World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:6, No:10, 2012

Intelligent Condition Monitoring Systems for Unmanned Aerial Vehicle Robots

A. P. Anvar, T. Dowling, T. Putland, A. M. Anvar, and S.Grainger

Abstract**—**This paper presents the application of Intelligent Techniques to the various duties of Intelligent Condition Monitoring Systems (ICMS) for Unmanned Aerial Vehicle (UAV) Robots. These Systems are intended to support these Intelligent Robots in the event of a Fault occurrence. Neural Networks are used for Diagnosis, whilst Fuzzy Logic is intended for Prognosis and Remedy. The ultimate goals of ICMS are to save large losses in financial cost, time and data.

Keywords**—**Intelligent Techniques, Condition Monitoring Systems, ICMS, Robots, Fault, Unmanned Aerial Vehicle, UAV, Neural Networks, Diagnosis, Fuzzy Logic, Prognosis, Remedy.

I. Introduction

ENGINEERING systems have experienced a surge of growth in terms of complexity and autonomy since the dawn of the information age. As these systems increase in complexity, the Condition Monitoring Systems required to monitor said systems must also evolve to compensate.

During the Vietnam War it has been estimated that seventy percent of aircraft losses could have been avoided if the fault tolerant flight control systems were designed properly [1]. Since that time, fault tolerant flight control systems have evolved into highly advanced designs, such as Intelligent Troubleshooting Systems. Intelligent Condition Monitoring Systems are made up of three critical subsystems; Fault Diagnosis, Prognosis and Remedy. Diagnosis is the process that is used to detect a fault. Prognosis is the investigation of the cause of and evaluation of the nature of the fault and Remedy is providing possible solutions to the fault.

In most of today's high performance aircraft (military and commercial) resides a triple redundancy system [2], which involves having extra sensors or actuators for any one application, such that if one or two of these sensors/actuators fail, there is still one more system that will be able to control a vital component. For instance, a Boeing 777 commercial airliner has a triple redundancy system, and millions of lines of software code, with close to sixty percent of this code dedicated to redundancy management [3]. However for smaller scale, less complex aircrafts that need to consider much lower budgets and weight considerations these systems are much too costly.

In these small scale applications (such as with a civilian UAV system) a software based approach is generally favoured. This is due to the lower cost of the implementation, and the little to no weight consideration needed. There are many different methods of implementing software control, but most can be ruled out when considering the application of the system. While many traditional linear, time invariant controllers (e.g Kalman Filters) can be used for some basic aircraft situations, for example at cruise, when the aircraft enters a dynamically changing situation, the classic linear, time invariant systems cannot model the dynamics correctly and hence cannot model faults in the actuators or sensors which causes these systems to lose their appeal. It is for these dynamic situations, that a non-linear, time variant process is needed to be implemented. It has been found that Neural Networks are more than able to handle these situations, as they are non-linear approximators [2] and will outperform conventional methods with fewer false alarms (detecting a fault that does not exist) and fewer missed faults [4].

In order to convert this diagnosis data into an action, some form of logic must be implemented into the system to make decisions based on fault information. An increasingly popular system for this is the Fuzzy Logic System. Fuzzy Logic is a generalisation of standard logic, in that Fuzzy Logic values have a range of 0 to 1, which is in contrast to the usual logical system of values either being 1 (true) or 0 (false). Fuzzy Logic is "an effective means of capturing the approximate, inexact nature of the world" [5]. What this means is that Fuzzy Logic is able to take inexact variables, such as linguistic variables (i.e a 'hot' day), and make decisions based on the variables. Because of the inexact nature of the diagnostic process (at what point does an engine go from being in the normal operating range to an abnormal operating range?), the Fuzzy Logic System is well suited to making informative decisions based on the diagnostic data using simple if-else logic [6].

Pawar et al (2007) [7] designed and implemented a system consisting of Neural Networks to detect faults, and Fuzzy Logic to implement a local remedy (isolating the fault) on helicopter. The system was designed to measure; flap and lag bending deflections, elastic tip deflection and three forces and moments about the rotor hub. The system was able to detect and isolate rotor faults with an accuracy of 90-100%. The system was also, "robust and gave excellent results even when some measurements were not available," [7]. When compared to a classic expert system, the Fuzzy Logic system was much more capable of dealing with situations where noise was present.

A.P. Anvar is a Post Graduate Candidate in the field of Mechatronics Engineering, from the School of Mechanical Engineering, the University of Adelaide 5005 AUSTRALIA (e-mail: amir.p.anvar@student.adelaide.edu.au).
T. Dowling is a Graduate Engineer from the School of Mechanical Engineering, the University of Adelaide 5005 Australia.
T. Putland is a Graduate Engineer from the School of Mechanical Engineering, the University of Adelaide 5005 Australia.
A.M. Anvar is with the School of Mechanical Engineering, the University of Adelaide 5005 AUSTRALIA (e-mail: amir.anvar@adelaide.edu.au).
and S. Grainger is with the School of Mechanical Engineering, the University of Adelaide 5005 Australia.

World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:6, No:10, 2012

II. Preliminary Design and Analysis

The two current structural designs for Maritime UAV Robots are Catapult Launched and Vertical Take-Off and Landing (VTOL). Figure 1 below shows an example of each type of these Robots respectively, pioneered by Students and Staff of the University of Adelaide. Although these Robots differ in structure and component types, their subsystems do not.

(a) (b)

Fig. 1 Maritime UAV-Robots during Operational Scenaios (a) Catapault Launched and (b) VTOL [8,9]

As a preliminary approach towards this research, for each UAV subsystem (i.e. Energy, Localisation, Actuation, Vision and Communication) there will be a corresponding Fault Diagnostic and Prognostic System. This architecture is shown below in Figure 2.

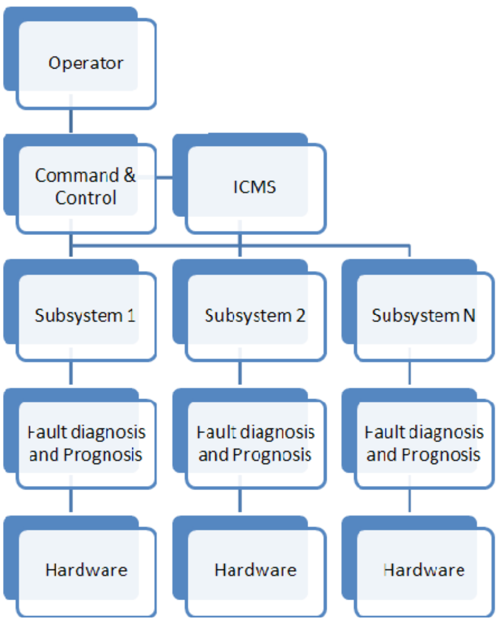


Fig. 2 A Preliminary Design of the General System Architecture [10]

III. Simulation and Testing

A. Introduction

The Intelligent Condition Monitoring System and Command and Control (C&C) are two separate entities that work alongside one another. Data is processed by the ICMS and communicated to the C&C whilst maintaining the human operator throughout the entire procedure. Global Remedies, for Global Faults that have a Major effect on the mission, are administered by Command and Control [10].

The subsystems of the UAV that were chosen for this routine were Localisation, Actuation and Battery Logistics. Analysis of these three subsystems covers the majority of the aircraft. Furthermore, a single representative component was selected for each subsystem for simulation.

These were an accelerometer (ADXL202EB), AUV thruster (engine) and a twelve Volt battery for the Localisation, Actuation and Energy Supply subsystems respectively. It should be noted that both the accelerometer and thruster components were used on an AUV (University of Adelaide Project #997), but for the demonstrative purposes of this model, it only mattered that there were some devices with easily found normal operating attributes. It should be further noted that as this was the initial stage of a long-term project, in most cases, only one means of failure was analysed, and further means would be investigated after verification of the simple model.

The accelerometer voltage was measured to determine if the component was working correctly or not. The normal operating conditions are 3-5.25 Volts. This would be considered a 'good' operating condition i.e. the device is working correctly. Any voltage outside this range would be considered 'bad' (not working correctly).

The engine would have both its voltage and current measured to determine correct working conditions. This is because the required power for hover (in ideal conditions) is 18 Watts, which corresponds to 1.5 Amperes of current and 12 Volts of voltage. The normal operating ranges of the current and voltage under these conditions were not found, so it was assumed for the sake of the simulation that the normal operating ranges were 1.25-1.75 Amperes and 11-13 Volts. It should be noted that once proper engine data is found it would be instituted into the model.

The Battery Logistics system would monitor a 12 Volt battery and compare its output voltage to the required voltage of a black box subsystem. The black box subsystem could potentially be any of the UAV's subsystems; for the brevity of this simulation it is not required that the subsystem is known. The battery subsystem will be either classified as red, yellow or green (this will be explained further in the Fuzzy Logic Systems section).

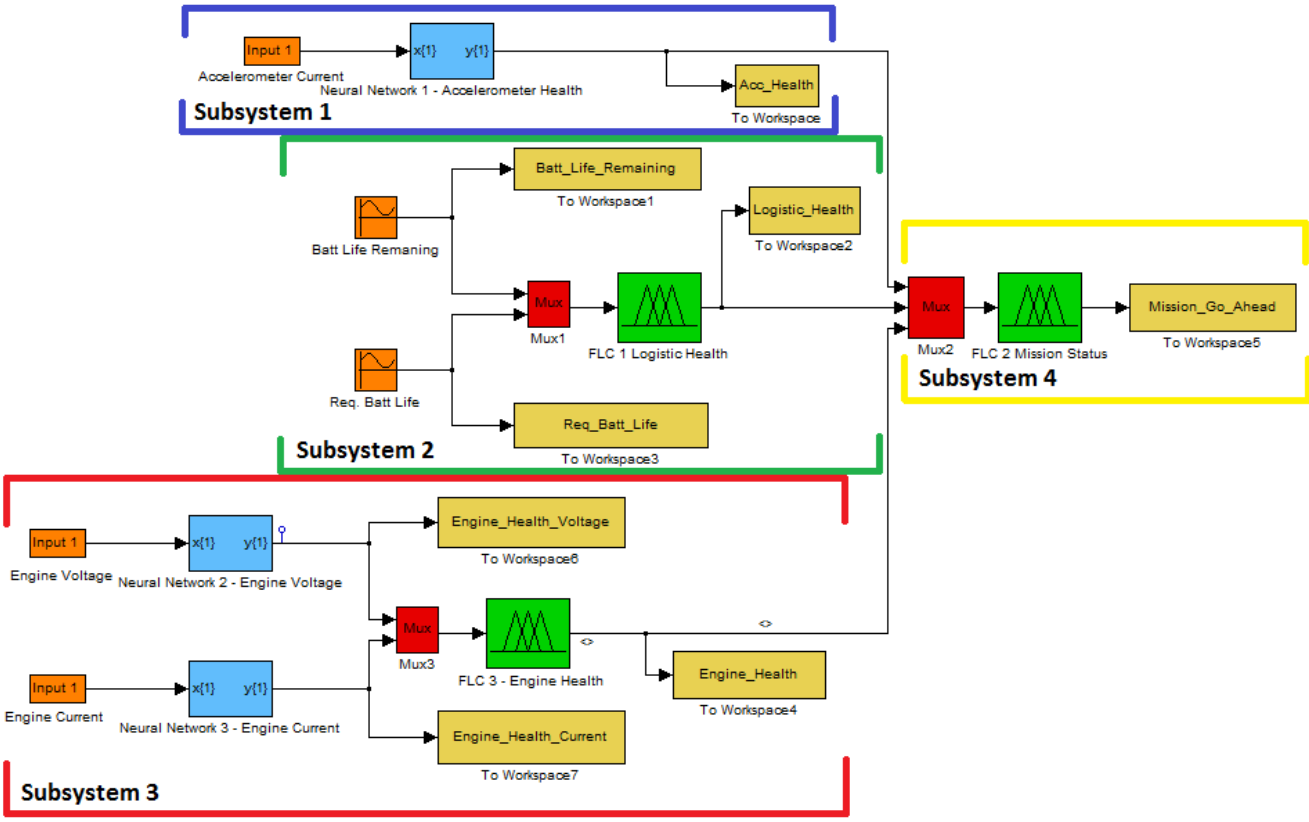


Fig. 3 Simulation Model

The simulation was constructed and tested in MATLAB (R2007B) and Simulink, as this program allows quick and easy construction of Fuzzy Logic systems, networks and many ways to portray test data for user friendliness. A View of the Simulink Model constructed can be seen in Figure 3.

B. Neural Networks

The Neural Networks used in this simulation are all cascade forward back-propagation Neural Networks. These use a Tan-Sigmoid transfer function to classify data based on network biases and weights. There are two variables that affect the accuracy of the classification of variables by the network, the number of hidden neurons, and the number of data points (samples). The number of hidden neurons affects the way the network classifies the data. When the number of hidden neurons is higher than optimum, the network tends to overcomplicate relations between variables, resulting in a loss of accuracy. Similarly if the number of hidden neurons is below the optimum number, the network simplifies the relations between variables, resulting in a loss of accuracy [11].

To determine the best number a MATLAB M-file was produced that created many networks (based on Neural Network 1) with different combinations of hidden neurons and sample sizes.

Figure 4 shows the MSE (mean squared error) verses the number of hidden neurons when the sample inputs are randomly chosen each time the test is run. It can be seen that the general trend is that as the number of hidden neurons increases, the mean squared error increases (especially in the 300 – 500 hidden neuron range). This is seen accross all of the Neural Networks. The crests and troughs that appear apparently randomly can be attributed to the randomly chosen inputs and the way in which each Neural Network learns. Figure 5 is the same test ran again but with a constant input of data. This will allow alleviation of concern over Figure 4 being a product of the random samples used. As can be seen the same trend of proportionality exists between MSE and number of hidden neurons. The random spikes and troughs can be attributed again to the way in which the networks learn. This would be reduced if the sample size was increased (it was 1000 for this test) to 10000 or more. We can further conclude from the figures that an optimum number of neurons would be less than 100 for each Neural Network. For the purposes of the simulation, 30 hidden neurons was decided upon.

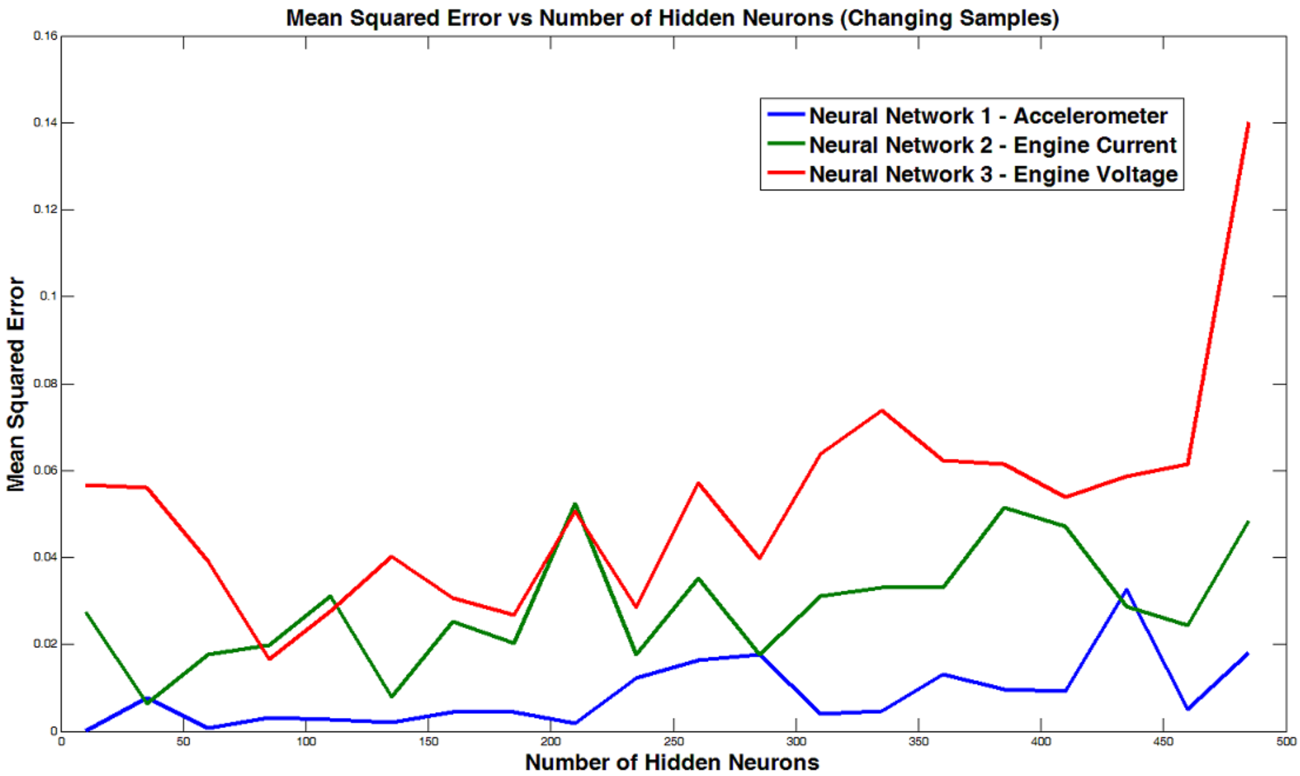


Fig. 4 MSE versus Number of Hidden Neurons for Changing Samples

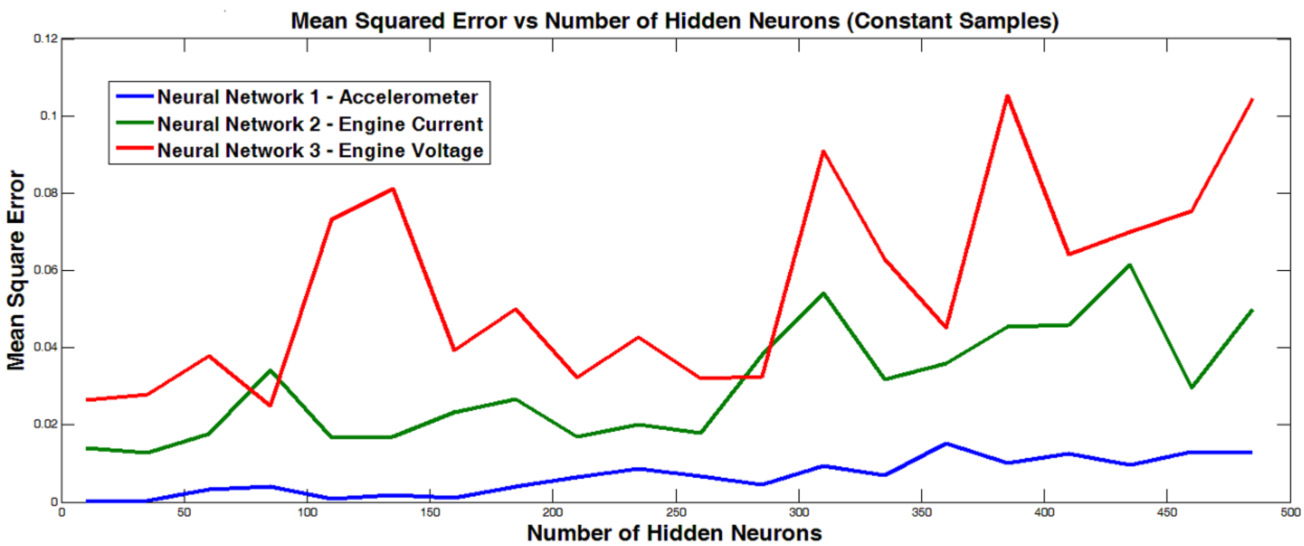


Fig. 5 Mean Square Error versus Number of Hidden Neurons for Constant Samples

C. Fuzzy Logic Controllers

There are three FLC's in the model. These are all Mamdani type systems. It was chosen over a Sugeno type system because of its intuitiveness, and ease of human input into the controller. A Sugeno type system was not chosen as it is only really advantageous when the FLC is very complex, and reduced computational load is needed (this is not the case in this simulation) [12].

Each of the Fuzzy Logic controllers take data from sensors/engines/batteries and convert the data into three outputs, red (bad), yellow (caution) or green (good). This allows quick response from a pilot or observer of the UAV when a red signal is produced, implying a critical situation.

D. Subsystem One (Accelerometer)

This subsystem can be seen in Figure 3 above and consists of a Neural Network only. This network has been trained to output a value of 1 if the input voltage is within acceptable operating conditions [3, 5.25] (Volts) and gives an output of 0 if outside the proper working conditions.

Using a MATLAB M-file, the network created was simulated 30 times with new data. The results are shown in Figure 6. The sample size was made to be 10000, and voltage data (produced using a random number generator) was fed into the system. The results show five and ten percent interval error rates. The ten percent errors correspond to the number of classification mistakes made by the network (i.e. if the output was supposed to be one, an acceptable answer may lie between 0.9 and 1.1); a similar approach was made for the five percent errors. On average, the network made 25 errors per 10000 inputs at the 10% interval, and 50/10000 at the 5% interval. This is an accuracy of 99.75% for a 10% interval and 99.50% accuracy at a 5% interval. It can be seen from Figure 6 that this is not constant, and depends on how the network responds to each input.

The data from the Neural Network is then fed into the Mission Status subsystem which decides whether or not the mission is able to be accomplished successfully.

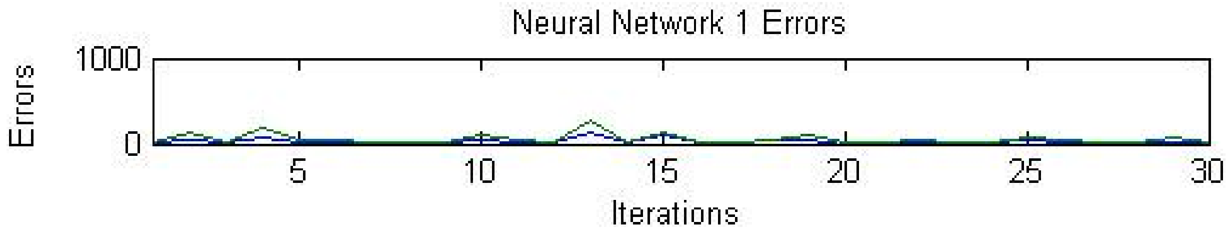


Fig. 6 Neural Network One Error

E. Subsystem Two (Battery Health)

This subsystem is concerned with the availability of power to a subsystem (black box subsystem). It takes two inputs, required battery voltage and available battery voltage and decides if the batteries are able to supply adequate power to the subsystem. For the simple simulation a FLC is capable of doing quite well. The FLC takes the two inputs and compares these values against a rule set (see Table I).

TABLE I
FLC ONE BATTERY HEALTH RULE SET

Rule 1	**IF:** Available Voltage is **Low** **AND** Req. Voltage is **Low** **THEN:** Logistic Health is **Red**
Rule 2	**IF:** Available Voltage is **Med** **AND** Req. Voltage is **Low** **THEN:** Logistic Health is **Yellow**
Rule 3	**IF:** Available Voltage is **High** **AND** Req. Voltage is **Low** **THEN:** Logistic Health is **Green**
Rule 4	**IF:** Available Voltage is **Low** **AND** Req. Voltage is **Med** **THEN:** Logistic Health is **Red**
Rule 5	**IF:** Available Voltage is **Med** **AND** Req. Voltage is **Med** **THEN:** Logistic Health is **Red**
Rule 6	**IF:** Available Voltage is **High** **AND** Req. Voltage is **Med** **THEN:** Logistic Health is **Yellow**
Rule 7	**IF:** Available Voltage is **Low** **AND** Req. Voltage is **High** **THEN:** Logistic Health is **Red**
Rule 8	**IF:** Available Voltage is **Med** **AND** Req. Voltage is **High** **THEN:** Logistic Health is **Red**
Rule 9	**IF:** Available Voltage is **High** **AND** Req. Voltage is **High** **THEN:** Logistic Health is **Red**

The highlighted rule (number 3) is a condition that responds with a green output, which is always desirable. The output is then defuzzified (reverted to a crisp variable) via the centroid rule. This crisp value is then an input into the mission status subsystem.

F. Subsystem Three (Engine Health)

The engine health in this simulation will be dictated by the voltage and current being drawn from the power supply. As stated previously, the current will have a defined (arbitrarily) normal operating range of [1.25, 1.75] (Amperes) and the voltage will have a normal operating range of [11, 12] (Volts). A Neural Network will classify the inputs (current and voltage) and discern whether or not the system is in a normal operating range. These values will then be fed into a Fuzzy Logic system, which will classify the engine health (red, yellow or green). Using MATLAB, thirty Neural Networks were created to identify an average accuracy for each Neural Network system (current and voltage). Below, in Figure 7, are two graphs showing the performance of both systems. The Neural Network concerned with classifying current made (on average) 125 errors per 10000 inputs at a 10% level (a correct classification is within the range [0.9, 1.1]) and 290 errors per 10000 at a 5% level (correct classification [0.95, 1.05]). These are errors of 1.25% and 2.90% respectively. Here we can see that the lack of optimisation of the Neural Network has led to errors significantly larger than in Neural Network 1 (0.25% and 0.5% error rates versus 1.25% and 2.90%). It would be beneficial in further research to optimise all Neural Networks.

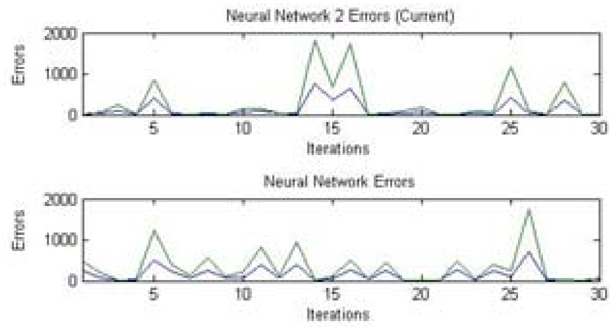


Fig. 7 Neural Networks Two (Current) and Three (Voltage) Errors

The Neural Network concerned with voltage classification had an average error rate of 150 per 10000 inputs at a 10% interval and an average error rate of 300 per 10000 inputs at a 5% interval (1.5% and 3% respectively). The spikes in errors seen in each graph can be attributed to situations where overfitting (loss of generality) occurs.

The Fuzzy Logic system for engine health has the input and output membership functions shown in Figure 8. Each input is categorized into three groups, **Bad**, **Unclassified** or **Good** and then the output is categorized into **Red**, **Yellow** and **Green**. Trapezoidal membership functions were implemented for all of the membership functions.

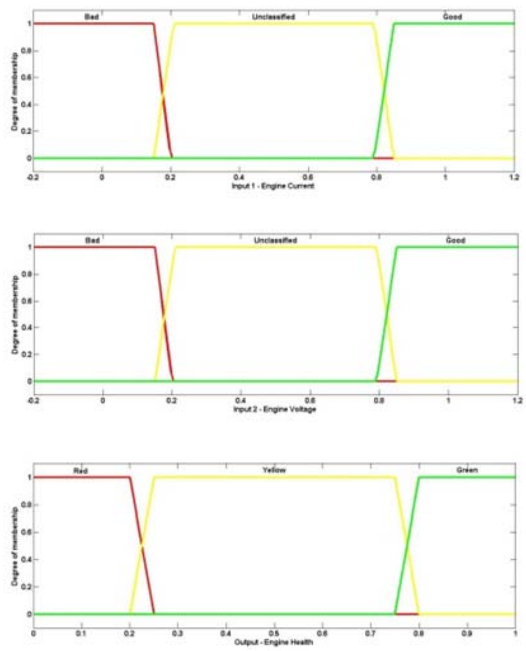
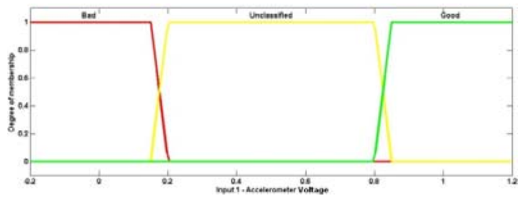
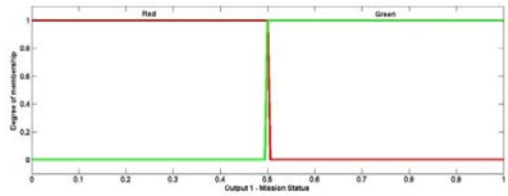


Fig. 8 Subsystem Four Mission Health Membership Functions



(a)



(b)

Fig. 9 Mission Status Membership Functions

The fourth and final subsystem, as seen in Figure 3 above, is an overall classification of the mission's health. This allows quick and easy identification by an operator on the current health of the system in relation to mission specifics (i.e. if a camera on the UAV is working for a surveillance specific mission). It comprises a Fuzzy Logic controller only. It does not need Neural Networks to classify initial states. The inputs (corresponding to the 3 former subsystems) determine the final health status of the system. The input membership functions for Battery Logistic Health and Engine Health are the same as the outputs for Subsystem 2 and 3 respectfully. The membership functions for the accelerometer health and output mission health can be seen in Figure 9.

IV. RESULTS

A. Testing

The model system will be tested with six situations (and expected results);

Test 1 – All systems working correctly (Green)

Test 2 – Accelerometer Voltage failure, all other systems working correctly (Red)

Test 3 – Engine Current Failure, all other systems working correctly (Red)

Test 4 – Engine Voltage Failure, all other systems working correctly (Red)

Test 5 – Battery Voltage/ Required Battery Voltage failure, all other systems working correctly (Red)

The results of this testing can be seen in Table II.

TABLE II
TEST SET UP AND RESULTS

	Test 1	Test 2	Test 3	Test 4	Test 5
Accelerometer Voltage	4 V	2 V	4 V	4 V	4 V
Battery Voltage	12 V	12 V	12 V	12 V	9 V
Battery Voltage Required	2 V	2 V	2 V	2 V	12 V
Engine Current	1.5 A	1.5 A	1 A	1.5 A	1.5 A
Engine Voltage	12 V	12 V	12 V	10 V	12 V
Mission Status (Expected Output)	**Green** (1)	**Red** (0)	**Red** (0)	**Red** (0)	**Red** (0)
Mission Status (Actual Output)	**Green** (0.75)	**Red** (0.25)	**Red** (0.25)	**Red** (0.25)	**Red** (0.25)

V. ANALYSIS

As can be seen in Table II, the actual linguistic output (Green or Red) is what was expected from the system, however, the actual crisp value is slightly off (i.e. in test 1 the predicted output is 1, but the actual output is 0.75). This is due to the centroid defuzzication rule implemented by the FLC. This problem would be reduced if a third membership function (yellow) was brought into the output or removed if the defuzzification rule was changed. But in terms of linguistic output the system works as expected.

VI. CONCLUSION

The results in Table II show the effectiveness of the model thus far. The model could be improved upon by increasing the accuracy of the Neural Networks. This would be done by further research into each individual Neural Network, testing combinations of the number of hidden neurons, the number of samples and the length of the training epochs. The FLC's could be improved by bringing expert knowledge into the production of the membership functions for the Fuzzy Logic controllers. This would allow a more powerful and optimum solution to be found for each specific case. Finally the model could be well improved upon by finding data from the UAV and implementing these into the design.

This simulation successfully justifies the use of Neural Networks in conjunction with Fuzzy Logic in the preliminary case put forward in this report.

ACKNOWLEDGMENT

The Authors would like to acknowledge the Defence Science and Technology Organisation (DSTO) for in-kind support in this project.

REFERENCES

[1] An, Y 1998, 'A Design of fault tolerant flight systems for sensor and actuator failures using online learning Neural Networks', Ph.D Thesis, Department of Mech. And Aerospace Eng. Morgantown, West Virginia.

[2] Napolitano, M, An, Y & Seanor, B 2000, *Fault Tolerant Flight Control System for Sensor and Actuator Failures Using Neural Networks*, Aircraft Design 3, pp. 103-128.

[3] Vanek, B 2009, *Future Trends in UAS Avionics*, 10th International Symposium of Hungarian Researchers on Computational Intelligence and Infomatics, pp. 791-802.

[4] Samy, I, Postlethwaite, I & Gu, D-W 2009, *Sensor Fault Detection and Accomodation Using Neural Networks with Application to a Non-Linear Unmanned Air Vehicle Model*, Journal of Aerospace Engineering, Vol. 224, pp. 437-447.

[5] Lee, C 1990, *Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part I*, IEEE Transactions on Systems, Man. And Cybernetics, Vol. 20, No. 2, pp. 404-418.

[6] Kasabov, N 1996, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, The MIT Press, Cambridge, Massachusetts.

[7] Pawar, P & Ganguli, R, *Helicopter Health Monitoring – a Review*, Journal of Aerospace Engineering, Vol. 221, pp.631-647.

[8] Li D, Wang, L & Zhu X, Design and Development of a new-version of Automated Systems Operations of Catapult-Launched Maritime UAV Robot, Final Report, The University of Adelaide, 2011.

[9] Xuning Z, Design, Build and Implementation of a new version of VTOL Maritime UAV Robot System, Final Report, The University of Adelaide, 2011.

[10] Anvar A.P. et al, Intelligent Condition Monitoring Systems for an AUV Robot, The Journal of Applied Mechanics and Materials Vols. 152-154 (2012) pp 1159-1164.

[11] Demuth H, Beale M & Hagan M 2007, *Neural Network Toolbox 5 User's Guide*, The MathWorks, Inc., Natick, Massachusetts.

[12] Author Uncited, 2007, *Fuzzy Logic Toolbox 2 User's Guide*, The MathWorks, Inc., Natick, Massachusetts.