〈Original Paper〉

A Design of a Fault Tolerant Control System Using On-Line Learning Neural Networks

온라인 학습 신경망 조직을 이용한 내고장성 제어계의 설계

Younghwan An*
안 영 환

(Received August 31, 1998 ; Accepted October 15, 1998)

Key Words : Fault Tolerant Control System (내고장성 제어계), On-Line Learning Neural Networks (온라인 학습신경망)

ABSTRACT

This paper describes the performance of a full-authority neural network-based fault tolerant system within a flight control system. This fault tolerant flight control system integrates sensor and actuator failure detection, identification, and accommodation (SFDIA and AFDIA). The first task is achieved by incorporating a main neural network (MNN) and a set of n decentralized neural networks (DNNs) to create a system for achieving fault tolerant capabilities for a system with n sensors assumed to be without physical redundancy. The second scheme implements the same main neural network integrated with three neural network controllers (NNCs). The function of NNCs is to regain equilibrium and to compensate for the pitching, rolling, and yawing moments induced by the failure. Particular emphasis is placed in this study toward achieving an efficient integration between SFDIA and AFDIA without degradation of performance in terms of false alarm rates and incorrect failure identification. The results of the simulation with different actuator and sensor failures are presented and discussed.

요 약

본 연구에서는 신경조직망을 이용한 항공제어계의 내고장성 성능에 대해 관점을 두었다. 이 내고장성 제어계는 감지기와 작동기의 고장 발견, 확인, 그리고 보완으로 이루어진다. SFDIA는 주 신경조직망과 n개의 국소 신경조직망으로 이루어지는데, 여분의 감지기없이 n개의 감지기로 내고장성 능력을 성취함을 목적으로 한다. 또한, AFDIA는 같은 주 신경조직망과 세개의 신경조직망 제어기들로 구성되며, 이 제어기들은 평형을 유지하는 역할을 하며 고장으로 인한 pitching, rolling, 그리고 yawing moment를 상쇄하는 기능을 한다. 본 연구에서는 특히 잘못된 경보와 고장 확인의 성능이 떨어짐이 없이 SFDIA와 AFDIA의 효과적인 통합 기능을 수행하는데 중점을 두었으며 여러가지 작동기와 감지기의 고장에 대한 연구 결과가 제시되었다.

* Members, West Virginia University

1. Introduction

The relatively low procurement of high performance military aircraft along with the cancellation of plans for building new aircraft has renewed interest in fault tolerant flight control systems with capabilities for accommodating sensor and actuator failures. For military purposes it is evident that such a feature can increase the chances of survivability following a control surface failure in a combat scenario. In fact, studies on the casualties by the US Air Force during the Vietnam war have revealed that up to 70 % of aircraft losses could have been avoided if reconfigurable flight control schemes were properly implemented in the flight control systems. For scientific applications, much emphasis has been placed on the design of fault tolerant flight control systems for low weight and unmanned aerial vehicles (UAVs) used for remote sensing purposes. In general, a full fault tolerant flight control system needs to perform:

(1) Sensor failure detection, identification, and accommodation (SFDIA)[1~4];

(2) Actuator failure detection, identification, and accommodation (AFDIA)[5~10];

Furthermore, the SFDIA task can be divided into:

- sensor failure detection, identification (SFDI), which monitors the degree of deterioration in the accuracy of the sensors.
- sensor failure accommodation (SFA), which replaces the faulty sensor with an appropriate estimation.

Similarly, the AFDIA task can be divided into:

- actuator failure detection and identification (AFDI), which detects significant abnormalities and searches for the cause or for a set of probable causes;
- actuator failure accommodation (AFA), which determines on-line what actions should be taken to recover the impaired aircraft.

Sensor failure detection and identification (SFDI) has been considered as an important issue, particularly when the measurements from a failed sensor are used in the feedback loop of a control law. Since the aircraft control laws use sensor feedback to establish the current dynamic state of the airplane, even slight sensor inaccuracies, if left undetected and unaccommodated for, can lead to closed-loop instability, at its worst leading to unrecoverable flight conditions.

For SFA purposes, most of today's high performance military aircraft as well as commercial jetliners implement a triple physical redundancy in their sensor capabilities. However, when reduced complexity, lower costs, and weight optimization are important factors in aircraft design, an analytical sensor redundancy approach is more appealing.

In terms of the AFDIA problem, an actuator failure may imply a locked surface, a missing part of the control surface, or a combination of both. Generally, an actuator failure may change trim conditions and induce a dynamic coupling due to the corruption of dynamic stability, which possibly leads to unrecoverable flight conditions. The objective of the AFDIA, if properly and successfully implemented, is to regain and maintain acceptable stability, trim conditions, and handling qualities following of an actuator failure. Therefore, the pilot can lower mission abort rates and even aircraft loss rates.

Theoretical FDI techniques, such as Generalized Likelihood Ratio and Multiple-Model Kalman Filtering, perform a continuous monitoring of the measurements from the sensors[11,12]. At nominal conditions, these measurements follow known patterns with uncertainties due to system and measurement noises. However, when sensor or actuator failures occur, the measurements deviate from the expected values computed on-line or off-line from state estimation schemes. The main problems associated with the application of these failure detection schemes are their suitability only for linear time invariant systems and their applicability only when the system model is identical to the filter or observer model and/or with high signal-to-noise ratio. As an alternative approach, the implementation of Neural Networks (NNs) to the SFDIA and AFDIA problems has been proposed and developed in recent years[3~7,10].

The previous decade has witnessed an increase in NN research, mostly induced by the introduction of the Back-Propagation Algorithm (BPA) for feed-forward NNs with supervised learning[13~15]. In recent years NNs have been proposed for identification and control of linear and non-linear dynamic systems[16~18].

An aircraft system in certain phases of its flight envelope can be considered a time-varying, non-linear system with system and measurement noises. Therefore, the control of such a system can be attempted with an adaptive scheme. For NN applicability to adaptive control systems, the following properties are important[3,16,17,19]:

· Learning and adaptation : NNs can be trained using past recorded or simulated data (off-line training) or current data (on-line learning).

· Applicability to non-linear systems : The applicability of NNs to non-linear systems originates from their demonstrated mapping capabilities.

· Application to multivariable systems : NNs are multi-input, multi-output (MIMO) entities and this, naturally, leads to their application to multivariable systems.

· Parallel distributed processing and hardware implementation : NNs have an inherent parallel architecture which, naturally, leads to high speed parallel hardware implementations.

It is clear that these properties of NNs are very appealing for the purpose of providing fault tolerance capabilities in a flight control system following sensor and/or actuator failures. A critical design choice in the use of NNs for both estimation and control purpose is on-line learning vs. off-line learning. Off-line learning implies that the NNs have a frozen numerical architecture and do not take advantage of the additional learning which can be provided by on-line data. With on-line learning neural networks two issues are of concern. These are the necessary amount of time required to achieve an acceptable learning level and the level of complexity of the NN architecture. Both these problems are related to the learning algorithm. Therefore, the performance and the acceptability of an on-line learning NNs depend on the performance of its training algorithm.

To date, the Standard Back-Propagation algorithm (SBPA), a gradient-based optimization method, has been widely used as a training algorithm for the NN architecture. However, there are some drawbacks associated with the SBPA in terms of learning speed and local minimums[15]. These problems may be solved by introducing a heterogeneous network, meaning that each neuron in the hidden and output layer of the NN has the capability of updating the output range (upper and lower bounds U, L) and the slope of the sigmoid activation function (T) as given by

$$f(x_{i,j},U_{i,j},L_{i,j},T_{i,j})=\frac{U_{i,j}-L_{i,j}}{1+e^{-x_{i,j}/T_{i,j}}}+L_{i,j} \quad (1)$$

where i, j are the indices for the generic neuron of the hidden and output layers and x is the same argument as in the SBPA sigmoid activation function. This learning algorithm has been named the Extended Back-Propagation algorithm (EBPA) and has demonstrated substantial performance improvements with respect to the SBPA in terms of accuracy and learning speed[20].

An important objective of this effort is to address the issues of the integration between the SFDIA and AFDIA schemes. Although both SFDIA and AFDIA problems have been extensively addressed and described in the technical literature in recent years, it was difficult to find a single reference describing the integration of the two schemes within flight control systems. This is mainly due to the fact that fault tolerance following actuator failure is a more important problem within the flight control community since physical redundancy in the sensory capabilities is typically available within flight control systems of both civil and military aircraft. However, as in the case of high altitude UAV performing scientific or military missions, an analytical sensor redundancy is preferred due to the reasons that the addition of physical redundant sensors may substantially increase the weight, the cost, and the complexity of the aircraft. Because of these facts, a reliable software-based integration between SFDIA and AFDIA schemes is a favorable approach to provide comprehensive sensor and/or actuator failure detection, identification, and accommodation capabilities against all probable sensor and/or actuator failure types.

The paper is organized as follows: the next sections review the SFDIA and AFDIA schemes followed by a section describing the results of the simulations for sensor and actuator failures. The final section summarizes the paper and provides some conclusions.

2. Neural Network-Based Sensor Failure Detection, Identification, and Accommodation

Using on-line learning NN estimators, the SFDIA problem can be approached by introducing multiple feed-forward NNs trained on-line with the EBPA. Particularly, the scheme consists of a main NN (MNN) and a set of n decentralized NNs (DNNs), where n is the number of the sensors in the flight control system without physical redundancy. The outputs of the MNN are the estimates of the same parameters measured by the n sensors at time 'k', using measurements from time instant k-1' to k-l'; these estimates are compared with the actual measurements at time k'. For the i-th of the n DNNs, the output is the estimate of the measurement of the i-th sensor, that is, the prediction of the state at time 'k', using measurements from 'k-1' to 'k-l' to be compared with the actual measurement at time 'k'. The inputs to the i-th DNN are the measurements from any number to up '$n-1$' sensors, in other words, all the n sensors excluding the i-th one.

For SFD purposes, when a quadratic estimation error parameter from the MNN exceeds some predefined threshold at a certain time instant, the scheme deduces that a sensor failure may be occurring or has already occurred. Following the positive sensor failure detection, the learning for each DNN is halted; then, a quadratic estimation error parameter from the DNNs exceeding, at the same time instant, another threshold provides the identification. For the accommodation phase, the i-th DNN output is used to replace the measurement from the faulty sensor; i-th DNN output is also used as input to the MNN for the purpose of allowing the MNN to provide detection capabilities until the end of the flight. This output is also passed to all other DNNs using the i-th sensor as an input parameter. This double trigger' approach using both MNN and DNNs has the purpose of reducing the rate of false alarms in the FDI process. Several options can be added to this scheme to add robustness for noisy measurements and/or intermittent sensor failures. For example, a lower and a higher threshold level can be introduced for the DNNs. If the estimation error for the i-th DNN exceeds the lower threshold once, the status of the corresponding i-th sensor is declared suspect and the numerical architecture of the i-th DNN is not updated. Should this status continue for a certain number of time instants and/or the estimation error in successive time instants exceeds the higher threshold, then the sensor is declared failed and is, therefore, replaced by the output from the i-th DNN.

The following quadratic parameters are used for sensor failure detection (SFD) purposes :

$$MQEE(k)=\frac{1}{2}\sum_{i=1}^{Num.\ of\ DNNs}(Y_i(k)-O_{i,MNN}(k))^2$$

$$=\frac{1}{2}\left[\begin{array}{l}(p(k)-\hat{p}_{MNN}(k))^2+(q(k)\\ -\hat{q}_{MNN}(k))^2+(r(k)-\hat{r}_{MNN}(k))^2\end{array}\right] \quad (2)$$

or

$$OQEE(k)=\frac{1}{2}\sum_{i=1}^{Num\ of\ DNNs}(O_{i,MNN}(k)-O_{i,DNN}(k))^2$$

$$=\frac{1}{2}\left[\begin{array}{l}(\hat{p}_{MNN}(k)-\hat{p}_{DNN}(k))^2+(\hat{q}_{MNN}(k)\\ -\hat{q}_{DNN}(k))^2+(\hat{r}_{MNN}(k)-\hat{r}_{DNN}(k))^2\end{array}\right] \quad (3)$$

where $\hat{p}_{,DNN}$, $\hat{q}_{,DNN}$ and $\hat{r}_{,DNN}$ are the estimates of p, q and r from the respective DNNs. For simplicity purposes, let us consider failures only for the pitch rate, the roll rate, and the yaw rate gyros. In general, MQEE provides better performance for step-type sensor failures whereas OQEE performs better for ramp-type of sensor failures[4].

The sensor failure identification (SFI) can instead be achieved by monitoring the absolute value of the estimation error of each DNN, defined as

$$DQEE_x(k)=\frac{1}{2}(x(k)-\hat{x}(k))^2$$

where $x=p$, q and r (4)

For sensor failure accommodation (SFA) purposes, the following classic parameters for the estimation error are instead evaluated:

$$DAEE_x=\frac{1}{N}\sum_{k=1}^{N}(x(k)-\hat{x}(k)) \quad (5)$$

$$DVEE_x=\frac{1}{N}\sum_{k=1}^{N}[(x(k)-\hat{x}(k))-DAEE_x]^2$$

where $x=p$, q and r (6)

where DAEE and DVEE represent the DNN estimation error mean and variance respectively. The 'N' refers to the number of time steps from the instant when failure of sensor is declared to the end of the simulation.

3. Neural Network-Based Actuator Failure Detection, Identification, and Accommodation

The occurrence of any actuator failure also implies that the parameter MQEE' defined above exceeds a selected threshold. Thus, the actuator failure detection (AFD) can be achieved by spotting substantial changes in the aircraft angular velocities following any type of actuator failure. Next, the actuator failure identification (AFI) can be performed by analyzing specific cross-correlation functions. In general, for two random processes $Y(k)$ and $X(k)$, a cross-correlation function is defined by :

$$R_{YX}(n)=E[Y(k)X(k+n)] \quad (7)$$

For AFI purposes the cross-correlation functions R_{pq}, R_{pr} and R_{qr} are used, taking advantage of the fact that any type of actuator failure on any aircraft control surface involves a loss of symmetry. This loss is then followed by a dynamic coupling between longitudinal and lateral-directional aircraft dynamics. Furthermore, the auto-correlation function, R_{rr} has shown capabilities as a useful identification tool in the event of a rudder actuator failure.

Following a positive AFD and AFI, the immediate objective is to regain equilibrium and to compensate for the pitching, rolling, and yawing moments induced by the failure. Toward this goal, three separate NN controllers are introduced: i.e., a NN pitch, a NN roll, and a NN yaw controller.

The output of the NN pitch controller is the compensating deflection for the remaining healthy elevator (elevator failure case) or the symmetric elevators (aileron and/or rudder failure case). The on-line learning for the NN pitch controller is initiated as the simulation starts. Under nominal conditions, the controller is trained to emulate the actual control deflections for the symmetric elevators. Therefore it minimizes the cost function :

$$J_{pitch_{nom}}=(\delta_{H_{L,R}}-\hat{\delta}_{H_{L,R}}) \quad (8)$$

Following a positive AFD and AFI showing the need for a longitudinal AFA, the on-line learning NN pitch controller switches its target to minimize the cost function:

$$J_{pitch_{AFA}} = k_1(q - q_{ref}) + k_2(\theta - \theta_{ref}) + k_3(\dot{q} - \dot{q}_{ref}) \quad (9)$$

where $q_{ref} = 0$, $\theta_{ref} = \theta_{trim}$, and $\dot{q}_{ref} = 0$.

It should be noted that this cost function resembles a controller with a PID error formulation. Also, it should be mentioned that the on-line learning at nominal conditions (that is with the AFDIA process not activated) has no physical meaning; in fact, the NN pitch controller is just "emulating" control deflections at nominal conditions. However, this procedure has shown the benefit of improving the transient response by having the NN output within the same order of magnitude of the NN output necessary when the AFA process is turned on following a positive AFD and AFI.

The NN roll and yaw controllers operate in a similar fashion. Under nominal conditions, these controllers learn to replicate the actual control deflections for the ailerons and the rudder by minimizing the cost functions:

$$J_{roll_{nom}} = (\delta_{A_{L,R}} - \hat{\delta}_{A_{L,R}}) \quad (10)$$

$$J_{yaw_{nom}} = (\delta_R - \hat{\delta}_R) \quad (11)$$

Following positive AFD and AFI, the on-line learning NN roll and yaw controllers switch their targets to minimize the cost functions:

$$J_{roll_{AFA}} = k_4(p - p_{ref}) + k_5(\phi - \phi_{ref}) \quad (12)$$

$$J_{yaw_{AFA}} = k_6(r - r_{ref}) \quad (13)$$

where the $p_{ref} = \phi_{ref} = r_{ref} = 0$ at trim conditions.

Figure 1 shows a block diagram of the AFDIA process integrating with SFDIA.

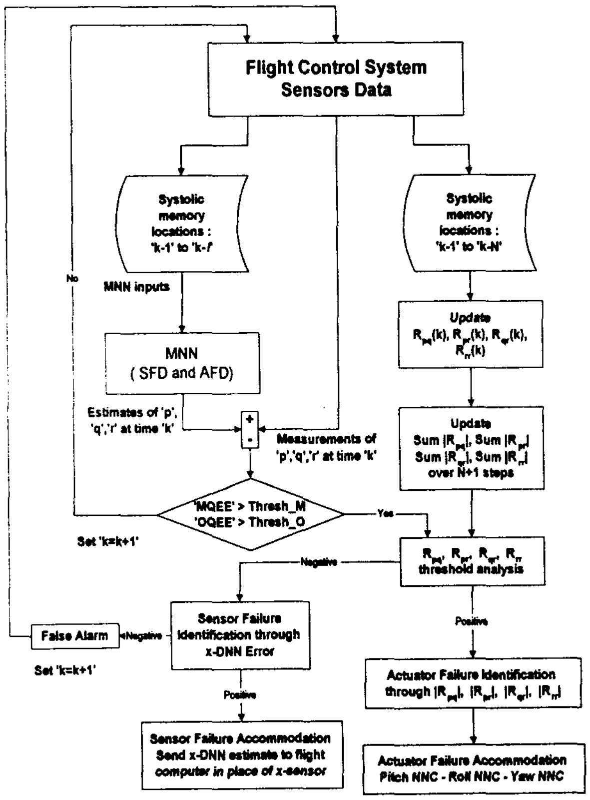

Fig. 1 Block diagram of the AFDIA scheme integrating with SFDIA

4. Analysis of the Integration between the SFDIA and the AFDIA Schemes

As stated in the previous section, the SFDIA scheme is simulated only for failures of the pitch, roll, and yaw rate gyros. Therefore, the NN-based SFDIA scheme uses one main NN (MNN) with three decentralized NNs (DNNs). The architectures for the MNN and the q-DNN are shown in Table 1. Note that input parameters of q-DNN do not include pitch rate measurements to have reliable estimates. Similarly, the p-DNN and the r-DNN do not use 'p' and 'r' as input parameters, respectively. The AFDIA scheme is simulated for failures of the actuators of elevators, ailerons, and rudder. Therefore, the AFDIA scheme consists of three NN controllers and shares the MNN with the SFDIA as also shown in Table 2.

To ensure robustness in the performance, the SFDIA scheme requires a better understanding of the typical nature of sensor failures. Angular rate gyro accuracy is characterized by two parameters: drift and scale factor. Gyro drift characterizes the ability of the gyro to reference all rate measurements to the nominal zero points. It appears as an additive term on the gyro output, so sensor failure can be modeled as,

$$x_{failure,\,i}=x_{nom,i}+\rho n_i \tag{14}$$

where n_i is the direction vector for the i-th faulty sensor, and ρ is the magnitude of the failure which can be positive or negative. The scale factor describes the capability of the gyro to accurately sense angular velocity at different angular rates. System imperfections can cause variations in the scale factor which appear as a factor at the output of the gyro. Therefore, sensor failure can be modeled as,

$$x_{failure,\,i}=(1.0+kn_i)x_{nom,i} \tag{15}$$

where k is a amplitude for the i-th faulty sensor.

According to the direction vector, n_i sensor failure can be modeled as:

- for step-type sensor failures, $n_i=1$;
- for ramp-type sensor failures,

$$\frac{t-t_{f1}}{t_{f2}-t_{f1}} \qquad (t_{f1}\le t\le t_{f2})$$

$$n_i=1 \qquad (t\ge t_{f2})$$

where t_{f1} and t_{f2} indicate the initial and final time instant of ramp-type sensor failure, respectively. The different types of sensor failures considered in this study based on the above formula are represented by :

Type #1&2 : Large/small sudden bias;
Type #3&4 : Large/small drifting bias with fast transient period (<1 sec);
Type #5&6 : Large/small drifting bias with slow transient period (<5 sec);

The maximum control surface deflections in the simulation code are ± 15 degrees for the elevators and ± 20 degrees for the ailerons and the rudder. The code also models the actuation rate for each of the control surfaces with a maximum deflection rate of ± 20 deg/sec. The actuator failures are assumed to occur randomly during high speed cruise conditions. The considered actuator failures are given by :

Case #1&2 : Stuck R/L elevator at trimmed/untrimmed deflections with/without missing surface;
Case #3&4 : Stuck R/L aileron at trimmed/untrimmed deflections with/without missing surface;
Case #5&6 : Stuck rudder at trimmed/untrimmed deflection with/without missing surface;

It is important to note that the mathematical modeling of the actuator failures in the event of a missing part of the control surface is derived through a set of closed form expressions for the non-dimensional stability and control derivatives. These expressions are functions of the normal force coefficient of the control surface whose

Table 1 Neural network architecture for SFDIA

	MNN	$q-DNN$
Input parameter	I_{MNN}	I_{q-DNN}
Data Pattern	5	5
Total number of input	40	20
Number of HLs	1	1
Number of HL neurons	25	20
Outputs	$\hat{p},\ \hat{q},\ \hat{r}$	$\hat{q}$
Learning rates	0.01	0.01
Momentum coefficients	0.005	0.005
Parameters to be updated	1187	504
$I_{MNN}=p,\ q,\ r,\ \phi,\ \theta,\ \delta_E,\ \delta_A,\ \delta_R$		
$I_{q-DNN}=\alpha,\ a_n,\ a_x,\ \dot{w}$		

Table 2 Neural network architecture for AFDIA

	$q-NNC$	$r-NNC$
Input parameters	I_{q-NNC}	I_{r-NNC}
Data Pattern	2	3
Total number of inputs	6	15
Number of HLs	1	1
Number of HL neurons	20	20
Outputs	$\hat{\delta}_{E(L,R)}$	$\hat{\delta}_R$
Learning rates	0.2	0.5
Momentum coefficients	0.1	0.3
Parameters to be updated	224	404
$I_{q-NNC}=q,\ \theta,\ \delta_E$		
$I_{r-NNC}=r,\ p,\ \beta,\ \phi,\ \delta_r$		

actuator is assumed to have failed[7,10].

As described in previous sections, the SFDIA and the AFDIA schemes share the same detection mechanism. Once the detection-alert is triggered, then the critical task is to decide among the occurrence of a sensor failure, an actuator failure, and a false alarm. For this purpose a key role is played by the trends shown by the on-line calculated and stored auto correlation(R_{rr}) and cross correlation functions (R_{pq}, R_{qr}, R_{pr}). Only when any of these functions exceeds a threshold an actuator failure declared, in lieu of a false alarm or a sensor failure. This is true even if at the same time any of the quadratic estimation errors from the DNNs is exceeding its thresholds.

The mathematical model used for this study is the model of a B747-200 aircraft. Using aerodynamic and thrust data[21], a simulation code was developed. This model features non-linear dynamics, linearized aerodynamics and it includes system and measurement noise[22]. The system noise is modeled as zero mean, white, Gaussian gust disturbance on the angle of attack and on the sideslip angle. The sensor noise is also assumed to be Gaussian and white. The primary control surfaces consist of two differential elevators, two differential ailerons, and rudder.

The numerical simulation starts at typical cruise conditions, defined by an altitude of 40,000 ft and at an airspeed of 871 ft/sec. An on-line learning process is simulated following 30,000 sec. of off-line training. For the purpose of showing the complicated SFDIA-AFDIA integration scheme, a sequence of different failures is introduced within a 150 sec. simulation:

Failure #1: an actuator failure on the right elevator with a stuck surface at +10 deg. leading to a 25% reduction in aerodynamic effectiveness. Failure occurrence: $t=30$ sec.

Failure #2: a pitch rate gyro failure involving additive large drifting bias (〈 5 deg/sec : 2.5 deg/sec) with slow transient period (〈 5 sec : 2 sec). Failure occurrence: $t=60$ sec.

Failure #3: an actuator failure on the rudder with a stuck surface at +7 deg. with a 25% reduction in aerodynamic effectiveness. Failure occurrence: $t=100$ sec.

Fig. 2(a) shows the trend of the MQEE parameter around the occurrence of failure #1. Although failure #1 is an actuator failure, it could be interpreted as a sensor failure from the Fig. 2(b). However, the suspected sensor failure is overruled by the trend of the cross correlation functions. In other words, a comparison of the magnitudes of the sums of the absolute values of the cross correlation R_{pq}, R_{pr} and R_{qr} allows

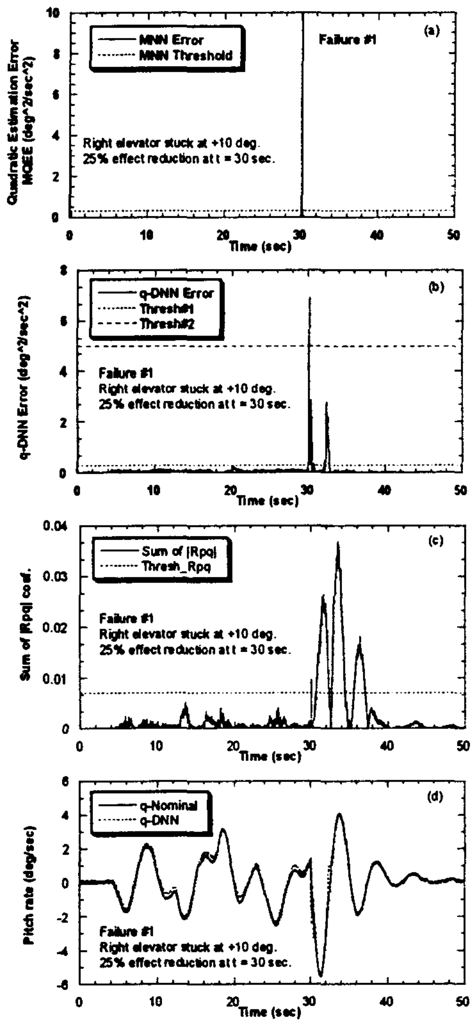

Fig. 2 Plots of AFDIA phase following right elevator failure

the identification of the failure as an "elevator actuator failure", as shown in Fig. 2(c). It should be noted that the decreasing trend in Fig. 2(c) is due to the successful AFA, following the positive AFDI, provided by the combined actions of the on-line learning pitch, roll, and yaw neural controllers. Fig. 2(d) shows the time history of the pitch rate, which confirms that the fault tolerance schemes can regain equilibrium of the aircraft following the elevator actuator failure.

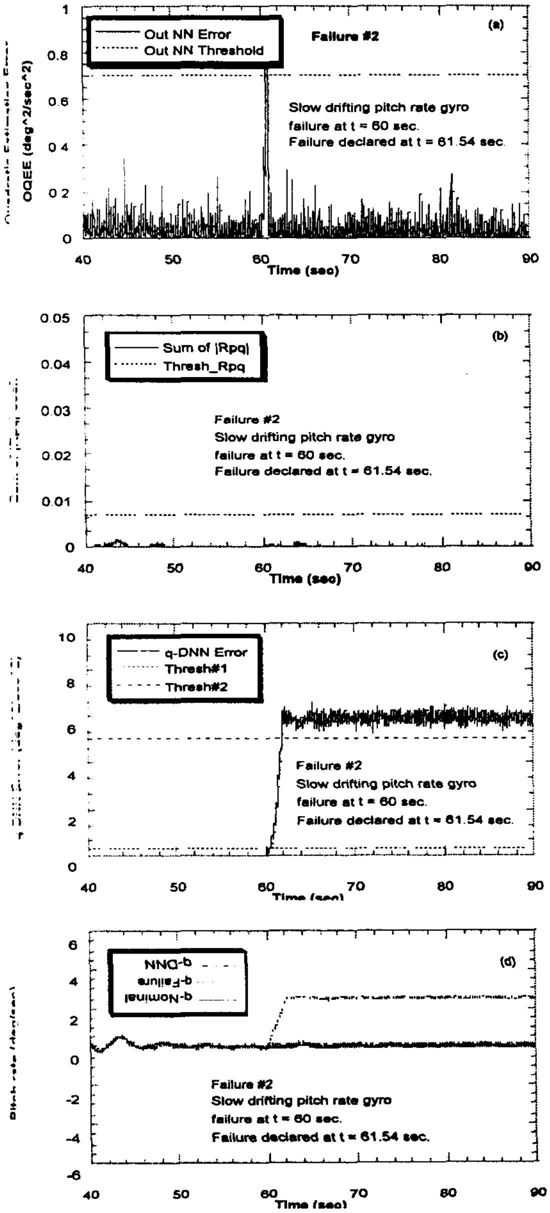

Fig. 3 Plots of SFDIA phase following pitch rate gyro failure

Fig. 3(a) shows that the "OQEE" parameter exceeds the predefined threshold during the observation period. It triggers a "state-of-alert" for both actuators and sensors. However, since failure #2 is a sensor failure, this time a substantial longitudinal-lateral dynamic coupling, R_{pq}, has not occurred, as revealed in Fig. 3(b). Furthermore, the clear trend of the q-DNN error shown in Fig. 3(c) identifies the failure #2 as a "pitch rate gyro failure". It should be noted that from the instant q-DNN error exceeds threshold #1 the on-line learning for the q-DNN is halted; furthermore, from the instant q-DNN error exceeds threshold #2, the q-DNN error becomes meaningless since the output of the q-DNN replaces the reading from

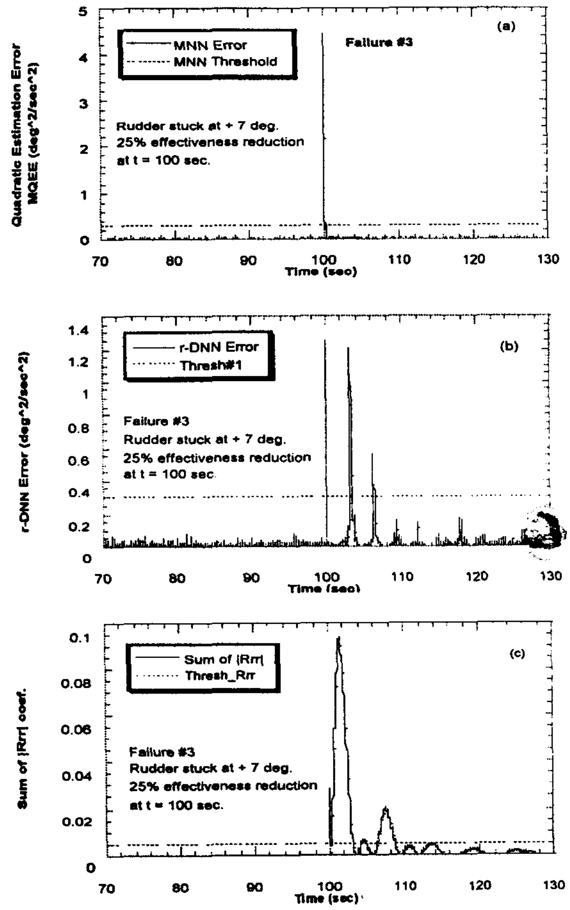

Fig. 4 Plots of AFDIA phase following rudder failure

the pitch rate gyro. Fig. 3(d) shows a successful SFA following by a positive SFDI. Thus, the time history of q-DNN is in good agreement with the nominal value for pitch rate gyro.

Failure #3, the last failure in the simulation, is a rudder actuator failure. Once again, the "MQEE" parameter exceeds its threshold triggering detection failure for both actuator and sensor in Fig. 4(a). Once again the r-DNN error exceeds the lower threshold, as shown in Fig. 4(b): this halts the on-line learning for the r-DNN. However, the suspected 'sensor failure' is overruled by the trend of the sum of the absolute values of the coefficients of the auto correlation function R_{rr} which exceeds its threshold, as shown in Fig. 4(c). Therefore a 'rudder actuator failure' is declared. As in Fig. 2(c), a decreasing trend can be noticed in Fig. 4(c) due to the successful AFA achieved by the on-line learning roll controller.

The overall effectiveness of the AFA scheme for both elevator and rudder failures is more clear from the trends in Fig. 5(a) through 5(c), which show the deflections of the left and right elevators, left and right ailerons, and rudder. As stated above, the primary goal following an actuator failure is to regain a trimmed equilibrium condition for the aircraft. For that purpose, following failure #1 the left elevator provides the necessary pitching deflection as calculated by the on-line learning pitch neural controller, as shown in Fig. 5(a). Fig. 5(b) shows instead the compensating aileron deflections canceling the rolling moment induced as a cross-effect by the

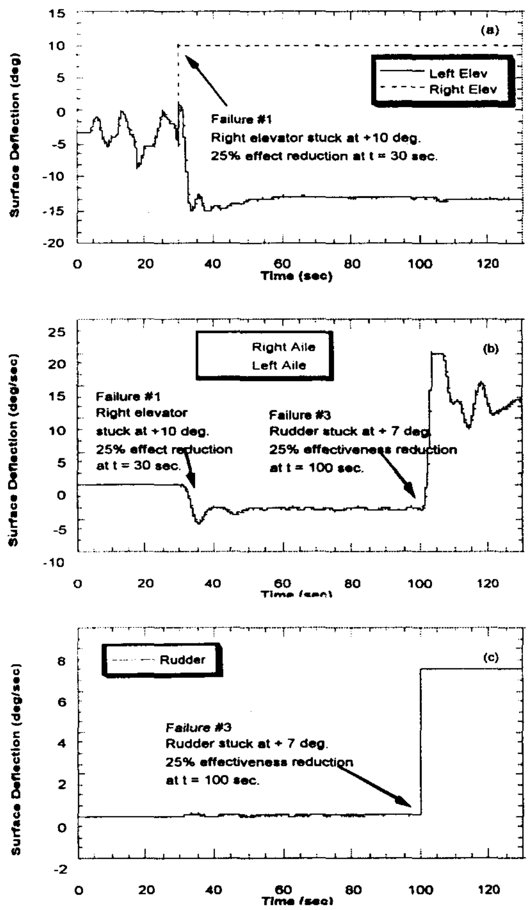

Fig. 5 Plots of control surface deflection following failures

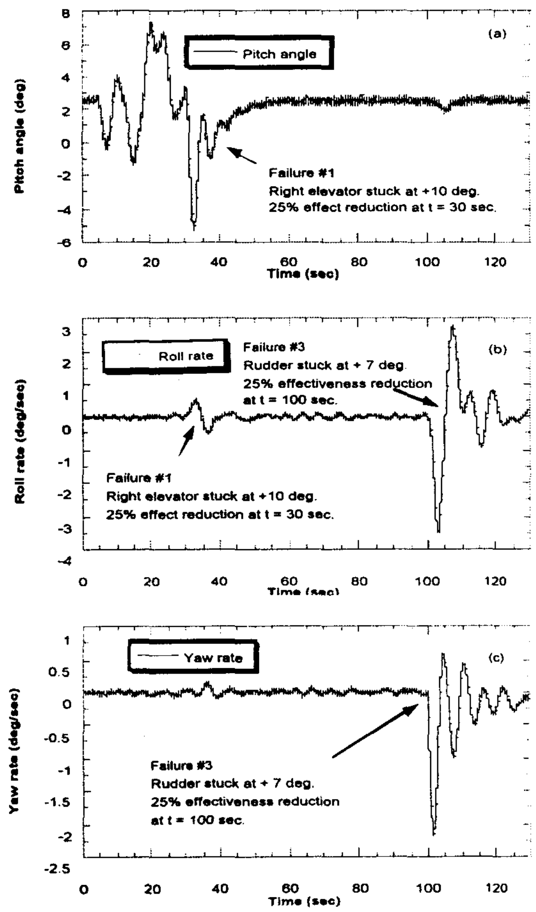

Fig. 6 Plots of accommodation phases following sensor and actuator failures

right elevator failure; also shown after $t=100$ sec. is the aileron deflection canceling the yawing moment induced as a cross-effect by the rudder failure.

Figure 6(a) through 6(c) show the time histories of key aircraft parameters, which is the dynamic scenario of the simulation. That is, the pitch angle(θ), roll rate(p), and yaw rate(r), respectively. Again, these time histories verified that the fault tolerant schemes were able to regain equilibrium of the aircraft following the different failures.

5. Conclusions

This paper has presented a neural network-based fault tolerant flight control system with capabilities for detecting, identifying, and accommodating sensor and actuator failures. A particular logic is introduced allowing the integration of the two schemes with the goal of minimizing the false alarm rate as well as incorrect failure identification. The results confirm the potential offered by on-line learning NNs for both state estimation and control purposes within fault tolerant systems. A further study for the assessment of reliability of this integration technique is required.

References

(1) Friedland, B., 1982, "Maximum Likelihood Failure Detection of Aircraft Flight Control Sensors", Journal of Guidance, Control, and Dynamics, Vol. 5, No. 5, pp. 498～503.

(2) H. Baruh, K. Choe, 1987, "Sensor-Failure Detection Method for Flexible Structures", Journal of Guidance, Control, and Dynamics, Vol. 10, No. 5, pp. 474～482.

(3) Napolitano, M. R., Neppach, C. et. al., 1995, "Neural-Network-Based Scheme for Sensor Failure Detection, Identification, and Accommodation", Journal of Guidance, Control, and Dynamics, Vol. 18, No. 6, pp. 1280～1286.

(4) M.R. Napolitano and Younghwan An, 1998, "On-Line Learning Neural Network Approaches for Sensor Validation", 14th IFAC Symposium on Automatic Control in Aerospace.

(5) Ha, C. M., Wei, Y., Bessolo, J.A., 1992, "Reconfigurable Aircraft Flight Control System Via Neural Networks", Proceedings of the 1992 Aerospace Design Conference, AIAA Paper 92～1075.

(6) Huang, C., Tylock, J., Engel, S., Whitson, J., Eilbert, J., 1992 "Failure-Accommodating Neural Network Flight Control", Proceedings of the AIAA GNC Conference, 92～4394.

(7) Napolitano, M. R., Chen, C.I., et. al., 1993, "Aircraft Failure Detection and Identification Using Neural Networks", Journal of Guidance, Control, and Dynamics, Vol. 16, No. 6, pp. 999～1009.

(8) Ho, H.S., S. N. Balakrishnan, 1995, "Fuzzy Logic in Restructurable Flight Control Systems", Proceedings of the American Control Conference, Part 2, pp. 1362～1366.

(9) Y.Ochi, K. Kanai, 1995, "Application of Restructurable Flight Control System to Large Transport Aircraft", Journal of Guidance, Control, and Dynamics, Vol. 18, No.2, pp. 365～370.

(10) Napolitano, M. R., Casdorph, V., 1996, "On-line Learning Neural Architectures and Cross-Correlation Analysis for Actuator Failure Detection and Identification", International Journal of Control, Vol. 63, No. 3, pp. 433～455.

(11) Wilsky, A. S., 1980, "Failure Detection in Dynamic Systems", Agard LS-109, pp. 2.1～2.14.

(12) Kerr, T. H., 1982, "False Alarm and Correct Detection Probabilities over a Time Interval for Restricted Classes of Failure Detection Algorithms", IEEE Transactions of Information Theory, Vol. IT-28, No. 4, pp. 619～631.

(13) Rumelhart, D., and McClelland, J., 1986, "Parallel Distributed Processing, MIT Press", Cambridge, MA.

(14) Nielsen, R. H. 1989, "Neurocomputing", Addison Wesley Publishing Company.

(15) Simpson, P. K., 1990, "Artificial Neural Systems: Foundations, Paradigms, Applications", and Implementations, Pergamon Press, New York.

(16) Narendra, K. S., Partasarathy, K.1990, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, Vol. 1, No 1, pp. 4~27.

(17) Hunt, K. J.,Sbardato, D., Zbikowski, R., and Gawthrop, P.J. 1992, "Neural Networks for Control Systems-A Survey", Automatica, Vol 28, No 6, pp 1083~1112.

(18) Levin, A. U., Narendra, K.S., 1993 "Control of Non-Linear Dynamical Systems Using Neural Networks: Controllability and Stabilization", IEEE Transactions on Neural Networks, Vol. 4, No. 2, pp. 124~142.

(19) Cybenko, G., 1989, "Approximation by Superposition of Sigmoidal Functions", Mathematics of Control Signals and Systems, Vol. 2, No. 4, pp. 303~309.

(20) Chen, C. L., Nutter, R.S., 1992, "An Extended Back-Propagation Learning by Using Heterogeneous Processing Units", International Joint Conference on Neural Networks, pp. III-988~993.

(21) Roskam, Jan, 1995, Airplane Flight Dynamics and Automatic Flight Controls, Design, Analysis and Research Corporation, Lawrence, Kansas.

(22) Motyka, P., W. Bonnics, 1985, "The Evaluation of Failure Detection and Isolation Algorithms for Recontstructurable Control", NASA Contractor Report 177983.