration and exploitation for adequate convergence and robustness. Details regarding the design of the EA and its general characteristics are presented in [30].

Starting with an initial set of candidate detectors, located randomly in the nonself of an n-dimensional hyperspace, the algorithm performs a selection process based on two criteria: no overlapping with the self and maximum coverage of the nonself. At every iteration, the radius of each detector is computed using the distance between the candidate detector and the nearest self cluster. The radius of these detectors is set as the maximum possible without overlapping with a self cluster. Since a minimum radius r_m is permitted for detectors, the distance between centers must be greater than or equal to the sum of r_m and the radius of the cluster r_s. Because

a better coverage is achieved when a minimum overlapping among detectors is ensured, an overlapping measure w_i of a detector with respect to the others is calculated during the maturation process [31]. For an overlapping threshold value w_{thr}, every detector is selected as mature if the condition $w_i \leq w_{\text{thr}}$ is satisfied. Eventually, if $w_i = 0$, that particular detector is selected to have a number of $N_{\text{clon}} = 2n$ clones around it. The center of the first clone is placed at a distance equal to one radius and at a random unitary direction. The remaining clone centers are generated at 90° angles with respect to the first one at n different planes. If $0 < w_i \leq w_{\text{thr}}$, only one center clone is generated at a direction opposite to the nearest element (mature detector or cluster self). For this operation, an updating rule is used to determine how far the clone element is located at every iteration. Additionally, the N_{MOV} smallest rejected detectors are selected to be moved in opposite direction of the mean center of the k-nearest elements. The same updating rule used in the detector cloning operation is used in the detector movement step as well. Finally, a set of N_{RD} random centers is inserted; the radius of the mature detectors calculated, and the coverage and overlapping computed. All these parameters constitute the design options for the algorithm, which allow tuning the detector generation process. The process can be stopped after a prescribed number of iterations, when a prescribed maximum number of acceptable detectors has been reached, or when a desired coverage of the nonself has been achieved. The algorithm can optimize the requirements for no overlapping among nonself detectors and self and minimum uncovered areas in the nonself.

Using the ENSA-RV, several sets of detectors are generated for different self configurations. They are then tested and classified in terms of detection rate and false alarms. Those configurations that ensure high detection for specific failure are identified. In Fig. 2, a typical execution of the algorithm for a two-dimensional self configuration after 2 and 50 iterations is shown for illustration. The thick line circles represent the self clusters and the thin line ones, the generated detectors.

D. Integration of Self Patterns

Once the capabilities of different sets of features for the detection of specific types of failures have been analyzed, the selves with best results are organized and integrated within a hierarchical scheme such that a high detection rate, low number of false alarms, and correct identification are achieved for all failure categories. The process is performed offline and can be customized continually until a configuration is obtained, which ensures the desirable performance of the AIS scheme.

E. Detection and Identification Scheme Using a Hierarchical Multiself Strategy

As shown in Fig. 3, the online detection and identification process is performed using two main components. The first one uses an integrated block of self patterns which performs the detection phase. The second component, where the identification phase is performed, attempts to ensure the correct identification of categories and subcategories of failures at different levels.

The hierarchical multiself (HMS) strategy scheme relies on the assumption that within a class of failures, differences between failed elements may be captured by different numbers and/or types of features as compared to the ones necessary to detect the class. Thus, a specific set of parameter could favor the identification of some particular failures better than others. This approach requires building failure specific selves and identification of low dimension spaces for hierarchical selves definitions.

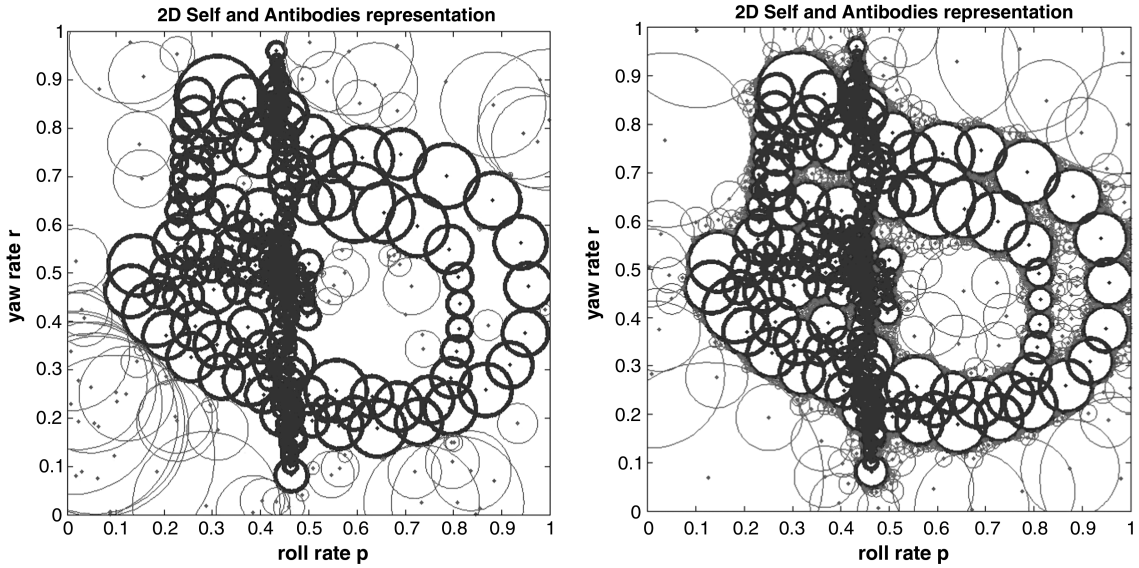
III. Framework for FDI Scheme Testing

A. Description of the Failure Modeling

Four types of failures were modeled to support the development and testing of the FDI scheme: actuator, sensor, propulsion, and structural failures/damages. A brief description of the modeling approach is presented next.

1. *Actuator Failure Modeling*

Within this effort, failures on left or right individual stabilator, aileron, or rudder (since the aircraft considered is equipped with a dual fin) have been considered. Two types of control-surface failure are modeled: stuck aerodynamic control surface and physically damaged aerodynamic control surface. The first failure type corresponds to an actuator mechanism failure and results in a locked surface; in fact, at the failure occurrence, the control surface remains fixed in the current position/deflection or moves to a predefined position and remains fixed there. A failure involving a blockage of the control surface at a fixed deflection does not alter the aerodynamic properties of the control surface. However, each surface in a pair (left and right) will have different deflections and the resulting moments and forces are computed individually. The second failure type corresponds to a physical destruction and/or deformation of the control surface. It consists of a deterioration of the aerodynamic efficiency of the control surface starting at the failure occurring moment. A control failure that involves physical destruction of the control surface may alter the aerodynamic properties in manners that can be both qualitative (affecting the



a) *2nd iteration* b) *50th iteration*

Fig. 2 Enhanced NSA-RV with 2-D variable detectors.

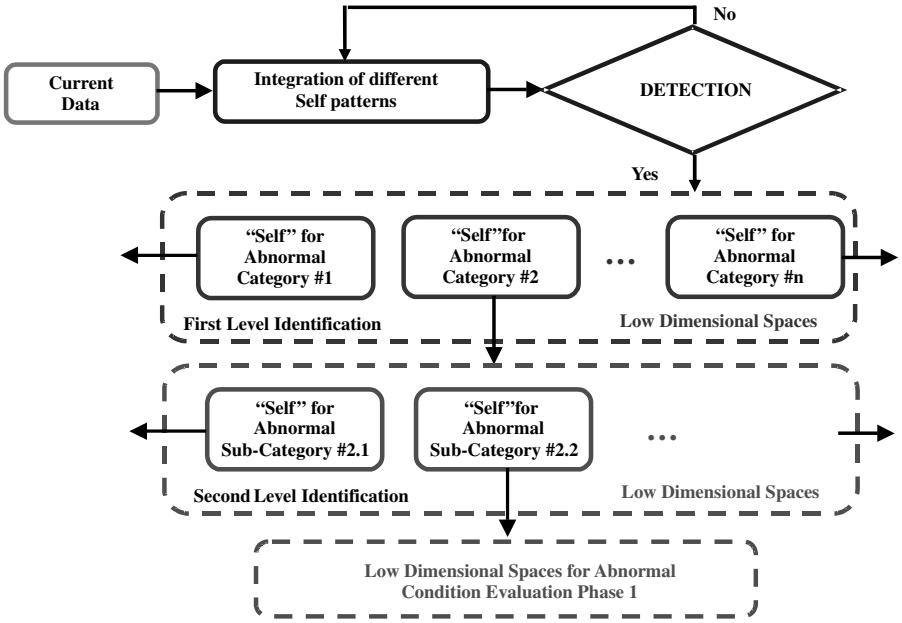


Fig. 3 Hierarchical multiself strategy for online failure detection and identification.

nature of the aerodynamic phenomena involved) and quantitative (affecting the magnitude of characteristic parameters). More details and complete models are presented in [32,33].

2. Sensor Failure Modeling

Failures of the gyros on the three channels have been considered within this effort, since the outputs of these sensors are used for control purposes. The simulated sensor failure implemented consists of an output bias. The transition to the biased sensor output can be instantaneous (step bias) or over a certain transient (drifting bias) [33]. Different transients as well as different sizes of the bias can be defined.

3. Aerodynamic-Surface Damage Modeling

For the purpose of this paper, only the damage of the wing is modeled separately. Damages to other aerodynamic surface may be considered as failures of the respective actuators (loss of aerodynamic efficiency). A simple model of wing damage is developed considering both aerodynamic and gravimetric effects. The failure type corresponds to a total or partial physical destruction and/or deformation of the wing and different percent values along the wing can be selected as damage affected area.

4. Engine Failure Modeling

Simple models for the following engine failures/malfunctions have been implemented: stuck throttle, thrust runaway, and power/thrust-reduced control efficiency. The stuck throttle failure implies normal operation of the engine but no response to power lever actuation. The thrust runaway failure models a malfunction of the fuel control system, which causes the increase of the fuel flow to maximum and the increase of the thrust as a result. This is modeled by increasing the throttle to maximum with first order dynamics and time constant setup by the user. Finally, the power/thrust-reduced control efficiency is modeled by scaling down the throttle input by a constant factor selected by the user. For this paper, only the latter type of engine failure is considered.

B. Flight Simulator Experiment

The aircraft aerodynamic model used was derived from a nonlinear model of a high-performance military aircraft distributed by NASA to academic institutions in 1990 within a student design competition [34]. This generic model was customized through the addition of the aerodynamics modeling of canard surfaces for the purpose of simulating the NASA Intelligent Flight Control System (IFCS) F-15 research aircraft [23]. The aerodynamic and thrust characteristics are provided through 42 lookup tables. The lookup tables have been subdivided to isolate the contribution of individual aerodynamic surfaces and control surfaces in order to be able to simulate structural damage and control-surface failure.

The experimental data were generated in the WVU six-degree-of-freedom (6-DOF) motion-based flight simulator. The Motus 600 flight simulator and was interfaced with an external computer on which the customized NASA IFCS F-15 research aircraft was run within the MATLAB/Simulink environment to drive the entire simulator system (see Fig. 4). This offers a very realistic flight environment allowing true motion cues flight simulation capabilities and high-quality visual cues.

To define the self as completely and accurately as possible, adequate coverage of the state space must be achieved. Different flight scenarios were considered over a wide area of the flight envelope. However, only the results of a subset of the flight envelope are shown in this paper. The flight test is first defined based on three specific reference points for Mach numbers between 0.75 and 0.9 and altitudes between 20,000 and 31,000 ft. The sequence starts with the aircraft flying in level flight at point 1 (Mach 0.75 and 20,000 ft), ascend at constant speed to point 2 (Mach 0.75 and 31,000 ft), accelerate at constant altitude to point 3 (Mach 0.9 and 31,000 ft), and then return to points 2 and 1. The flight tests, lasting between 10 and 20 min each, were designed to include steady-state flight conditions, transitions between steady-state conditions, and mild to moderate maneuver such as doublets, coordinated turns at progressive bank angles. These flight scenarios were simulated under normal flight conditions. Then, they were repeated under various failure scenarios for development purposes. Only one failure at a time is considered to capture/isolate the dynamic fingerprint of each type of failure and generate detectors appropriately. Additional tests were also collected to be used as validation data. The data from the simulator were acquired at a rate of 50 Hz.

The simulated failures tested for the purpose of this paper are outlined in Table 2. The training data set was created by taking all of the data from nominal conditions, and the test data were generated from flight tests with failure. An additional validation set of data at normal conditions was acquired to evaluate the level of false alarms.

IV. Immunity-Based Detection and Identification

The 10 sets of features presented in Table 1 were used to define the self as a set of hyperspherical clusters. Corresponding detectors were

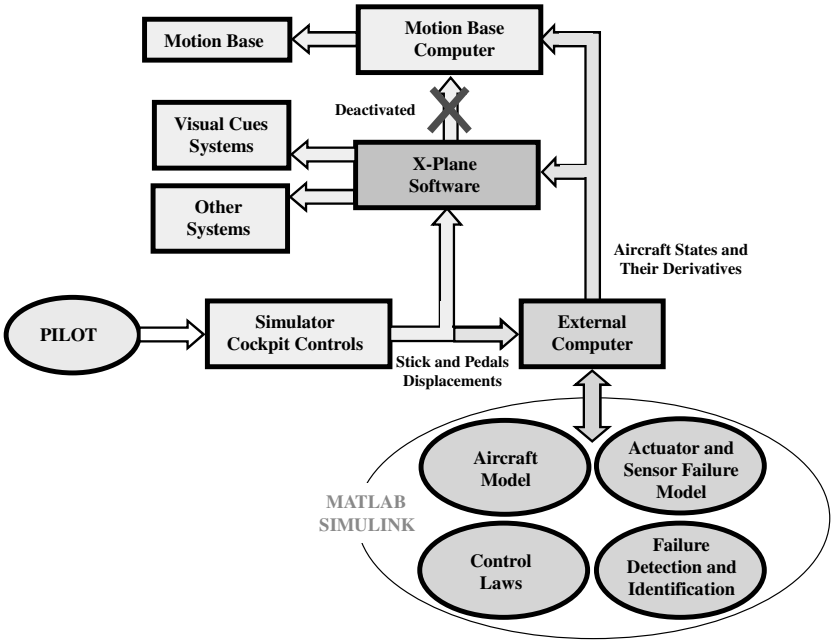


Fig. 4 Interface of the WVU flight simulator with external models.

generated in each case. Their capability to detect abnormal conditions was determined for every type of failure. The quantitative evaluation has been defined by using a specific metric. Assuming typical binary outcomes, the results of the detection can be categorized as follows:

1) With true positives (TP), abnormal data points are detected as abnormal.

2) With true negatives (TN), normal data points are not detected as abnormal.

3) With false positives (FP), normal data points detected as abnormal.

4) With false negatives (FN), abnormal data points are not detected as abnormal.

The detection rate (DR) is defined as the ratio between abnormal data points detected as abnormal divided by the total amount of abnormal data points:

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100 \tag{7}$$

The false alarm rate is defined as the ratio between normal data points detected as abnormal divided by the total amount of normal data points:

$$\mathrm{FA} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \times 100 \tag{8}$$

The detection performance for every self and failure has been determined and is presented in Table 3 in terms of detection rate and false alarms. These results show that only self 3 achieves an acceptable detection performance for all failure cases, but with a high number of false alarms. In contrast, the worst case is presented for self 4, in which only the yaw gyro sensor failure is correctly detected. Note that every self case presents at least one acceptable performance in one of the failures considered. For instance, self 7 shows poor detection for the roll rate sensor, pitch rate sensor, and aileron failures, but an acceptable detection performance for the other failures with very low false alarms. The fact that different selves favor the detection of particular types of failures is used in this paper to develop an integrated scheme in which different self configurations ensure overall high detection rate and low number of false alarms.

The differences in the DR of different classes of failure are attributed primarily to the capabilities of the respective sets of features to capture the signatures of each and every failure. This is a necessary but not a sufficient condition for successful FDI. Additional conditions include adequate coverage of the self hyperspace by the test data and the self clusters and adequate coverage of the nonself hyperspace by the detectors through the detector generation and optimization process. For instance, in the case of self 8, only 2.86% of DR is achieved for the right engine failure, which is very low compared with 44.13% DR achieved for the left engine failure, although one would expect the results to be similar, due to symmetry. This shows that the detectors fail to cover very well the volume corresponding to the right engine failure due to the fact that the self clusters are inadequately covering it and/or the EA has not converged to an optimized set of detectors. The first cause appears to be more probable, because the discrepancies between left and right engine are present for several of the different self configurations. One interesting way to mitigate such situations could be to include in the self clustering process additional information such as the symmetry of the engines. However, in this paper,

Table 2 Simulated failures

Failure type	Description
	Actuator failure
Stabilator, aileron, and rudder	Blockage of any, left or right, control surface at 8 deg
	Sensor failure
Large step bias (LSB) and large fast drifting bias (LFDB)	Step bias of 10 deg/s in the roll and pitch rate gyro sensors and 3 deg/s in the yaw rate gyro sensor
	Structural failure
Wing damage	Loss of 35% of any, left or right wing, affecting the efficiency of the aileron control surface
	Engine failure
Power/thrust-reduced control efficiency	Loss of 98% of the power in any left or right engine.

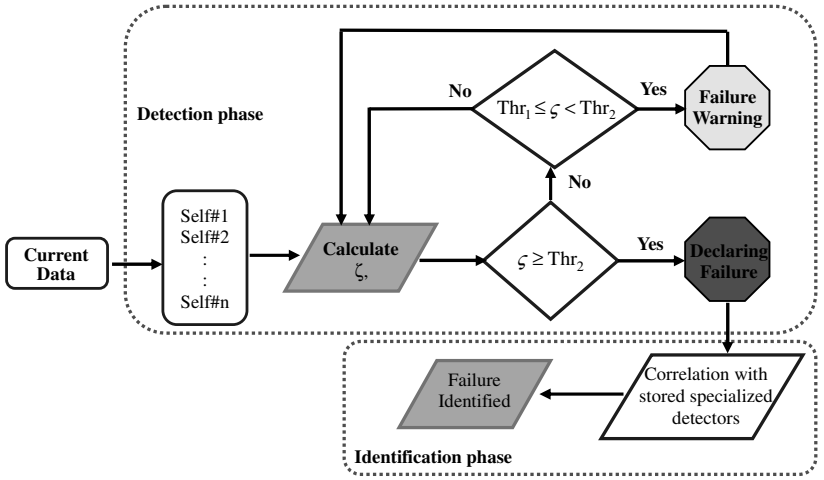


Fig. 5 Block diagram of the proposed HMS strategy for online failure detection and identification.

additional processing was not necessary, because selves 6 and 7 provided enough engine failure capabilities.

Detection is considered to be the process leading to declaring that an abnormal condition in any of the categories is present. During this phase, sets of current values of the identifiers measured in flight at a certain sampling rate are compared against the detectors that have been generated for every self configuration, as shown in Fig. 5. A detection parameter ς is calculated, which represents the number of consecutive points over a window ω that trigger detectors, summed over all selves. If ς is within a certain range, a failure warning is issued, but if ς exceeds a threshold, a failure is declared and the identification phase starts.

The identification is performed analyzing which of the detectors have been activated. As shown in the Fig. 6, the detectors have been labeled previously to represent specific categories of failures through an offline process that is equivalent to detectors generation based on positive selection strategy performed repeatedly on selves at each failure category. The identification decision is then taken based on a majority vote.

The identification phase is performed in two steps:

1) In the preidentification step, the failure is attributed to one of the four categories: control surface, sensor, structure, or engine failure. As shown in Fig. 7, the category is determined based on the number of times each set of detectors is activated for a particular failure category. For instance, if one set of specialized detectors of an individual/isolated detector sets are more frequently being activated, the failure category corresponding to those specific detectors is identified. The preliminary result is compared with the output of the other individual/isolated detector sets. The most repetitive result will determine the identified category.

2) In the identification step, if the failure is classified as actuator failure, the failure is identified to be a left or right stabilator, aileron, or rudder failure. If preidentified as a sensor failure, it is identified to be a roll, pitch, or yaw rate sensor failure. If preidentified as a structural failure, the abnormal condition may be identified as affecting the left or right wing. Finally, if a propulsion failure was

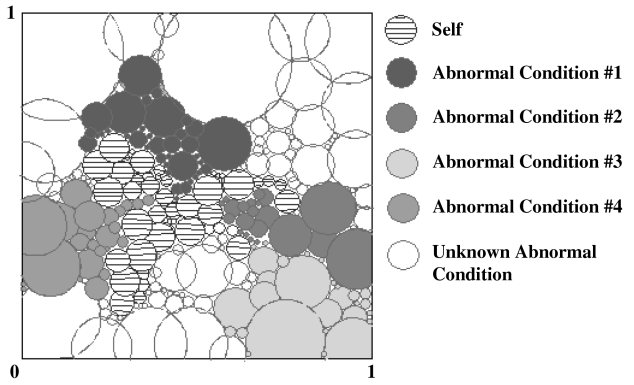


Fig. 6 Identification using specialized detector approach.

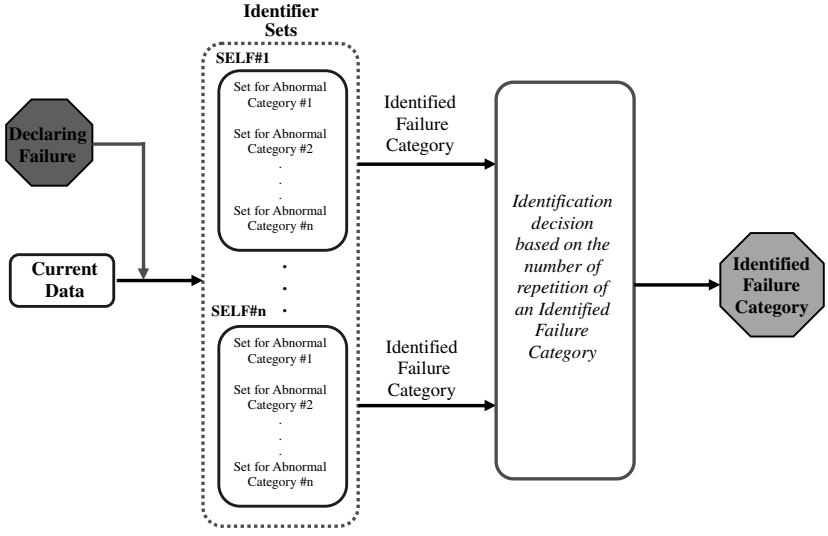


Fig. 7 Block diagram of the identification phase of proposed HMS strategy.

Table 3 Detection performance of different self configurations, %

Self configuration	No. of detectors	Actuator failure 8 deg						Sensor failure								Structural failure		Engine failure		Nominal
		Stabilator		Aileron		Rudder		LSB				LFDB								
		L	R	L	R	L	R	p	q	r	p	q	r	L	R	L	R	FA
Self 1	499	99.79	99.98	86.52	70.62	90.85	71.81	92.87	92.01	99.75	87.62	95.93	99.38	99.68	99.84	73.82	36.32	2.49
Self 2	213	99.97	99.97	96.84	99.97	97.01	95.70	69.59	83.13	98.03	52.89	87.06	98.03	99.98	72	72.30	52.90	5.25
Self 3	516	99.07	99.97	99.96	99.96	98.51	96.39	60.80	94.93	98.97	41.02	94.78	99.28	99.96	87.34	80.62	34.19	19.59
Self 4	507	6.44	1.66	0.23	0.80	0.91	1.60	92.74	57.62	99.73	89.15	27.78	99.81	12.44	8.25	0.84	1.02	0.78
Self 5	504	99.9	99.9	79.8	57.8	92.3	76.9	95.6	90.5	99.8	90.8	95.58	99.8	99.5	99.9	59.8	10.8	2.5
Self 6	280	96.48	99.51	3.73	14.35	98.90	99.01	24.89	2.45	98.87	10.22	10.01	99.75	88.12	96.83	96.51	94.11	3.52
Self 7	389	99.96	99.44	81.33	32.61	98.84	98.88	5.05	1.33	95.38	59.73	4.33	97.53	94.04	100	96.48	99.75	1.77
Self 8	507	97.97	98.14	70.63	73.04	73.72	69.76	57.61	97.97	99.79	14.21	97.88	97.56	79.56	78.10	44.13	2.86	0.56
Self 9	515	99.19	99.70	3.39	9.56	64.28	7.55	90.13	49.54	99.54	88.83	56.17	99.79	98.31	99.64	2.73	6.03	5.83
Self 10	504	99.98	99.75	99.97	68.71	98.38	98.57	97.01	96.80	99.77	91.06	98.81	99.82	97.40	73.75	68.12	20.03	0.06

declared, it must now be determined if it affects the left or the right engine.

To test the detection and identification capabilities of the proposed HMS strategy, the first five of the eight selves outlined in the Table 3 have been combined in an integrated scheme as described in the Fig. 5. The results presented and analyzed in the next section have been obtained using a hypersphere self/nonself representation; however, the approach can be expanded to other hypershapes as hypercubes, hyperellipse or even a combination of them.

V. Test Results, Analysis, and Evaluation of the FDI Scheme Performance

A. Detection Performance of the HMS Strategy

The detection outcome is a binary output produced at the sampling rate ν based on a moving time window of ω samples for each self with a detection threshold thr. The detection threshold thr represents the minimum number of points N over the window ω that trigger detectors, summed over all selves N_s, necessary for a positive detection outcome. A relative detection threshold is also defined as

$$\text{Thr} = \frac{\text{thr}}{\omega \cdot N_s} \tag{9}$$

Thus, a failure can be declared if the following condition is satisfied:

$$N \geq \text{Thr} \cdot \omega \cdot N_s \tag{10}$$

The time window ω and the relative detection threshold Thr are important design parameters that must be determined such that high detection performance is achieved. This was performed through an iterative search process by changing the values of the parameters until the detection rate DR, the number of false alarms FA, and the detection time DT achieved desired values for a set of nominal and failure flight test data.

A relationship between the design parameters and the detection time can be obtained. Assuming a perfect detection performance of each of the N_s, self configurations, then $N = Ns$ and a detection time DT $= 0$ s (detection simultaneous with the occurrence of the failure) can be obtained if

$$\omega \leq \frac{1}{\text{Thr}} \tag{11}$$

A detection time equal to the size of the sampling time, DT $= 1/\nu$ will be obtained if $\omega \leq 2/\text{Thr}$. In general,

$$\text{DT} = \frac{n}{\nu} \quad \text{if } \omega \leq \frac{n+1}{\text{Thr}}, \quad \text{for } n = 0, 1, 2, 3, \ldots \tag{12}$$

A limited set of training data at nominal and failure conditions and the first self configuration were used to determine ω and thr such that FA $\leq 2\%$, DR $\geq 90\%$, and DT ≤ 0.04 s for a fixed sampling rate $\nu = 50$ Hz. The search was performed for increasing values of n. The set of values that produced the best performance for most of the cases in the training data was $\omega = 5$ and thr $= 10$, which corresponds to Thr $= 40\%$ for the first self configuration and 29% for the second self configuration. Finally, with these values, the time elapsed from the occurrence of the abnormal situation until the moment when the detection threshold is reached was DT ≤ 0.04 s for all the failure cases investigated. The results obtained for FA and DR for the two different self configurations are summarized in Table 4.

For the first configuration, selves 1 through 5 as outlined in Table 2 were selected and integrated. For the second configuration, the selves 6 and 7 are added to the first five ones. As compared to the results presented in Table 3, for individual/isolated detector sets, the detection performance is significantly improved for any of the two configurations. For the stabilator, aileron, and structural failures, for example, the percent of detection rate even reaches 100%, while the average detection rate for the individual detector sets in which, respectively, 89.8, 57.5, and 84.2%. Not only the detection rate is improved, the false alarms are also reduced significantly, from an average of 4.2 to 2.6%.

Table 4 Detection performance of the HMS strategy, %

		Actuator failure						Sensor failure										
		Stabilator		Aileron		Rudder		LSB			LFDB			Structural failure		Engine failure		Nominal
ω	thr	L	R	L	R	L	R	p	q	r	p	q	r	L	R	L	R	FA
Configuration 1: self 1 to self 5																		
5	10	100	99.40	99.27	100	97.75	95.51	97.26	95.42	99.81	92.00	95.42	99.82	100	100	77.27	40.02	2.50
Configuration 2: self 1 to self 7																		
5	10	100	100	99.39	100	99.35	98.87	97.34	95.40	100	92.62	97.76	100	100	100	98.26	96.95	2.63

Table 5 Preidentification performance of the HMS strategy, %

	Actuator						
Failure type	Stabilator	Aileron	Rudder	Sensor	Structural	Engine	Unknown
L stabilator	98.47	0	0	0	0	1.52	0.01
R stabilator	96.61	0.42	0	0	1.87	0.18	0.92
L aileron	0	93.02	0	0	0	0	6.98
R aileron	0	91	0	4.41	3.24	1.35	0
L rudder	0	5.09	60.03	28.81	0.24	5.69	0.14
p LSB	1.31	0.90	2.76	95.03	0	0	0
q LSB	0	4.92	0.59	93.87	0	0	0.62
r LSB	0	0	0.88	97.89	0	0.31	0.92
L structural	4.88	0.43	0	0.77	93.23	0.69	0
R engine	0	4.48	0	2.33	3.19	90	0

Table 6 Identification performance of the HMS strategy: actuator subcategory failures

	Stabilator		Aileron		Rudder	
Failure type	L	R	L	R	L	R
R stabilator	100	0	——	——	——	——
L stabilator	0	100	——	——	——	——
R aileron	——	——	100	0	——	——
L aileron	——	——	0	100	——	——
R rudder	——	——	——	——	91.32	8.68
L rudder	——	——	——	——	3.2	96.80

Table 7 Identification performance of the HMS strategy: sensor subcategory failures

	LSB		
Failure type	p	q	r
p LSB	91.76	5.11	3.13
q LSB	2.52	89.53	7.95
r LSB	0.98	0.54	98.48

Table 8 Identification performance of the HMS strategy: structural subcategory failures

Failure type	R	L
R structural	100	——
L structural	——	100

Table 9 Identification performance of the HMS strategy: engine subcategory failures

Failure type	R	L
R engine	100	——
L engine	10.81	89.19

Note that the engine failure is the only one that presents a lower DR in configuration 1. In fact, none of the five selves selected for this configuration achieves a very good detection performance for this type of failure. However, configuration 2 shows a significant improvement on the detection of this type of failure. This is due to the fact that additional features relevant to engine operation such as longitudinal acceleration have been considered in self 6 and self 7. The average performance of configuration 2 in detecting engine failure is 97.6% compared to 47.7% for the individual selves.

In general, these results confirm the fact that better detection performance can be achieved by using an integrated multiple-self scheme instead of considering self configurations separately.

B. Identification Performance of the HMS Strategy

Once a failure condition is declared by the detection phase scheme, the identification phase starts to perform a preclassification according to the four categories of failures considered. Tables 5–9 summarize results for the identification of different type of failures using self configuration 2. The rate is computed with respect to the number of points for which detection was successful. Note that unknown abnormal conditions have been considered as well. This unknown condition is presented when some antibodies are activated but they do not belong to any of the specialized detector sets.

These results show that the HMS strategy can potentially lead to significant increase in performance for both the failure preidentification and identification phases.

VI. Conclusions

An integrated artificial-immune-system-based failure detection and identification scheme using a hierarchical multiself strategy has been developed, implemented, and tested using data from a 6-DOF flight simulator. Failures/damages of aircraft actuators, sensors, propulsion, and structure have been considered.

The detection and identification capabilities have been demonstrated in terms of low false alarm and high detection rates for different categories of failures. The results confirm the fact that using an integrated multiple-self approach instead of considering self configurations separately can improve significantly the detection performance while maintaining the multidimensionality of the identifier space manageable.

The HMS can produce a flexible scheme and extract the best characteristics of different features for FDI purposes. It allows consistent integration leading to improved FDI performance.

The proposed approach can potentially have a significant impact in the areas of failure detection and adaptive flight control systems by providing the tools for a comprehensive/integrated solution to the problem of aircraft subsystem FDI with reduced online computational effort.

Acknowledgments

This research effort was sponsored by NASA Aviation Safety Program through a grant within the Integrated Resilient Aircraft Control project. Special thanks to the volunteer pilots Alejandro Posada, Steven Mullins, Steven Hard, and Ondrej Karas for performing the tests in the flight simulator.

References

[1] Jones S. M., and Reveley M., "An Overview of the NASA Aviation Safety Program Assessment Process," 3rd AIAA Aviation Technology, Integration, and Operations (ATIO), AIAA Paper 2003-6706, Denver, CO, Nov. 2003.

[2] KrishnaKumar, K., and Gundy-Burlet, K., "Intelligent Control Approaches for Aircraft Applications," *JANAFF Modeling and Simulation Subcommittee*, Vol. 1, Destin, FL, 2002, pp. 189–197.

[3] KrishnaKumar, K., "Intelligent Systems for Aerospace Engineering—An Overview," *Proceedings of the von Karman Lecture Series on Intelligent Systems for Aeronautics*, May 2002.

[4] Dasgupta, D. (ed.), "Artificial Immune Systems and Their Applications," Springer–Verlag, Berlin, 1999.

[5] D'haeseleer, P., and Forrest, S., "An Immunological Approach to Change Detection: Algorithms, Analysis and Implications," *IEEE Symposium on Security and Privacy*, IEEE Computer Society, Washington, D.C., 1996, pp. 110–119.

[6] Dasgupta, D., Majumdar, N., "Anomaly Detection in Multidimensional Data Using Negative Selection Algorithm," *Proceedings of the Congress on Evolutionary Computation CEC'02*, Vol. 02, IEEE Computer Society, Washington, D.C., 2002, pp. 1039–1044.

[7] Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R., "Self-Nonself Discrimination in a Computer," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 202–212.

[8] Dasgupta, D., and Forrest, S., "Artificial Immune Systems in Industrial Applications," *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials*, IEEE Press, Piscataway, NJ, 1999, pp. 257–267.

[9] Gonzalez, F., and Dasgupta, D., "Anomaly Detection Using Real-Valued Negative Selection," *Genetic Programming and Evolvable Machines*, Vol. 4, Kluwer Academic, Norwell, MA, 2003, pp. 383–403.

[10] Farmer, J., Norman, H., Packard, S., and Perelson, A. S., "The Immune System, Adaptation, and Machine Learning," *Physica D*, Vol. 22, 1986, pp. 187–204. doi:10.1016/0167-2789(81)90072-5

[11] Karr, C., Nishita, K., and Kenneth, S., "Adaptive Aircraft Flight Control Simulation Based on an Artificial Immune System," *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, Vol. 23, 2005, pp. 295–308. doi:10.1007/s10489-005-4614-z

[12] Ko, A., Lau, H., and Lau, T., "An Immuno Control Framework for Decentralized Mechatronic Control," *Proceedings of the Third International Conference on Artificial Immune Systems (ICARIS 2004)*, Lecture Notes in Computer Science, Vol. 3239, Springer, Berlin, 2004, pp. 91–105.

[13] De Castro, L., and Timmis, J., "Artificial Immune Systems: A Novel Paradigm to Pattern Recognition," *Artificial Neural Networks in Pattern Recognition*, edited by J. M. Corchado, L. Alonso, and C. Fyfe, Univ. of Paisley, Paisley, England, U.K., pp. 67–84.

[14] Stibor, T., Timmis, J., and Eckert, C., "On the Appropriateness of Negative Selection Defined over Hamming Shape-Space as a Network Intrusion Detection System," *Proceedings of the Congress on Evolutionary Computation (CEC-2005)*, Vol. 2, IEEE Press, Washington, D.C., Sept. 2005, pp. 995–1002.

[15] Dasgupta, D., KrishnaKumar, K., Wong, D., and Berry, M., "Negative Selection Algorithm for Aircraft Fault Detection," *Proceedings of the Third International Conference on Artificial Immune Systems (ICARIS 2004)*, Lecture Notes in Computer Science, Vol. 3239, Springer, Berlin, 2004, pp. 1–13.

[16] KrishnaKumar, K., "Artificial Immune System Approaches for Aerospace Applications," 41st Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA 2003-0457, 2003.

[17] 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," AIAA Guidance, Navigation, and Control Conference, Chicago, AIAA Paper 2009-6266, Aug. 2009.

[18] Perhinschi, M. G., Moncayo, H., and Davis, J., "Integrated Framework for Aircraft Sub-System Failure Detection, Identification, and Evaluation Based on the Artificial Immune System Paradigm," AIAA Guidance, Navigation, and Control Conference (submitted for publication).

[19] 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," AIAA Guidance, Navigation, and Control Conference, Austin, TX, AIAA Paper 2003-5644, Aug. 2003.

[20] Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R., "Self-Nonself Discrimination in a Computer," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 202–212.

[21] Dasgupta, D., and Attoh-Okine, N., "Immunity-Based Systems: A Survey," *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, IEEE Computer Society, Washington, D.C., 1997, pp. 369–374.

[22] Coico, R., Sunshine, G., and Benjamini, E., *Immunology, A Short Course*, 5th ed., Wiley–Liss, New York, 2003, pp. 1–10.

[23] Perhinschi, M. G., Napolitano, M. R., Campa, G., and Fravolini, M. L., "A Simulation Environment for Testing and Research of Neurally Augmented Fault Tolerant Control Laws Based on Non-Linear Dynamic Inversion," AIAA Modeling and Simulation Technologies Conference, Providence RI, AIAA Paper 2004-4913, 2004.

[24] Perhinschi, M. G., Napolitano, M. R., Campa, G., Seanor, B., and Gururajan, S., "Design of Intelligent Flight Control Laws for the WVU F-22 Model Aircraft," AIAA Intelligent Systems Technical Conference, Chicago, AIAA Paper 2004-6282, 2004.

[25] Perhinschi, M. G., Napolitano, M. R., Campa, G., Fravolini, M. L., and Seanor, B., "Integration of Sensor and Actuator Failure Detection, Identification, and Accommodation Schemes within Fault Tolerant Control Laws," *Control and Intelligent Systems*, Vol. 35, No. 4, Dec. 2007, pp. 309–318.

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

[27] Gonzalez, F., Dasgupta, D., and Niño, L., "A Randomized Real-Valued Negative Selection Algorithm," *Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS 2003)*, Lecture Notes in Computer Science, Vol. 2787, Springer, Berlin, 2003, pp. 261–272.

[28] Perhinschi, M. G., and Moncayo, H., "Integrated System for Immunity-Based Failure Detection, Identification, and Evaluation," NASA Langley Research Center, Hampton, VA, Nov. 2008.

[29] Elkan, C., "Using the Triangle Inequality to Accelerate *k*-Means," *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, AAAI Press, Menlo Park, CA, 2003, pp. 147–153.

[30] Davis, J., Perhinschi, M. G., and Moncayo, H., "Evolutionary Algorithm for Artificial Immune System-Based Failure Detector Generation and Optimization," AIAA Guidance, Navigation, and Control Conference, Chicago, AIAA Paper 2009-5891, Aug. 2009.

[31] Wong, D., D Dasgupta, D., KrishnaKumar, K., and Berry, M., "Negative Selection Algorithm for Aircraft Fault Detection," *Proceedings of Third International Conference on Artificial Immune Systems (ICARIS 2004)*, Springer, New York, 2004, pp. 1–13.

[32] 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 Modelling and Simulation*, Vol. 26, No. 1, Jan. 2006, pp. 1–10.

[33] Perhinschi, M. G., and Napolitano, M. R., "A Simulation Environment for Design and Testing of Aircraft Adaptive Fault-Tolerant Control Systems," *Aircraft Engineering and Aerospace Technology*, Vol. 80, No. 6, Dec. 2008, pp. 620–632. doi:10.1108/00022660810911563

[34] Antoniewicz, R. F., Duke, E. L. and Patterson, B. P., "User's Manual for Interactive LINEAR, a Fortran Program to Derive Linear Aircraft Models", NASA TP 2835, 1988.