ADAPTIVE VERIFICATION FOR AN ON-LINE LEARNING NEURAL-BASED FLIGHT CONTROL SYSTEM

Ronald L. Broderick, Graduate School of Computer and Information Science, Nova Southeastern University, Ft. Lauderdale, FL

Abstract

This paper presents a complex adaptive systems approach for the verification of an adaptive, online learning, sigma-pi neural network that is used for the Intelligent Flight Control System (IFCS) that has the potential of commercial aviation application. This paper reports on the partial completion of my doctoral dissertation proposal at Nova Southeastern University, in the Graduate School of Computer and Information Sciences. The most significant shortcoming of the prior and current approaches to verifying adaptive neural networks is the application of linear approaches to a non-linear problem. The project will use a MatLab simulation of the sigma-pi adaptive neural network and an aircraft simulation to fly a series of simulated flight tests. As a result of the flight simulations, a statistical analysis of the neural network weights is performed as input to both a complexity analysis and a neural network rule extraction analysis. Complex adaptive methods are a novel approach to overcome previous linear analysis limitations. Future work will be required to analyze emergent behavior of the neural network weights to show stability and convergence characteristics. Advances in computational power and neural network techniques for estimating aerodynamic stability and control derivatives provide opportunity for real-time adaptive control. New verification techniques are needed that substantially increases trustworthiness in the use of these neural network systems in life critical systems. Verification of neural-based IFCS is currently an urgent and significant research and engineering topic since these systems are being looked upon as a new approach for aircraft survivability, for both commercial and military.

Introduction

Adaptive neural networks have shown great promise in their application to address very complex, dynamical, real-time intractable problems. The purpose of this paper is to propose a complex adaptive systems approach for the verification of an adaptive, online learning, sigma-pi neural network that is being developed by NASA for IFCS applications. For commercial avionic systems certification, the Radio Technical Commission for Aeronautics (RTCA) industry standard DO-178B [1] titled "Software Considerations in Airborne Systems and Equipment Certification" is recognized and adhered to by the Federal Aviation Administration as the means of demonstrating compliance with the Federal Aviation Regulations. For commercial flight control systems, certification is mandatory. This certification requirement imposes stringent verification requisites. Any attempt at the verification of an adaptive neural network system must address substantially different issues compared to traditional software.

The most significant issue of all verification efforts is trustworthiness. Traditional verification methods do not guarantee correctness; they only provide objective evidence that the system performs as specified by its requirements. On-line learning neural network (OLNN) systems are able to address very complex problems, but they raise very difficult verification issues. New verification techniques are needed that substantially increases confidence in the use of these neural network systems in life, safety, and mission critical systems. This paper will investigate the application of methods, models and techniques from the science of complex adaptive systems. It will attempt to use complex adaptive system theory to determine an approach to define the rules of the adaptive sigma-pi neural network.

Background

For the past several years, one of the most interesting applications of neural network systems is the IFCS application. For the IFCS application, artificial neural networks have been extensively researched at NASA Ames Research Center because of their powerful ability in approximating

0-7803-9307-4/05/$20.00 ©2005 IEEE.

non-linear dynamical functions. Current research being performed by Georgia Institute of Technology [2], Boeing Phantom Works [3], and NASA Dryden Flight Research Center [4] is based on a Generation II IFCS, as shown in Figure 1.

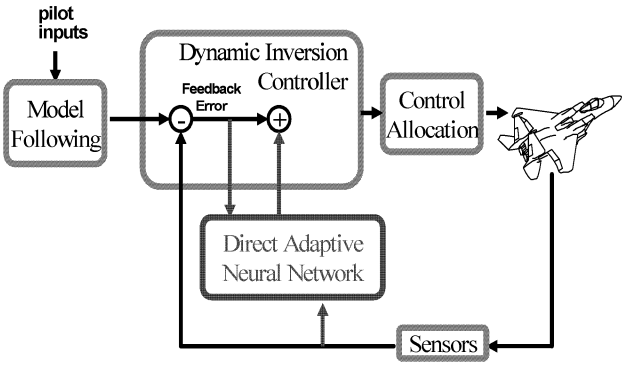

Figure 1. Gen II IFCS Architecture

Based on a dynamic inversion model, the pilot commands are combined with the aircraft sensor readings to calculate a desired command output. The desired command output is fed into the inversion model of the aircraft which provides the actual command values that are provided to the aircraft's control surfaces. In the nominal case, the inversion model is an exact inversion function of the aircraft dynamics. Due to changes in the aircraft dynamics from system failure or damage, there are deviations in which the adaptive sigma pi neural network attempts to minimize the deviation by learning from the sensor readings and previous command output. The neural network output is an adjusted value of the command output.

Prior Research

Because of the importance of neural network applications, there is a rich collection of prior and current research to reference. The major themes of this research have been statistical methods, formal methods, rule extraction, numerical analysis, stability analysis and run-time monitoring. The goal is to decrease both system output error and system output error variability. The usefulness of neural networks has motivated the implementation of many systems. Unfortunately, the uncertainty of neural network output has not been addressed in many implementations. In safety-critical systems, output error variability must be evaluated.

A decade ago, in the paper titled "A foundation for neural network verification and validation" G. E. Peterson, [5] proposed to enhance the current verification and validation approach. This paper also introduced the idea of statistically evaluating true and apparent error. In addition, Peterson addressed the subject of a neural networks boundary of acceptable behavior and used a confidence statistic to express the performance within the boundary.

Formal methods have been proposed by a number of research papers as a verification approach for neural networks. High quality software often uses formal method techniques from the disciplines of logic and discrete mathematics to develop system specifications, design and construction. Software developed using formal methods enable the formalization of verification and validation. When used for testing purposes, formal methods reduce the reliance on human intuition and judgment. Traditional formal methods include model checking and theorem proving. The serious limitation of formal methods is the rigor of mathematical proof which decreases the practical size of the software under consideration [6].

Rule Extraction is a method that has been used to determine the functionality of a neural network by mapping its inputs to its outputs. The rules extracted represent a set of if-then statements. Rule extraction applies best to fixed neural networks that have been trained for a specific application. In the passed, rule extraction has been successfully applied to multilayered perception [7], local cluster and radical basis function neural networks. A current research paper supporting rule extraction is [8]. The major limitation of rule extraction is that it supports static neural networks but does not support adaptive neural networks.

In the paper titled "Towards Developing Verifiable Neural Network Controller" [9], the authors raise the question of completeness of training data and flight envelope. The generation of training data cannot rely solely on recorded flight data. This data would not contain the environment the aircraft would experience in a damaged condition. Also, what is the performance of the neural network in the "vicinity of the boarders of the flight envelope?" In a more recent paper, "Towards V&V of Neural Network Based

Controllers", [10], authors propose modification of current verification methods and attention to numerical aspect of neural network verification. In the recent paper titled "Certifying Adaptive Flight Control Software" [11], the authors take a fault tolerance approach including both sensor failure and actuator failure. With the goal of establishing the "foundations for software certification of adaptive flight control systems", the authors approach is to evaluate the limitations of various combined verification techniques, including formalization of requirements specifications, their formal verification, and techniques for proving the convergence and stability properties of neural networks.

Lyapunov stability analysis has played an important role in the verification of neural networks. Lyapunov stability analysis can be used during a network's development or during its operation. Lyapunov's direct (second) method is widely used for stability analysis of linear and non-linear systems, both time invariant and time varying (neural networks are non-linear, time-varying systems). Viewed as a generalized energy method, it can provide insight into a system's behavior and is used to determine if a system is stable, unstable, or marginally stable [12]. The Lyapunov stability analysis is used to prove the self-stabilizing characteristics of the IFCS [13]. This analysis proves that the neural network is convergent, but to ensure system stability, run time monitoring is required to assure system robustness.

Run-Time Monitoring is used to address the possibility of the adaptive neural network encounters unusual data patterns. The run-time monitor is used to detect deviations of state that could lead to unstable behavior. In the current research literature, run-time or operational monitoring methods appear to be the current evolution of verification for neural networks [14]. Run-time monitoring involves evaluating the neural network during execution, and/or evaluating information such as event logs collected during execution. Information collected can be used to detect either violation of system constraints or to manage resources at runtime. Run-time monitoring or data sniffing can help assess the validity of input or output of the neural network [15]. The pre-processing of neural network input determines if the input will cause unexpected adaptations in the system. The post-processing approach evaluates the neural network outputs and before they are used determines if it is outside the normal range of values and prevents it from being used. In actuality, this monitoring is an indication of an incomplete method of adaptive neural network verification.

The Approach

The approach of this paper is to use an adaptive analysis to verify the online, learning sigma-pi neural network of the Intelligent Flight Control System (IFCS). The following statistical analysis will provide a framework to determine, for a given set of flight condition inputs, which neural network weights actually change to produce a stable range of command error. The adaptive analysis will attempt to determine the rules that govern the weights changes of the sigma-pi neural network using the techniques of the new science of complex adaptive systems.

The Generation II IFCS uses a direct adaptive approach in which the command error is nulled. The command errors are use to transform the pilot input to the computed aircraft control surface commands in the aircraft closed-loop control system. As shown in Figure 2, the process of using state information to govern the control inputs is known as closing the loop, and the resulting system as a closed-loop control or feedback control.

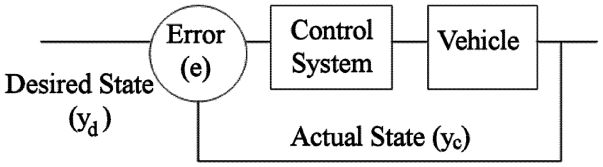

Figure 2. A Simple Closed-Loop Control System

The aircraft controller compares the desired control commands (y_d) to the computed control commands (y_c) and determines the aircraft controller error (e) to be used in the next cycle of aircraft control.

Statistical Analysis

For certification purposes, the certification flight envelope would be a limited set of the neural

networks inputs: Mach (M), altitude (h) and aircraft pitch angle (α). Thus

$$x(t) = (M(t), h(t), \alpha(t))^T , 0 \leq t \leq T$$

where T is the observation time interval. For a realistic flight envelope, environmental conditions must also be modeled to account for pressure, density, temperature, and speed-of-sound variations with altitude. The test space of all x(t) is infinite and is far to rich to be covered by a reasonable number of test cases. To cover a reasonable set of certification tests at each point in the flight envelope, the certification flight envelope set would be further reduced to a finite interval and to eliminate impossible combinations of inputs. The neural network certification database should range from $0.1 < M < 1.3$, at a minimum recovery altitude (h=5000 ft.) to $0.3 < M < 2.0$ at 50,000 ft altitude (h). The aircraft pitch angle would range from -180 to +180 degrees. For each point in the database or flight envelope, as shown in Figure 3, a specific M (t), h (t) and α(t) would be input to the neural network.

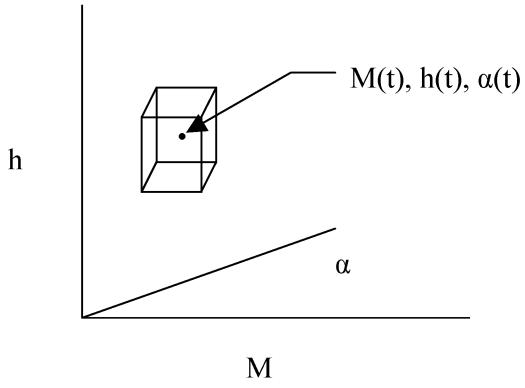

Figure 3. A Point in the Flight Envelope

A computational study is required to determine the increment of each parameter to determine the instance of each computer run and the scaling of each input value. For example, the increment of altitude could be 1000 ft, Mach could be 0.25 Mach and angle of attack could be 6 degrees. Larger increments could be used to limit the time required for simulations.

Adaptive Analysis

It is believed that the greatest deficit to verifying neural networks has been the attempt to apply linear methods to non-linear systems. In the previous section, a method to develop statistical information about the behavior of the sigma-pi neural network weights was proposed. In the following section, adaptive analysis is proposed to understand the behavior of the neural network weighs for verification purposes. Complex adaptive systems have an extensive set of algorithms and techniques that can be used to develop a new verification approach. The OLNN is a fully adaptive sigma-pi network with one hidden layer. Sigma-pi is a product unit neural network that was introduced by Durbin and Rumelhart [16]. A normal neural network used a unit summation function:

$$Y = \sum_{i=1}^{n} W_{ij} X_i$$

Sigma-pi neural networks use a unit product function:

$$Y = \prod_{i=1}^{n} W_{ij}^{X_i}$$

where X is the activation value of the unit and W is the weight. Figure 4, taken from [17], shows a sigma-pi network:

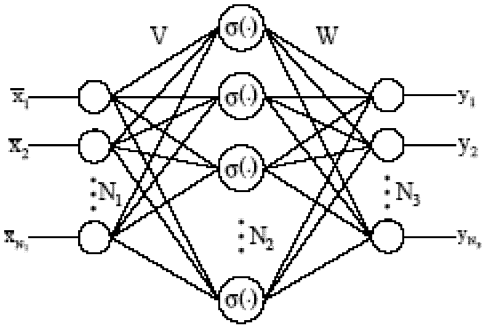

Figure 4. Sigma-Pi Net with One Hidden Layer

where V are the first to second layer interconnection weights, W are the second to third interconnection weights, and σ is the hidden layer activation function. An example of a hidden layer activation function is the sigmoidal function:

$$\sigma(z) = 1 / (1+ e^{-az})$$

The damage adaptive controller uses a backpropagation controller to change the interconnection weights to optimize the control

response to compensate for damage and failure conditions of the aircraft [3]. For specific cases of M (t), h (t), α(t) and damage condition, how do the weights change as a function of time? Using a network fitness distribution approach, it is proposed that a small number of the weights change significantly, thus the control of the outcome of the neural network would be dominated by a few weights as shown in Figure 5.

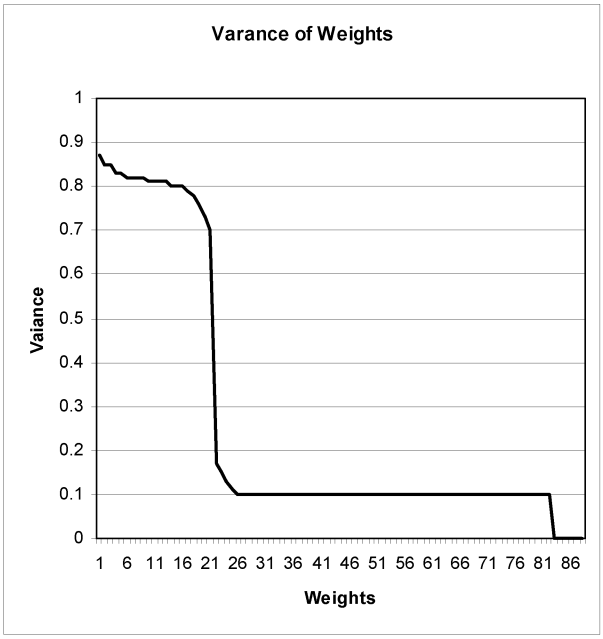

Figure 5. Neural Network Weight Variance

Since, it is proposed that networks weights are characterized by the power law, there are certain weights in the neural network that dominate. A command error timeline would be simulated which would characterize the control error before the damage condition, at the time of damage, at the time of the neural network engagement, and after the pilot has regained control of the aircraft. As the system changes, the point x(t) will trace a trajectory. At each point during the simulation, the value of each weight of the neural network would be recorded. Using complex adaptive system methods it could be determined whether the weights naturally converge. Using this approach, rules that govern the neural network weights could be determined.

Neural Network Weights

In the study of complex adaptive systems [18] a concept of interest is emergence. Emergence is the study of complex adaptive systems that are governed by elements that follow low-level rules but perform sophisticated behavior. Schools of fish, flocks of birds, bees, ant and termite colonies, and herds of land animals are subjects of emergent study since they all exhibit complex, adaptive behavior [19].

As shown in Figure 6, the neuron is the processing unit. Each neuron is characterized by an activity level, representing the state of polarization of the neuron, an output value – representing the firing rate of the neuron, a set of input connections – representing synapse on the cell and its dendrite, a bias value – representing an internal resting level of the neuron, and a set of output connections – representing a neuron's axonal projections. Each aspect is represented mathematically by real numbers. Each connection has an associated weight (synaptic strength) that determines the effect of the incoming input on the activation level of the unit. The weights may be positive (excitatory) or negative (inhibitory) Input lines are product units that yield an activation value for neuron j at time t.

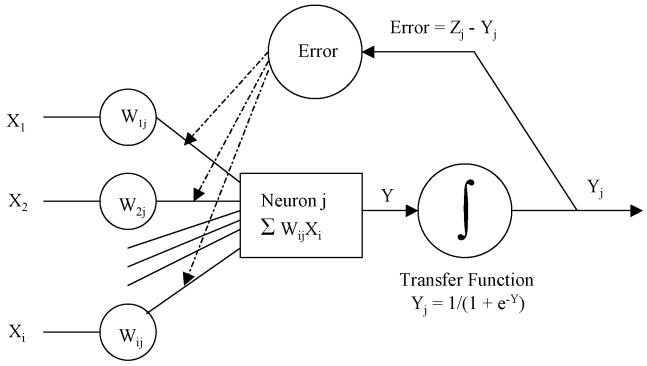

Figure 6. Backpropagation Learning

Neural network learning can be defined as simply the adjustments necessary to the set of weights that allow the network to calculate the desired output. The most popular form of neural network learning is backpropagation, as shown in Figure 6. In sigma-pi architecture, the weights of the network are adjusted as a function of error between the actual output and the desired output values. In the error correction learning procedure, if the actual output equals the desired output, the

weights of the neural network are not changed. However, if the output differs, change must be made to some of the weights. The problem is to determine which weight contributed to the error. The total error of the system is defined as:

$$E = \sum_{p,\,i} (d_{i,p} - y_{i,p})^2$$

where i indexes the output units, p indexes the input/output pairs, $d_{i,p}$ is the desired output $y_{i,p}$ is the actual output and E is the total error for the system. The goal is to change the weights to minimize this function. That is, change the weights of the system in proportion to the derivative of the error with respect to the weights. The change in w_{ij} is thus proportional to:

$$\frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial w_{ij}}$$

With this understanding of weight change and the proposed statistical analysis, the non-linear, dynamical system behavior can possibility be characterized using complex adaptive system methods. The following subsections will outline possible techniques that can possibility be used.

Chaos Theory Analysis

According to [20], complex dynamic behavior has the following characteristics: a) independence: a large number of relatively independent components, b) dynamic: each component responds to its fellow component, c) adaptiveness: the system conforms to new situations to bring about some realignment, d) self-organization: order forms, e) local rules: govern each component, and f) hierarchical nature of structure. Developing a non-linear, dynamical algorithm for a system determination could be based on chaos theory. The initial effort in the determination of a new system algorithm based on chaos theory is to identify at least one control parameter [21]. There is a large number of published examples of non-linear equations that can lead to chaos. Using a quadratic map approach, Devaney [22]:

$$x_{t+1} = c + x_t^2$$

where c is a constant parameter; or Lorenz [23] quadratic map approach:

$$x_{t+1} = a(3x_t - 4 x_t^3)$$

where a is a constant parameter; or Thompson and Stewart [24]:

$$\theta_{t+1} = \theta_t + (a/2\pi)\sin 2\pi\, \theta_t + b$$

where θ is an angular variable and b is a constant.

One of the most used non-linear equations that can characterize steady state, bifurcation and chaos is the logistic equation, by May [25]:

$$x_{t+1} = k x_t (1 - x_t)$$

where x_{t+1} is the updated value of the control parameter during a time period. This equation, originally developed to model long-term population dynamics, provides a one-dimensional feedback system with discrete time intervals. To use the logistics equation, would require the substitution of population with the control parameter and a suitable delta time. The logistics equation is suitable in depicting order (convergence to a constant value), complexity (constant osculation between two values) and chaotic activity (persistent instability), as shown in Figure 7.

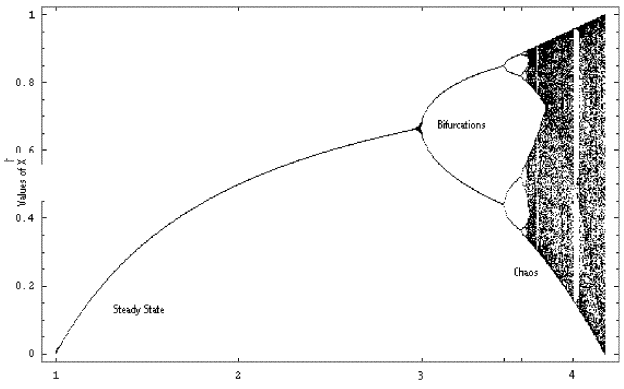

Figure 7. Chaos Theory Phases

The logistics equation than plots a parabola with x_t on the abscissa and x_{t+1} on the ordinate and the value of x between 0 and 1. The constant k governs the steepness and height of the parabola above the abscissa. The peak chosen is k = 4, with the range of k from 0 to 4.0.

The characteristics of the logistic equation with these limits are very interesting. For values of k from 0 to <3.0 the time series trajectory always converges to a constant value or steady state. This condition is true for any initial condition. In chaos theory terms, the forward trajectory is a single attractor, which means that for any value of x_0 the resulting set of points will move towards a constant. An attractor is a dynamical system's set of stable conditions. It is the equilibrium state for the system, such that if the system is started from another state it will evolve until it arrives at the attractor. Each weight will be analyzed to determine if it has a k value from 0 to < 3.0 and is convergent.

Chaos theory contents that there are particular universal phases that dynamical systems transition from regular motion to irregular motion. For k from 3.0 to 3.7, the dynamics of the time series system is unstable, transitioning from stable to chaotic. In this phase, systems will continually oscillate which represents complexity. For k < 3.7 to 4.0 the time series is chaotic. In the chaotic phase, the system will continually oscillate chaotically which represent an unstable system.

Thus, if a weight is analyzed using the logistic equation to determine k as:

$$k = x_{t+1} / (x_t (1-x_t))$$

First consider $0 < k < 3$. Whatever value of k is used the time series using that value will converge to constant value. If k = 2.43 and the control parameter x_t is 0.40, then $x_{t+1} = kx_t(1-x_t)$ yields $2.43(0.4)(1-0.4) = 0.5832$. Using the calculated value for x_t , then $x_{t+1} = 0.59076$. As shown in Table 1 below, this series converges to a steady state value of 0.58847, which in chaos theory represents stability. Trials show that all iteration at constant k for k < 3.0 decay to a steady state, regardless of x_0. Therefore, if the value of k calculated the control parameter is less than 3.0, the weight can determine as convergent or stable.

Table 1. Normalized Control Parameter x_0=0.40

x_0 = 0.4	Order	Complexity	Chaos
Delta t	k=2.43	k=3.30	k=3.95
1	0.58320	0.79200	0.94800
2	0.59067	0.54362	0.19471
3	0.58751	0.81871	0.61937
4	0.58888	0.48978	0.93121
5	0.58830	0.82465	0.25302
6	0.58855	0.47717	0.74656
7	0.58844	0.82328	0.74737
8	0.58849	0.48011	0.74578
9	0.58847	0.82369	0.74887
10	0.58847	0.47948	0.74284
11	0.58847	0.82357	0.75455
12	0.58847	0.47948	0.73155
13	0.58847	0.82361	0.77570
14	0.58847	0.47941	0.68724
15	0.58847	0.82360	0.84900
16	0.58847	0.47942	0.50636
17	0.58847	0.82360	0.98734

Similarly, Table 1 demonstrate the case when k = 3.30 and shows that persistent oscillation occurs. Finally, Table 1 demonstrates the case where k = 3.95 and shows a profile of chaotic oscillation occurring.

Linear Time Series Statistical Analysis

The adaptive neural network is a dynamical system, thus a linear statistical time series analysis can be preformed. Even through this adaptive neural network in non-linear, a first attempt will be made to characterize it behavior using linear statistical time series model [26]. If the nonlinearities are cooperative enough they may appear as noise and allow a model to be determined. If this approach does not produce sufficient result the knowledge gained will be applicable to the non-linear statistical dynamic time series analysis.

Statistical Learning and Data Mining Analysis

Statistical learning and data mining methods will be employed which have extended statistical analysis beyond the low-dimensional, independent data. Recent developments in the science of complex adaptive systems provide statistics and machine learning techniques that will allow this research the possibility to infer a reliable, predictive

model from data, even when the data from the weights are strongly dependent variables. Such data mining can possibility disclose the nature of the patterns that the values of the weight change show. The basic idea of data mining is to fit a model with minimal assumptions about what the correct model should be, or how the variables in the data are related. This differs from such classical statistical questions as testing specific hypotheses about specific models, such as the presence of interactions between certain variables. This is facilitated by the development of an extremely flexible class of models called nonparametric or mega-parametric. They are able to approximate any function. This research will attempt to use these mega-parametric models to derive a method to characterize the behavior of an adaptive neural network. Numerous mega-parametric models and their applications are described by Hastie, Trbshirani and Friedman [27].

Causal Inference Analysis

Causal inference is currently an active area of research in the field of complex adaptive systems science. In a causal model, the goal is to predict how changes will propagate through the system. The main difficulty is to determine if the predictive relationships are confounded by the influence of other variables and other relationships that are not recognized in the causal model. Causal inference attempts to overcome this limitation. This research will investigate the many recently developed causal inference models [28] to characterize the behavior of the sigma-pi adaptive neural network that will provide an approach to verification.

Non-Linear Dynamic Filter Analysis

Non-linear dynamic filter is a model of a state-space of a time series. A non-linear dynamic filter can help characterize dynamical system emergent behavior. A filter is a function which provides a future estimate base on the past. Filters are suited to on-line use, since a complete history of previous observations is not required. The problem of optimal linear filters for stationary processes was solved by the founders of complex systems science, Kolomogorov and Wiener. In the 1960s, Kalman and Bucy solved the problem of optimal recursive filtering, assuming linear dynamics, linear observation and additive noise. In the resulting Kalman filter, the new estimate of the state is a weighted combination of the old state, extrapolated forward, and the state which would be inferred from the new observation alone. Non-linear solutions go back to the late 1960s. Unlike the Kalman filter, which gave point estimates, the Stratonovich-Kushner approach calculates the complete conditional distribution of the state and point estimates take the form of the mean or the most probable state. Current developments of non-linear filters have made sufficient improvements including approaches which exploit the geometry of the non-linear dynamics [29] and approaches that yield tractable numerical approximations to the optimal filters [30].

Conclusion

This research will investigate the application of methods, models, and techniques from the new science of complex adaptive systems to determine a new verification methodology for the adaptive sigma-pi neural network that is used for the IFCS. Verification is defined as a method that provides objective evidence that the online, learning sigma-pi neural network system performs consistently as specified by its requirements. The hypothesis of this research is an adaptive neural network is a non-linear, dynamical system and a complex adaptive approach should be used to discover a verification methodology. The logistic equation is an example of a mathematical characterization of complexity and can be used to characterize the change of the sigma-pi neural network weights. The logistic equation or similar equations can be used to define the rules in which the weights of the neural network change and can be use to determine their convergent or divergent nature. In addition, adaptive methods and models provided including linear time series statistical analysis, statistical learning and data mining analysis, causal inference analysis and non-linear dynamic filter analysis are powerful tools to characterize the non-linear dynamics of an adaptive neural network. This research will lead to a new verification methodology for the adaptive, online learning sigma-pi neural network that is currently being used for aircraft damage adaptive flight control.

References

[1] RTCA Software Considerations in Airborne Systems and Equipment Certification, DO-178B. RTCA, Washington DC, 1992.

[2] Calise, A. J., S. Lee, M. Sharma, 2000, Development of a Reconfigurable Flight Control Law for the X-36 Tailless Fighter Aircraft, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 14-17, 2000.

[3] Urnes, James. Sr., R. Davidson, S. Jacobson , 2001, A Damage Adaptive Flight Control System Using Neural Network Technology, *Proceeding of the American Control Conference*, June 25-27, 2001.

[4] Totah, Joseph J, 1996. "Simulation Evaluation of a Neural-Based Flight Controller", AIAA 96-3503.

[5] Peterson, Gerald. E. 1993, "A Foundation for Neural Network Verification and Validation", in Science of Artificial Neural Networks II, Dennis W. Ruck, Editor, Proc. SPIE 1966, pp. 196-207.

[6] Schumann, J., 1999, Automated theorem proving in high quality software design. In S. Hölldobler, editor, *Intellectia and Computational Logic: Papers in Honor of W. Bibel,* Kluwer.

[7] Andrew, R., J. Diederich, & A. B. Tickle, 1995. A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks, *Knowledge Based Systems*, 8.

[8] Plikynas, D., 2004, Decision Rules Extraction From Neural Network: A Modified Pedagogical Approach, *ISSN 1392-124X proceedings of the Informacinës Technologijos IR Valdymas,* Nr. 2(32)

[9] W. Wen, J. Callahan, and M. Napolitano, 1996, "Towards Developing Verifiable Neural Network Controller", *ICTAI '96 Workshop on Artificial Intelligence for Aeronautics and Space,* Toulouse, France.

[10] Schuman, J. & S. Nelson, 2002, Towards V&V of Neural Network Based Controllers, *Proceedings of the First ACM Workshop on Self-healing Systems*.

[11] Cortellessa, V., B. Cukic, D. Del Gobbo, A. Mili, M. Napolitano, M. Shereshevsky, & H. Sandhu, 2003, *Certifying Adaptive Flight Control Software*, Department of Mechanical and Aerospace Engineering, West Virginia University.

[12] Schumann, J, P. Gupta & K. A. Loparo 2003, Lyapunov Stability Analysis of Neural Network Based Flight Controller, *Proceeding of the International Joint Conference on Neural Networks*

[13] Yerramalla, S., E. Fuller, M. Mladenovski, & B. Cukic, 2003, Lyapunov Analysis of Neural Network Stability in an Adaptive Flight Control System, *Proceeding of the 3rd Annual NASA Office of Safety and Mission Assurance Symposium.*

[14] Taylor, B. J. & M.A. Darrah, 2003, Verification and Validation of Neural Networks: A Sampling of Research in Progress, In *Proceedings of AeroSense,* Orlando, FL.

[15] Liu, Y., T. Menzies, & B. Cukic, 2002, Data Sniffing – Monitoring of Machine Learning for Online Adaptive Systems, In *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*.

[16] Durbin, R. & Rumelhart, D., 1989. A Computationally Powerful and Biologically Plausible Extension to Backpropagation Networks, *Neural Computation*, Vol. 1: 133-142.

[17] Calise, A. J. & R. T. Rysdyk 1998. Nonlinear Adaptive Flight Control using Neural Networks, *IEEE Control Systems Magazine,* 21(6): 14-26.

[18] Cannady, James, 2002, DCIS 790 Complex Adaptive Systems. (Class Lecture Viewgraphs), Nova Southeastern University, Graduate School of Information Sciences Web site, URL: http://scis.nova.edu/~cannady

[19] Holland, J. H, 1998, *Emergence: From Chaos to Order*, Perseus Books, Cambridge, MA.

[20] Williams, G. P, 1997, *Chaos Theory Tamed,* Washington DC, Joseph Henry Press.

[21] Ott, E, 2002, *Chaos in Dynamical Systems*, New York, Cambridge University Press.

[22] Devaney, R. L., 1988 "Fractal Patterns Arising in Chaotic Dynamical Systems", Peitgen, H-O., D. Saupe (eds), *The Science of Fractal Images*, Springer, Berlin,

[23] E. N. Lorenz, 1976, "Nondeterministic Theories of Climate Change", *Quaternary Research*, 6, pp 495-506.

[24] Thompson, J. M. & H. B. Stewart, 1986, *Nonlinear Dynamics and Chaos: Geometrical Methods for Engineers and Scientists*, John Wiley, New York.

[25] R. M. May, 1976, "Simple Mathematical Models with Very Complicated Dynamics", Nature 261, pp. 459-467.

[26] Eyink, G. L. 1998, Linear Stochastic Models of Nonlinear Dynamical Systems. *Physical Review E*, 58: 6975-6991

[27] Hastie, T., R. Tibshirani, & J. Friedman, 2001, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

[28] Pearl, J., 2000, *Causality: Models, Reasoning, and Inference*. Cambridge, England: Cambridge University Press.

[29] Darling, R. W. R., 1998, *Geometrically Intrinsic Nonlinear Recursive Filters*, Tech. Reports 494 & 512, Statistics Department, University of California-Berkeley.

[30] Ahmed, N. U., 1998, *Linear and Nonlinear Filtering for Scientists and Engineers*. Singapore: World Scientific.