An Approach to V&V of Embedded Adaptive Systems

Yan Liu[1], Sampath Yerramalla[1], Edgar Fuller[2], Bojan Cukic[1], and Srikanth Gururajan[3]

[1] Lane Department of Computer Science and Electrical Engineering
{yanliu, sampath, cukic}@csee.wvu.edu
[2] Department of Mathematics and Institute for Math Learning
ef@math.wvu.edu
[3] Mechanical and Aerospace Engineering Department
srikanth@web.cemr.wvu.edu
West Virginia University
Morgantown WV 26506

Abstract. Rigorous Verification and Validation (V& V) techniques are essential for high assurance systems. Lately, the performance of some of these systems is enhanced by embedded adaptive components in order to cope with environmental changes. Although the ability of adapting is appealing, it actually poses a problem in terms of V&V. Since uncertainties induced by environmental changes have a significant impact on system behavior, the applicability of conventional V& V techniques is limited. In safety-critical applications such as flight control system, the mechanisms of change must be observed, diagnosed, accommodated and well understood prior to deployment.

In this paper, we propose a non-conventional V&V approach suitable for online adaptive systems. We apply our approach to an intelligent flight control system that employs a particular type of Neural Networks (NN) as the adaptive learning paradigm. Presented methodology consists of a novelty detection technique and online stability monitoring tools. The novelty detection technique is based on *Support Vector Data Description* that detects novel (abnormal) data patterns. The Online Stability Monitoring tools based on *Lyapunov's Stability Theory* detect unstable learning behavior in neural networks. Cases studies based on a high fidelity simulator of NASA's Intelligent Flight Control System demonstrate a successful application of the presented V&V methodology.

1 Introduction

The use of biologically inspired soft computing systems (neural network, fuzzy logic, AI planners) for online adaptation to provide adequate system functionality in changing environments has revolutionized the operation of realtime automation and control applications. In the instance of a safety-critical adaptive flight control system, these changes in the environment can be, for example, a stuck stabilator, broken aileron and/or rudder, sensor failure, etc. Stability and safety are two major concerns for such systems. In recent years, NASA conducted series of experiments evaluating adaptive computational paradigms for providing fault tolerance capabilities in flight control systems following sensor and/or actuator failures. Experimental success suggest significant potential for developing and deploying such fault tolerant controllers for futuristic airplanes [1–4].

The non-probabilistic evolving functionality of realtime controllers, through judicious online learning, aid the adaptive system (aircraft) to recuperate from operational damage (sensor/actuator failure, changed aircraft dynamics: broken aileron or stabilator, etc). This adds an additional degree of complexity and system uncertainty. Since it is practically impossible to estimate and analyze beforehand all possible issues relative to adaptive system's safety and stability, these systems require a non-conventional, sophisticated V&V treatment. While adaptive systems in general are considered inherently difficult to V&V, system uncertainties coupled with other real time constraints make existing traditional V&V techniques practically useless for online adaptive systems and implementation of a non-conventional V&V technique a challenging task [5, 6]. This (in)ability to provide a theoretically valid and practically feasible verification and validation remains one of the critical factors limiting wider use of neural networks based flight controllers [5–7].

We propose a non-conventional V&V approach and derive a validation methodology suitable for online adaptive systems. We apply our approach to an adaptive flight control system that employs Neural Networks (NN) as the adaptive learning paradigm. Presented V&V methodology consists of an online novelty detection technique and online stability monitoring tools. The novelty detection technique is based on Support Vector Data Description (SVDD) in order to detect novel (abnormal) data patterns. As a one-class classifier, the support vector data description is able to form a decision boundary around the learned data domain with very little or even zero knowledge outside the boundary. The online stability monitoring tools based on Laypunov stability theory are designed to detect unstable (unusual) NN behavior. The underlying mathematics of the online monitoring tools is a rigorous mathematical stability verification technique. This technique emphasizes the need for a precise stability definition for adaptive systems and reasons about the self-stabilizing properties of the adaptive neural network within the control system's architecture.

1.1 Paper Overview

We propose a V&V framework that is suitable for online adaptive systems in Section 2. The presented validation approach requires an understanding of two complementary novelty detection and stability analysis techniques that are discussed in detail in Sections 3 and Section 4. In Section 5, test cases and simulation results describing the operational behavior of the online novelty and stability analysis are discussed in detail. We conclude the paper with a brief discussion on the prospects of the presented V&V approach for other online adaptive systems in Section 6.

2 A V&V Framework

One of the goals of our V&V and safety assurance approach is to ensure the correct diagnosis followed by blocking/permitting of novel (abnormal or unreliable) data from entering the online adaptive component, the neural network. We propose to use novelty detectors and safety monitors as online filters [8]. Figure 1 illustrates the V&V framework. The SVDD data analysis technique is capable of detecting anomalies in the neural network's inputs and outputs. Safety monitors disallow the propagation of unsafe controller gains (adjustments) from entering the controller. It is evident that such a device must require a wide range of system (aircraft) domain-knowledge. Therefore, we seek to define a control error adjustment and detection technique suitable for alerting from anomalous, unstable, and eventually unsafe aircraft behavior if the outputs from neural network adaptation were to enter the controller.

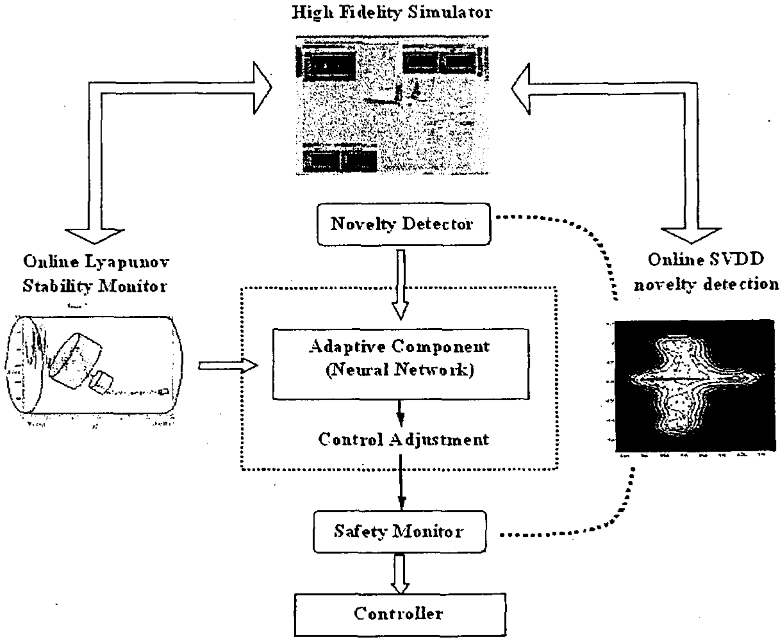

High Fidelity Simulator
Novelty Detector
Online Lyapunov Stability Monitor
Online SVDD novelty detection
Adaptive Component (Neural Network)
Control Adjustment
Safety Monitor
Controller

Fig. 1. Adaptive Flight Control System's V&V Framework.

Another key step of the validation framework is the runtime stability monitor. Its goal is to determine whether, under given flight conditions, the neural network converges, i.e., if its state transition trajectories lead to a stationary state. The online monitor is complemented by mathematical stability proofs [9] that can define its engagement or disengagement. In other words, to preserve computational resources the online monitor may not be engaged in flight conditions that are considered to be *a priori* safe.

3 Novelty Detection Technique

In general, novelty detection techniques require beforehand knowledge of both nominal and off-nominal flight domains. However, for the validation of NN in online adaptive systems, it is impossible to anticipate all possible adverse environmental conditions and/or failure modes. Under flight failure scenarios, the performance of most regular classification models deteriorate due to restrictions in their generalization capabilities and low quality data. As a one-class classification tool, Support Vector Data Description (SVDD) technique is derived from Support Vector learning theory by Tax et. al. [10,11]. Differing from general support vector classifiers that decide the maximum margin hyperplane to separate two classes, SVDD method tries to find an optimal decision boundary for a given data set. Thus, it provides the best possible representation of the target-class and offers inferences that can be used to detect the outliers from the nominal feature space. This, for our validation purposes, can be defined as the *"safe region"*, relating to nominal flight conditions.

SVDD is developed from the concept of finding a sphere with the minimal volume to contain all data [12–14]. Given a data set S consisting of N examples $x_i, i = 1, .., N$, the SVDD's task is to minimize an error function containing the volume of this sphere. With the constraint that all data points must be within the sphere, which is defined by its radius R and its center a, the objective function can be translated into the following form by applying Lagrangian multipliers,

$$L(R, a, \alpha_i) = R^2 - \sum_i \alpha_i \{R^2 - (x^2 - 2ax_i + a^2)\}$$

where $\alpha_i > 0$ is the Lagrange multiplier. L is to be minimized with respect to R and a and maximized with respect to α_i. By solving the partial derivatives of L, we also have:

$$\sum_i \alpha_i = 1;$$

and

$$a = \sum_i \alpha_i x_i,$$

which gives the Lagrangian with respect to α_i:

$$L = \sum_i \alpha_i (x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j)$$

where $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$.

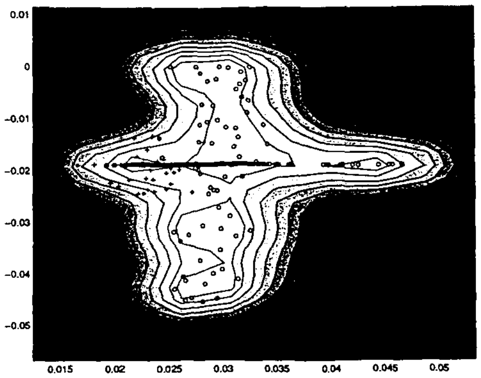


Fig. 2. SVDD with different distances from the center.

In the solution that maximizes L, a large portion of α_i's become zero. The rest of α_i's are greater than zero and their corresponding objects are those called support objects. They lie on the boundary that forms a sphere that contains the data. Hence, object z is accepted by the description when:

$$\|z-a\|^2 = (z - \sum_i \alpha_i x_i)(z - \sum_i \alpha_i x_i) \leq R^2.$$

Real world systems usually produce multi-dimensional highly nonlinear data that are inseparable by a linear discriminant. This makes the data description harder to obtain. Similar to the Support Vector Machine (SVM) [10], by replacing some kernel function $K(x,y)$ with the product of (x,y) in the above equations, we are able to map our data from a high dimensional space onto a Hilbert space, which is also referred to as the "feature space". In the feature space, objects can be classified with lower complexity. Selecting the well-known Gaussian kernel function, where $K(x,y) = exp(-\|x-y\|^2/s^2)$, we now have:

$$L = 1 - \sum_i \alpha_i^2 - \sum_{i \neq j} \alpha_i \alpha_j K(x_i, x_j).$$

The formula of checking object z now becomes:

$$1 - 2\sum_i \alpha_i K(z, x_i) + \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \leq R^2.$$

Since the SVDD is used as a one-class classifier, in practice, there are no actual outliers well defined other than those randomly drawn from the rest of the space outside the target class. However, by applying the SVDD method, we can obtain a relatively sound representation of the target class. To detect outliers, a more precise criteria should be inferred from empirical testing or pre-defined thresholds. By setting the boundaries to a certain distance from the center, Figure 2 illustrates the different boundaries with respect to different parameter settings. A rule of thumb here is that the greater the value of the distance from the center, the rougher the boundary. Therefore, the number of the outliers that can be detected decreases. In practice, a pre-defined threshold can be used as the furthest distance of a data point from the center, which the system can tolerate. Such pre-defined thresholds need sufficient testing within each specific data domain.

4 Online Monitoring

Self-organizing neural networks, introduced by Kohonen [15] and modified by several others [17–19] over the last twenty years, offer topology-preserving adaptive learning capabilities that can, in theory, respond and learn to abstract from a much wider variety of complex data-manifolds, the type of data encountered in an adaptive flight control system.

The adaption of neural networks can successfully model the topology and abstract the information from data patterns that have a predictable structure. However, during online adaptation, the data patterns may be presented to the network at a varying sampling rates. The presented data can exhibit pathological dimensional stratification, such as uniformity or functional discontinuities. It has been observed (experimentally) that under these circumstances, the neural network encounters difficulties in learning and abstracting information from the presented data, eventually leading to a deteriorating network performance. In such cases the neural network might fail in its primary goal *"to successfully learn and provide a better estimate of the learnt parameters to the flight controller"*. This degradation in the network's performance is depicted in a loss of its self-stabilizing properties. The goal of an online stability monitor is to capture and analyze the self-stabilizing properties of the network in the hope that it will be able to detect unstable neural network behavior and warn the pilot/system of the imminent threat to the controller.

The construction of an online stability monitor is based on rigorous mathematical stability analysis methodology, *Lyapunov's direct method* [16]. According to this method, a system is said to be stable near a given solution one can construct a Lyapunov function (scalar function) that identifies the regions of the state space over which such functions decrease along some smooth trajectories near the solution. In the discrete sense, Lyapunov stability can be defined as follows:

Definition 1. ***Lyapunov Stability***
If there exists a Lyapunov function, $V : \mathbb{R}^O \rightarrow \mathbb{R}$, defined in a region of state space near a solution of a dynamical system such that

1. $V(0) = 0$
2. $V(x) > 0 : \forall x \in O, x \neq 0$
3. $V(x(t_{i+1})) - V(x(t_i)) = \Delta V(x) \leq 0 : \forall x \in O$

then the solution of the system is said to stable in the sense of Lyapunov.

$x = 0$ represents a solution of the dynamical systems and $\mathbb{R}^O$, O represent the output space and a region surrounding this solution of the system respectively.

According to the above definition a system is stable if all solutions of the state that start nearby end up nearby. A good distance measure of nearby must be defined by a Lyapunov function (V) over the states of the system. By constructing V, we can guarantee that all trajectories of the system converge to a stable state. The function V should be constructed keeping in mind that it needs be scalar ($V \in \mathbb{R}$) and should be non-increasing over the trajectories of the state space. This is required in order to ensure that all *limit points* of any trajectory are stationary.

Definition 2. ***Asymptotic Stability (AS)***
If in addition to conditions 1 and 2 of Definition 1, the system has a negative-definite Lyapunov function

$$\Delta V(x) < 0 : \forall x \in O \tag{1}$$

then the system is Asymptotically Stable.

Asymptotic stability adds the property that in a region surrounding a solution of the dynamical system trajectories are approaching this given solution asymptotically.

Definition 3. ***Global Asymptotic Stability (GAS)***
If in addition to conditions 1 and 2 of Definition 1, the Lyapunov function is constructed such that,

$$\lim_{t \to \infty} V(x) = 0 \tag{2}$$

*over the **entire** state space then the system is said to be Globally Asymptotically Stable.*

A notable difference between AS and GAS is the fact that GAS implies any trajectory beginning at *any* initial point will converge asymptotically to the given solution, as opposed to AS where only those trajectories beginning in the neighborhood of the solution approach the solution asymptotically. The types of stability defined above have increasing property strength.

Global Asymptotic Stability $\Longrightarrow$ Asymptotic Stability $\Longrightarrow$ Lyapunov Stability.

The reverse implication does not necessarily hold as indicated by the Venn diagram of Figure 3. Thus a strict Lyapunov function should force every trajectory to asymptotically approach an equilibrium state. Even for non-strict Lyapunov functions, it is possible to guarantee convergence by LaSalle's invariance principle. In mechanical systems a Lyapunov function is considered as an energy minimization term. In economic and finance evaluations it is considered

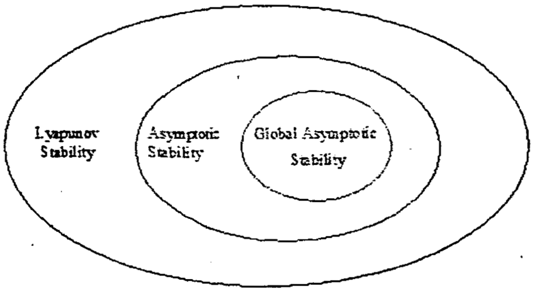


Fig. 3. Relative strengths of Stability

as a cost-minimization term, and for computational purposes it can be considered as an error-minimization term. Figure 4 shows a Lyapunov function for the NN operation where the decreasing cylinder radii indicate a converging, stable operation. The online stability monitor essentially computes Lyapunov and Lyapunov-like functions (similar to the one shown in Figure 4) based on the current states of the neural network learner and analyze each function to evaluate the overall network stability. Thus, online stability monitoring complements analytical stability analysis techniques by being being able to detect system states that deviate away from stable equilibria in real-time.

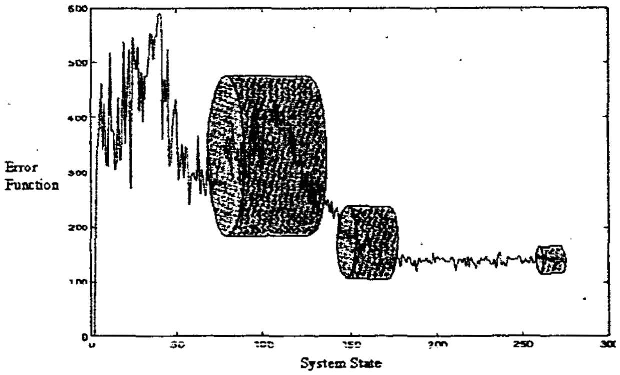


Fig. 4. A converging Lyapunov-like function.

5 Case Study

The knowledge gained through the design and evaluation of new control schemes is of direct use in performance verification and validation. Proper experimentation is required to justify realism and applicability of the proposed techniques into actual practice.

5.1 The Intelligent Flight Control System

The Intelligent Flight Control System (IFCS) was primarily developed by NASA as a novel flight control system with the primary goal to *"flight evaluate control concepts that incorporate emerging soft computing algorithms (NN or AI techniques) to provide an extremely robust aircraft capable of handling multiple accident and/or an off-nominal flight scenario"* [1, 2, 7].

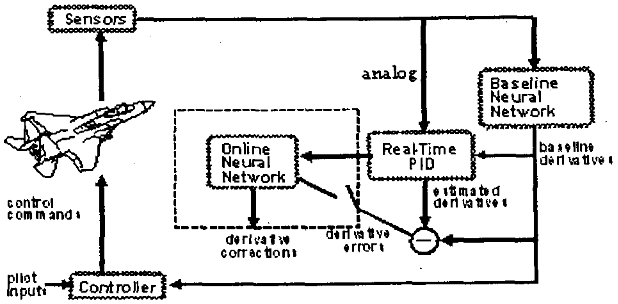


Fig. 5. The Intelligent Flight Control System.

The diagram of Figure 5 shows the architecture of the IFCS using Dynamic Cell Structure (DCS) neural network, referred to as the Online Learning Neural Network (OLNN). The control concept can be briefly described as follows. Notable discrepancies from the outputs of the the Baseline (Pre-trained) Neural Network (PTNN) and the Real-time Parameter Identification (PID), either due to a change in the aircraft dynamics (loss of control surface, aileron, stabilator) or due to sensor noise/failure, are accounted by the Online Learning Neural Network. The primary goal of OLNN is to learn online and provide a better estimate for future use of these discrepancies, commonly known as Stability and Control Derivative errors. The critical role played by the online learning neural network in fine-tuning the control parameters and providing a smooth control adjustments is the motivation for the need for a practical, nonconventional validation methodology.

Major advances in the development of modern control laws have generated the need for developing very detailed and sophisticated simulation environments for R&D purposes. Novel techniques for adaptive flight control achieves maturity through extensive experimentation in simulated environments. Figure 6 shows the interface of the IFCS F-15 simulator developed by the WVU research team. The control framework of the simulator is based on the IFCS architecture, shown in Figure 5. Through the high fidelity simulator, we are able to collect valuable data representing nominal flight conditions as well as some failure scenarios.

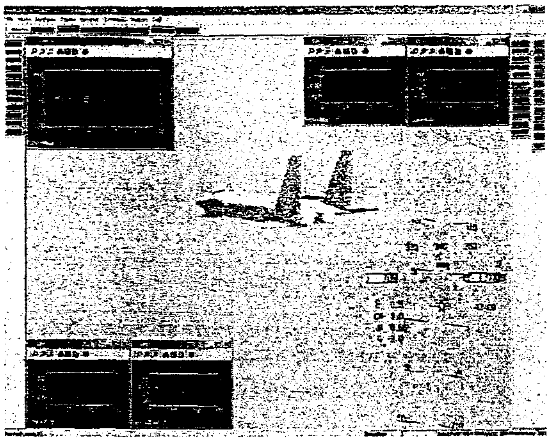


Fig. 6. NASA-WVU F-15 Simulator

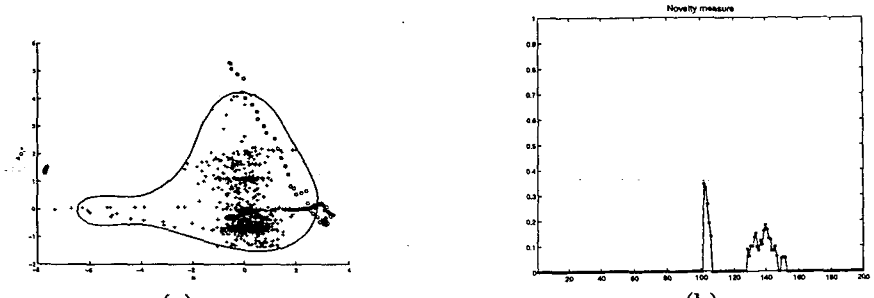
5.2 Flight-Data Description

The simulation data depicts nominal and off-nominal flight conditions of approximately 10 seconds of flying time corresponding to 200 frames of data at the simulation rate of 20Hz. A data frame is a point in a seven-dimensional space corresponding to 4 sensor readings (independent variables) and 3 stability and control derivative errors from PID and PTNN (dependant variables). The NN tested here is the $DCS - C_z$ network, one of the five DCS-subnetworks of the IFCS. The independent variables are Mach number (the ratio of the speed of the aircraft to the local speed of sound), alpha (aircraft's angle of attack) and altitude of the aircraft. The dependent variable are three stability and control derivative errors generated by the difference between PID and PTNN.

In the following sections, we first present novelty detection results using SVDD on the NN training data. Online stability monitoring results for NN learning are described next. Both tools are tested on two failure mode data sets obtained from the simulator. The two specific types of failures induced in the IFCS simulator are control surface failures (stuck aileron, stabilator) and loss of control surface. A control surface failure (locked left stabilator, stuck at +3 Degree) is simulated into the system at the 100^{th} data frame. In another simulation, a loss of control surface (50% missing surface of right aileron) failure is also induced at the 100^{th} data frame.

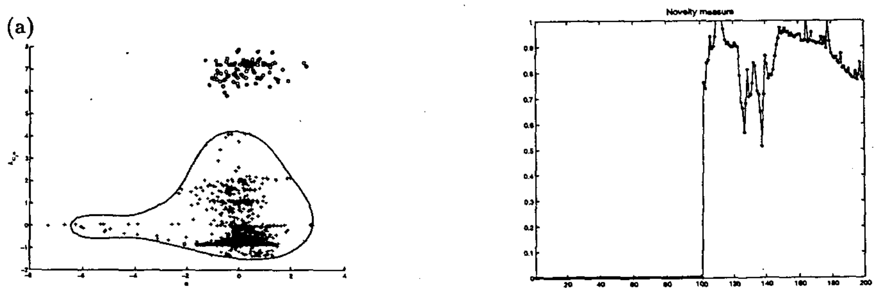
5.3 Novelty Detection Using SVDD

We first simulate one run of nominal flight conditions of 40 seconds with a segment of 800 data points saved. After running SVDD on the nominal data, we obtain a sound data description of nominal flight conditions. The data description is then used to detect novel data that falls outside the boundary. The crosses in Figure 7(a) and Figure 8(a) represent the nominal data points on which the boundary is found by SVDD.



(a) (b)

Fig. 7. Novelty detection results using SVDD on control surface failure simulation data. (a): SVDD of nominal flight simulation data is used to detect novelties. (b): Novelty measures returned by SVDD tool for each testing data point.



(a) (b)

Fig. 8. Novelty detection results using SVDD on loss of control surface failure simulation data. (a): SVDD of nominal flight simulation data is used to detect novelties. (b): Novelty measures returned by SVDD tool for each testing data point.

We then use the boundary formed by SVDD to test on failure mode simulation data. Novelty detection results for control surface failure simulation data and loss of control surface failure simulation data are shown in Figure 7 and Figure 8 respectively. Circles in Figure 7(a) and Figure 8(a) represent failure mode simulation data. In the plot of Figure 8(a), depicting the loss of control surface failure, a large portion of failure mode data falls outside the boundary. The loss of control surface failure indicates a more substantial damage than the stuck-at type of failure. Consequently, the data points in Figure 8(a) fall further outside the nominal data boundary than the data points in Figure 7(a). The novelty measures shown in Figure 7(b) and Figure 8(b) are probability-like measures computed for each data point based on the distance from the SVDD boundary formed on the nominal flight condition data. Correspondingly, in plots of Figure 7(b) and Figure 8(b), we can see that the novelty measures of loss of control surface failure data after 100^{th} data frame are larger than those of control surface failure data. In both figures, after the 100^{th} point, when failures occurred, SVDD detects the abnormal changes and returns with the highest novelty measures. This demonstrates the reasonably effective and accurate detection capabilities of our SVDD detector.

5.4 Online Stability Monitoring

Described novelty detection mechanisms provide an independent approach to reliable failure detection, thus enhancing the ability of the system analyst to evaluate the mechanisms in charge of the activation adaptive component(s). Online stability monitors serve the purpose of evaluating whether adaptive subsystem provide adequate accommodation capabilities that address specific environmental conditions. In other words, the monitors track the adaptation process and continually evaluate the difference between the current state abstraction provided by the learning device (DCS neural network in our case study) and its desired goal.

Adaptive systems are associated with uncertainty, many degrees of freedom and high noise-level in real flight conditions. Due to their complexity, we may not always be able to check to see if each dimension of the input data is effectively abstracted and represented by the neural network. Lyapunov theory provides the tool to collapse the multidimensional evaluation criteria into one or a few meaningful bounded functions. The data sets being modeled in the case study represent short data sequences for one out of five neural networks in the intelligent flight control system. We constructed four Lyapunov-like functions to reduce the need for checking effective learning by each dimension. Rather than looking onto several dozen graphs, the adequacy (stability) of learning can be assessed from the analysis of these four graphs, representing the Lyapunov functions.

The four Lyapunov-like functions are specific for the DCS neural network of the Intelligent Flight Control System. Their formal description would require detailed presentation of the DCS learning algorithm, which is outside of the scope of this paper. In general terms, the DCS network is a so called self-organizing map. Self-organizing maps evolve their topology to reflect as closely as possible the topological characteristics of the data set being approximated. Therefore, by measuring euclidian distances within the evolving network and comparing them with actual distances in the training data set, we may derive the measure of the goodness of approximation. The four Lyapunov like functions were defined because they evaluate different aspects of DCS adaptation: the Kohonen's rule and the competitive Hebbian rule [19, 20]. Furthermore, we noticed that these four functions react with different intensities to different training data sets. Given that these data sets represent actual aircraft failure scenarios, selected Lyapunov-like functions complement each other.

As the neural network starts to adapt to the presented failure mode data, the run-time monitor is engaged. It continually monitors the behavior of the neural network. Figure 9 shows the plots of the four Lyapunov-like monitors before a control surface failure (locked left stabilator, stuck at +3 Degree) is induced into the system, and before it propagates into the neural network. Figure 9 shows no predominant spikes in the individual monitors, indicating the lack of intense adaptation in nominal conditions. Because the neural network does not attempt to change the control input to the flight control system, its output bears very limited overall system risk during this period.

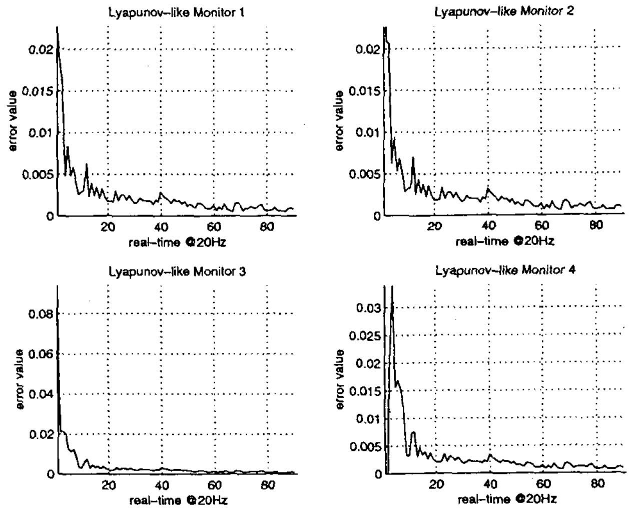


Fig. 9. Online Monitors: Pre-control Surface Failure

Figure 10 shows the plots of the four Lyapunov-like monitors after the control surface failure (locked left stabilator, stuck at +3 Degree) is induced into the system and after the failure propagates into the neural network. Figure 11 shows the plots of the four Lyapunov-like monitors after the loss of control surface (50% missing surface of right aileron) is simulated into the system and after the failure propagates into the neural network. The plots show a predominant spike at time frame 100 (the time of the failure). The spikes indicate the successful detection of the unusual (failed) environmental condition by monitoring the internals of the neural network. In the short term, the neural network undergoes a significant degree of adaptation. The high values of the Lyapunov-like functions indicate that the neural network needs additional time (and learning cycles) to faithfully represent its newly arrived (in real-time) input data set. During this period, the confidence on neural network's output diminishes drastically, i.e., the network is not providing the desirable failure accommodation. But, Within the next 50 or so frames in Figure 10, the values of Lyapunov-like monitors approach 0, indicating that the failure has been accommodated through adaptation. The failure accommodation delay is longer in Figure 11, an expected indication of the severe failure condition (the loss of a control surface). At this point, a verification and validation engineer needs to assess the adequacy of the failure accommodation mechanism with respect to the overall system safety requirements, evaluate alternative designs, and prepare suitable V&V recommendations to the safety board.

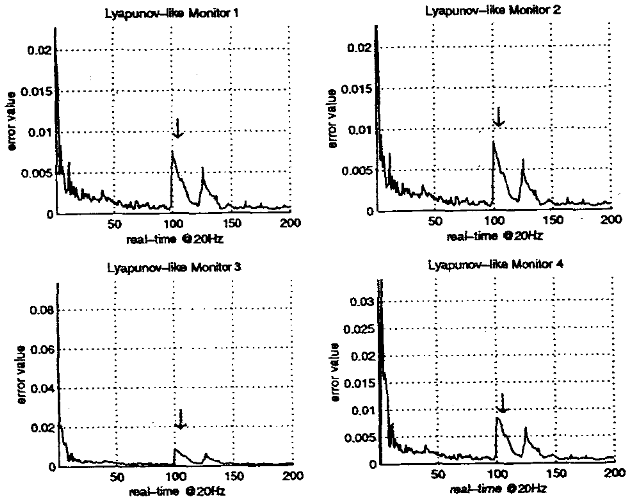


Fig. 10. Online Monitors: Post-control Surface Failure

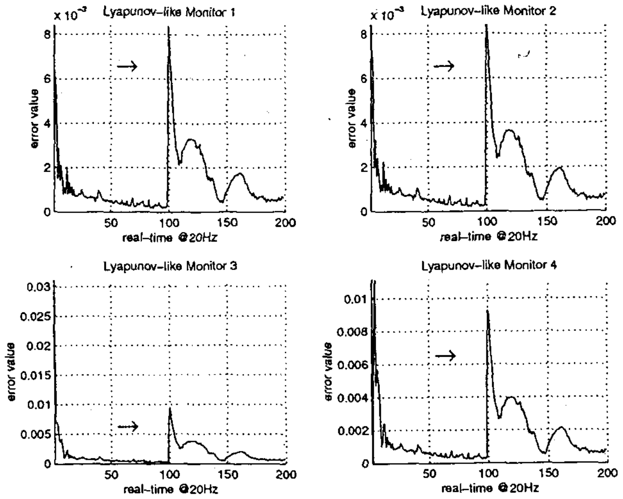


Fig. 11. Online Monitors: Post-Loss of Control Surface Failure

6 Conclusions

We developed a non-conventional approach for validating the performance adequacy of the neural network embedded in an online adaptive flight control system. The validation framework consists of

- Online filters (novelty detectors) that check the validity of inputs and control outputs, and
- Runtime stability monitors that examine the stability properties of the neural network adaptation.

Experimental results from the data collected on an F-15 aircraft flight simulator show that:

1. SVDD can be adopted for defining nominal performance regions for the given application domain. Our techniques provided successfully automated separation between faulty behaviors and normal system events in real-time operation.
2. Based on the originally developed concept of Lyapunov-like functions applied for the first time to neural network learning, the online stability monitors have shown a successful realization of convergence tracking of adaptation error towards a stable (or unstable) and safe (or unsafe) state in the adaptive flight control system.

We conclude that the proposed methodology provides a good approach for validating online adaptive system's safety, stability and performance. The observed efficiency and scalability of both methods give us the expectation that the proposed V&V method can be successfully applied to other types of online adaptive learners.

Acknowledgment

This work was funded in part by grants to West Virginia University Research Corp. from the NASA Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP) managed through the NASA Independent Verification and Validation (IV &V) Facility, Fairmont, West Virginia, NASA research grant No. NAG4-163; and from the NSF CAREER award CCR-0093315.

References

[1] The Boeing Company, Intelligent flight control: advanced concept program, Technical report, 1999.

[2] Charles C. Jorgensen. Feedback linearized aircraft control using dynamic cell structures, *World Automation Congress* (ISSCI), Alaska, 1991, 050.1-050.6.

[3] M. Napolitano, C. D. Neppach, V. Casdorph, S. Naylor, M. Innocenti, and G Silvestri, A neural network-based scheme for sensor failure detection, identification and accomodation, *AIAA Journal of Control and Dynamics*, Vol. 18, No. 6, 1995, 12801286.

[4] Institute of Software Reseach. Dynamic cell structure neural network report for the intelligent flight control system, Technical report, Document ID: IFC-DCSR-D002-UNCLASS-010401, January 2001.

[5] Johann Schumann, and Stacy Nelson, Towards V&V of neural network based controllers. *Workshop on Self-Healing Systems*, 2002.

[6] D. Mackall, S. Nelson, and J. Schumann. Verification and validation of neural networks of aerospace applications, Technical report, CR-211409, NASA, 2002.

[7] M.A. Boyd, J, Schumann, G. Brat, D. Giannakopoulou, B. Cukic, and A. Mili, Validation and verification process guide for software and neural nets, Technical report, NASA Ames Research Center, September 2001.

[8] Y. Liu, S. Yerramalla, E. Fuller, B. Cukic and S. Gururajan. Adaptive Control Software: Can We Guarantee Safety? *Proc. of the* 28^{th} *International Computer Software and Applications Conference, workshop on Software Cybernetics*, Hong Kong, September 2004.

[9] S. Yerramalla, E. Fuller, B. Cukic. Lyapunov Analysis of Neural Network Stability in an Adaptive Flight Control System, *6th Symposium on Self Stabilizing Systems (SSS-03)*, San Francisco, CA, June 2003.

[10] V. N. Vapnik, Statistical learning theory, Wiley, 1998.

[11] D.M.J. Tax, and R.P.W. Duin, Outliers and data descriptions, *Proc. ASCI 2001, 7th Annual Conf. of the Advanced School for Computing and Imaging* (Heijen, NL, May 30-June 1), ASCI, Delft, 2001, 234-241.

[12] D.M.J. Tax, and R.P.W. Duin, Support vector domain description, *Pattern Recognition Letters*, Vol. 20, No. 11-13, 1999, 1191-1199.

[13] D.M.J. Tax, and R.P.W. Duin, Data domain description using support vectors, *Proc. European Symposium on Artificial Neural Networks* (Bruges, April 21-23, 1999), D-Facto, Brussels, 1999, 251-257.

[14] D.M.J. Tax, A. Ypma, and R.P.W. Duin, Support vector data description applied to machine vibration analysis, *Proc. 5th Annual Conference of the Advanced School for Computing and Imaging* (Heijen, NL, June 15-17), ASCI, Delft, 1999, 398-405.

[15] Teuvo Kohonen, The self-organizing map, *Proc. of the IEEE*, Vol. 78, No. 9, September, 1990, 1464-1480.

[16] V. I. Zubov, Methods of A. M. Lyapunov and their applications, U.S. Atomic Energy Commission, 1957.

[17] Bernd Fritzke, Growing cell structures - a self-organizing network for unsupervised and supervised learning, *Neural Networks*, Vol. 7, No. 9, May 1993, 1441-1460.

[18] Thomas Martinetz, and Klaus Schulten, Topology representing networks, *Neural Networks*, Vol. 7, No. 3, 1994, 507-522.

[19] Jorg Bruske, and Gerald Sommer, Dynamic cell structure learns perfectly topology preserving map, *Neural Computation*, Vol. 7, No. 4, 1995, 845-865.

[20] N.Rouche, P.Habets, and M.Laloy, Stability theory by Liapunov's direct method, Springer-Verlag, New York Inc., 1997.