SENSOR FAULT TOLERANT CONTROL SYSTEM USING ONLINE MULTI-LAYER FEED-FORWARD NEURAL NETWORK

Sudesh Kumar Kashyap[1] *and Girija G*[2]

[1]*Scientist E1, FMCD, NAL, Bangalore 560 017, sudesh@css.nal.res.in*
[2]*Deputy Head, FMCD, NAL, Bangalore 560 017, ggirija@css.nal.res.in*

EXTENDED ABSTRACT

Sensor Fault Detection Identification and Accommodation (SFDIA) is an important part of safety critical systems used in aircraft. SFDIA can be achieved either by hardware redundancy or analytical redundancy technique. The advantages like reduced complexity, cost and weight of analytical redundancy over hardware redundancy encourages the designers to follow the former technique. Analytical redundancy techniques could use either model based or non-model based approaches. Model based techniques include observer based residual generation, parity based and parameter based approaches [1]. Fuzzy decision-making and artificial neural networks are used for building analytical redundancy in non-model based approaches. Due to the learning and adaptation capability of Neural Network (NN) [2-4], applicability to nonlinear and multivariable systems, parallel distributed processing and hardware implementation, Artificial NNs are very appealing for the purpose of providing fault tolerance capabilities in a flight control system following sensor failures.

In this paper, the SFDIA is achieved by using a Main Neural Network (MNN) and n Decentralized Neural Networks (DNNs) for a system with n sensors. Here MNN is used to detect the fault and DNN is used for identifying the fault. The reconfiguration of faulty sensor can be achieved by feeding back the DNN estimate for the faulty sensor instead of sensor measurement to the flight control system. The SFDIA scheme is realized using MATLAB/SIMULINK® for closed loop decoupled linearised models (Refer Appendix A for the longitudinal and lateral motion models) of Aerosonde UAV [5-6] having pitch and roll angle autopilots with rate feedback [7]. The SFDIA algorithm is evaluated for pitch and roll rate sensor faults of constant bias type.

Figure 1 shows the block diagram of proposed algorithm for longitudinal motion. A similar approach has been used for lateral motion.

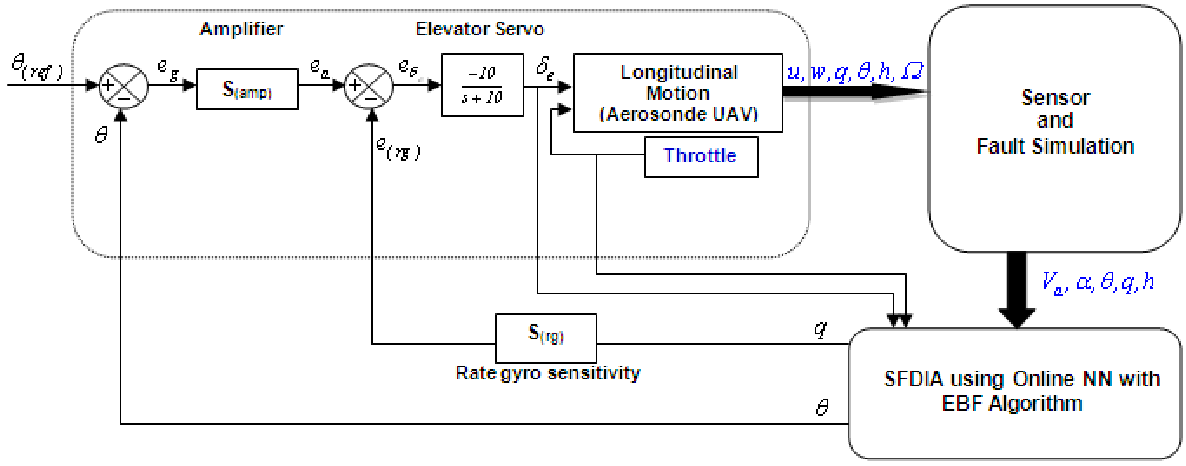

Fig. 1: Proposed SFDIA Algorithm for Longitudinal Motion of UAV

In present case, the gain $S_{(amp)}$ of outer loop is kept at 5, whereas the value of rate gyro sensitivity $S_{(rg)}$ is chosen based on trial and error to meet the general requirements of rapid response with minimal overshoot. It was found that $S_{(rg)}$ at 0.9 yields fairly satisfactory response of pitch angle as compared to its reference value. The non-model based SFDIA scheme is evaluated using linearised longitudinal model of Aerosonde UAV (available in Aerosim blockset of MATLAB) at flight condition (airspeed = 23 [m/s], altitude 200 [m], bank angle = 0 [rad], fuel mass = 2 [kg], flap setting = 0). The state defining longitudinal motion of UAV are u,w,q,θ,h,Ω, where, i) u,w are ground speed along x and z- axis respectively, ii) q,θ,h,Ω are pitch rate, pitch angle, altitude and propeller rotation speed respectively. The measurements from sensors are total Airspeed (V_a), angle of attack (α), pitch angle, pitch rate and altitude. For the results presented in this abstract, it is assumed that these measurements are noise free and fault is introduced in pitch rate sensor only. Results pertaining to the case with realistic sensor noise values will be presented at the workshop.

Sensor fault detection, isolation and reconfiguration are carried out using multi-layered feed-forward Neural network with extended back propagation (EBP) as a learning algorithm. The problem of slow speed of learning and local minima in BP can be solved with EBP, which is a heterogeneous network where each neuron in the hidden and output layer of Neural network has its own output capability of updating some new parameter giving the overall architecture increased mapping and adaptation capabilities. In a heterogeneous network each neuron is able to change its output range (upper and lower bounds) and the slope of the sigmoid activation function. The details of SFDIA algorithm would be provided in final paper. The validation of SFDIA algorithm is carried out using different types of fault in pitch rate sensors injected at different instants of time. In this abstract, the results are shown for a step fault of 0.1 rad/sec introduced at 20th second onwards of simulation.

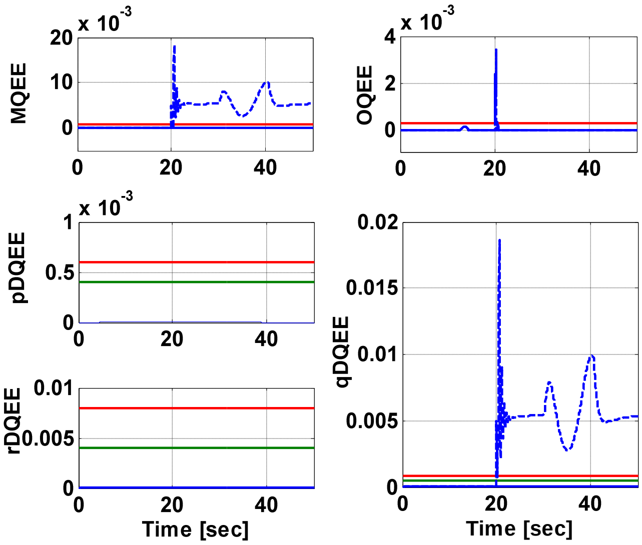

Fig. 2: Sensor Fault Detection and Isolation

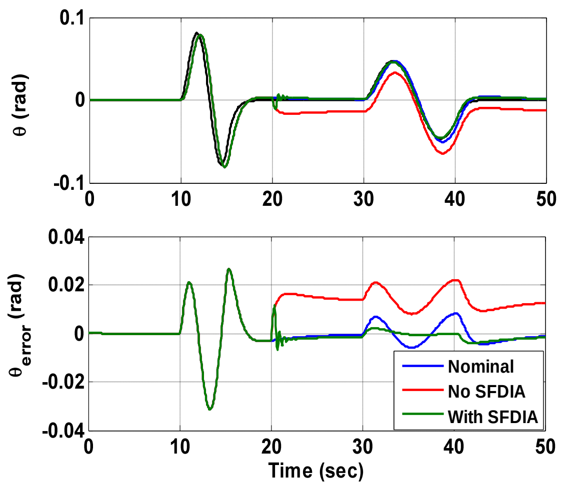

Fig. 3: Pitch Angle Responses

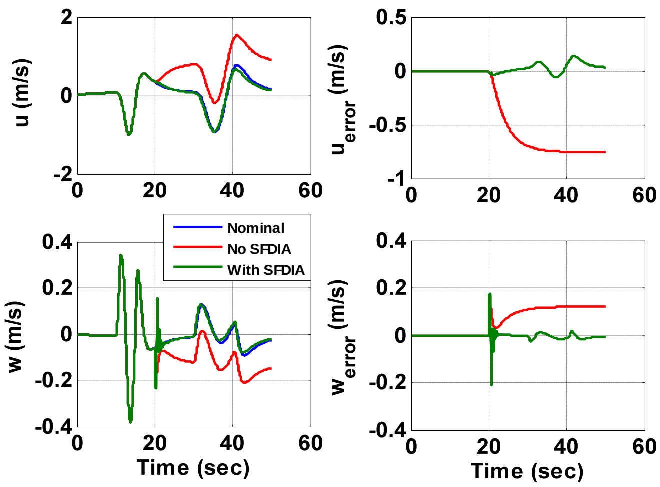

Fig. 4: Forward and Vertical Speed Reponses

Figure 2 shows the plots of MQEE – main quadratic estimation error generated by the MNN, the DQEE – Decentralized quadratic estimation errors at the p, q, r DNNs and the OQEE - output quadratic estimation errors along with the corresponding thresholds for sensor fault detection and isolation. The thresholds are selected based on visual inspection of these errors under no fault condition. It is clear from the plots that sensor fault is detected at the 20th sec. by the MNN. The qDQEE error clearly identifies the fault as being in the pitch rate sensor. Figure 3 shows the comparison of reference pitch angle with pitch angle achieved by UAV under three conditions i) no fault in pitch rate sensor-nominal , ii) a step fault of 0.1 rad/sec at 20th second onwards but without SFDIA and iii) when NN based SDFIA used. It is observed from the figure that there is a definite improvement in UAV response when SFDIA is introduced in the loop. Similar observation can be made from Figure 4, where ground speeds along x and z-axis are compared. It can been seen from the plots that without SFDIA ground speeds are significantly away from their true values, whereas, with SFDIA they are comparable to true values.

Conclusion and Future works

SFDIA for Aerosonde UAV using online Neural network with EBP has been proposed. The results are found to be satisfactory. The exercise carried out was for demonstrating the capabilities of the NN based SFDIA algorithm. Further validation of the methodology is planned with multiple faults in sensors and control surfaces.

References

1. Takahisa Kobayashi, Donald.L.Simon, (2003), “Application of Bank of Kalman Filter For Aircraft Engine Fault Diagnosis”, American Society of Mechanical Engineers and the International Gas Turbine Institute, Atlanta, Georgia.
2. Brian M.Stolarik, (2001), “A Neural Network- based Sensor Validation Scheme in Aircraft Control Laws”, Lane Department of computer Science and Electrical Engineering, The College of Engineering and mineral resources, West Virginia University.
3. Marcello.R.Napolitano, Giovanni Silvestri, Dale.A.Windson II, Jose Luise Casanova, Mario Innocenti, (1998), “Sensor Validation Using Hardware-Based On-Line Learning Neural Networks”, IEEE transactions on aerospace and electronic systems vol. 34, no. 2 April 1998.
4. Paul.J.Werbos, (1990),”Back Propagation Through Time: What It Does and How to Do It”, Proceedings of the IEEE, Vol.78, No.10, October 1990.
5. www.mathworks.com, AeroSim Blockset User Guide.
6. http://en.wikipedia.org/wiki/Instu_Aerosonde
7. John H. Blakelock, (1990),”Automatic Control of Aircraft and Missiles”, Second Edition, Wiley inter-science publication.
8. Phalguna Kumar Rachinayani, (2006), “Robust Fault-Tolerant Control For aircraft Systems”, The Department of Electrical and Computer Engineering, Louisiana State University and Agricultural and Mechanical College.
9. Yu Gu, (2004), “Design and Flight Testing Actuator Failure Accommodation Controllers on WVU YF-22 research uav’s”, Department of Mechanical and Aerospace Engineering, The College of Engineering and mineral resources, West Virginia University.
10. Sudeesh. P.D, Sudesh K. Kashyap and R.Kayalvizhi, “A Scheme of SFDIA Using Multilayer Perceptron and EBP for Unmanned Aerial Vehicle”, International Conference on Aerospace Science and Technology, CSIR, National Aerospace Laboratories, 26-28th June 2008, Bangalore

Appendix A

Longitudinal Motion

State vector: x = [u w q θ] // Input vector: u = δ_e // Output vector: y = [V_a α q θ]

$$A = \begin{bmatrix} -0.2197 & 0.6002 & -1.4882 & -9.7967 \\ -0.5820 & -4.1204 & 22.4024 & -0.6461 \\ 0.4823 & -4.5284 & -4.7512 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0.0658 & -0.9978 & 0 & 22.9997 \end{bmatrix}; \; B = \begin{bmatrix} 0.3246 \\ -2.1520 \\ -29.8216 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.9978 & 0.0658 & 0 & 0 \\ -0.0029 & 0.0434 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

Lateral Motion

State vector: x = [v p r ϕ φ] // Input vector: u = [δ_a δ_r] // Output vector: y = [β p r ϕ φ]

$$A = \begin{bmatrix} -0.6373 & 1.5135 & -22.9498 & 9.7967 & 0 \\ -4.1919 & -20.6283 & 9.9282 & 0 & 0 \\ 0.6798 & -2.6757 & -1.0377 & 0 & 0 \\ 0 & 1.0000 & 0.0659 & -0.0000 & 0 \\ 0 & 0 & 1.0022 & -0.0000 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -1.2510 & 3.1931 \\ -109.8373 & 1.9763 \\ -4.3307 & -20.1754 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.0435 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}$$