

A Systematic PHM Approach for Anomaly Resolution: A Hybrid Neural Fuzzy System for Model Construction

Piero Bonissone, Xiao Hu, and Raj Subbu

General Electric Global Research, Niskayuna, NY, 12309, USA

[\[bonissone; hux; subbu\]@research.ge.com](mailto:[bonissone; hux; subbu]@research.ge.com)

ABSTRACT

We analyze potential causes of anomalies, as they vary from incipient system failures to malfunctioning sensors, operating the asset in unusual regions, using inappropriate anomaly detection models, etc. For each cause, we follow the PHM cycle, creating an anomaly resolution action. Within this systematic approach, we focus on one of the most neglected causes for anomalies: the inadequate accuracy of anomaly detection models. We describe a hybrid approach based on a fuzzy supervisory system and an ensemble of locally trained auto associative neural networks (AANN's). The supervisory system will manage the transition among local AANN's during operating regime changes. This approach is illustrated with experiments with a simulated aircraft engine.

1. INTRODUCTION

The main goal of Prognostics & Health Management (PHM) for assets such as locomotives, medical scanners, and aircraft engines is to maintain these assets' operational performance over time, improving their utilization while minimizing their maintenance cost. This tradeoff is critical for the proper execution of contractual service agreements (CSA) offered by OEM's to their valued customers.

1.1 PHM: The Big Picture

PHM can be divided into two main components:

- **Health Assessment:** the evaluation and interpretation of the asset's current and future health state, and
- **Health Management:** the control, operation, and logistic plans to be implemented in response to such assessment.

As originally presented in references (Bonissone 2007; 2008a, Bonissone and Iyer 2007), the PHM data flow can be summarized by the functional diagram

shown in Figure 1. The first two tasks, (1) *remote monitoring*, and (2) *input data pre-processing*, are platform-dependent, as they need domain knowledge to identify and select the most informative input, scrub them, aggregate them, and prepare them to become suitable inputs for the models. The remaining decisional tasks could be considered platform-independent (at least to the extent that their functions could be accomplished by pure data-driven models. They are: (3) *anomaly detection and identification*; (4) *anomaly resolution*; (5) *diagnostics*; (6) *prognostics*; (7) *fault accommodation*; and (8) *logistics decisions*.

1.1.1. Asset Health Assessment

Using platform-deployed sensors, the data are remotely collected and preprocessed (e.g., segmented, filtered, validated, etc.). Then these data are summarized by a subset of features that provide a more informative, robust representation of the information contained in the data. These features could contain any combination of categorical and numerical values.

Anomaly Detection (AD). These features are analyzed by an anomaly detection module to assess the degree of abnormal behavior for each asset in the fleet. If the degree of abnormality exceeds a defined threshold, the module will identify the asset, determine the time when the anomaly was first noticed and the possible cause(s) of the anomaly (usually a coarse identification at the systems/subsystem level). Anomaly detection leverages unsupervised learning techniques, such as clustering. Its goal is to extract the underlying structural information from the data, define normal structures and regions, and identify departures from such regions.

Anomaly Identification (AI). After detecting an abnormal change, (e.g. a departure from a normal region), we need to identify its cause. There are many factors that could cause such change:

(a) A **system fault**, which could eventually lead to a failure;

(b) A **sensor fault**, which is creating an incorrect measurement;

(c) An **inadequate anomaly detection model**, which is falsely reporting an anomaly due to bad design, inadequate model update, execution outside the model's region of competence, etc.

(d) A sudden, unexpected **operational transient**, which is stressing the system by creating an abrupt load change. In turn, this transient could be originated by an operator error, who is requesting such sudden change; by an incorrect reference (set-up) vector – in case of operation automation - which is also requesting such abrupt change; or by a bad controller, which is over- or under-compensating for some perceived state change.

Diagnostics. This information allows a diagnostic module to focus on a given platform subsystem, analyze key variables associated with the subsystem, and try to match their pattern with a library of signatures associated with faults or incipient failure modes. The result is a ranked list of possible faults.

Diagnostics leverages supervised learning techniques, such as classification. Its goal is to extract potential signatures from the data, which could be used to recognize different failure modes

Prognostics. A prognostics module updates a deterioration index for the platform (sub-) system, and modifies the expected Remaining Useful Life (RUL) or time-to-failure (TTF) from a linear, normal wear trajectory to an exponentially decaying one. The fault time and incipient failure mode determine the inflection point in such curve and the steepness in deterioration, respectively. A prerequisite to leverage this RUL estimation is to have a narrow confidence interval, such that this information is actionable and can be used in the asset health management part of PHM as a horizon to optimize the logistics/maintenance scheduling plan.

Prognostics leverages prediction techniques. Its goal is to estimate, update, and forecast the asset's health index, which is mapped into RUL. Originally, this index reflects the expected deterioration under normal operating conditions. Later the index is modified by the occurrence of an anomaly/failure, reflecting faster RUL reductions.

1.1.2. Asset Health Management

All these functions are interpretations of the system's health state. These interpretations lead to an on-board control action and an off-board logistics, repair and planning action. On-board control actions are usually focused on maintaining performance or safety margins, and are performed in real-time. Off-board maintenance/repair actions cover more offline decisions. They require a decision support system (DSS) performing multi-objective optimizations,

exploring Pareto frontiers of corrective actions, and combining them with preference aggregations to generate the best decision tradeoffs.

Anomaly detection (AD) is the first, critical step in the chain of PHM decisional tasks. Figure 1 illustrates such chain, when a system anomaly is the detected source.

1.2 Paper Focus on Soft Computing

PHM is a multi-discipline field, as it includes facets of Electrical Engineering (reliability, design, service), Computer Science and Decision Sciences (Artificial Intelligence, Soft Computing, Machine Learning, Statistics, OR), Mechanical Engineering (geometric models for fault propagation), Material Sciences, etc.

Within this paper we will focus on the role that Soft Computing plays in PHM functionalities. When addressing real-world PHM problems, we usually deal with systems that are difficult to model and possess large solution spaces. So we augment available physics-based models, which are usually more precise but difficult to construct, customize, and adapt, with approximate solutions derived from Soft Computing methodologies. In this process we leverage two types of resources: problem domain knowledge of the process (or product) and field data that characterize the system's behavior. The relevant available domain knowledge is typically a combination of first principles and empirical knowledge. This knowledge is often incomplete and sometimes erroneous. The available data are typically a collection of input-output measurements, representing instances of the system's behavior, and are generally incomplete and noisy. Soft computing is a flexible framework in which we can find a broad spectrum of design choices to perform the integration of knowledge and data in the construction of approximate models.

1.3 Paper Structure

We will use Soft Computing to explore the concept of Anomaly Identification and Resolution, and extend the PHM cycle, illustrated in Figure 1, beyond system failures. In Section II we will analyze the most common anomaly sources - while limiting the scope of our analysis to a subset of them – and for each case we will propose their associated resolution within the PHM framework above described. In section III we will focus on one of the most neglected causes for anomalies, the inadequate accuracy of anomaly detection models, and we will describe a solution based on a fuzzy supervisory system and an ensemble of locally trained auto associative neural networks (AANN's).

In Section IV we will illustrate this approach with a set of experiments within a simulated aircraft engine environment. Finally, in section V we will discuss potential extensions and future work.

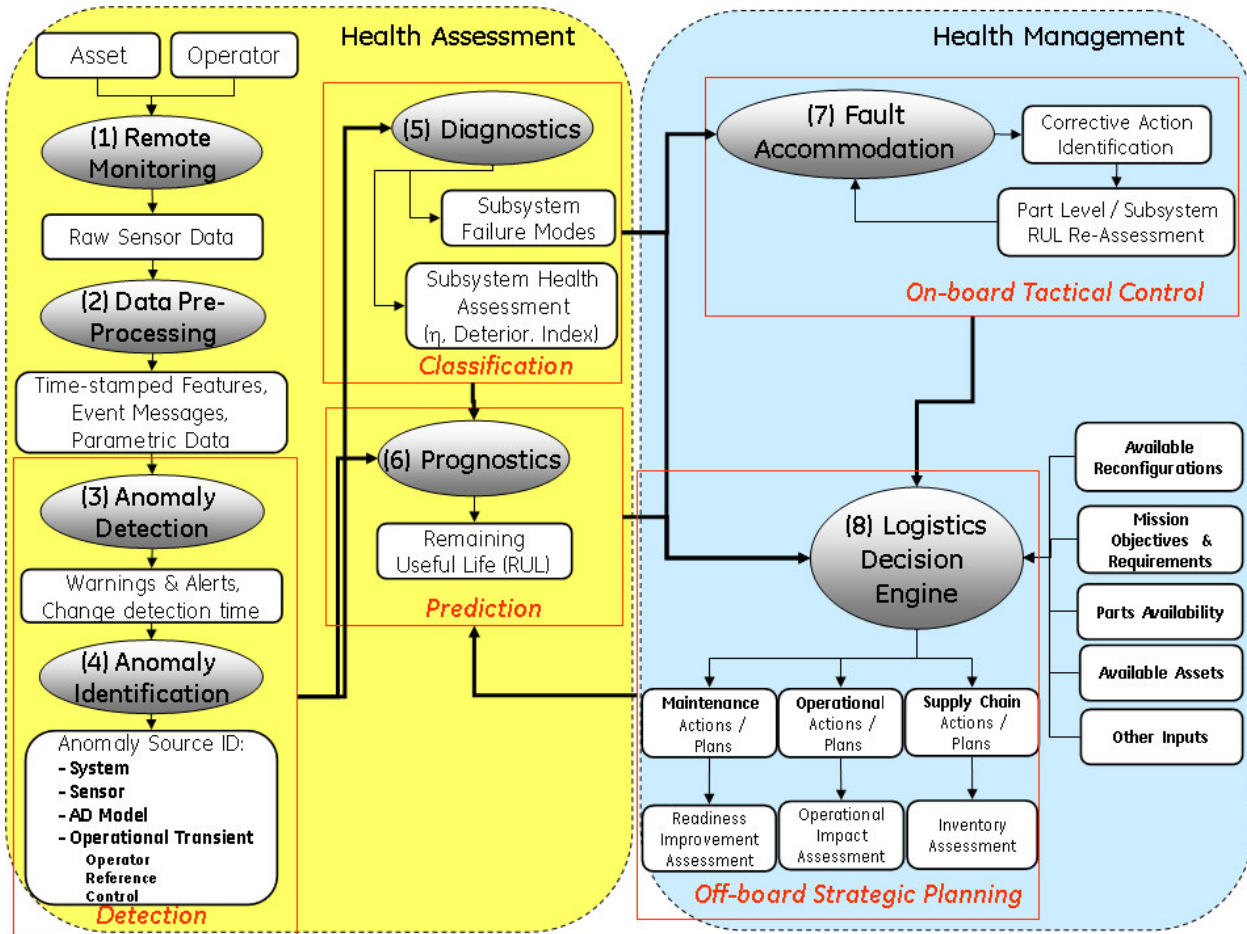


Figure 1. PHM Architecture showing the decisional tasks triggered by a system anomaly.

2. AMBIGUITIES IN ANOMALY IDENTIFICATION

2.1 Possible Sources of Anomaly for Dynamic Systems

Figure 2 shows a typical system diagram for a controlled dynamic system (the asset), for which we want to provide an effective PHM service. From this diagram we can see that the anomalies could be caused by incipient system failures, sensor failures, AD model failures, extreme operational transients (cause by an operator or a reference generator), or malfunctioning controllers or actuators.

2.2 System Failures

This is the textbook situation in which the asset being monitored exhibits an anomalous behavior, which is a precursor to a failure mode. Many books and papers have been devoted to this case, so we will not cover in this paper. The associated PHM cycle is the one described in section 1 and illustrated in Figure 1.

2.3 Sensor Failures

There are situations in which the instrumentation used to monitor the asset experiences a failure mode, such as intermittent signals, offset shifts, drifting, saturations, etc. Faulty sensors usually produce signatures/residuals that profoundly affect the variable being measured by the sensor, when compared with the other variables. This situation can be disambiguated by using specialized modeling techniques, such as the auto-associative neural networks (AANN's) that we will describe in Section III. Because of its auto-association property, AANN can be used to infer nominal sensor values from raw measurements when information in the saturated measurements is analytically redundant in the sense that if one measurement is missing, it can be replaced with an estimate from the remaining valid sensors (Mattern et al., 1998.) After diagnosing a sensor failure, the PHM cycle consists in determining the sensor's RUL, which could range from zero (as in the case of a broken or saturated sensor) to a reasonable operational horizon, as in the case of a slow drifting

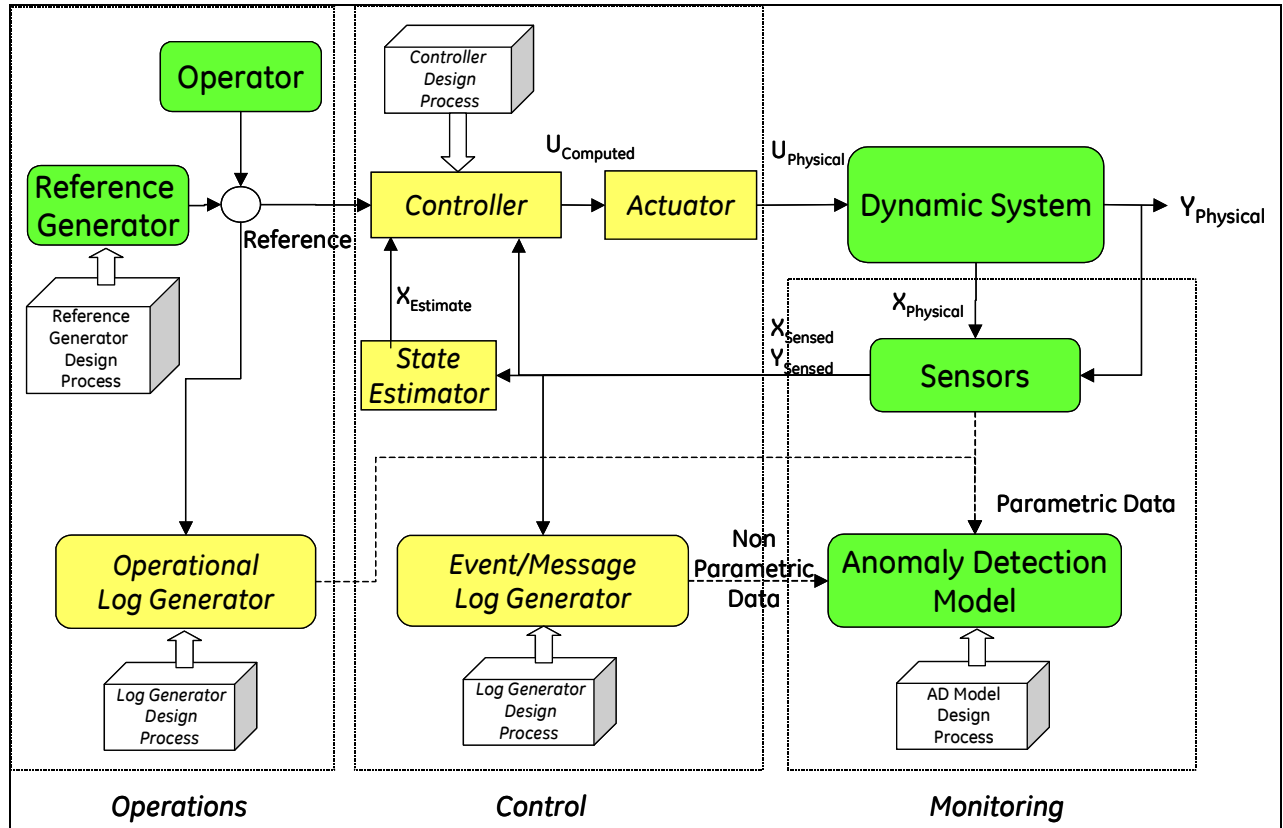


Figure 2. Typical architecture for controlled dynamic system (asset) including operations and monitoring.

sensor. The fault accommodation – if needed – would consist of switching to a virtual sensor (such as an AANN or a feedforward NN) until it is possible to replace or repair the sensor.

2.4 Inadequate Anomaly Detection (AD) Model

To avoid false alarms, we need to verify the correctness of the AD model that generates the anomaly signals. This verification requires the satisfaction of several conditions related to the model design and lifecycle management:

- **Accuracy.** The AD model must be designed to achieve the required accuracy, representing the tradeoff between False Positives (FP) and False Negatives (FN). Sometimes, it is not possible to achieve this accuracy with a single global model (whose large variance would make most anomalies indistinguishable from normal cases). In these situations, we can use a collection of local models (with limited, overlapping regions of applicability) aggregated by a (fuzzy) supervisory system. This will allow us to leverage the performance of customized local models, and combine their outputs using a smooth interpolation mechanism as we move across adjacent operating regions (Hu et al., 2009). This approach will be further described in the next section. This accuracy could be further

improved by fusing multiple AD models, providing that they are diverse, i.e. their errors are mostly orthogonal (Varma et al., 2007.)

- **Region of competence.** The model is operating within its region of competence, which is determined by the domain of its training set. If that is not the case, it is likely that we are experiencing model extrapolation errors (Bonissone 2008b).

- **Updated version.** The model's performance is within the boundary established by the test and validation errors observed during the design phase. Otherwise we should update the model with the most recent data to prevent model obsolescence (Patterson et al., 2005.)

Once an AD model has been deemed inadequate, its associated PHM cycle is quite simple. Fault accommodation can be achieved if the model is part of an ensemble of models. In such case the output of the faulty model can be excluded or discounted within the aggregation performed by the fusion mechanism, allowing the remaining models to provide a more accurate classification. Finally, the inadequate model must undergo an updating process consisting of training, testing, and validation using recent data, possibly in the new operation regions.

2.5 Other sources of anomalies

There are at least three other possible causes that could trigger the output of an Anomaly Detector. We are not considering them within the scope of this paper but we are listing them for the sake of completeness:

- **Extreme operational transients.** This situation is caused by a sudden reference change, which is either requested by the operator (in manual mode) or by the reference generator (in automated mode).

- **Malfunctioning Controller.** This situation could have many possible causes, ranging from control software glitches, control operating outside its region of competence (for which it was designed), etc.

- **Actuator Failure.** The controller's output is incorrectly interpreted or executed by the actuator. This situation has also many possible causes, all of which are considered outside the scope of this paper.

3. IMPROVING THE ACCURACY OF THE ANOMALY DETECTION (AD) MODEL WITH A HYBRID SC MODEL

Global models, trained on the entire operating space of the asset, are designed to achieve a compromise among completeness (for coverage), high-fidelity (for accuracy), and transparency (for maintainability). As a result, we typically end up with models that have small biases but large variability. This variability might be too large to distinguish between model error and anomalous system behavior.

This section focuses on a different design tradeoff for such model. We want to guarantee coverage throughout the state space by developing many local AD models, each of which has been trained on overlapping regions of the state space. We develop each model with techniques that minimize its variance within its region of competence. Finally, we capture the criteria for model applicability by using a fuzzy supervisory model approach that leverages linguistic fuzzy rules to integrate local models to better represent system dynamics as system transits from one operating regime to another. Through this fuzzy supervisory approach, the magnitude of residuals caused by the operating regime transition can be significantly reduced so that false alarms can be avoided.

We will briefly illustrate the various components used in this approach. We will start with the Component Level Model (CLM), a simulator that replicates the behavior of the dynamic system, sensors, and controllers. Then, we will describe the auto-associative neural network (AANN) and the fuzzy supervisory system used to implement the anomaly detection model. Each rule defines a mapping between a fuzzy state vector and a corresponding fuzzy action.

3.1 Component Level Model (CLM)

For this work, we leverage the Component Level Model (CLM), a physics-based thermodynamic model that has been widely used to simulate the performance of an aircraft engine. Flight conditions, such as altitude, Mach number, ambient temperature, and engine fan speed, and a large variety of model parameters, such as module efficiency and flow capacity are inputs to the CLM (see Figure 3). The outputs of the CLM are the values for pressures, core speed and temperatures at various locations of engine, which simulate sensor measurements. Realistic values of sensor noise can be added after the CLM calculation. In this study, a steady state CLM model for a commercial, high-bypass, twin-spool, turbofan engine is used. The objective is to use engine data collected under cruise conditions to monitor engine health changes.

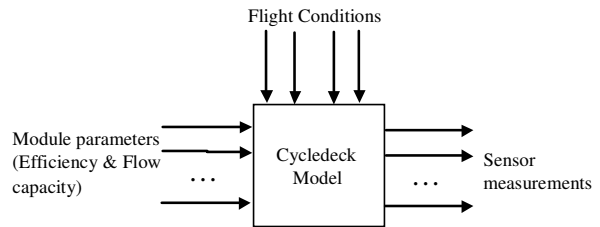


Figure 3. Physics-based Component Level Model (CLM)

3.2 Auto-Associative Neural Networks

Auto-associative neural networks (AANN) are basically feed-forward neural networks with network structure satisfying requirements for performing restricted auto-association. The inputs to the AANN go through a dimensionality-reduction, as their information is combined and compressed in intermediate layers. For example, in Figure 4 the 7 nodes in the input layer are reduced to 5 and then 3, in the 2nd layer (encoding) and 3rd layer (bottleneck), respectively. Then, the nodes in the 3rd layer are used to recreate the original inputs, by going through a dimensionality-expansion (4th layer, decoding, and 5th layer, outputs). In the ideal case, the AANN outputs should be identical to the inputs. Their difference (residuals) and their gradient information are used to train the AANN to minimize such difference.

This network computes the largest Non-Linear Principal components (NLPCA's) – the nodes in the inter-mediate layer – to identify and remove correlations among variables. Besides the generation of residuals this type of network can also be used in dimensionality reduction, visualization, and exploratory data analysis. As noted in reference (Kramer 1991): “while [Principal Component Analysis] PCA identifies only linear correlations between variables, NLPCA uncover both linear and nonlinear correlations, without

restriction on the character of the nonlinearities present in the data". NLPKA operates by training a feed-forward neural network to perform the identity mapping, where the network inputs are reproduced at the output layer. The network contains an internal "bottleneck" layer (containing fewer nodes than input or output layers), which forces the network to develop a compact representation of the input data, and two additional hidden layers.

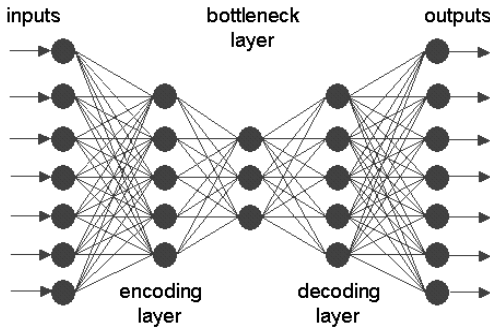


Figure 4. Architecture of a 7-5-3-5-7 Auto Associative Neural Network

In reference (Hu et al., 2007), we used AANN's to estimate sensor measurement under normal conditions and then the residual between raw measurement and normal measurement were used to infer the conditions of the components and systems. Additional information about AANN's can be found in references (Kramer 1991; Kramer 1992; Mattern et al., 1998; Lerner et al., 1999; Berenji et al., 2004).

3.3. Fuzzy Logic Systems

Fuzzy logic (FL) gives us a language, with syntax and local semantics, within which we can translate qualitative knowledge about the problem to be solved (Zadeh 1978; Ruspini et al., 1998). In particular, FL allows us to use linguistic variables to model dynamic systems. These variables take fuzzy values that are characterized by a label (a sentence generated from the syntax) and a meaning (a membership function determined by a local semantic procedure). The meaning of a linguistic variable may be interpreted as an elastic constraint on its value. These constraints are propagated by fuzzy inference operations, based on the generalized modus-ponens. This reasoning mechanism, with its interpolation properties, gives FL a robustness with respect to variations in the system's parameters, disturbances, etc., which is one of FL's main characteristics.

The most common definition of a fuzzy rule base R is the disjunctive interpretation initially proposed by Mamdani and found in most Fuzzy Controller applications (Mamdani and Assilian, 1975). R is a

disjunction of m rules. The Cartesian product operator represents each rule.

$$R = \bigcup_{i=1}^m r_i = \bigcup_{i=1}^m (\bar{X}_i \rightarrow Y_i) \quad (1)$$

The inference engine of a FC can be defined as a parallel forward-chainer operating on fuzzy production rules. An input vector \bar{I} is matched with each n -dimensional state vector \bar{X}_i , i.e., the Left Hand Side (LHS) of rule $(\bar{X}_i \rightarrow Y_i)$. The degree of matching indicates the degree to which the rule output can be applied to the overall FC output. The main inference issues for the FC are: the definition of the fuzzy predicate evaluation, which is usually a possibility measure (Zadeh 1978); the LHS evaluation, which is typically a triangular norm (Schweizer and Sklar 1983; Bonissone 1987); the conclusion detachment, which is normally a triangular norm or a material implication operator; and the rule output aggregation, which is usually a triangular conorm for the disjunctive interpretation of the rule base, or a triangular norm for the conjunctive case. Under commonly used assumptions we can describe the output of the Fuzzy System as

$$\mu_Y(y) = \text{Max}_{i=1}^m \{ \text{Min}[\lambda_i, \mu_{Y_i}(y)] \} \quad (2)$$

where λ_i is the degree of applicability of rule r_i

$$\lambda_i = \text{Min}_{j=1}^n \Pi(X_{i,j}, I_j) \quad (3)$$

and

$$\Pi(X_{i,j}(x_j), I_j(x_j)) = \text{Max} \{ \text{Min}[X_{i,j}(x_j), I_j(x_j)] \} \quad (4)$$

is the possibility measure representing the matching between the constraints on the state variables and the actual inputs.

These three equations describe the generalized modus-ponens, which is the basis for interpreting a fuzzy rule set. Let's provide a brief explanation for these equations. As stated in Eq. (1), we consider a rule base to be the union of m rules. Therefore, the output $\mu_Y(y)$ of a fuzzy system, as described in Eq. (2), is the union (i.e., the maximum operator) over the contributions of each of the m rules. Each rule's contribution is derived by weighting its original output $\mu_{Y_i}(y)$, using the minimum operator, with λ_i , the degree of applicability of rule r_i . Equation (3) shows that λ_i is the intersection (minimum operator) of the degrees of matching between each input I_j , and the constraint $X_{i,j}$ on the corresponding state variable for each rule r_i . In other words, λ_i represents the degree to which the rule LHS is satisfied by the input vector. The possibility measure of Eq. (4) is the maximum of the intersection between the membership function of the input and its corresponding constraint for that rule.

3.4 Hybrid Fuzzy Neural Anomaly Detection Model

Usually, when a system is operated under different operating regimes, it is better to train multiple local models for each operating regime. These models represent system dynamics more accurately than a global model applicable over the entire operating space.

AANN's are one realization of empirical local models because of its auto-association property. They embed system dynamics through training into network weights matrix. If the system operates normally in the regime, where the AANN model was built upon, the sensor estimations from AANN output should approximately be the same as raw sensor measurements, resulting in very small residuals. Conversely, if the system operated outside of its defined operating regime, large residuals are usually generated indicating "anomalous" behavior and triggering alerts.

Multiple AANNs can be customized and trained individually to model system within multiple operating regimes of system, respectively. However, none of these local models can accurately capture system dynamics as system transits from one operating regime to another. In this case, residuals generated during the transition phase will most likely exceed the pre-specified alarming threshold and cause false alarms. One common solution deployed to this problem is to ignore the alarms if it is known that the system is undergoing operating regime transition phase. One disadvantage of this approach is it causes the interruption of system monitoring using local models and having the risk of missing true fault alarms generated during the transition phase.

We propose to implement a fuzzy supervisory model to control the transition of local models when operating regime changes. Then we demonstrate the proposed approach using data generated from a CLM model of aircraft engine. This is illustrated in Figure 5.

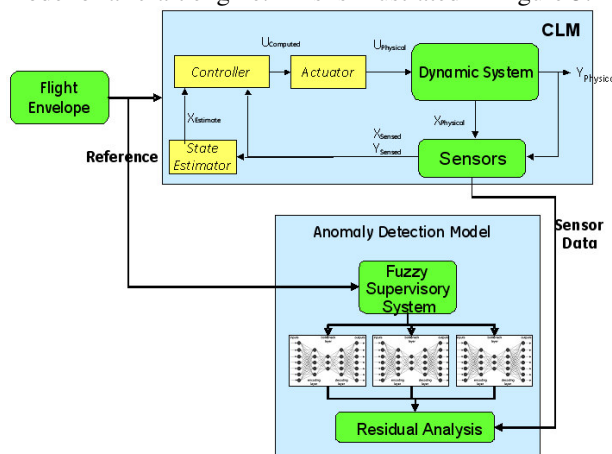


Figure 5. Connecting the CLM simulator with the hybrid fuzzy neural Anomaly Detection (AD) model

4. TESTING THE HYBRID SC MODEL WITH A SIMULATED AIRCRAFT ENGINE (CLM)

We will now test the proposed architecture using a CLM model to simulate a commercial, high-bypass, twin-spool, turbofan engine. Within the normal flight regimes, we specified three flight envelopes (FE) within the typical cruise flight regime, defined by altitude (ALT), ambient temperature (T1A) and mach number (XM) to represent three local operating regimes.

4.1 Global Anomaly Detection Model

First, one AANN with a 9-5-3-5-9 structure was built to model the entire cruise flight regime, which includes the three defined local operating regimes. By configuring the CLM parameters, which includes ALT, T1A, XM, model efficiency and flow, nine simulated sensor measurements were acquired. The selection of the bottle-neck layer is critical to obtain the desired performance of eliminating redundancies in the data. The training of AANN has two phases. In phase one we trained the network using normal data until we reached a reasonable conversion in the MSE training metric. In phase two, we modified the training set by introducing random noise into one or multiple training inputs, to represent faulty measurements. This phase is important to allow the AANN model to learn how to filter noisy information and restore true measurements.

After the global AANN model was properly trained, a new set of data were generated to simulate the transition among operating regimes along the trajectory depicted in Figure 6(b). Figure 6(a) shows the values of flight envelope variables through the transition phase. Since aircraft engine exhibits different system dynamics in the difference local flight regimes, the global model cannot very well capture their characteristics well As expected, the performance of the global AANN model was not very satisfactory.

4.2 Local Anomaly Detection Models

Three AANN's with same structure as 9-5-3-5-9 were built to model local dynamics within individual operating regimes. The similar training process as in the global was repeated in each local AANN model. After the local AANN models are properly trained, a new set of data were generated to simulate the transition of operating regimes along the trajectory depicted in Figure 6(b).

Figure 7 defines the fuzzy membership functions for "Low", "Medium" and "High" of flight envelope variables. The scales of the plots in this figure have been normalized using their range [0 to 100] to protect proprietary information. Then we can specify a set of fuzzy rules, such as the ones described in the table in Figure 8, which describe the applicability of local

models under different operating regimes defined in fuzzy terms.

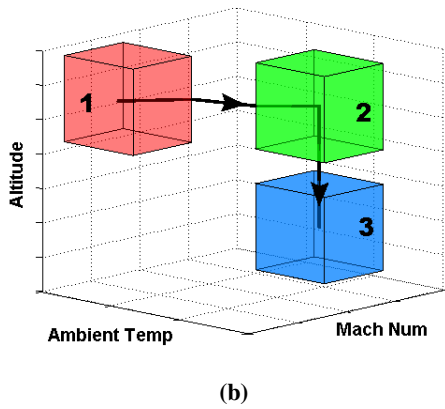
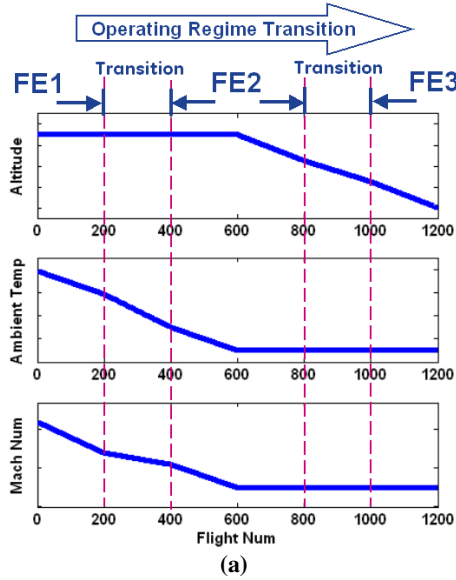


Figure 6. The illustration of system operating regime transition – across 3 flight envelopes (FE).

Figure 8 depicts the scheme of using a fuzzy supervisory approach to control the fusion of local models and assure the smoothness of residuals by interpolation as operating regime transits. Raw sensor measurements are presented to three local AANN models to generate the residuals, respectively. The three variables (T1A, ALT and XM) that define operating regimes are fed through the three fuzzy rule set to determine the applicability of each local AANN model. The three rules define three operational regions. Their associated weight w_j represents the degree of compatibility of the rules LHS and, similarly to λ_j , it is computed using equation (3) and (4). The normalized applicability of each model is then used to perform a weighted average of the residuals from each individual local model to generate an integrated residual. Alerts should be issued only if the aggregated residual

exceeds predefined thresholds regardless of the behavior of residuals from individual local models.

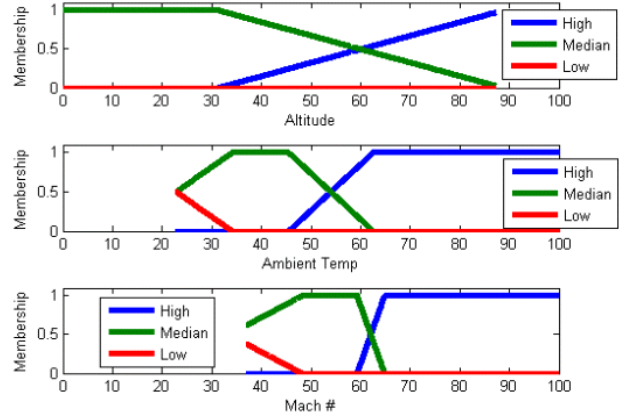


Figure 7. Fuzzy membership functions for variables defining flight operating regimes. For each of the three plots, the scale [0, 100] indicates a percentage of their range of values.

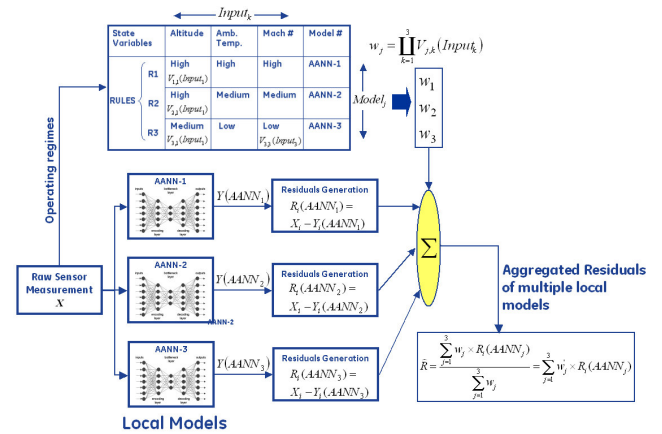


Figure 8. The scheme of model selection/fusion by fuzzy supervisory model (expanding the AD box in Figure 5).

In Figure 9 (a) – (c) we show examples of residuals between actual sensor measurements and estimations from local model AANN-1, AANN-2 and AANN-3, respectively as the flight regime transits along the trajectory defined in Figure 3. Note that each variable has the same range in y-axis of each subplot. Clearly, a local model can only minimize the residuals within the flight regime for which it was trained. However, the fuzzy supervisory model can leverage the superior performance of individual local models in their corresponding flight regimes, and blend their outputs to ensure the smooth interpolation of their residuals during operating regime transitions.

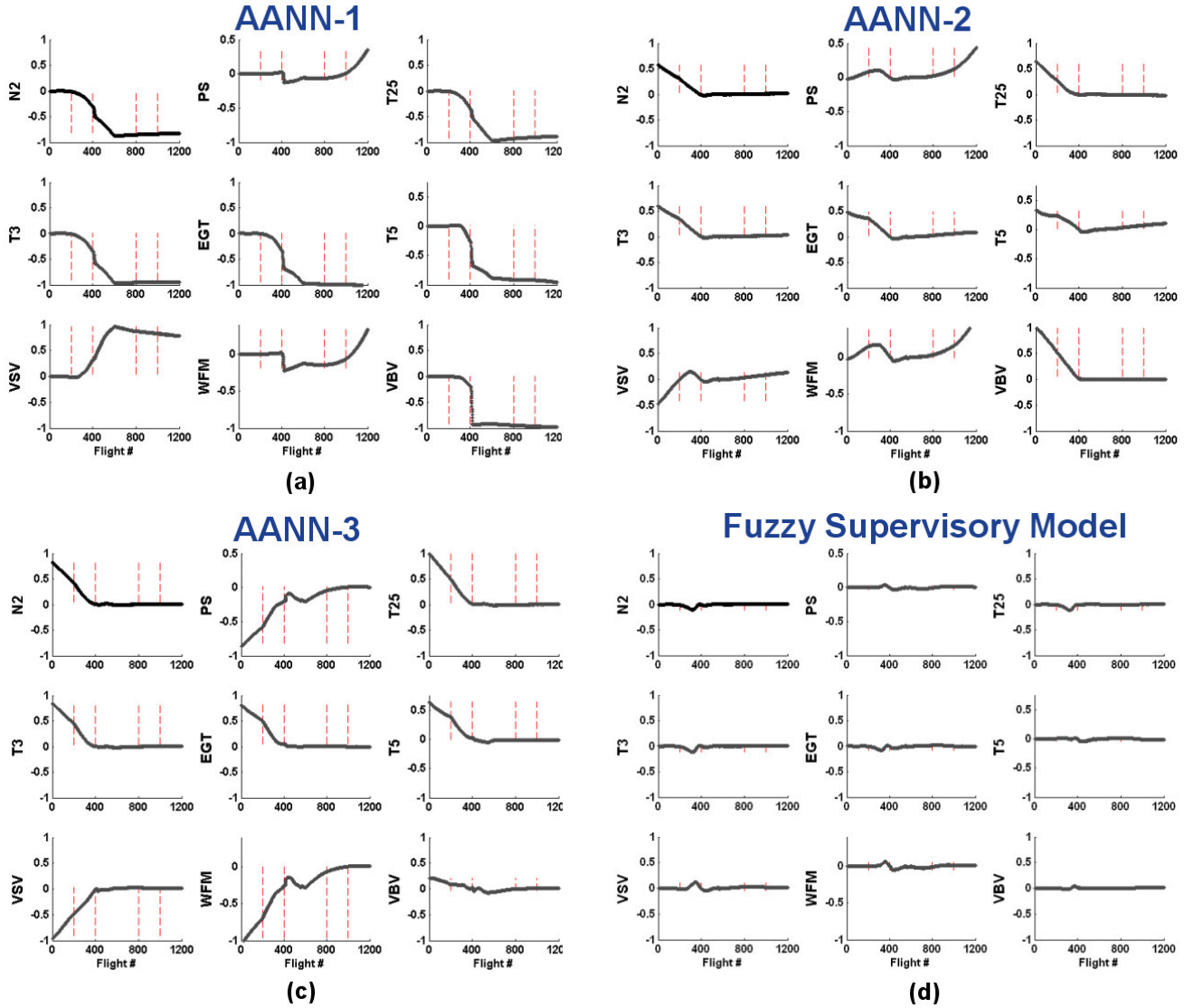


Figure 9. (a)- (c): Residuals from the local model AANN-1, AANN-2 and AANN-3 as the flight regime transits along the trajectory defined in Fig. 3; (c) residuals by applying fuzzy supervisory model. Note that each variable has the same range in the y-axis of each subplot.

To automate the detection process, we suggest normalizing R_{ij} - the residuals of variable i at data point j - using the average of the raw data measurements, i.e., $E_{ij} = R_{ij} / \hat{X}_i$ where \hat{X}_i is the average of variable i . Then we can use a figure of merit (FOM) such as

$$FOM = \sqrt{\frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m (R_{ij} / \hat{X}_i)^2} \quad (5)$$

where n is the number of variables and m is the number of data points, to evaluate the overall magnitude of the residuals. If FOM is smaller than a pre-specified threshold, we can declare that no anomalies are present.

Otherwise, the anomaly is detected. When there is a large (in percentage) residual only from one particular variable, we identify the anomaly as a sensor fault. When residuals from all the variables are larger than the baseline but are roughly equally contributing to the FOM , then there are two possible cases:

1. It is a system fault; or
2. The set of local models are not sufficient to capture the system dynamics (i.e., the current local models were trained in different regions of operating regimes from the one where the test data have been extracted.)

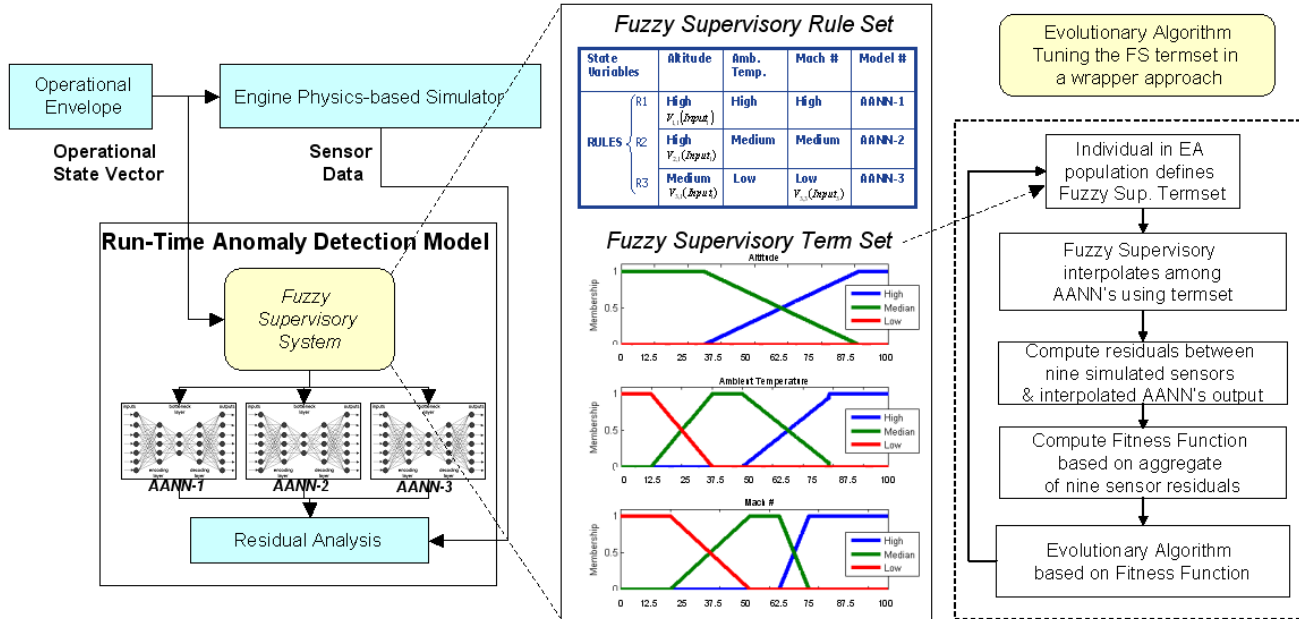


Figure 10: EA tuning the term set of the Fuzzy Supervisory System that interpolates within an ensemble of local AANN’s.

In this second case, we will need to identify additional operating regimes and build corresponding additional local models to better capture the system characteristics.

4.3 Improving the Anomaly Detection Models

The membership functions shown in Figure 7 were handcrafted, and the remarkably improved residuals behavior of the fuzzy supervisory model that uses these membership functions is shown in Figure 9 (d). While this handcrafted membership function set generates very good residuals management behavior across the operating regime, we need a systematic way of achieving similar or better outcomes in an unsupervised manner. To this end we have developed an evolutionary algorithm (EA) wrapper that identifies an optimal set of membership functions.

The left part of Figure 10 shows the run-time anomaly detection (AD) model. The center part of Figure 10 shows an instance of the term set used by the fuzzy supervisory system (the scale of the operational state variables was normalized as a percentage of the range of values to preserve proprietary information). In the right part of figure 10, we can see the evolutionary algorithm (EA) in a wrapper configuration, used to tune the shapes of the membership functions (term sets). Each individual in the EA population is a set of parameters that represents an implementable term set configuration. The parameters varied are the intersection points of the membership functions, and the length of the base of the lower triangle whose upper vertex is the intersection point. The added restriction is that this lower triangle is an isosceles triangle.

However, we do not require the base of the isosceles triangle to be fully contained within the supports of the membership functions. This is allowed to enable maximum horizontal range for the intersections point. As a result, for Altitude, we vary 2 parameters. For each of Ambient Temperature and Mach #, we vary 4 parameters, with a total of 10 search parameters. Each individual is a set of 10 parameters that creates a corresponding set of membership functions that control residuals behavior of the fuzzy supervisory model. The fitness of each individual is computed based on the aggregate of the nine sensor residuals, with a goal towards maximizing fitness or minimizing overall residuals. The EA used is based on the GAOT toolkit*. The population size is set at 500, and the generation count is set at 1000. The EA execution is very efficient taking only about 2 hours of execution time on a standard desktop machine.

Figure 11 shows the residuals management behavior of the optimal membership function set shown in Figure 12. There is an appreciable but not very significant improvement shown in Figure 11 over Figure 9 (d). This can be attributed to two factors. Firstly, the original partition - derived from domain knowledge - provided a reasonable initial segmentation of the model regions of applicability. Secondly, the most significant “glitches” in Figure 11 cannot be solved at the supervisory level. They show intrinsic shortcomings by AANN-1 and AANN-2 in covering the region between flight numbers 200 and 400. This problem could be solved either by extending the

* www.ise.ncsu.edu/mirage/GAToolBox/gaot/

training regions of the two models to provide some coverage for this operating space or by developing a fourth local model (AANN-4) trained in such operating space. While demonstration of significant performance improvement is not the goal of the wrapper-EA approach, demonstration of a reliable unsupervised means to achieve optimal membership functions is. To this latter end, the wrapper-EA approach is a powerful and efficient system tuning approach.

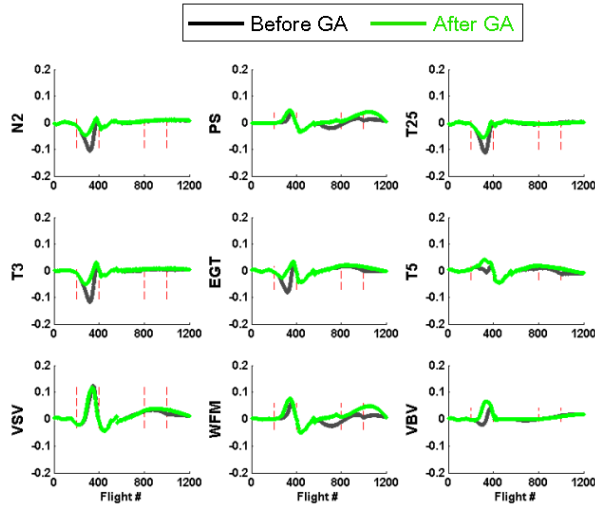


Figure 11. Residuals by applying fuzzy supervisory model with GA tuned membership functions.

Note that the scale of the y-axis in Figure 11 has been modified (as compared to the subplot in Figure 9) to enhance the *Before GA* versus *After GA* comparison.

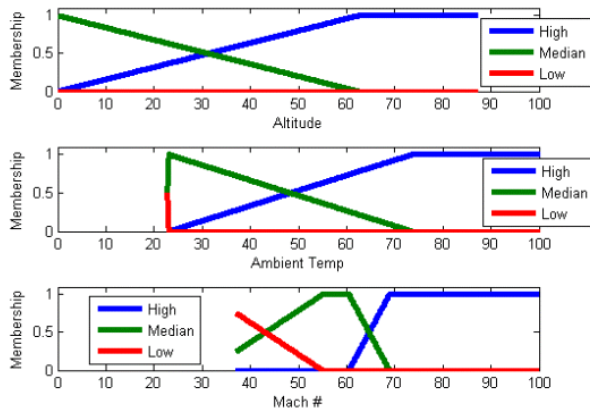


Figure 12. GA-tuned fuzzy membership functions for variables defining flight operating regimes. For each of the three plots, the scale [0, 100] indicates a percentage of their range of values.

5. FUTURE WORK AND CONCLUSIONS

5.1 Prerequisite of Deployment of AANN

AANN model leverages covariance information to reconstruct the network input. For it to work properly there must be dependencies (correlations or interactions) among the variables being monitored. This prerequisite is generally met for most of complex industrial systems we are interested, such as sensor data collected from turbine, aircraft engine and etc. However it is worthwhile to confirm the correlation of system-associated variable before applying AANN.

5.2 Improvement of Fuzzy Supervisory Model

There are two main factors affecting the performance of the fuzzy supervisory model. One of them is related to local operating regimes and local models built on them, how local operating regimes are defined, i.e. how well local empirical models perform with individual operating regime boundaries. The other is closely dependant to fuzzy rules, i.e. how to define the applicability of local models as operating regime changes.

To that end, fuzzy membership functions that interpret crisp parametric values into fuzzy terms play a critical role. Fuzzy membership function defines the fuzzy space and then determines the degree of matching to each rule. In the experiments, we have done some heuristic tuning of fuzzy membership functions in Figure 7 and were able to improve the overall performance of the supervisory model. To optimize the performance, we need to introduce membership functions that can be parameterized. One possibility is to use a Generalized Bell Function (Jang et al., 1997):

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad (6)$$

where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped function varied accordingly, thus exhibiting various forms of membership functions for a fuzzy set A. Figure 13 illustrates examples of a bell-shaped membership function and the traditional trapezoidal membership function. By using differentiable membership functions we can then apply learning algorithms such as backpropagation to tune the parameters in the (generalized) bell function and achieve optimal performance of the overall supervisory model.

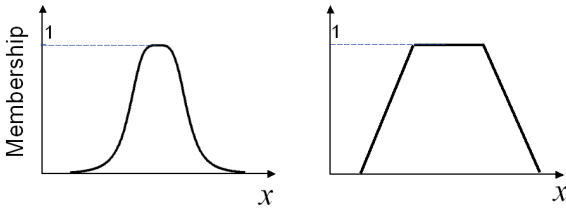


Figure 13. Examples of bell-shaped and trapezoidal membership functions.

5.3 Conclusions

We proposed a systematic approach to analyzing the potential causes of anomalies in dynamic systems, ranging from incipient system failures to malfunctioning sensors, operating the system in unusual regions, using inappropriate anomaly detection models, etc. For each cause, we extended the PHM cycle, creating an anomaly resolution action. Within this approach, we focused on the inaccuracy of the anomaly detection models and proposed a hybrid approach based on a fuzzy supervisory system and an ensemble of locally trained auto associative neural networks (AANN's).

In this approach we interpolate among the outputs of local models to assure smoothness in operating regime transition and then provide continuous condition monitoring to the system. Experiments on simulated data from a high bypass, turbofan aircraft engine model demonstrated promising results.

REFERENCES

- P. Bonissone (2008a). Soft Computing Applications in PHM, *Proc. FLINS 2008*, Madrid, Spain – in *Computational Intelligence in Decision and Control* (Da Rua, Montero, Lu, Martinez, D'hondt, Kerre, eds.), pp 751-756, World Scientific 2008.
- P. Bonissone (2007). Soft Computing Applications in Prognostics and Health Management: A Time and Knowledge Framework with Selected Case Studies, *Proc. AAAI Fall Symposium on Artificial Intelligence for Prognostics*, Arlington, VA..
- P. Bonissone, N. Iyer (2007). Soft Computing Applications to Prognostics and Health Management (PHM): Leveraging field data and domain knowledge, *Proc. 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, pp. 928-939, San Sebastián (Spain).
- D.L. Mattern, L.C. Jaw, T.-H. Guo, R. Graham, W. McCoy (1998). Using Neural Networks for Sensor Validation, *Proc. 34th Joint Propulsion Conference*, Seattle, WA, 1998; AIAA-98-3547; NASA/TM-1998-208483.
- X Hu, P. Bonissone, R. Subbu (2009). Robust Model Selection Decision-making using a Fuzzy Supervisory Approach, *Proc. IEEE MCDM 2009*, Nashville, TN.
- A. Varma, P. Bonissone, W. Yan, N. Eklund, K. Goebel, N. Iyer, S. Bonissone (2007). Anomaly Detection using Non-Parametric information, *Proc. ASME Turbo Expo 2007: Power for Land, Sea and Air*, Montreal, Canada.
- P. Bonissone (2008 b). Research Issues in Multi Criteria Decision Making (MCDM): The Impact of Uncertainty in Solution Evaluation, *Proc. IPMU 2008*, Malaga, Spain.
- A. Patterson, P. Bonissone, and M. Pavese (2005). Six Sigma Quality Applied Throughout the Lifecycle of and Automated Decision System", *Journal of Quality and Reliability Engineering International*, 21(3):275-292.
- M.A. Kramer (1991). Nonlinear Principal Component Analysis Using Auto-associative Neural Networks, *AIChE Journal*, 37(2): 233-243.
- X. Hu, H. Qiu, N. Iyer, (2007). Multivariate Change Detection for Time Series Data in Aircraft Engine Fault Diagnostics, *Proc. 2007 IEEE International Conference on Systems, Man, and Cybernetics*, Montreal, Canada.
- L.A. Zadeh (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3-28.
- E. Ruspini, P. Bonissone, and W. Pedrycz, *Handbook of Fuzzy Computing*, Institute of Physics, Fall 1998, ISBN: 0750304278
- E.H. Mamdani and S. Assilian (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Machine Studies*, 7(1):1-13.
- B. Schweizer and A. Sklar (1983). *Probabilistic Metric Spaces*, North Holland, New York.
- P.P. Bonissone (1987). Summarizing and Propagating Uncertain Information with Triangular Norms, *International Journal of Approximate Reasoning*, 1(1):71-101.
- J.-S.R. Jang, C.-T. Sun, E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing- A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall.
- B. Lerner, H. Guterman, M. Aladjem, I. Dinstein (1999). A Comparative Study of Neural Network Based Feature Extraction Paradigms, *Pattern Recognition Letters*, 20, pp 7-14.
- M.A. Kramer (1992). Autoassociative neural networks, *Computers & Chemical Engineering*, 16 (4):313-328.
- J.W. Hines, I E. Uhrig (1998). Use of Autoassociative Neural Networks for Signal Validation, *Journal of Intelligent and Robotic Systems*, 21(2): 143-154.
- H. Berenji, W. Yan, D. Vengerov, R. Langari, M. Jamshidi (2004). Using gated experts in fault diagnosis and prognosis, *Proc. 2004 IEEE International Conference on Fuzzy Systems*, pp. 463-467.



Piero Bonissone received a Ph.D. degree in EECS from the University of California at Berkeley in 1979. A Chief Scientist at GE Global Research, he has been a pioneer in the field of fuzzy logic, AI, soft computing, and approximate reasoning systems applications since 1979. Recently he led a Soft Computing group in the development of SC application to diagnostics and prognostics of processes and products, including the prediction of remaining life for each locomotive in a fleet, to perform efficient assets selection. His current interests are the development of multi-criteria decision making systems applied to PHM issues, and the automation of intelligent systems lifecycle, i.e. the development of processes to create, deploy, and maintain smart SC-based systems that provide customized performance while adapting themselves to avoid obsolescence. He is a Fellow of the IEEE, AAAI, IFSA, and a Coolidge Fellow at GE Global Research. In 2008 he received the II Cajastur International Prize for Soft Computing from the European Centre of Soft Computing. He served as Editor-in-Chief of the *International Journal of Approximate Reasoning* for thirteen years. He is on the editorial board of five technical journals and is Editor-at-Large of the *IEEE Computational Intelligence Magazine*. He co-edited six books and has over 150 publications in refereed journals, book chapters, and conference proceedings, with an H-Index of 20. He received 45 patents issued from the US Patent Office (plus 44 pending patents). From 1982 until 2005 he has been an Adjunct Professor at RPI, in Troy NY, where he has supervised five Ph.D. theses and 33 Master theses. He has co-chaired 12 scientific conferences and symposia focused on Multi-Criteria Decision-Making, Fuzzy sets, Diagnostics, Prognostics, and Uncertainty Management in AI. In the past, while serving as President of the IEEE Neural Networks Society (now Computational Intelligence Society) he was also a member of the IEEE Technical Board Activities (TAB). He has been an Executive Committee member of NNC/NNS/CIS society since the past 16 years.



Xiao Hu received the B.S. degree in electrical engineering from Sichuan University, China, in 1998, and the M.S. and Ph.D. degrees in computer engineering from the University of Missouri-Rolla in 2001 and 2004, respectively. Prior to joining the Computing and Decision Sciences

Group as a Research Scientist at General Electric Global Research Center in 2005, he spent three summers at Boeing Phantom Works, developing algorithms to diagnose aircraft engine health conditions. In the GE Global Research Center, he is currently engaged in the activities redeveloping anomaly detection, diagnostics and prognostics algorithms for industrial equipments and processes. His research interests also include developing new and improving existing machine learning/computational intelligence algorithms for system modeling, optimization and decision-making.



Raj Subbu received the Ph.D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2000. Since 2001, he has been a Senior Computer Scientist in the Industrial Artificial Intelligence Laboratory within the Computing and Decision Sciences division of GE Global Research, Niskayuna, NY, where he leads research and development in artificial intelligence, evolutionary algorithms, optimization, visualization, multi-criteria decision-making, and intelligent control. His work at GE Global Research has focused on the application of these methods to GE's engineered assets and financial processes, and towards technology management for creating new products leveraging widely distributed teams. He led the creation of Kn3, a real-time intelligent model-based multi-objective optimization and control software architecture and product that is applicable across a wide variety of power plants, industrial processes, energy trading, and engineered assets. Kn3 has been deployed internationally. Dr. Subbu has authored over fifty publications and proceedings, has received eight U.S. patents, and has twenty six U.S. patents pending. He is the principal coauthor of the book *Network-Based Distributed Planning Using Coevolutionary Algorithms* (World Scientific, 2004). Dr. Subbu received the Andrew P. Sage Best Transactions Paper Award of papers published in 2005 in the *IEEE Transactions on Systems, Man, and Cybernetics*, and the Best Paper Award at the IEEE International Conference on Fuzzy Systems in 2003. He is an Associate Editor of the *IEEE Transactions on Systems, Man and Cybernetics Parts A and C*, is an IEEE Senior Member, and serves on the program committees of international conferences.