Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Neural Network Based Sensor Validation Scheme Demonstrated on an Unmanned Air Vehicle (UAV) Model

Ihab Samy, Ian Postlethwaite and Dawei Gu

Abstract— **Nowadays model-based fault detection and isolation (FDI) systems have become a crucial step towards autonomy in aerospace engineering. However few publications consider FDI applications to unmanned air vehicles (UAV) where full-autonomy is obligatory. In this paper we demonstrate a sensor fault detection and accommodation (SFDA) system, which makes use of analytical redundancy between flight parameters, on a UAV model. A Radial-Basis Function (RBF) neural network (NN) trained online with Extended Minimum Resource Allocating Network (EMRAN) algorithms is chosen for modelling purposes due to its ability to adapt well to nonlinear environments while maintaining high computational speeds. Furthermore, in an attempt to reduce false alarms (FA) and missed faults (MF) in current SFDA systems, we introduce a novel residual generator. After 47 minutes (CPU running time) of NN offline training, the SFDA scheme is able to detect additive and constant bias sensor faults with zero FA and MF. It also shows good global approximation capabilities, essential for fault accommodation, with an average pitch gyro estimation error of 0.0075 rad/s.**

I. Introduction

FAULT tolerant flight control systems (FTFCS) can be found in many air-vehicles nowadays. Their purpose is to detect, identify, and accommodate for any type of failure that may occur on-board the vehicle. In fact, studies on the causalities by the US Air Force during the Vietnam War have revealed that upto 70% of aircraft losses could have been avoided if FTFCS were properly designed and implemented [1]. The two recognized classes of critical faults are sensor and actuator failures. In this paper we focus on sensor failures in an unmanned air vehicle (UAV).

Fault detection techniques have been applied to large manned aircrafts [1]-[5], underwater vehicles [6], and autonomous helicopters [7], while few have been extended to fixed wing UAVs. Nowadays, UAVs are used in applications deemed dangerous or unreachable by manned vehicles which can lead to high risks of failure. Furthermore, current trends in UAV design have shown that cheap and low-weight UAVs are more likely to be accepted by the civil aviation industry [8]. As a result, these factors become essential guidelines for control engineers designing any UAV FTFCS.

Reliable sensor fault detection and accommodation (SFDA) becomes even more critical when faulty measurements are used in control feedback loops. Traditionally, fault detection (FD) and fault accommodation (FAc) are accomplished via physical redundancy where identical sensors are used to measure the same parameter, and based on a voting-scheme, FD and FAc can be employed [9]. However, it is apparent that this method has serious weight, power and cost implications especially for UAVs. For this reason, over the past two decades analytical redundancy has become a more appealing approach for SFDA where a residual, which at its simplest form computes the difference between the model estimate and sensor measurement, is generated and a fault is then declared when it exceeds a predefined threshold.

In the event of fault detection, FAc is implemented by simply replacing the faulty sensor with the model-estimate. However in the event of multiple sensor faults, model-estimates can be inaccurate and so real-world applications tend to maintain a slight degree of physical redundancy.

Over the past two decades many SFDA publications have targeted fixed model-based approaches, with parameter estimation and observer-based methods being the most popular [10]. While proving to be successful they generally are limited to linear time-invariant systems. Novel approaches include nonlinear online adaptive schemes where the model is continuously tuned to fit the time-varying system. This is why the use of neural networks (NN) with online learning capabilities is steadily growing in the fault detection field [10].

Chow and Willsky (1984) first defined model-based FDI to consist of two main stages; residual generation and residual evaluation [11]. Patton *et al.* (1989) also outlined the criteria for selecting a suitable FDI approach, two of which were low false alarm rates and fewer missed faults [12]. Ideally a residual is nonzero only when a fault is present. However in real applications, the residual will always be nonzero due to unknown inputs (e.g. measurement noise and disturbances etc.). This can increase the risk of false alarms especially if simple threshold logic is implemented for residual evaluation. Of course one can raise the threshold but this can also increase the number of missed faults. Ways to improve model-based FDI robustness to unknown inputs is a widely studied topic. Examples include adaptive thresholds, originally proposed in Emami-Naeini *et al.* (1988), or the application of alternative residual evaluation techniques which do not rely on simple threshold

I. Samy is a PhD student at the Engineering Department, Leicester University, UK (phone: +44 0116 252 2522; e-mail: isar1@leicester.ac.uk).

I. Postlethwaite is the Head of the Control and Instrumentation Group, Leicester University, UK (e-mail: ixp@leicester.ac.uk).

D. Gu is a Professor at the Control and Instrumentation Group, Leicester University, UK (e-mail: dag@leicester.ac.uk).

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

logic, e.g. statistical tests on the Kalman filter innovation sequence [13]-[14]. Another famous approach is the unknown input observer which attempts to decouple the effects of unknown inputs on the residual [15]. Alternatively, we suggest a novel residual processing approach which will be referred to as residual padding.

In an attempt to widen the scope of NN-based SFDA schemes we design and apply such a scheme to a nonlinear UAV model. The NN structure chosen is based on the Extended-Minimum Resource Allocating Network (EMRAN) Radial Basis Function (RBF), due to its good generalization ability and fast performance [16]. Comparisons of a conventional residual generator approach and our novel approach are carried out under different levels of sensor noise and fault classes in order to test their robustness and sensitivity respectively.

II. UAV Model

The UAV is based on the Eclipse class vehicle with conventional control surfaces (ailerons, elevators and a rudder) and is powered by a small gas turbine engine. Some of the flight characteristics are shown in Table 1. The model used to describe the UAV is an open-loop nonlinear 6DOF model. Aircraft dynamics are modeled as a set of twelve first order decoupled nonlinear differential equations with twelve states; linear velocities, angular rates, attitude, and position vectors. The input data consisted of preset manoeuvres in the pitch axis with the elevator demand following a 3-2-1-1 input sequence. The model features both system and measurement noise where the system noise is modelled as zero mean, white, Gaussian gust disturbances on the angle of attack and sideslip with a 0.1 deg standard deviation. The sensor/measurement noise is also assumed to be Gaussian and white.

Table 1. UAV characteristics

Characteristics	Value
Wing Span	2.20 m
Fin Span	0.43 m
Mass	39.7 kg
Ground Velocity (max.)	44 m/s
Flight Altitude (max.)	150 m
Flight Time (max.)	30 mins

III. NN strucure

A. NN Inputs

Longitudinal motion of the UAV model is considered and sensor faults are introduced only in the pitch gyro. The NN input set must therefore include the longitudinal flight parameters of the aircraft. These include angle of attack (α), normal-acceleration (a_z), elevator demand (δ_e), throttle (τ), airspeed (V_t) and altitude (H). The analytical redundancy between these measurements and the pitch rate is ensured by the aircraft equations of motion [17]. The input a_z can be replaced by $a_z/\ V_t^{\ 2}$ to smoothen the nonlinear estimations caused by the nonlinear aircraft dynamics [2].

B. Network Size Reduction

Including all the longitudinal flight parameters in the NN input set ensures accurate pitch rate (q) estimations but can also result in a large network size. To avoid this we carried out a series of test runs to find out which flight parameters have the least effect on the NN output estimation error. The resulting NN can be described by the following input/output relation:

$$q_i = \mathrm{NN}\big[\ \alpha_i, \delta_{ei}, V_{ti}, a_{zi}/V_{ti}^{\ 2}\ \big] \tag{1}$$

where NN is the NN model and i is the sampling instant. Note also that pitch rate is not included in the NN input set to avoid contaminating the NN structure when a fault is present.

C. RBF NN Structure and Training

The NN structure chosen is the EMRAN RBF proposed in [16]. It consists of a group of input signals, a hidden layer and an output layer (Fig 1). It starts with zero hidden units (neurons) and only adds hidden units if *all* of the following three criteria are met (for a single output NN):

$$e_i = |\, y_i - ys_i \,| \qquad > E1 \tag{2}$$

$$e_{iRMS} = \sqrt{\sum_{j=i-(M-1)}^{i} \frac{e_j^{\ 2}}{M}} \qquad > E2 \tag{3}$$

$$d_i = \|\, \mathbf{x}_i - \mu_{ir} \,\| \qquad > E3 \tag{4}$$

where y_i and ys_i are the real sensor output and NN estimation at sample instant i respectively, and μ_{ir} is the centre of the hidden unit *closest* to the current NN input vector $\mathbf{x}_i$. *E1* ensures that the NN estimation error is below a set threshold (i.e. good enough), *E2* checks if the root mean square (RMS) of the past M errors is low enough and *E3* checks if the minimum distance between the current input vector and the centres of the hidden units is significantly small.

If less than three of the criteria (2)-(4) are met, then the NN training algorithm updates (i.e. tunes) the free parameters (centres, widths and weights) of *only* the most active (so called 'winner') neuron. This reduces the number of parameters to be updated which speeds up the training process. An additional reason why the EMRAN RBF NN is superior to many other NN designs is that it can automatically prune hidden units which contribute the least to the NN estimates [16].

The training algorithm used to develop the NN is based on the popular gradient descent algorithm and is described as follows [18]:

$$\mathbf{\Theta}_{\mathbf{i+1}} = \mathbf{\Theta}_{\mathbf{i}} - \eta \left.\frac{\partial ys_i}{\partial \mathbf{\Theta}_{\mathbf{i}}}\right|_i e_i - \eta\sigma\mathbf{\Theta}_{\mathbf{i}} \qquad (5)$$

where $\mathbf{\Theta}$ is the vector of NN free parameters and η is the NN learning rate. The differential in (5) is the gradient of the estimated output *ys* with respect to the free parameters, *e* is as in (2) but without the absolute operator, and σ is a positive constant known as the stabilizing factor. The second term on the right hand side of (5) is simply the delta rule while the third term is designed to counteract casual parameter drifting by slowly driving the free parameters towards zero.

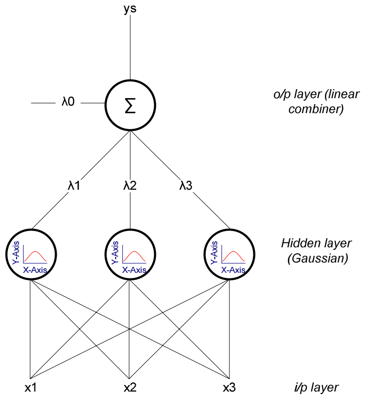

Fig. 1 RBF-NN structure example. **x**, λ, *ys* are inputs, weights and output respectively.

IV. Residual Generation

A. Conventional Residual Generator Approach

In their simplest form, residuals are generally functions of the squared difference between model estimate *ys* and real sensor output *y* [7]. Averaging is generally applied to reduce the effects of noise, and the residual *r* can be described as the weighted (moving) average of the past *M* measurements:

$$r_i = w \sum_{j=i-(M-1)}^{i} \frac{(y_j - ys_j)^2}{M} \qquad (6)$$

where *w* is the weight deduced from experiments and *i* is the current measurement. Ideally residuals are only nonzero when a fault is introduced. In this case selecting a threshold is simple; a low threshold would be suitable. However, the presence of residual *noise* (caused by measurement disturbances, process disturbances and model uncertainties) also causes the residuals to become nonzero which can lead to false alarms. In current health management systems which employ SFDA schemes there is an average false alarm rate of 1 per 100 flight hours [19].

In low-weight UAV applications, the vehicle can be more susceptible to sudden deviations from its set flight envelope caused by random disturbances. This can result in high amounts of residual noise and consequently high false alarm rates. Therefore a more robust residual generation approach is needed. Here we demonstrate a novel residual *processing* technique which we will refer to as residual ‘padding’.

B. Residual Padding

We can think of the residual as *n* sets (*S1, S2…Sn*) with *D* data points in each set. With padding, we manage to extend each of the *n* sets with artificial data. Each data point is the squared difference between model estimate *ys* and real sensor output *y*. Padding maps a length *D* set to a length $D_{pad}>D$ set, where D_{pad}-*D* artificial data are added. Each artificial point added is equal to the *minimum* magnitude (min {*Sn*}) in the nth set. The aim is to reduce the overall average of each residual set so as to minimize the effect of residual noise especially ones causing large spikes. ‘min {*Sn*}’ is chosen instead of ‘zeros’ because the latter will also damp the genuine fault peaks while ‘min{*Sn*}’ is more specific of the individual residual sets. In conclusion, the first step of the novel residual generator is padding and only then is (6) implemented.

Consider this simple example where the following assumptions are made (no units are used):

1) Residual threshold is of magnitude 5.
2) Typical residual set is {0 0 0 0}.
3) Faults result in a continuous residual magnitude of 6.
4) Residual noise is large and sudden (i.e. resembling spikes) and results in a residual magnitude of 24.
5) 2 padding points can be added to the residual set.
6) Padding points have a magnitude equivalent to the minimum magnitude in the residual set.

Let us consider three scenarios; scenario 1 is when faults and noise are not present, scenario 2 is when a fault occurs and scenario 3 is when residual noise is present. In the event of scenario 1 the residual set is {0 0 0 0} with an average of zero. Therefore the threshold is not exceeded as desired. In the event of scenario 2, the residual set becomes {6 6 6 6} and so the padding points would be of magnitude 6. The modified residual becomes {6 6 6 6 *6 6*} with an average of 6 and so the threshold is exceeded and a fault is declared. In the event of scenario 3, the residual set is {24 0 0 0}. If padding is not applied and the simple average of this residual is taken, the average would be 6 and so the threshold is exceeded resulting in an undesired fault alarm, i.e. false alarm. However if instead we pad the residual set using the same technique as scenario 2, the modified residual set becomes {24 0 0 0 *0 0*}. The average is now 4 and so the threshold is not exceeded.

Obviously this is a simple example with no consideration to issues such as drifting faults or adjacent residual noise patterns; however the underlying concepts are the same. Residual padding can be seen as a way of damping only the residual noise and not the fault peaks.

It is generally understood that small and slow-drift faults (referred to as incipient faults) are hard to detect [1]–[4], [7]. The reason for this is that with residual noise being present, thresholds are generally set at a high level to avoid false alarms. Consequently, small magnitude faults do not exceed this threshold and pass by undetected. A common solution is to amplify the residual but this can also have the adverse effect of amplifying residual noise. However with our padding approach, amplification should be a feasible solution as residual noise is consistently damped. It is also crucial that excessive amounts of padding are avoided as this can increase the fault detection time or even completely damp the fault peaks. This is demonstrated and discussed in section VIII.

V. Fault Scenarios

The pitch gyro sensor failures considered here are:

1) *Constant bias faults:* The sensor output gets stuck and outputs a constant bias.

2) *Additive faults:* Additive faults are very common. A term is added to the normal sensor value as a result of temperature changes or calibration problems.

Additive faults can be described by ramp functions (drift) and can be of soft or hard nature depending on the duration of the ramp T_R. Typically T_R is 1s for hard additive faults, and 4 seconds for soft additive faults [1].

VI. SFDA Outline

A. Offline Training

Prior to any SFDA tests, the NN structure must be initialised. This is done through offline training where one data set from the UAV model is used to build the NN structure. However one must be careful not to overtrain the NN or else it will perform poorly when exposed to novel data. To avoid overtraining the NN, two sets of flight data of 225s each were collected from the UAV model:

1) Train set: This is used to train the NN
2) Test set: This is used to query the NN, i.e. with learning switched off.

Offline training continues until the point where the NN RMS estimation error from the test set reaches a minimum (Fig 2).

B. SFDA Application to the UAV Model

Once the NN is initialised via offline training it can then be used in the SFDA tests. Three sets of flight data of 800s each were used for the SFDA tests, where each set had different sensor noise configurations. *Configuration 1* models zero mean, white Gaussian sensor noise. C*onfiguration 2* and *configuration 3* include the same sensor noise model but with twice and ten times greater noise standard deviations respectively. This allowed us to test the robustness of the residual 'padding' approach, mentioned earlier, to the different sensor noise levels. In all cases, a NN learning rate of 0.0007 was found most suitable and the data was filtered using a 1st order Butterworth low pass filter, before being used in the NN. The fault classes tested include; constant bias, hard additive and soft additive faults. The fault magnitudes included; 20% and 5% (percentage of sensor range; 30deg/s).

VII. Results

The mean detection time (MT) and number of undetected faults (UD) were recorded for each test. Additionally the *percentage* of time that the residual remains above the set threshold prior to a fault being introduced is recorded. This is an indication of the false alarm characteristics, and will be referred to as 'false alarm' (FA). A careful analysis of the results reveals the following:

1) Fig. 2 shows the NN RMS estimation error history during offline training where 1 epoch represents one pass through the entire data set. After 1981 epochs (47 mins CPU time) offline training was stopped to avoid overtraining the NN structure.
2) From Table 2 we notice that the conventional residual generation approach ('no padding') struggles to detect all faults, with 2 UDs. The two undetected faults were found to be the hard and soft additive faults of sensor range magnitude 5%. Furthermore regardless of the noise configurations implemented, false alarms remain present with an average FA of 0.11 %. On the other hand, residual padding has managed to remove false alarms completely as well as allowing the detection of all faults tested. However it has also caused an increase in MT by almost 55%.
3) Fig. 3 shows an example plot for each residual generation approach. The top plot is when the residual is generated with no padding (i.e. conventional approach) and the bottom plot is when residual padding is applied. The first thing we notice is the higher set threshold when padding is applied (0.006 rad^2/s^2) in comparison to the conventional approach (0.001 rad^2/s^2). This is because residual amplification is possible when padding is applied, as noise is successfully damped. Consequently small faults are easier to detect and false alarms are avoided. On the other hand we can see that false alarms are present in the top plot and therefore further residual amplification is not advisable.
4) From Table 3, we see a breakdown of the fault detection properties of our SFDA scheme using the novel residual generator approach. Small magnitude and slow-drift faults generally take longer to detect in comparison to constant bias faults.
5) From Fig. 4 and 5, a FAc plot can be seen. Note that NN

training was stopped when the fault was detected to avoid learning faulty data. Here we observe the good global approximation capability of the RBF-NN. Overall an average RMS pitch rate estimation error of 0.39 deg/s, 0.42 deg/s and 0.47 deg/s were achieved for sensor noise configurations 1, 2 and 3 respectively.

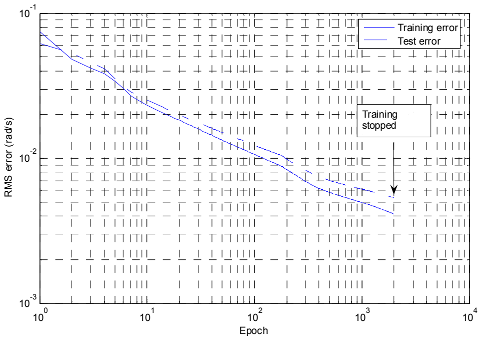

Fig. 2 Offline training error history.

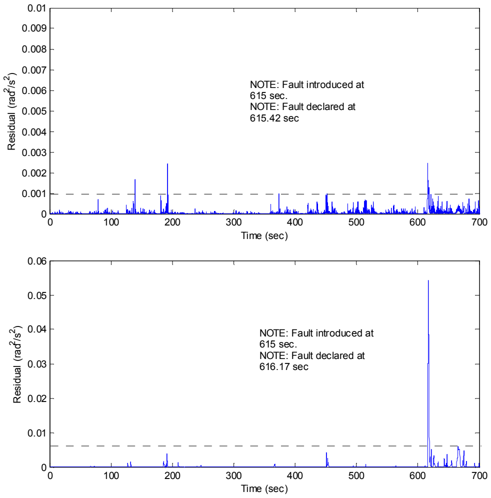

Fig. 3 Example residual plots. Top: No padding, Bottom: With padding. Fault type 20% hard additive; noise configuration 1.

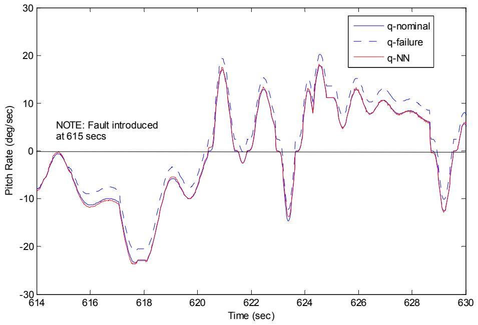

Fig. 4 Simulation of pitch gyro failure. Failure type 20% hard additive; noise configuration 1. Plot shows nominal pitch rate (no fault), failed pitch gyro, and RBF-NN estimation.

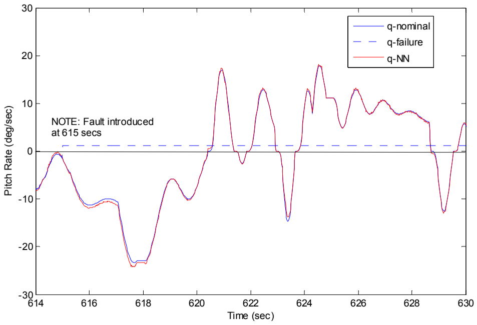

Fig. 5 Simulation of pitch gyro failure. Failure type 5% constant bias; noise configuration 1. Plot shows nominal pitch rate (no fault), failed pitch gyro, and RBF-NN estimation.

TABLE 2. SUMMARY OF FD RESULTS.

	No padding			With padding		
	MT (s)	UD	FA (%)	MT (s)	UD	FA (%)
Confg 1	0.72	2	0.10	1.30	0	0
Confg 2	0.77	2	0.09	1.32	0	0
Confg 3	0.79	2	0.14	1.53	0	0

TABLE 3. FD RESULTS (WITH PADDING). TIME IN SECONDS

	Hard Additive		Soft Additive		Constant Bias	
	20%	5%	20%	5%	20%	5%
Confg 1	1.17	1.42	1.52	1.70	1.00	1.00
Confg 2	1.20	1.42	1.58	1.71	1.00	1.00
Confg 3	1.36	1.58	2.06	2.18	1.00	1.00

VIII. SUMMARY OF RESULTS

Two residual generator approaches were compared. One considered the conventional approach which included residual averaging and weight tuning (6). The other was based on a novel 'padding' approach (refer to section IV). It was found that the conventional residual generator approach performed best for constant bias faults. It also performed well for hard and soft additive faults which had a sensor range magnitude greater than 5%. However, it failed to detect hard and soft additive faults which had a sensor range magnitude of less than or equal to 5%. This is because small faults resulted in small residuals which did not exceed the set threshold. Lowering the threshold or amplifying the residual as a solution would also result in the increase in false alarms (Fig 3; top plot). Instead, a novel residual generation approach was implemented with the aim of damping residual noise (caused by measurement disturbances, process disturbances and model uncertainties). As a result it was possible to amplify the residual and consequently detect small faults. Also notice from Fig. 3 that the fault detection peak is higher in the bottom plot, as a result of residual amplification. This is desirable as it provides a more robust fault detection scheme.

From Table 2 we also find that the novel approach manages to remove false alarms regardless of the amount of sensor noise present in the system, while the conventional approach struggles to do so. The main drawback of the padding approach was the increase in the fault detection time. In comparison to the conventional residual generator approach, padding caused an average increase of 55% in the fault detection time. However, with reference to other SFDA work, the detection times remain within acceptable limits [1]–[4], [7].

From the tests we found that false alarms were removed by padding each residual set with same number of residual points. So if we assume that for a residual set *S1* there are *D* data points, an artificial data set of size *D* would be added to the residual set *S1*, where each artificial data point added would have magnitude 'min{*S1*}'. The weighted average of the modified residual set $S1_{pad}$ can then be taken as in (6).

The ability to detect slow-drift faults is also dependant on the NN learning rate implemented. A high learning rate results in learning of slow-drift faults, whereas a low learning rate prevents the learning of new aircraft operating conditions. A trade-off is therefore needed, and in our application an offline and online learning rate of 0.04 and 0.0007 were found most suitable respectively.

In real-time implementation, it is important that the NN learning time (t_s) for one sample of data is lower than the sampling time (T). For our tests, the NN (maximum of 9 hidden neurons) was run on a 1.6GHz Pentium processor with T equal to 20ms. It was found that the maximum t_s was equal to 0.2 ms which is desirably lower than T.

As a general remark, it must be noted that the test conditions (e.g. thresholds, amount of padding, etc.) were heuristically selected and are not intended to give optimum SFDA performance. They are chosen to facilitate the analysis of the results. If the test conditions are to be changed, the results would naturally change, however the general conclusions drawn would be the same.

IX. Conclusion

We have managed to demonstrate a SFDA scheme on a nonlinear UAV model using an EMRAN-RBF NN in an attempt to widen the scope of NN-based SFDA schemes to fixed-wing UAVs. A novel residual generator approach was presented and shown to outperform the conventional method, with fewer false alarms and missed faults. The NN designed showed good global approximation characteristics with an average pitch rate estimation error of 0.43 deg/s. Further work must explore the SFDA properties under different levels of process noise, and simultaneous faults. The fault accommodating system must also be tested for closed loop stability when used in feedback control. Real flight data and different flight input sequences can help to generalize the results obtained here.

Acknowledgment

The work presented here is part of project FLAVIIR which is funded by British Aerospace (BAE) Systems. Support for the 1st author was given by the Overseas Research Scheme (ORS).

References

[1] Y. An, "A design of fault tolerant flight control systems for sensor and actuator failures using on-line learning neural networks," Ph.D. dissertation, Dept. of Mech. And Aerospace Eng., Morgantown, West Virginia, 1998.

[2] G. Campa, M.L. Fravolini, M. Napolitano and B. Seanor, "Neural networks sensor validation for the flight control system of a B777 research model," *American Control Conference,* vol. 1, pp. 412-417, 2002.

[3] M. Napolitano, Y. An, B. Seanor., S. Pispistos and D. Martinelli, "Application of a neural sensor validation scheme to actual Boeing B737 flight data," *Proc. Of the AIAA Guidance Navigation and Control Conference,* 1999.

[4] M. Napolitano, Y. An, B. Seanor, "A fault tolerant flight control system for sensor and actuator failures using neural networks," *Aircraft Design,* vol. 3, pp. 103-128, 2000.

[5] P. Motyka, W. Bonnice, S. Hall and E. Wagner, "The evaluation of failure detection and isolation algorithms for restructurable control." NASA Contractor Report 177983, 1985.

[6] A. Alessandri, M. Caccia and G. Veruggio, "Fault detection of actuator faults in unmanned underwater vehicle," *Control Engineering Practice,* vol. 7*,* pp. 357-368, 1999.

[7] G. Heredia, A. Ollero, R. Mahtani, V. Remub and M. Mausial, "Detection of Sensor Faults in Autonomous Helicopters," *Proc. Of the 2005 IEEE International Conference on Robotics and Automation*. Barcelona, Spain, 2005.

[8] P. La Franchi, "Grand designs: An EC-funded research project has unveiled its proposals for a new generation of aircraft that are intended to give Europe the edge in the civil UAV sector," *Flight International,* pp. 109-114, 2005.

[9] A. S. Willsky, "A survey of several failure detection methods," *Automatica,* vol. 12 no. 6, pp. 476-484, 1976.

[10] R. Isermann and P. Balle, "Trends in the applications of model-based fault detection and diagnosis of technical processes," *Control Engineering Practice,* vol. 5, pp. 709-719, 1997.

[11] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust detection systems," *IEEE Trans. Automat. Contr.*, vol. 29, pp. 603-614, 1984.

[12] R. J. Patton, P.M. Frank and R. N. Clark, "Fault diagnosis in dynamic systems, theory and application," London, Control Engineering Series, Prentice Hall, 1989.

[13] A. E. Emami-Naeini, M. M. Akhter, and S. M. Rock, "Effect of model uncertainty on failure detection: the threshold selector," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 1106-1115, 1998.

[14] R. K. Mehra and J. Peschon, "An innovations approach to fault detection and diagnosis in dynamic systems," *Automatica*, vol. 7, pp. 637-640, 1971.

[15] K. Watanabe and D. M. Himmelblau, "Instrument fault detection in systems with uncertainties," *Int. J. Sys. Sci.*, vol. 13, pp. 137-158, 1982.

[16] Y. Li, N. Sundararajan and P. Saratchandran, "Analysis of minimal radial basis function network algorithm for real time identification of nonlinear dynamic systems," *IEEE Contr. Theory and Application*, vol. 4, pp 476-484, 2000.

[17] M. V. Cook, "Flight dynamics principles," Great Britain: Arnold, pp. 55-79, 1997.

[18] M. L. Fravolini, G. Campa, M. Napolitano and Y. Song, "Minimal resource allocating networks for aircraft SFDIA," *Advanced Intelligent Mechatronics*, vol. 2, pp. 1251-1256, 2001.

[19] J. Zakrajsek, E. Huff, M. Augustin, R. Safa, A. Duke, P. Ephraim, P. Grabill, H. Decker, "Rotorcraft health management issues and challenges," *NASA ISHEM Forum,* Napa, CA, 2005.