Applications of Artificial Intelligence and Decision-Making Methods in PHM

Paddy Conroy[1], Jacek Stecki[2], Joshua Bautista[3], Alex Ringeri[4]

[1,2,3,4] *PHM Technology, Melbourne, Victoria, 3068, Australia*

paddy.conroy@phmtechnology.com
jstecki@phmtechnology.com
joshua.bautista@phmtechnology.com
alex.ringeri@phmtechnology.com

ABSTRACT

Developing PHM capability for a system is a multi-staged process. This paper explores genetic algorithms, neural networks, fuzzy logic systems, AHP (Analytical Hierarchy Process), and Boolean logic to synthesize and fuse complex decisions arising in PHM design. Tools for PHM analysis are typically introduced and utilized towards the end of a products design or potentially after design. The methods proposed are tools that can be implemented during conceptual and early stage preliminary design prior to specific hardware design decisions being made. As a result, diagnostic capability can be developed along with the broader system allowing better embedded design of diagnostic instruments into the system and giving PHM a greater role in operation rather than being a secondary consideration of system development.

1. BACK GROUND

PHM being a discipline of engineering focused on the detection of failures in mechanical systems for the purpose of maintenance, reliability, and safety. Prognostics refers to the estimation of life remaining in the item before functional failure (inability to perform a function) occurs. Equally important in the PHM process is the diagnosis of failures. Diagnosis referring to the detection (awareness that a failure is present) and isolation (awareness of which item in the system is failing and how).

A comprehensive PHM process or system incorporates elements of condition monitoring, state assessment, diagnostics, failure progression analysis, prognostics, and maintenance considerations (Sheppard, Kaufman &Wilmering, 2009). Focusing on implementing PHM into the design of engineering systems during the conceptual or preliminary design phase, this paper contains details and discussions of several methods to aid in design for PHM. Assessing and designing a PHM system through the lens of diagnostic capability lends itself to the early stage of product development as at this point the general configuration and functions of the system are known, however the physical structures and failures may not. Prognostics may not be possible at a preliminary design stage due to the requirements for physics of failure or data driven trend analysis, unavailable at this point in design.

PHM's place in the design process can be varied however the prevalent approaches to PHM result in a largely complete system design prior to the introduction of PHM considerations.

Data driven diagnostics take available data/information and via methods, such as neural networks, seek to form diagnostic rules from patterns in the data. Whilst valid, there are downsides to an approach that is inherently biased towards the later side of design such as limited sensor placement potential due to inflexible design, lack of system understanding, misattribution of physical causes of failure, and the inability to adequately estimate costs.

Physics of failure models tracking degradation over time offer increased fidelity over data driven methods however suffer many of the same problems.

Additionally, the computationally expensive process limits the broad, system scale application of physics of failure (Sheppard et al, 2009).

By introducing PHM considerations earlier allows design for PHM, earlier costings, sensor placement flexibility, and development of diagnostic design along with criticality and failure analyses.

Failures within an engineering system can be sensed in a multitude of way. Using sensors to detect failures ultimately requires the presence of observable symptoms that occur as a result of a failure. A symptom being a manifested condition occurring as the result of a failure. Symptoms can be the directly indicative of a failure (for example an observation that a shaft has fractured), or they may be indirect (for example increased power consumption as a result of wear in bearings).

To diagnose a failure as having occurred a uniquely identifiable syndrome that is associated to that failure must be observed. In the case that a non-uniquely identifiable syndrome is observed a diagnosis has not been made, or the result is ambiguity in the diagnosis. Ambiguity referring to the inability to distinguish between multiple failures (this may sometimes be acceptable if the ambiguity narrows down potential diagnoses in a controlled manner).

In the case of this paper the focus is specifically in failures in engineering systems. The observables being discussed are failure mode responses (properties of an item indicative of function/performance) due to the conditions imposed during modelling (early stage design using a logical modelling framework).

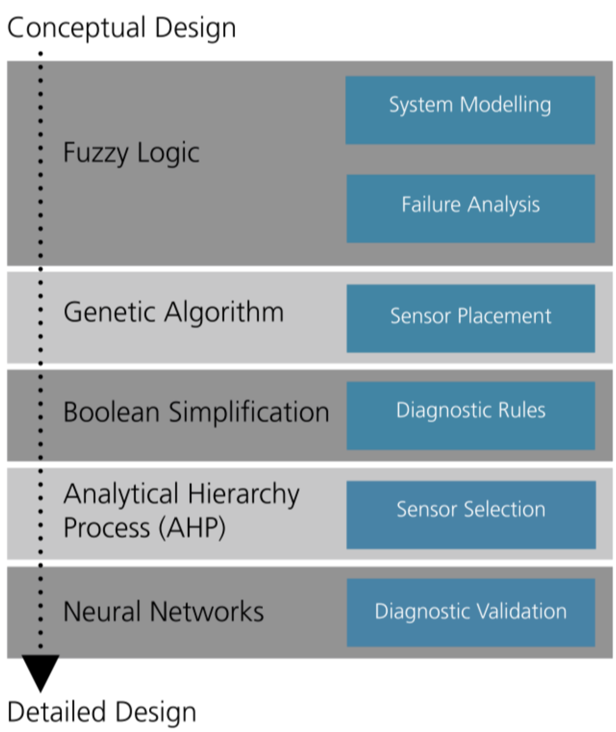

Figure 1: Use of methods across phases of diagnostic design

The initial success metric from which to judge a prospective diagnostic capability / sensor set is coverage (coverage being the proportion of failures in the system that can be diagnosed without ambiguity).

Once hardware has been introduced (in the form of physical sensor allocation) then sensor parameters such as cost, weight, probability of detection, and reliability can be utilized as prominent metrics for comparing and selecting an optimal solution.

Ultimately, the questions that require answering during sensor set design (particularly from an early stage) are:

Where should sensors be placed? How to diagnose a failure? What physical sensor types should be used at a given location?

2. METHODS

2.1. Fuzzy Logic

Fuzzy logic is a method for translating knowledge and expertise into a consistent, rule-based form for the purpose of some analysis. It can be used to map human language, with the vagueness and imprecision inherent within, to a crisp value based process via a set of rules.

The application of fuzzy logic discussed is twofold; firstly, modelling of a system, secondly, prioritization of failure diagnosis based on engineering risk.

2.1.1. Fuzzy Cognitive Maps (FCM)

FCM is a methodology of modelling that represents a network of interconnected concepts in order to understand the network's performance. Each of the concepts in the network may be connected to other concepts through causal connections that represent the performance dependencies between the concepts. Concepts are factors in the network that may exert influence on one another through causational relationships (e.g. increase in concept A causes decrease in concept B). Utilizing FCM allows complex networks of interdependent concepts to be modelled and the impacts of changes in magnitude to one or more of the concepts simulated as they impact upon the network.

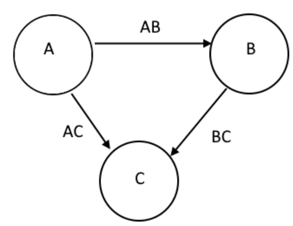

Figure 2: Basic FCM diagram with three concepts

In order to simulate the behaviour (or response) of the system, FCM looks at all initial state of the concepts in the system, the causal connections between the concepts and any perturbations affecting the concepts (Palaez & Bowles, 1995). The process used to simulate the system response can be represented using an iterative set of matrix multiplications (Stylios & Groumpos, 1999):

$$\boldsymbol{A}^{t-1}\boldsymbol{W} + \boldsymbol{P} = \boldsymbol{A}^{t} \quad (1)$$

Where:

$\boldsymbol{A}^{t-1}$ is the system state vector from the previous time-step. During the first time-step of the analysis $\boldsymbol{A}^{t-1}=\boldsymbol{A}^{0}$. $\boldsymbol{A}^{0}$ represents the initial state of the system (the value of each concept at the beginning of the analysis)

$\boldsymbol{W}$ is the weighting matrix that describes all the inter-connections between concepts in the system

$\boldsymbol{P}$ is the perturbation vector that describes any permanent changes or deviations to the concepts in the system

$\boldsymbol{A}^{t}$ is the system state vector that represents values of the flow properties after a time-step

FCM is used in modelling to predict or capture the syndrome that will be tested for using sensors.

The FCM modelling framework is the equivalent of a logical model with functions and flow properties assigned to each item. The functional flow properties being the concepts of the FCM model and connections being assigned with weights (or causal strengths) indicating how the concepts influence one another (Styblinski & Meyer, 1988).

These causal connections define the interactions between item flow properties. Failures of the system are ultimately simulated using the model. In order to simulate, failures are "injected" into the model by perturbing flow properties. To get the predicted/ simulated "syndrome" of failure, the state of the flow properties in the model after the failure has been perturbed through the model is compared to the state of the flow properties in the model before the failure was perturbed. This requires two consecutive simulations of the FCM model:

- The first simulation does not include the perturbation vector, simulating the operational or undisturbed system. This establishes a nominal system response.
- The second simulation introduces the perturbation vector representing a failure within the system. The results of this simulation are compared against that of the nominal system response. The relative change in response represents the potentially observable syndrome of the failure that was introduced.

Table 1: Failures and corresponding test points

	Test Point		
Failure	**1**	**2**	**3**
A	High	Nominal	Nominal
B	High	Low	Nominal
C	Nominal	Low	High

End output of this is to obtain a "fuzzy", or qualitative, syndrome that maps failures to symptoms that can be sensed. These syndromes can be collectively presented in a diagnostic table for the system (Hess, Stecki & Rudov-Clark, 2008).

2.2. Fuzzy Criticality

Fuzzy criticality uses the fuzzy logic framework to assess the risk associated with each failure. Risk generally being a measure of each failure's potential of severity, it's probability of occurrence and the ease of which that failure can be detected.

Defining and applying a fuzzy rule base for criticality offers an easily implemented criticality assessment at the early stage of design within the scope of this discussion (Fonseca & Knapp, 2001).

Risk Priority Number (RPN) is a ubiquitous example of criticality in which each failure is allocated severity (S), occurrence (O), and difficulty of detection (D) values on a scale ranging from 1 to 10 (with 1 representing least critical, 10 representing most). The values assigned to each metric can be based on a qualitative or quantitative set of guidelines corresponding to each integer in the 1 to 10 scales and determining these guidelines forms the basis of the criticality method being applied. The product of the three metrics (O*S*D) represents the failure's criticality.

Fuzzy criticality seeks to extend the RPN methodology by incorporating measures of failure progression and causal probability into the criticality assessment, similar metrics are used in work by Liu, Yang, Wang, Sii, and Wang (2004). This is particularly advantageous as the FCM model being utilized can act as the structure onto which these new metrics are assigned.

Fuzzy Criticality metrics:

- *Apparent Occurrence* is the overall likelihood or frequency of a failure occurring and resulting in an end-effect. Calculated as a mapping between Occurrence and Causal Probability (of the failure resulting in end-effect).
- *Apparent Severity* is the severity of the end-effect considering the relative rate at which the initial failure develops and progresses to that end-effect. Calculated as a mapping between Severity and Progression Rate.
- *Difficulty of Detection* as in RPN criticality, the Difficulty of Detection is a measure of the ease at which the operator of the system may diagnose the specific failure.

Collectively these metrics are then mapped to an overall Fuzzy Criticality measure of risk.

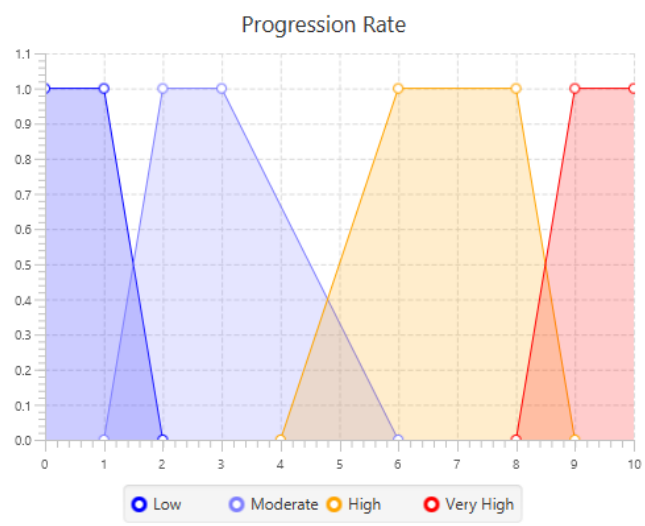

Figure 3: Example of fuzzy memberships for the Progression Rate parameter

Mappings between parameters are based on rules that are determined prior to commencement of criticality analysis. Example of mappings between parameters are shown in the application section of the paper.

The application of Fuzzy Criticality as a tool for sensor set design is as a measurement of risk and prioritization for the failures the sensor set is seeking to diagnose. The sensor set can be designed in order to prioritize the diagnosis of those most critical failures, and in the process of applying a sensing methodology decreasing the risk of failures by reducing the Difficulty of Detection.

2.3. Genetic Algorithm

A methodology that mimics nature's evolutionary process, a genetic algorithm is a form of heuristic that can generate high quality solutions over a large potential solution space.

Evolution in a genetic algorithm is an iterative process that functions as a search through the potential solutions. The genetic algorithm initially randomly generates a number of candidate solutions which are evolved into better solutions with each iteration. Every iteration, the highest fitness candidate solutions are selected to produce the next generation of solutions. The selection process gives preference to the solutions that best meet the criteria.

In application to diagnostics, genetic algorithms can be used to select sensor sets (combinations of sensors) based on the diagnostic table generated in using the FCM model. The criteria for preferencing a generated candidate sensor set is given to solutions with high coverage, and low sensor count, these can be adjusted and specified depending on the analysts own preferences (criteria). The selected solutions are combined and modified to produce the next generation. This process continues until the specified iteration limit has been reached.

The fitness function used to assess sensor sets generated uses diagnostic coverage and number of sensors. There are many potential fitness functions based on strategies for sensor set optimization. A simple strategy is described below, with other strategies being variants that bias the fitness function to optimize for a specific level of coverage or number of sensors.

$$f(c,s) = w_c . f_c(c) + w_s . f_s(s) \tag{2}$$

Where:

$c =$ coverage

$s =$ sensors in sensors set

$f(c,s) =$ fitness function

$w_c = 5$, coverage weighting

$f_c(c) = \frac{c}{100}$

$w_s = 1$, sensor weighting

$f_s(s) = \frac{s_{max} - s}{s_{max}}$

$s_{max} =$ maximum possible value of s (equal to the number of columns in the diagnostic table)

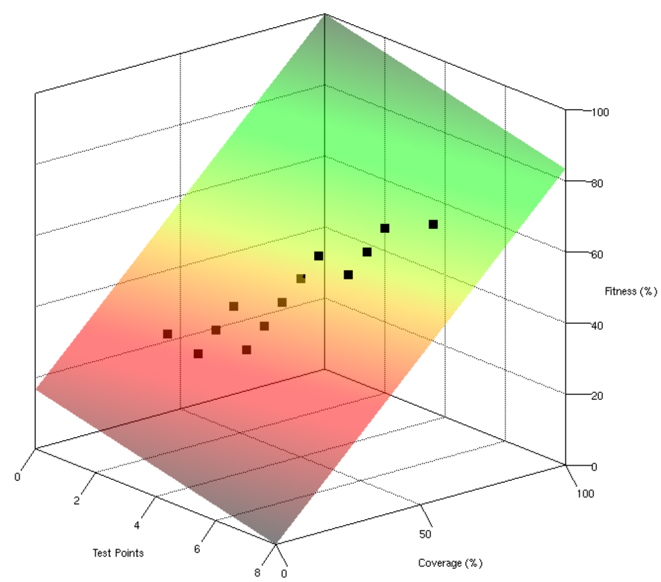

Figure 4: Plot of potential fitness, dots represent generated Sensor Sets for an example system

Analytical solution to this problem are possible, however with large / complex diagnostic tables, the time to solve these problems increases to the point where it is computationally impractical.

Advantages of a genetic algorithm in this context are relatively low time requirement for analysis and the ability to produce both broad and specific results by manipulation of the fitness function as a mechanism to change the analysis. The process will result in a list of Sensor Set's that are able to uniquely identify the failure responses found in the Diagnostic Table.

2.4. Boolean Simplification

Boolean simplification or logic optimization is a process to simplify a Boolean function into its smallest form via application of logic.

In practice this allows the expression of failures and the sensor locations required to diagnose them to be expressed as a truth table.

A broad definition of sensors as observable elements in the system, meaning a sensor can be either an onboard unit or a manual inspection point, this highlights the want for a minimal number of steps (inspections) to diagnose a specific failure (Kohda, Ohki & Inoue, 1991).

The diagnostic rules serve two purposes:

1. To confirm whether a specific failure has occurred
2. To determine which failure has occurred

Implemented by comparing rows against one another and finding the unique combinations within. A diagnostic rule is defined as the minimum number of sensor location that require observation to confirm the presence of a failure and to specifically distinguish a failure from all other failures. To that end, each diagnostic rule includes the locations that are unique for that failure when compared to each other failure's response at that location. If a single location does not uniquely identify the failure, then a set of locations are used that can uniquely identify the failure.

The diagnostic rules can be applied to the development of maintenance/detection instructions (such as those contained within a fault detection and isolation manual). This aids in sorting through and managing non-essential information and finding failures with fewer inspection steps.

Steps in Boolean Minimization:

1. Define functions for each failure based on the available sensors (the sensors identified by the genetic algorithm)
    a. Compare each failure with each other failure
    b. Find which inspection locations can be used to distinguish between each failure pairing
    c. The failure's function is constructed by combining all the inspection locations from each failure pairing
2. Use Boolean properties to reduce the functions to their smallest form
3. Convert the functions to diagnostic rules

2.5. Analytical Hierarchy Process (AHP)

The Analytical Hierarchy Process (AHP) is a multicriteria decision-making tool used to evaluate both quantitative and qualitative criteria. The AHP arranges criteria and sub-criteria into a hierarchical structure, similar to a family tree, in general these criteria can be anything that is comparable between the alternatives being compared (Saaty, 1990). In the context of engineering systems, criteria may include design parameters such as performance, cost, constraints, and so on. The purpose of this arrangement is to organise the most pertinent requirements into a series of simplified comparisons and rankings, followed by the synthesis of results. This process provides a traceable basis for the choices made that lead to a ranking the most important criteria and alternatives that are considered the "best-fit" to meet such criteria.

As a method for decision making, AHP presents a framework for determining which sensors are best utilized from available options.

2.6. Neural Networks

Artificial Neural Networks (ANNs) or simply 'Neural Nets' (NN) are computing systems used to determine a solution by analysing large datasets without the use of traditional, task-specific programming (Roemer, Byington & Schoeller, 2007). These systems are loosely based on the concept of neurons in biological neural networks.

Neural Networks consist of individual 'artificial neurons' called perceptrons (also known as sigmoid neurons). These perceptrons take in multiple binary inputs to produce a single binary output – either 0 or 1. This output is calculated with using weights in the form of real numbers to express the importance of each input to the output

A typical layout for a neural net consists of three elements:

- An Input Layer (Left-side)
- Input Neurons (Hidden Layer)
- Output Neurons (Right-side)

By using a multi-layered neural net, multiple outputs act as a single input for perceptrons in the next layer. Learning algorithms can be used to tune weights and biases of a perceptron network which responds autonomously in response to external stimuli.

The purpose of using Neural Networks in the context of diagnostics is to use a generated diagnostic set from a sensor analysis, use this dataset to 'train' a neural network which would then be able to identify the probability of a sensor giving a false reading or false alarm.

3. EXAMPLE / CASE STUDY

Here a case study of the above methodology is presented. The case study features a Driveline System belonging to an 8-wheel drive ground vehicle, such as a military armored personnel carrier. The Driveline has been developed to an expectable level for mid-preliminary design. The logical framework of the system has been determined and as such components can be modelled in terms of the functional properties they provide; however physical hardware has not been allocated or designed.

3.1. Fuzzy Logic Applied

3.1.1. Fuzzy Cognitive Maps

The Driveline System has system inputs of compressed air (transferred through the airline), a control signal (to actuate the air input), and rotational torque (from the engine providing power to the wheels via differentials and gearboxes).

The power into the system is split via a Transfer Case to two Driveshafts each leading to Differentials. The two Differentials (Differential Rear and Differential Front) supply the rear and front wheels with power (each Differential is responsible for 4 wheels). Whilst all 8 wheels are driven, the front 4 wheels are used for steering.

The full Driveline System contains 29 components and is too large to graphically display here. Below is an example of a section of system in logical format and then devolved into a FCM diagram.

The connections in the diagram are typically of a value of 1.0 for directionally forward connections and -0.4 for directionally backward connections (referred to as feedbacks). This results in a FCM simulation that reaches an equilibrium (i.e. each failure injected into the system results in each flow property in the system reaching a constant response).

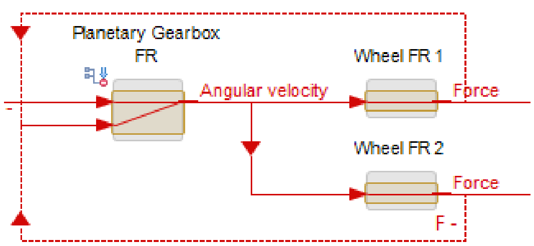

Figure 5: Logical structure of a section of the Driveline. Displayed is the front right Planetary Gearbox delivering angular velocity to two wheels

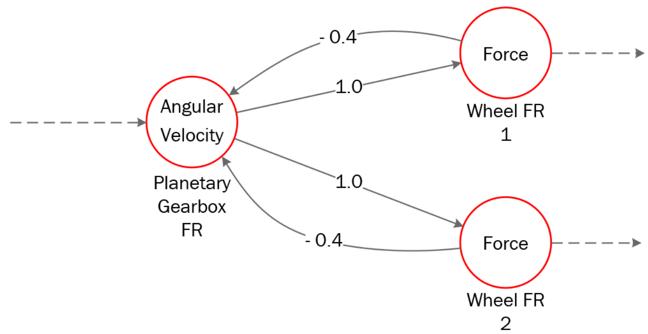

Figure 6: The logical structure is converted to a FCM diagram with strength of connection displayed on the connections. Dotted lines refer to connections leading to and from outside the diagram to other components in the Driveline

A diagnostic table is derived from the whole Driveline System by injecting failures into the model via perturbation of a single flow property and then propagating it through the system. Simulation steps:

1. Run FCM simulation with assumption of nominal behaviour represented by a value of 0.0 at each flow property concept. Note that the initial simulation does not include the perturbation vector (representing that no failures are being injected).
2. Introduce single failure as a perturbation of 1.0 in either positive or negative direction. A positive perturbation is equivalent to an increase in the flow property, a negative perturbation is equivalent to a decrease in the flow property. For the purposes of this system every component is assumed to have one functional failure, a decrease or loss of flow.
3. Run simulation until equilibrium is found.
4. Compare equilibrium response to nominal value (0.0) of each flow property in the simulation.
5. If the response is higher than the equilibrium point, the response is high, if lower the response is lower.
6. Diagnostic table is built by running a simulation for each failure and capturing the responses across the system. Each failure is represented by a row in the table.

3.1.2. Fuzzy Criticality Applied

Each failure (taken from the list of FCM simulated failures) can be assigned criticality. The goal of performing a Fuzzy Criticality analysis (or any alternative criticality analysis) is to prioritize the failures that carry the highest risk in order to design the diagnostic capability so that these failures may be isolated when they occur/are occurring (Tay & Lim, 2006).

As an example of the Fuzzy Criticality concept, four components will be analyzed:

- Driveshaft, which supplies the Differential Rear with power
- Driveshaft 2, which supplies the Differential Front with power
- Differential Rear, which transfers power to the four rear wheels via two gearboxes
- Differential Front, which transfers power to the four front wheels via two gearboxes

Fuzzy criticality containing the following parameters; Occurrence (Occ.), Causal Probability (CP), Apparent Occurrence (App. O), Severity (Sev.), Progression Rate (PR), Apparent Severity (App. Sev.), Difficulty of Detection (DoD), and Fuzzy Criticality.

Table 2: Criticality inputs and outputs

	Occ.	CP	App. O	Sev.	PR	App. Sev.	DoD	Fuzzy Criticality
Driveshaft	2.5 (L)	10 (H)	2.5 (L)	6 (M)	9 (VH)	7.5 (H)	8 (H)	5 (M)
Driveshaft 2	2.5 (L)	10 (H)	2.5 (L)	10 (VH)	9 (VH)	9.2 (VH)	8 (H)	5 (M)
Diff. Rear	8 (H)	10 (H)	7.5 (H)	6 (M)	7 (H)	7.5 (H)	8 (H)	7.5 (H)
Diff. Front	8 (H)	10 (H)	7.5 (H)	10 (VH)	7 (H)	9.2 (VH)	8 (H)	9.2 (VH)

Where:

L = Low, M = Medium, H = High, VH = Very High

The allocations of the raw criticality metrics (Occ., CP, Sev., PR, and DoD) are based upon historical knowledge of similar systems given the preliminary stage of design. The logic behind the allocations:

Occ. – Occurrence is allocated as low to the Driveshafts as known Driveshaft failures (primarily stress and torsional failures) are low occurrence under assumed operation. The Differentials experiencing wear alongside fatigue are prone to lower time to failure.

CP – The High CP value indicates direct causality with the end-effect.

App. O – Is produced as a mapping between Occ. and CP.

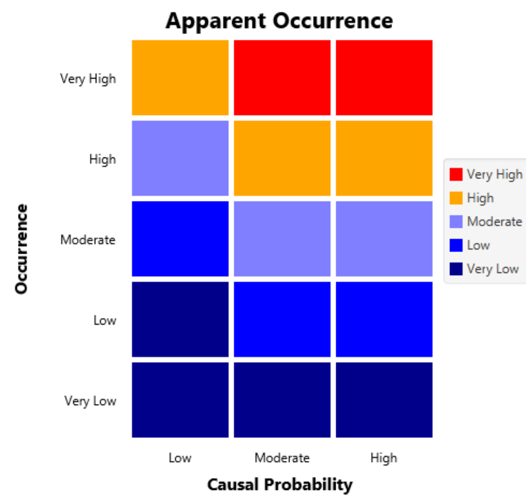

Figure 7: Mapping between Causal Probability and Occurrence to obtain Apparent Occurrence

Sev. – The front wheels providing steering mean the associated items (Driveshaft 2 and Differential Rear) are of higher relative criticality.

PR – As the primary cause of failure in the Differentials is wear as compared to stress in the Driveshafts, the Differentials experience a relatively slower progression to complete failure.

App. Sev. – Is produced as a mapping between Sev. and PR.

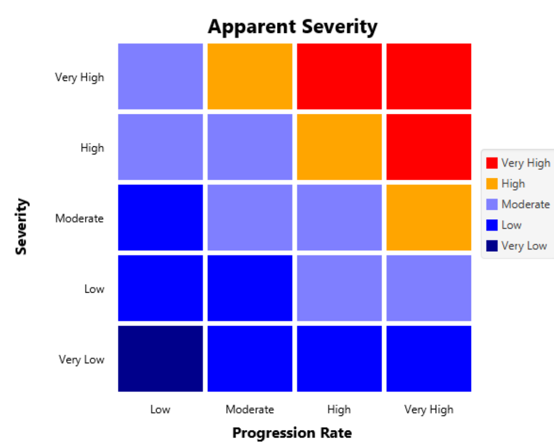

Figure 8: Mapping between Progression Rate and Severity to obtain Apparent Severity

DoD – All failures experience a similarly difficulty of detection.

Fuzzy Criticality – Is produced as a mapping between App. O, App. Sev., and DoD.

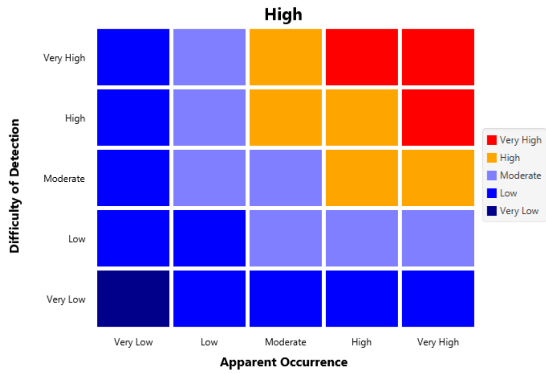

Figure 9: Mapping between Apparent Occurrence and Difficulty of Detection for the case of High Apparent Severity to obtain Fuzzy Criticality

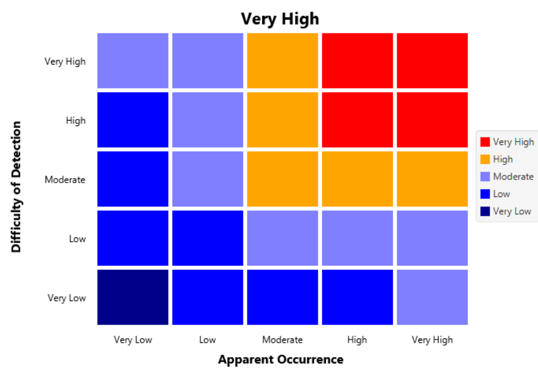

Figure 10: Mapping between Apparent Occurrence and Difficulty of Detection for the case of Very High Apparent Severity to obtain Fuzzy Criticality

Given the criticality analysis it is established that failures of the Differential Front component should specifically be covered by the diagnostic system being designed. Similarly, failures of the Differential Rear are of relative importance. These are the important deliverables from the Fuzzy Criticality analysis.

3.2. Genetic Algorithm Applied

The genetic algorithm is set-up to generate sensor sets for the system based on the diagnostic table produced via FCM and utilizing information from the Fuzzy Criticality analysis.

Initially, the genetic algorithm is run with the fitness criteria as default (equations mentioned in prior section). This will generate sensor sets with maximal coverage and minimal sensor location. Using this fitness, it is found that in order to achieve 100% coverage 16 sensors are required.

Fitness equation for maximum coverage and minimum sensor count:

$$f(c,s) = w_c.f_c(c) + w_s.f_s(s) \qquad (3)$$

Table 3: Summary of fitness calculation for sensor set generation targeting 100% coverage

Term	**Value**	**Description**
w_c	5	Coverage weighting
$f_c(c)$	1	Coverage function, $= \frac{c}{100}$
w_s	1	Sensor count weighting
$f_s(s)$	0.448276	Sensor count function, $= \frac{s_{max} - s}{s_{max}}$
s_{max}	29	Maximum number of sensor locations. Based on the model's table this is 29 (equivalent to the number of columns)
c	100	Coverage achieved by the sensor set generated
s	16	Sensor locations in the sensor set
$f(c,s)_{max}$	6	Maximum theoretical fitness value (if 0 sensors and 100% coverage)
$f(c,s)$	5.448276	Fitness value, based on sensor set generated
$f(c,s)$ %	90.8046	Percentage of maximum fitness

Limiting the number of potential sensor locations to 8 via adjusting the fitness, and the best possible coverage (with 8 sensor locations) is found to be 65.52% (equivalent to x out of y failures being covered).

In calculating fitness when a specific number of sensors is being targeted the same top-level fitness function is used however the coverage weighting is reduced to 1 and changes to the sensor count function are made as follows:

Table 4: Changes to the sensor count function

Term	**Value**	**Description**
$f_s(s)$	1	Sensor count function, $= \frac{p(s)}{p(s_{target})}$
$p(s)$	0.3982	Probability density function for sensor set generated, $= PDNormal\ (s, s_{target}, s_{range})$
$p(s_{target})$	0.3989	Probability density function for target number of sensors, $= PDNormal\ (s_{target}, s_{target}, s_{range})$
s_{target}	8	Sensor count targeted
s_{range}	1	Range ($\pm$ no. of sensors)

However, if a maximum of 8 sensor locations are going to cover failures of the two differential components, then only a 51.72% coverage is possible. Fitness is calculated in the same way as for the limited number of sensors strategy, however results are generated so that every sensor set created will contain the nominated failures of the Differentials.

When producing sensor sets via a genetic algorithm each fitness strategy can produce a multitude of sensor sets that fit the criteria, the differences between the sets being the sensor locations used and the failures being covered.

3.3. Boolean Simplification Applied

The output of the Genetic Algorithm is a sensor set with a number of identified required sensors to diagnose failures in the system. This can be displayed as a cut-down version of the diagnostic table with the unused sensor locations (columns) being removed.

In order to find the minimum number of sensors (or observations) required to diagnose each failure a process of Boolean minimization is undertaken (Kohda et al, 1991).

To simplify the example three failures and their sensor locations will be used to show the process.

The following cut down prop table was output from the GA and the first three failures will be used as an example, each sensor location has been assigned a letter:

Table 5: Sensor set specific diagnostic table

Component	Failure	Air Line 1 Static pressure	Air Line 3 Static pressure	Driveshaft 1 Torque	Wheel FL 1 Force	Wheel FL 2 Force	Wheel FR 1 Force	Wheel RL 1 Force	Wheel RR 1 Force
		A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Diff. Front (F_1)	Low	0	0	0	-1	-1	-1	0	0
Diff. Rear (F_2)	Low	0	0	0	0	0	0	-1	-1
Drive shaft (F_3)	Low	0	0	-1	1	1	1	-1	-1

Steps for developing diagnostic rules using Boolean expressions:

1. Define functions for each failure based on the available sensors (the sensors identified by the genetic algorithm)
    a. Compare each failure with each other failure
    b. Find which inspection locations can be used to distinguish between each failure pairing

$D_{12} = \{D, E, F, G, H\}$

$D_{13} = \{C, D, E, F, G, H\}$

$D_{23} = \{C, D, E, F\}$

    c. The failure's function is constructed by combining all the inspection locations from each failure pairing

$F_1 = (D \vee E \vee F \vee G \vee H) \wedge (C \vee D \vee E \vee F \vee G \vee H)$

$F_2 = (D \vee E \vee F \vee G \vee H) \wedge (C \vee D \vee E \vee F)$

$F_3 = (C \vee D \vee E \vee F \vee G \vee H) \wedge (C \vee D \vee E \vee F)$

2. Use Boolean properties to simplify the functions to their smallest form

$F_1 = D \vee E \vee F \vee G \vee H$

$F_2 = (C \wedge (G \vee H)) \vee D \vee E \vee F$

$F_3 = C \vee D \vee E \vee F$

3. Convert the functions to diagnostic rules

$IF\ D = -1\ OR\ E = -1\ OR\ F = -1\ OR\ G = 0\ OR\ H = 0$
$THEN\ Differential\ Front\ has\ failed\ Low$

$IF\ (C = 0\ AND\ G = -1\ OR\ H = -1) OR\ D = 0\ OR\ E = 0\ OR\ F = 0$
$THEN\ Differential\ Rear\ has\ failed\ Low$

$IF\ C = -1\ OR\ D = 1\ OR\ E = 1\ OR\ F = 1$
$THEN\ Driveshaft\ has\ failed\ Low$

This process is applied to each entire sensor set's diagnostic table yielding rules for diagnosis for each covered failure.

3.4. Analytical Hierarchy Process (AHP) Applied

The Analytical Hierarchy Process is used to select the best sensor for the identified locations from the GA analysis. Sensors are selected from a library of candidate sensors based upon the preferences of the analyst, those preferences being captured in the AHP framework.

For this example, a set of sensors will be assessed in terms of their applicability to sensing the rotational motion of the Driveshaft. The properties of interest are the cost, size, and power requirements of the sensors.

A difficulty in this process is finding accurate sensor data as data supplied by vendors is typically incomplete or incomparable. As such the properties will be assessed in terms of a qualitative taxonomy, which itself will be ranked using AHP. The qualitative taxonomy and general impressions of each type of sensor are derived from a Fleming (2001) paper and will be used as an example of the subjective information inherent in these types of problems. The purpose of AHP is to bring traceability and objectiveness to the decision making.

The first step in AHP is to establish candidates (also known as alternatives) and list the properties that will be compared. Each property for each candidate sensor has a value assigned to it (in this case all values are qualitative). Candidate sensors are Inductive (1), Wiegand Effect (2), Hall Effect (3), Magnetoresistor (4), AMR Magnetoresistive (5), and GMR Magnetoresistive (6).

Table 6: Sensor candidates with criteria values

Criteria	**Sensor Candidates**					
	1	2	3	4	5	6
Cost	Low	High	Low	Medium	Medium	Medium
Size	Small/Moderate	Moderate	Small	Moderate	Moderate	Moderate
Power	Passive	Passive	Active	Active	Active	Active

The next step is to get relative rankings between each criterion in terms of the priority they should be given, the greater the priority, the higher the weighting for the criteria in deciding on the candidate sensor.

Similarly, rankings are established for the level of impact that each property value has relative to the other potential values.

The inputs to the rankings are based on individual or organizational weightings.

	Cost	Size	Power
Cost	1	1.25	1.50
Size	0.80	1	3.00
Power	0.67	0.33	1

Figure 11: Relative importance between the three criteria is input. Here Size is being weighted as 1.25x more important than Cost

	Low	Medium	High
Low	1	1.50	2.00
Medium	0.67	1	1.50
High	0.50	0.67	1

Figure 12: Inputs for Cost weightings

	Small	Small/Moderate	Moderate
Small	1	1.20	1.50
Small/Moderate	0.83	1	1.25
Moderate	0.67	0.80	1

Figure 13: Input for Size weightings

	Passive	Active
Passive	1	3.00
Active	0.33	1

Figure 14: Inputs for Power weightings

Each pairwise weighting matrix is normalized via the following procedure in order to get an importance of each criteria/value as relative to the other criteria/values in the matrix.

To obtain normalized outputs from each matrix the following steps are followed:

1. Square the matrix

2. Sum each value then normalize

3. Create eigenvectors after each iteration

4. The previous three steps are repeated until the eigenvector converges – when this occurs the result is recorded

Table 7: Normalized criteria weightings

Criteria	***Normalized Weighting***	***Ranking***
Cost	0.388052448	2
Size	0.42133456	1
Power	0.190612992	3

Table 8: Normalized cost values

Cost Values	***Normalized Weighting***	***Ranking***
Low	0.459958088	1
Medium	0.318917126	2
High	0.221124785	3

Table 9: Normalized size values

Size Values	*Normalized Weightings*	*Ranking*
Small	0.4	1
Small/Moderate	0.333333333	2
Moderate	0.266666667	3

Table 10: Normalized power values

Power Values	*Normalized Weightings*	*Rankings*
Passive	0.75	1
Active	0.25	2

To obtain an overall ranking of the candidate sensors the normalized weightings of the criteria values are used to replace the qualitative terms in the original candidate-criteria table. The criteria weightings matrix is multiplied with that table and each sensor candidate is given a ranking based on the summation of its constituent rankings, yielding the following results.

Table 11: Ranking of sensor candidates

Sensors	1	2	3	4	5	6
AHP Outputs	0.46	0.34	0.39	0.28	0.28	0.28
Results	1	3	2	4	4	4

Based on the input selection criteria and the weightings provided it can be shown that the Inductive sensor is the most appropriate for sensing rotational motion at the Driveshaft.

3.5. Neural Nets Applied

The system model has been created, diagnostic rules (in indicating sensor locations) have been generated and sensors have been selected for their specific use case. Once complete, an ANN can be configured. A training session is run which continues until the number of cycles converge to the required result. Finally, the trained ANN displays the fault responses with its associated symptom responses.

The item of interest is the failure of the Driveshaft component, and its effect on Driveshaft 1 and Wheel components (RL1 & RR1) downstream.

Having generated sensor sets and diagnostic rules the ANN can now be trained by setting the number of hidden nodes (input & output nodes are fixed based on the number of sensor locations in the system).

Failure of the neural net to converge is indicated, and an analyst can repeat heuristic process until the ANN training yields a successful result. Upon successful completion of ANN training, the analyst then runs the neural net which outputs detected fault responses (red – UP, blue – DOWN, green – NO CHANGE) and their correlated symptom responses.

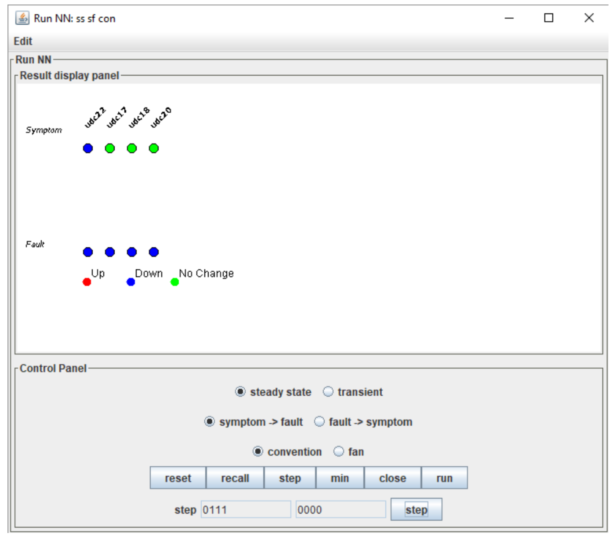

Figure 15: Running ANN to derive fault responses associated with symptom responses

Finally, different steps can be cycled through which are different configurations of faults and corresponding symptoms by using the trained ANN.

By observing the failure responses indicated by sensors, we can ideally detect which failures occur. There are cases where syndromes (responses of sensors) are not observed in a diagnostic set – this could be due to faulty sensors and would prevent a correct diagnostic observation.

The purpose of the Neural Network in this context is to determine, based on current failure responses, what the most probable failure would be during system operation. Diagnostic rules and Neural Network outputs should provide the same diagnosis when all sensors are working, while degraded states where sensors are detecting inconsistent failure responses are identified by the Neural Network outputs and a diagnosis made based on incomplete information.

4. CONCLUSIONS

Identified is a process for the design and development of diagnostic capability beginning in early stage design. The basis for diagnostic design begins with the formation of a model to act as a framework for capturing failures and associated observables. Answering questions such as where sensors should be placed, how to diagnose a failure when it occurs, and which physical hardware should be selected can be answered using a set of complementary tools (genetic algorithm, Boolean simplification, and AHP respectively).

Concluding with a set of diagnostic rules that are applied through the usage of a trained Neural Network.

REFERENCES

Fleming, W. J. (2001). Overview of automotive sensors. *IEEE Sensors Journal*, 1 (4), 296-308

Fonseca, D. J., Knapp, G. M. (2001). A fuzzy scheme for failure mode screening. *Fuzzy Sets and Systems*, 121, 453-457

Hess, A., Stecki, J., Rudov-Clark, S. (2008). The maintenance aware environment: the development of an aerospace PHM software tool. *IEEE International Conference on Prognostics and Health Management (PHM08)*

Kohda, T., Ohki, S., Inoue, K. (1991). Construction of failure diagnosis rules from diagnosis tables. *International Conference on Industrial Electronics, Control and Instrumentation*, 137-142

Liu, J., Yang, J., Wang, J., Sii, H., Wang, Y. (2004). Fuzzy rule-based evidential reasoning approach for safety analysis. *International Journal of General Systems*, 33 (2-3), 183-204

Pelaez, C. E., Bowles, J. B. (1995). Using fuzzy cognitive maps as a system model for failure modes and effects analysis. *Information Sciences*, 88 (1), 177-199

Roemer, M. J., Byington, C. S., Schoeller, M. S. (2007) Selected artificial intelligence methods applied within an integrated vehicle health management system. *AAAI Fall Symposia*, 87-95

Saaty, T. L. (1990). How to make a decision: the analytical hierarchy process. *European Journal of Operational Research*, 48, 9-26

Sheppard, J. W., Kaufman, M. A., Wilmering, T. J. (2009) IEEE Standards for Prognostics and Health Management, *IEEE A&E SYSTEMS MAGAZINE*, 34-41.

Styblinski, M. A., Meyer, B. D. (1988). Fuzzy cognitive maps, signal flow graphs, and qualitative circuit analysis. *IEEE International Conference on Neural Networks*, 549-556

Stylios, C. D. Groumpos, P. P. (1999). Mathematical formulation of fuzzy cognitive maps. *Proceedings of the 7th Mediterranean Conference on Control and Automation*, 2251-2261

Tay, K. M., Lim, C. P. (2006). Fuzzy FMEA with a guided rules reduction system for prioritization of failures. *International Journal of Quality and Reliability Management*, 23 (8), 1047-1066