Virtual Sensor for Fault Detection and Isolation in Flight Control Systems – Fuzzy Modeling Approach

Marcel Oosterom and Robert Babuška
Delft University of Technology, Faculty of Information Technology and Systems
Control Engineering Laboratory, Delft, The Netherlands
e-mail: marcel@oosterom.nl

Abstract

A virtual sensor for normal acceleration has been developed and implemented in the flight control system of a small commercial aircraft. The inputs of the virtual sensor are the consolidated outputs of dissimilar sensor signals. The virtual sensor is a fuzzy model of the Takagi–Sugeno type and it has been identified from simulated data, using a detailed, realistic Matlab/Simulink™ model used by the aircraft manufacturer. This virtual sensor can be applied to identify a failed sensor in the case that only two real sensors are available and even to detect a failure of the last available sensor.

Keywords: Fault detection and isolation, analytical redundancy, virtual sensor, fuzzy modeling, flight control systems.

1 Introduction

In order to assure the integrity of modern fly-by-wire flight control systems (FBW FCSs), replication of equipment and supporting systems is required. This is partly because the fault detection and isolation (FDI) is based on majority voting of like signals. Cost effective benefits are therefore expected form more integrated FDI algorithms, where the validation of signals is performed using dissimilar signals also. Along these lines a Brite/EuRam project "Affordable Digital Fly-by-wire Flight Control Systems for Small Commercial Aircraft" (ADFCS) was initiated [1]. The partners are from the United Kingdom (BAE Systems), Israel (Israel Aircraft Industries, Israel Institute of Technology), Italy (Alenia Aerospazio, Centro Italiano Ricerche Aerospaziali), and The Netherlands (National Aerospace Laboratory, Delft University of Technology).

The application of FBW FCSs for small commercial aircraft (SCA) is being investigated in order to proliferate proven benefits of FBW technology, in terms of increased safety, commonality, productability, and maintainability. The application of FBW technology in primary flight control systems is the state of the art for medium to large civil transport aircraft [2]. The major cost-drivers of FBW FCSs are the hardware repetition to obtain the required reliability, dissimilarity in hardware and software to avoid generic failures, and the design effort of the flight control laws (FCLs). One part of the project investigates whether fuzzy-neural techniques can contribute to the development of a cost-effective FBW FCS for SCA.

In the literature, many applications of analytical redundancy for FDI in flight control systems are reported. Most frequently applied are observer-based techniques [3, 4], parity-space methods [5, 6, 7, 8], and parameter-estimation schemes [9, 10]. The use of virtual sensors in aerospace applications has not been widely investigated yet, although this technique has been successfully applied in other fields like process control and engine control [11, 12].

In this paper the design and implementation of a fuzzy logic (FL) based virtual sensor is introduced. This virtual sensor is used to identify the failed sensor in the duplex case (two like signals left) and to detect a failure in the simplex case (one like signal left). The proposed approach is to construct a fuzzy model of the signal of interest using dissimilar consolidated sensor readings as inputs. The virtual sensor is then completely independent of the two sensor signals that need to be validated. The use of fuzzy modeling is proposed, based on positive experience with similar applications of this technique [13]. The performance is evaluated in a number of test points in the flight envelope, using a realistic Matlab/Simulink™ model of the aircraft.

In Section 2, the conventional FBW FCS is presented as it is implemented in many modern commercial aircraft. The potential benefits of using virtual sensors are discussed. The design and open-loop evaluation of the performance of the fuzzy virtual sensor are described in Section 3. In Section 4 the close-loop performance is evaluated. Conclusions are drawn in Section 5.

2 Virtual sensors in FDI

The conventional approach to FDI, as implemented and certified on most modern aircraft is based on point consolidation of like signals. The assumption is that the ma-

jority of like signals represents the truth and that any single dissimilar signal is the result of a failure. This works well down to two signals, where any discrepancy can no longer be related to a 'majority'. In this instance, the system will either reject both signals, and reconfigure to not use this information, or for essential data a simple average will be used as the best compromise.
Obviously, this fault detection by majority voting does carry the penalty of replication of equipment and supporting services [14].

2.1 Voting

Depending on the number of (valid) signals, the level of the voter that is to be used is determined (quadruplex, triplex or duplex). The required number of sensors in the FCS is related to the level of degradation of the system in the case of losing the corresponding signal. For example, losing the pitch rate information is considered to be a catastrophic event, which implies that the probability of this event should be less than 10^{-9} per flight hour. In order to achieve this, four pitch rate sensors are required.

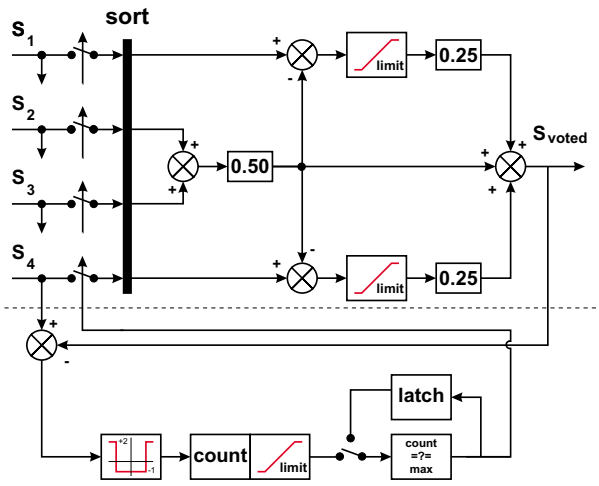

Figure 1: Conventional quadruplex voting/monitoring system (source: [14]).

For four signals, a *quadruplex* voter is used, see Figure 1. In this case, the two mid-value signals are averaged and the two extreme-value signal contributions are limited in their deviation from this average. When the limits are not invoked, the voted output is given by:

$$S_{\text{voted}} = 0.25 \sum_{i=1}^{4} S_i \,.$$

For three valid signals, a *triplex* voter is used. In this case, the mid-value signal is taken as the reference and the two extreme-value signal contributions are limited in their deviation from this value. When the limits are not invoked, the voted output is given by

$$S_{\text{voted}} = 0.25 \cdot (S_1 + 2S_2 + S_3) \,.$$

For two valid signals, a *duplex* voter is used. In this case, the two signals are averaged such that the voted output is

$$S_{\text{voted}} = 0.50 \cdot (S_1 + S_2) \,.$$

2.2 Monitoring

The difference between each input signal and the voted output signal is compared with a threshold. If the error is within the tolerance, the monitor count is decreased by one. If the error is outside the tolerance, the monitor count is increased by two. The updated count is bounded between zero and the *failure declaration value*. If the count value has reached the failure declaration value, a failure is declared and the signal is latched. The monitoring system is illustrated in Figure 1 for the quadruplex system.

In case of two valid signals (duplex), it is no longer possible to identify the failed sensor as there is no majority. However, there is information available from dissimilar sensors that could be used to identify the failed sensor in this case. A *virtual sensor* can be implemented to reconstruct the signal in question using readings of dissimilar sensors. The design of a FL based virtual sensor is illustrated in Section 3 for the normal acceleration signal.

2.3 Implementation of the virtual sensor

The virtual sensor can be used in the voting/monitoring (V/M) scheme in various ways. One option is to implement the virtual sensor in the same way as the real sensors. In this way the virtual sensor is contributing to the voted signal and will be latched in case of a failure. This might occur due to (temporary) degradation of its performance, for example, as a result of extreme turbulence or abrupt behaviour of one of its inputs.

Another possibility is to use the virtual sensor to validate the real sensors in duplex and simplex operation only. In this case, the input of the monitoring block is not the output of the voting scheme (Figure 1), but the output of the virtual sensor (Figure 2). The output of the virtual sensor is not an input of the voting scheme. During duplex operation, a failure of the virtual sensor should not lead to latching of both real sensors. In simplex operation, failure of the virtual sensor leads to reconfiguration of the FCLs.

Further research is needed to investigate how virtual sensors can best be implemented in (FBW) FCSs. In this paper the virtual sensor is used to validate the real sensors in duplex and simplex operation (Figure 2).

The benefit of using the virtual sensor in the same way as the real sensors is that the virtual sensor itself is subject to the V/M system as well. A failure of the virtual sensor is detected using the two real sensors. However, since the virtual sensor is not as accurate as the real sensors it might happen that a failure on the virtual sensor

is declared due to some (temporary) degradation of its performance, for example, as a result of extreme turbulence or abrupt behaviour of one of its inputs. In order to prevent these "false alarms" due to the virtual sensor, it is decided to use the virtual sensor only to monitor the two real sensors. This means that the voted output is simply the mean of the two real sensor readings, while the virtual sensor reading is used to identify the failed real sensor in case of a discrepancy between the two real sensor readings. This is illustrated in Figure 2.

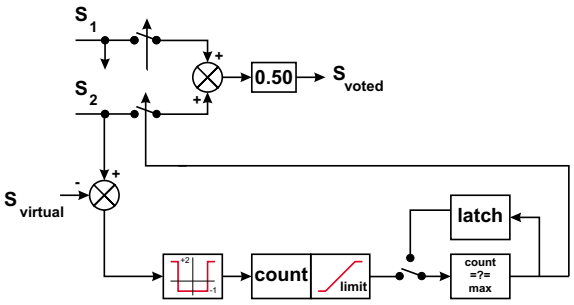

Figure 2: Duplex voter/monitor system with virtual sensor reference signal monitor.

3 Data-driven design of a FL based virtual sensor

The design and implementation a FL based virtual sensor is demonstrated for the normal acceleration signal, considering the aircraft motion in the vertical plane only (longitudinal motion). The virtual sensor is developed by using time histories of selected measured signals obtained from simulations at different flight conditions and for different pilot commands. The choice of the relevant inputs to the sensor, the generation of the data and development of the sensor are addressed in the subsequent sections.

3.1 Data generation

The data are generated by simulating a nonlinear model of the SCA implemented in Matlab/Simulink™. Only longitudinal motion is considered and sensor noise and moderate atmospheric turbulence are included.

Simulation runs are performed in a number of operating points, denoted by '+' in Figure 3, and for a number of pilot commands. The stick force input during simulation is the so-called '3-2-1-1' input, which is commonly used for identification. At each flight condition a simulation is performed with $[20\%, 40\%, 60\%, 80\%, 100\%]$ of the maximum stick force.

For each simulation run, the time histories of a number of variables were recorded. These variables are discussed in more detail in Section 3.3.

The generated data set is divided into training data and validation data. The training data set consists of the time histories obtained from the simulations applying 20%, 60%, and 100% of the maximum stick force, while the evaluation data set consists of time histories of the simulations applying 40% and 80% of the maximum stick force.

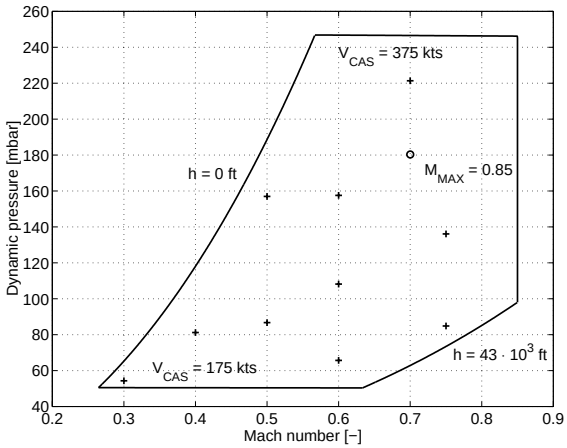

Figure 3: Operating points in the flight envelope.

3.2 Identification via fuzzy clustering

An effective approach to the identification of complex nonlinear systems is to partition the available data into subsets and approximate each subset by a linear model. Fuzzy clustering can be used as a tool to obtain a partitioning of data where the transitions between the subsets are gradual rather than abrupt.

The Takagi–Sugeno (TS) model [**?**] is used to smoothly combine the linear submodels. The TS rules have the following form:

$$R_i: \quad \textbf{If } x_1 \text{ is } A_{i1} \textbf{ and } \ldots \textbf{ and } x_p \text{ is } A_{ip} \quad \textbf{then } y_i = \mathbf{a_i^T x + b_i}, \qquad \mathbf{i = 1, 2, \ldots, K}. \quad (1)$$

This is a MISO model with inputs $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}]^\mathbf{T}$ and output y. A_{ij} are linguistic terms (like low, medium, high, etc.), represented by membership functions. Finally, $\mathbf{a_i}$ and b_i are real-valued consequent parameters.

TS rules are extracted from data by clustering in the product space of the inputs and outputs. By applying clustering algorithms that are capable of detecting linear substructures in data, a nonlinear regression problem is automatically decomposed it into several local linear subproblems. Each obtained cluster is represented by one rule in the TS fuzzy model. The antecedent membership functions are obtained by projection (Figure 4) and the consequent parameters can be estimated by various least-squares methods [**?**].

Illustrative example: Consider a nonlinear function

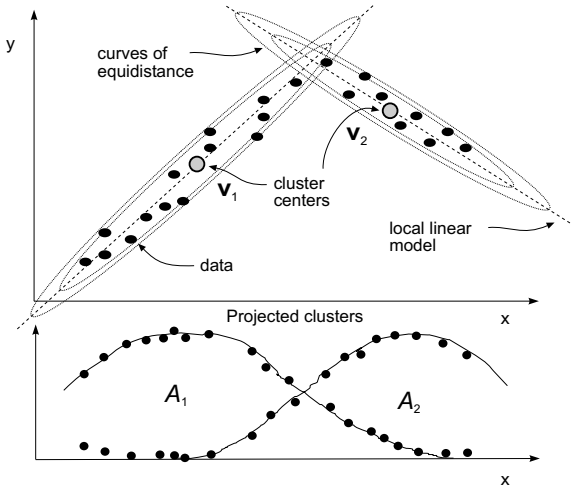

Figure 4: Extraction of Takagi–Sugeno rules by fuzzy clustering.

$y = f(x)$ defined piece-wise by:

$$\begin{aligned} y &= 0.25x, && \text{for } x \leq 3 \\ y &= (x-3)^2 + 0.75, && \text{for } 3 < x \leq 6 \\ y &= 0.25x + 8.25, && \text{for } x > 6 \end{aligned} \tag{2}$$

Figure 5 shows a plot of this function evaluated in 50 samples uniformly distributed over $x \in [0, 10]$. Zero-mean, uniformly distributed noise with amplitude 0.1 was added to y.

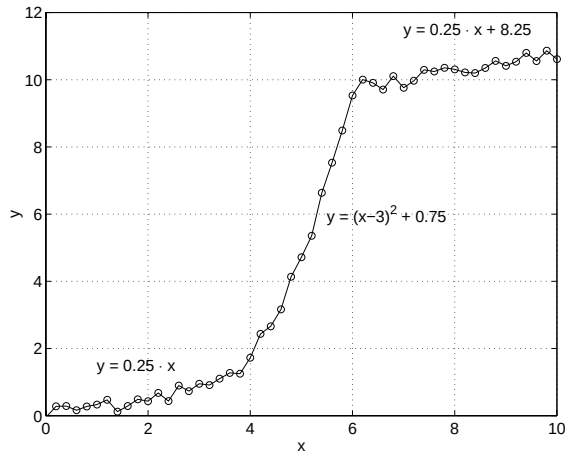

Figure 5: A nonlinear function (2).

The data set $\{(x_i, y_i)|i = 1, 2, \ldots, 50\}$ was clustered into four clusters. Figure 6 shows the local linear models obtained through clustering, the bottom plot shows the corresponding membership functions. In terms of TS rules, the fuzzy model is expressed as:

$$\begin{aligned} R_1&: \textbf{If } x \text{ is } C_1 \textbf{ then } y = 0.25 \cdot x + 0.05 \\ R_2&: \textbf{If } x \text{ is } C_2 \textbf{ then } y = 1.42 \cdot x - 3.76 \\ R_3&: \textbf{If } x \text{ is } C_3 \textbf{ then } y = 4.45 \cdot x - 17.40 \\ R_4&: \textbf{If } x \text{ is } C_4 \textbf{ then } y = 0.26 \cdot x + 8.19 \end{aligned}$$

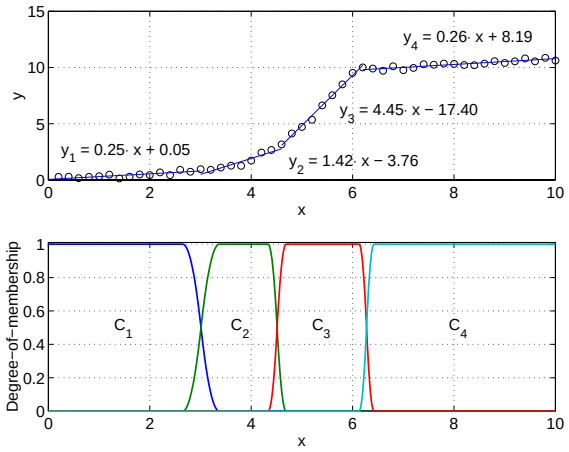

Figure 6: Cluster prototypes and the corresponding fuzzy sets.

Note that the consequents of R_1 and R_4 correspond almost exactly to the first and third equation (2). Consequents of R_2 and R_3 are approximate local linear models of the parabola defined by the second equation of (2).

3.3 Structure of the virtual sensor

The analytical equation for the normal acceleration n_z is

$$n_z = \frac{1}{g} \cdot (\dot{w}_b - u_b \cdot q_b + g \cdot cos\theta) \tag{3}$$

where u_b denotes the forward velocity, w_b the downward velocity, q_b the pitch rate, θ the pitch attitude, and g the gravitational acceleration. The time derivative of the downward velocity is estimated as follows

$$\dot{w}_b = V \cdot \dot{\alpha} \tag{4}$$

where V denotes the true airspeed and α is the angle-of-attack. The angle-of-attack is measured with vanes outside the aircraft. This causes the angle-of-attack signal to be very noisy, and therefore unsuitable for estimating the time derivative of the downward velocity.

To solve this problem, a TS fuzzy model is designed to estimate the time derivative of the downward velocity. The linear approximation of $\dot{w}_b$ is written as follows

$$\dot{w}_b = Z_u u_b + Z_w w_b + Z_q q_b + Z_\theta \theta + Z_{\delta_e} \delta_e + Z_{\delta_s} \delta_s \tag{5}$$

where δ_e denotes the elevator deflection and δ_s the stabilizer deflection. The parameters Z_u to Z_{δ_s} are the so-called aerodynamic derivatives. The local linear models of the TS fuzzy model are therefore in the form of (5) including an additional offset

$$\dot{w}_b = Z_0 + Z_u u_b + Z_w w_b + Z_q q_b + Z_\theta \theta + Z_{\delta_e} \delta_e + Z_{\delta_s} \delta_s \tag{6}$$

The nonlinearity in the aircraft dynamics is mostly due to variations in Mach number M, dynamic pressure Q_c and angle-of-attack α. These variables are used to identify the validity regions of the local linear models. Clustering is therefore performed in the data space defined

by $[\dot{w}_b, M, Q_c, \alpha]$. Two clusters were found to give good and comparable performance in both the training and evaluation data set. Projection of the clusters resulted in two membership functions for each of the antecedent variables. Finally the parameters that define the membership functions were fine-tuned by genetic algorithm optimization [16]. This improved the performance of the TS fuzzy model by 18.1% for the training set and by 13.7% for the evaluation set. The results will be discussed in more detail in Section 3.4. The resulting membership functions are shown in Figure 7.

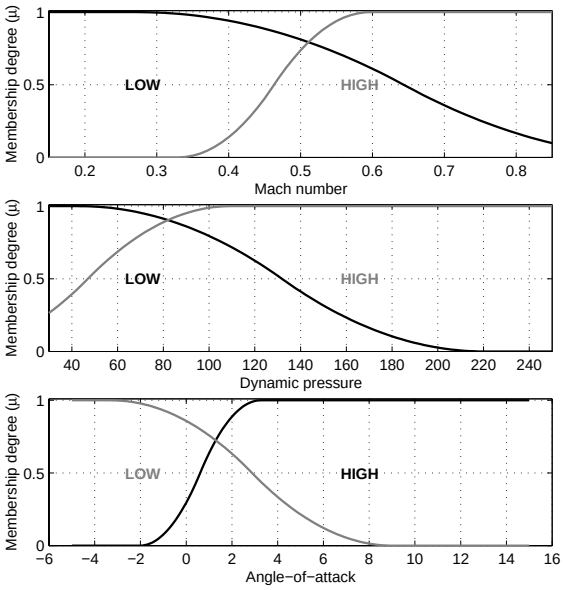

Figure 7: Membership functions for Mach number, dynamic pressure, and angle-of-attack.

In the TS fuzzy model, each of the clusters corresponds to a local linear model. In this case, the following rules are obtained:

1. **If** M is *LOW* **and** Q_c is *LOW* **and** α is *HIGH* **then** $Z_{0_1} = Z_{0_{1_0}} + Z_{0_{1_M}} M + Z_{0_{1_{Q_c}}} Q_c + Z_{0_{1_\alpha}} \alpha$
2. **If** M is *HIGH* **and** Q_c is *HIGH* **and** α is *LOW* **then** $Z_{0_2} = Z_{0_{2_0}} + Z_{0_{2_M}} M + Z_{0_{2_{Q_c}}} Q_c + Z_{0_{2_\alpha}} \alpha$

The degree of fulfillment of each of the rules is computed by taking the product of the membership degrees of each of the three statements

$$\begin{aligned} \mu_1 &= \mu_{\mathrm{LOW}}(M) \cdot \mu_{\mathrm{LOW}}(Q_c) \cdot \mu_{\mathrm{HIGH}}(\alpha) \\ \mu_2 &= \mu_{\mathrm{HIGH}}(M) \cdot \mu_{\mathrm{HIGH}}(Q_c) \cdot \mu_{\mathrm{LOW}}(\alpha) \end{aligned}$$

The output of the TS fuzzy model is the weighted output of the local linear models

$$Z_0 = \frac{\mu_1 \cdot Z_{0_1} + \mu_2 \cdot Z_{0_2}}{\mu_1 + \mu_2} \tag{7}$$

The same rules are applied for the other parameters in Equation 6. In total the TS fuzzy model is determined by 68 parameters, namely 12 parameters to define the membership functions and 56 parameters to define the two local linear models. The parameters of the local linear models are computed by least squares optimization.

3.4 Results

The performance of the TS fuzzy model is measured with the variance accounted for (VAF) index and the root mean square error (RMSE). The VAF is a dimensionless index (independent of the measurement units) which gives an idea whether the trends are correctly captured. It, however, does not account for offset errors.

$$\mathrm{VAF} = 100\% \cdot \left[1 - \frac{\mathrm{var}(y - \hat{y})}{\mathrm{var}(y)}\right] \tag{8}$$

Therefore, also the RMSE index is included

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (y(i) - \hat{y}(i))^2} \tag{9}$$

Table 1 shows the comparison between the TS virtual sensor using two local linear models and a linear virtual sensor in off-line simulation. The results are averaged over all the simulation runs. The benefits of the TS approach in terms of improved accuracy are clear.

Table 1: Average performance of the linear model and the Takagi–Sugeno model.

	Linear model		TS fuzzy model	
	VAF	RMSE	VAF	RMSE
training	77.78	1.7633	99.95	0.0860
evaluation	78.47	1.5904	99.94	0.0816

4 Closed-loop simulation

The TS fuzzy model discussed in Section 3 is implemented in the Matlab/Simulink™ environment. The virtual sensor for normal acceleration is implemented as follows

$$n_z = \frac{1}{g} \cdot (\dot{w}_{b_{TS}} - u_b \cdot q_b + g \cdot cos\theta) \tag{10}$$

where $\dot{w}_{b_{TS}}$ denotes the estimation of the time derivative of the downward velocity by the TS fuzzy model. As has been discussed in Section 2, the virtual sensor is used to identify the failed signal in the duplex operation and to detect a failure in simplex operation.

An example of a closed-loop simulation is illustrated in Figure 8. The flight condition is denoted by 'o' in Figure 3 ($M = 0.70$, $Q_c = 180.2$ mbar). The pilot command is a block-shaped command of 90% of the maximum stick force, initiated at t = 1 sec and terminated at t = 6 sec. In these conditions the normal acceleration signal is compared to the pilot command and fed back to the elevator

through a proportional/integral feedback path. At t = 1 sec a drift of 0.1 $g \cdot sec^{-1}$ is added to the output of one of the real sensors (black, dashed). When the difference between the real sensor output and the output of the virtual sensor (black, continuous) exceeds a certain limit, the count rate switches from minus one to plus two. At t = 4.1 sec the count reaches the failure declaration value and the corresponding signal is latched. The voted output (gray, continuous) jumps from the mean value of the real signal outputs to that of the output of the last available healthy signal (gray, dashed). It is shown in Figure 8 that in nonlinear, closed-loop simulation the virtual sensor is accurate enough to be used for FDI of sensor failures.

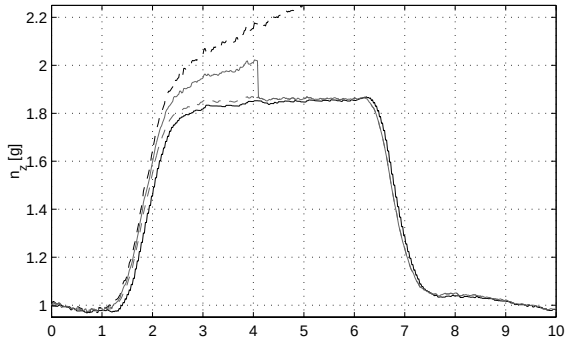

Figure 8: Nonlinear simulation illustrating the functionality of the virtual sensor based on a TS fuzzy model.

5 Conclusions

A fuzzy logic based virtual sensor has been proposed for the normal acceleration signal based on readings of dissimilar sensors. The application of virtual sensors in flight control systems makes it possible to distinguish between two real sensors in the case of a failure and therefore either increase the safety of the system or reduce the cost of the system because fewer hardware components may be used. In the latter case, part of the hardware redundancy is replaced by analytical redundancy.

The robustness of the virtual sensor with respect to variations in the aircraft weight and the centre-of-gravity need to be addressed in the near future. Final evaluation of the virtual sensor will take place with "pilot in the loop" simulations at the National Aerospace Laboratory (NLR) flight simulator.

6 Acknowledgements

This research was in part supported by the Brite/EuRam project "Affordable Digital Fly-by-wire Flight Control Systems for Small Commercial Aircraft" (ADFCS). The authors thank the partners of this project for their contributions and useful comments.

References

[1] Project Programme, "Affordable Digital Fly-By-Wire Flight Control System for Small Commercial Aircraft (ADFCS)", BE97-4098, 1997.

[2] Tischler M.B. (Ed.), "Advances in Aircraft Flight Control", Taylor & Francis Ltd, 1997.

[3] Patton R.J., P.M. Frank, and R.N. Clark, "Fault Diagnosis in Dynamic Systems, Theory and Applications. Prentice Hall, New York, 1989.

[4] Menke T.E. and P.S. Maybeck, "Sensor/actuator failure detection in Vista F-16 by multiple model adaptive estimation". IEEE Transactions on Aerospace and Electronic Systems, Vol. 31, No. 4, 1995.

[5] Patton R.J. and J. Chen, "A review of parity space approaches to fault diagnosis applicable to aerospace systems". Proceedings of the AIAA Guidance, Navigation and Control Conference, AIAA 92-4538, Hilton Head, pp. 1-10, 1992.

[6] Chow E.Y. and A.S. Willsky, "Analytical redundancy and the design of robust failure detection systems". IEEE Transactions on Automatic Control, Vol. 29, No. 7, pp. 603-614, 1984.

[7] Gopisetty S.M. and R.F. Stengel, "Detecting and identifying multiple failures in flight control systems", AIAA-98-4488, 1998.

[8] Schram G., S.M. Gopisetty, and R.F. Stengel, "A fuzzy logic logic–parity space approach to actuator failure detection and identification", 36th Aerospace Sciences Meeting & Exhibit, AIAA 98-1014, Reno, 1998.

[9] Isermann R., "Process fault detection based on modeling and estimation methods – A survey". Automatica, Vol. 29, No. 4, pp. 815-835, 1984.

[10] Frank P.M., "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy". Automatica, Vol. 26, No. 3, pp. 459-474, 1990.

[11] Leal R.R., P. Lane, and P.A. Payne, "Data Fusion and Artificial Neural Networks for Biomass Estimation", IEE Procedings D, Vol. 144, No. 2, pp. 69–72, 1997.

[12] Hanzevack E.L., T.W. Long, C.M. Atkinson, and M.L. Traver, "Virtual Sensors for Spark Ignition Engines Using Neural Networks", Proc. of the ACC, Vol. 1, pp. 669–673, Albuquerque, New Mexico, 1997.

[13] Babuška R., "Fuzzy Modeling for Control", Kluwer Academic Publishers, Boston, USA, 1998.

[14] Rosenberg K., "FCS Architecture Definition (Issue 1)", Deliverable 3.4, BE97-4098 ADFCS, 1998.

[15] Takagi T. and M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control", IEEE Transactions on Systems, Man, and Cybernetics, 15:116–132, 1985.

[16] Michalewicz Z., "Genetic Algorithms + Data Structures = Evolution Programs". Springer Verlag, New York, 1994