Development and Application of Sliding Mode LPV Fault Reconstruction Schemes for the ADDSAFE[1] Benchmark

Halim Alwi[a], Christopher Edwards[b]

[a] *Control Systems Research Group, Department of Engineering, University of Leicester, Leicester, LE1 7RH, UK. (e-mails: ha18@le.ac.uk)*

[b] *Center for Dynamical Systems and Control, University of Exeter, Exeter, Devon, 4EX 4QJ UK. (e-mails: C.Edwards@exeter.ac.uk)*

Abstract

This paper describes the development and the evaluation of a robust sliding mode observer fault detection scheme applied to an aircraft benchmark problem as part of the ADDSAFE project. The ADDSAFE benchmark problem which is considered in this paper is the yaw rate sensor fault scenario. A robust sliding mode sensor fault reconstruction scheme based on an LPV model is presented, where the fault reconstruction signal is obtained from the so-called equivalent output error injection signal associated with the observer. The development process includes implementing the design using AIRBUS's so-called SAO library which allows the automatic generation of flight certifiable code which can be implemented on the actual flight control computer. The proposed scheme has been subjected to various tests and evaluations on the Functional Engineering Simulator conducted by the industrial partners associated with the ADDSAFE project. These were designed to cover a wide range of the flight envelope, specific challenging manoeuvres and realistic fault types. The detection and isolation logic together with a statistical assessment of the FDD schemes are also presented. Simulation results from various levels of FDD development (from tuning, testing and industrial evaluation) show consistently good results and fast detection times.

Keywords: Sliding mode observer, Fault detection and diagnosis.

1. Introduction

The development of sliding mode schemes for fault tolerant control (FTC) and fault detection and diagnosis (FDD) has gained significant interest in the last decade – especially for aerospace applications. One of the reasons is due to its robustness properties with respect to certain classes of uncertainty – co-called matched uncertainty. Recent work in the area of fault tolerant control (see for example [1, 2]) has shown its potential through actual implementation on realistic engineering systems – despite the scepticism often associated with the so called 'chattering problem'. Recent work in the field of FDD under the auspice of an European project called 'Advanced Fault Diagnosis for Sustainable Flight Guidance and Control' (ADDSAFE) has attempted to take 'proof of concept' sliding mode schemes, to the next level by undertaking a formal industrial evaluation to assess its technical readiness.

In this paper, the development and application of a sliding mode observer for an ADDSAFE sensor fault benchmark problem will be presented. Specifically, an LPV based observer is proposed. The use of the LPV methodology is motivated by the need for the FDI system to be able to work robustly in a wide range of operating conditions. Recently the use of LPV concepts for fault detection have received a good deal of attention: see for example [3, 4, 5, 6, 7, 8, 9, 10]. The convenience of directly applying and extending linear techniques in an LPV framework, in order to deal with a wide range of operating conditions, is appealing. In certain situations, an LPV design is more applicable than direct nonlinear approaches due to complicated modelling issues [11]. With the development of generalized

[1] This work is supported by the EU (FP7-233815) grant: ADDSAFE (Advanced Fault Diagnosis for Sustainable Flight Guidance and Control).

methods for generating LPV models from a set of LTI systems, such as the recently proposed method in [12], LPV design has become more appealing.

In terms of sliding modes, although LPV based sliding mode control schemes were proposed a decade ago (see for example [13, 14]), until recently there were no LPV based sliding mode observers which had appeared in the literature. However recently higher order sliding mode methods have be employed to develop so-called interval observers [15], and an LPV observer has been proposed in a fault detection context [16]. The scheme first proposed in [16] considers pole placement methods to assign the design freedom. Subsequently the sensor fault reconstruction scheme employed in this paper, based on a more advanced approach, was developed in [17]. In [17, 18] a formal approach is introduced to create a robustness design, in order to deal with uncertainty or plant model mismatches, and the idea is to minimize the effect of uncertainty in the reconstruction of the faults by using LMIs methods. The fault reconstructions themselves are provided by the so-called output error injection signal associated with the observer. In [17, 18], the synthesis approach involves re-formulating the sensor fault problem as an actuator fault reconstruction problem. This is achieved by augmenting the original system with a filtered version of the potentially faulty measurements. By augmenting with only the potentially faulty measurements, the size of the augmented system is minimized which allows lower order system matrices to be considered in the LMI analysis during the observer design [17, 18]. This is different from the one in [19] where the plant is augmented with filtered versions of all the outputs.

The results which are presented in this paper are part of the activities undertaken within the ADDSAFE project [20]. The paper describes the development process from tuning, to testing and then industrial evaluations of the robust FDD scheme on the ADDSAFE Functional Engineering Simulator (FES) [21]. The results from the initial tuning and testing process are based on the simulation and verification FES which incorporates a high fidelity nonlinear aircraft model developed by AIRBUS [22]. This is intended to provide a design tuning and assessment tool for the proposed fault detection and diagnosis (FDD) schemes developed within the ADDSAFE consortium, and acts as a platform to test the robustness of the proposed schemes using a predetermined grid or sequence to cover a wide range of flight conditions and includes perturbations in the aircraft parameters. The FDI scheme has subsequently been evaluated in the Industrial Benchmarking and Verification FES which uses Monte Carlo methods to generate dispersions based on statistical distributions and covers all of the desired flight conditions, perturbations and uncertainties in the aircraft parameters under investigation.

The specific ADDSAFE benchmark problem which will be considered here is one of detecting and identifying faulty yaw rate sensors. A robust sliding mode reconstruction scheme based on a lateral axis LPV system representation is considered. The design utilizes LMIs to synthesize the gains of the observer which is selected to minimize the effect of uncertainty on the reconstruction signal. The fault reconstruction is obtained using the nonlinear output error injection signals from the robust sliding mode observer. The observer gain associated with the linear injection is parameterized by the LPV variable, while the gain associated with the nonlinear injection is fixed. In the paper the detection and isolation logic for the FDD scheme, which processes the reconstruction signals from the observer, is also discussed. This logic is used to declare the presence of a fault, and enables statistical assessment of the overall scheme. Results will be presented in this paper for all the various stages of the fault detections scheme's development, to show the efficacy of the proposed scheme and also to provide insight into the tuning process.

2. ADDSAFE

The Advanced Fault Diagnosis for Sustainable Flight Guidance and Control (ADDSAFE) project seeks to address the challenges associated with future 'sustainable' (cleaner, quieter, smarter and more affordable) aircraft [20, 23]. The aim of ADDSAFE is to demonstrate the applicability of advanced FDD methods to support the development of sustainable aircraft. It poses challenges to improve the FDD schemes which support new 'green technologies' thus allowing optimization of the aircraft structural design, improving aircraft performance, and reducing the environmental footprint

[24, 20, 23].

The FDD challenges considered in the ADDSAFE project consist mainly of actuator and sensor malfunctions and include the flight parameter management system, abnormal aircraft behaviour and the servo-loop actuator/sensor faults as described in Table 1. For further details and definitions of the benchmark scenarios considered in ADDSAFE see [20, 23]. The results in this paper concentrate on the monitoring of the Air Data Inertial Reference System (ADIRS) associated with the yaw rate sensors [20, 23].

Category	Fault scenario	Description
F1		**ADIRS Monitoring**
	r	yaw rate sensor fault
	α	angle of attack sensor fault
	n_z	Vertical load factor sensor fault
F2		**Aircraft abnormal configuration detection**
	liquid jamming	left inboard aileron liquid jamming at an offset position
	solid jamming	left inboard aileron solid jamming at an offset position
	disconnection	left inboard aileron disconnected from the hydraulic rod
F3		**Servo-loop actuator/Sensor fault**
	RIA liquid OFC	right inboard aileron liquid Oscillatory Failure Case (OFC)
	RIA solid OFC	right inboard aileron solid OFC
	LE liquid OFC	left elevator liquid OFC
	LE solid OFC	left elevator solid OFC
	RE runaway	right elevator runaway
	RE jamming	right elevator jamming at null position

Table 1: The ADDSAFE benchmark problem [25]

3. ADDSAFE Yaw Rate Sensor fault Benchmark Problem

The ADIRS yaw rate sensors in the nonlinear ADDSAFE model have triple redundancy, and the actual measurement used for the controller calculations is decided by data fusion processing known as 'consolidation' [26]: i.e. a consistency check (a vote) between the three individual sensors measurements. The result of the 'consolidation' process is a unique signal (the so-called triplex signal) sent to the controller for feedback control, while at the same time, the selected signal is compared to the three sensor measurements in order to detect and isolate any faulty measurements.

In the case when one of the redundant sensors in the air data and inertial reference system (ADIRS) is faulty, the consolidation process can reliably be used to detect and isolate the faulty sensor. However, if two sensors are faulty exactly at the same time, this can potentially become a problem as the 'consolidation' process may choose one of the two corrupted measurements as the triplex output [27]. This is the main facet of the scheme proposed in this paper – to introduce analytical redundancy to be able to correctly identify if the chosen triplex signal is truly fault free – especially in the event of two simultaneous sensor faults.

An erroneous triplex signal used by the controller could propagate through the controller and subsequently erroneous signals would be sent to the actuators and control surfaces. This may lead to abnormal or non-optimized aircraft behaviour, thus causing higher than normal drag and an unnecessarily high fuel burn in the longer term if not mitigated. Robust and reliable fault monitoring for the ADIRS signals, allows for better optimization of the aircraft structural configuration, therefore reducing weight and consequently fuel burn and carbon footprint.

The main idea presented in this paper is to differentiate between a situation in which one or two sensors are faulty by feeding the triplex (voted) signal used by the controller into a sliding mode observer and to reconstruct faults in the triplex signals (if present). If the reconstruction signal is 'zero', then the triplex signal does not contain a faulty signal, but when the reconstruction is 'nonzero', the triplex signal contains a fault and therefore the unselected signal is the fault free one. Furthermore not only should the presence of a fault be confirmed, the proposed scheme should also provide isolation of the faulty measurements.

Table 2 shows the benchmark scenarios associated with the ADIRS yaw rate fault monitoring problem. There are 6 different fault free manoeuvres, including lateral and longitudinal motions, and 10 different potential fault types (under cruise conditions). The fault types cover a wide range of possible fault scenarios from slow frequency low amplitude faults such as oscillatory and jam faults, to high frequency and large magnitude faults such as non-return to zero (NRZ), runaways and noise type faults. The fault cases also consider that the fault may occur on one or two of the measurements, differentiated by the abbreviation A (one faulty sensor) and B (two faulty sensors). The proposed scheme must adhere to a strict requirement[2] of very low probabilities of false or missed alarm, a low computational load (to enable it to be implemented in the Flight Control Computer (FCC)) and relatively fast fault detection times (especially in the case when two sensors are faulty).

fault type	no. of faulty sensors	amplitude or rate (%)	failure time (sec)
fault free			
cruise ADIRS	0	N/A	N/A
pitch protection 1	0	N/A	N/A
yaw angle mode	0	N/A	N/A
AoA protection 1	0	N/A	N/A
turn coordination 2	0	N/A	N/A
nose up	0	N/A	N/A
fault cases (manoeuvre: cruise)			
Oscill-A	1	1.25%	2
Oscill-B	2	3.75%	2
Jam-A	1	1.25%	2
Jam-B	2	5.00%	2
Runaway-A	2	2.50%	2
Runaway-B	2	50.00%	2
NRZ-A	1	320.00%	2
NRZ-B	2	320.00%	2
Noise-A	1	$\sigma^2 = 1.25\%$	2
Noise-B	2	$\sigma^2 = 50.00\%$	2

Table 2: The ADDSAFE yaw rate sensor fault benchmark problem - test cases [25]

Remark: Note that, throughout this paper, the magnitude or rate of the faults are expressed in terms of a percentage from the maximum admissible range. This is due to industrial confidentiality restrictions.

4. LPV Sliding Mode scheme for sensor fault reconstruction

Consider an LPV model of the plant represented by

$$\dot{x}_p(t) = A_p(\rho)x_p(t) + B_p(\rho)u(t) + M_p\xi_p(t, y, u) \quad (1)$$

$$y_p(t) = C_p x_p(t) + N_p f_o(t) + d_p(t) \quad (2)$$

where $A_p(\rho) \in \mathbb{R}^{n_n \times n_n}, B_p(\rho) \in \mathbb{R}^{n_n \times n_m}$ are parameter varying matrices, while the fixed matrices $C_p \in \mathbb{R}^{n_p \times n_n}, N_p \in \mathbb{R}^{n_p \times n_q}$ and $M_p \in \mathbb{R}^{n_n \times n_k}$ where $n_q < n_p < n_n$. Assume that the columns of

[2] Note, due to the industrial confidentiality constraints, explicit values for the allowable false or missed alarms, computational costs, and processing cycle and detection times are not divulged in this paper.

N_p are from the standard basis for $\mathbb{R}^{n_p}$ and N_p has full column rank. This models as an additive perturbation, the effect of faults on up to n_q out of the n_p measurements. The matrix C_p is assumed to have full row rank and it is also assumed that the inputs $u(t)$ and the output measurements $y_p(t)$ are available for use in the FDI scheme. The signal $f_o(t) \in \mathbb{R}^{n_q}$ represents the (unknown) vector of sensor faults: here $f_o \equiv 0$ is associated with fault–free conditions, while $f_o \neq 0$ indicates a fault exists associated with at least one of the measurements. Assume that, a possibly very conservative, upper bound on the size of f_o is known: specifically

$$\|f_o(t)\| < a(t) \tag{3}$$

where $a(t) : \mathbb{R}_+ \mapsto \mathbb{R}^+$ is a known function. The signal $d_p(t) \in \mathbb{R}^{n_p}$ represents a corruption of the true outputs and results in imperfect measurements even in the fault free case where $f_o \equiv 0$. A clear distinction is therefore to be made between ever-present corruptions represented by $d_p(t)$, (archetypically 'noise'), which is a factor taken into account in the controller design, and infrequent but serious corruptions/biases/drifts etc represented by $f_o(t)$. In this paper, it will be assumed that some of the sensors are fault-free while some are occasionally prone to exhibiting faults. This is a valid assumption since some sensors may be inherently less reliable or more vulnerable to external effects which may result in inaccurate readings. For design purposes, it will be assumed that the signal $d_p(t)$ is low pass in terms of its frequency content. This is important because as part of the observer design process, a distinction must be made between d_p and f_o so that the ever-present d_p do not cause false alarms. For this reason assume

$$d_p(s) = G_p(s)\phi(s) \tag{4}$$

for some stable transfer function matrix $G_p(s)$ with low pass filter characteristics. In (4) the driving signal $\phi(s)$ is assumed to be unknown but bounded.

The signal $\xi_p(t, y, u) : \mathbb{R}_+ \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_m} \to \mathbb{R}^{n_k}$ in (1) encapsulates the uncertainty in the system model and represents the plant model mismatch. It is assumed that $\xi_p(\cdot)$ is unknown but bounded such that $\|\xi_p(t, y, u)\| < b$ where the scalar b is known. In this paper it is assumed that the varying parameters ρ are known and range between known extremal values $\underline{\rho_i} < \rho_i < \overline{\rho_i}$ for $i = 1, 2, \ldots n_r$.

Because it was assumed the columns of N_p are formed from the standard basis for $\mathbb{R}^{n_p}$ by permutating the order of the outputs, without loss of generality, it can be assumed

$$\begin{aligned} y_p(t) &= \begin{bmatrix} y_{p,1}(t) \\ \hline y_{p,2}(t) \end{bmatrix} \begin{matrix} \}\text{fault free} \\ \}\text{prone to fault} \end{matrix} \\ &= \underbrace{\begin{bmatrix} C_{p,1} \\ \hline C_{p,2} \end{bmatrix}}_{C_p} x_p(t) + \underbrace{\begin{bmatrix} 0 \\ \hline I_{n_q} \end{bmatrix}}_{N_p} f_o(t) + \underbrace{\begin{bmatrix} d_{p,1}(t) \\ \hline d_{p,2}(t) \end{bmatrix}}_{d_p} \end{aligned} \tag{5}$$

where $C_{p,1} \in \mathbb{R}^{(n_p - n_q) \times n_n}$ and $C_{p,2} \in \mathbb{R}^{n_q \times n_n}$. This canonical form will be used as the basis for the observer design which follows. First filter the potentially faulty output set of measurements $y_{p,2}(t)$ to create a new state $z_f(t) \in \mathbb{R}^{n_q}$ according to

$$\dot{z}_f(t) = -A_f z_f(t) + A_f y_{p,2}(t) \tag{6}$$

where $-A_f \in \mathbb{R}^{n_q \times n_q}$ is a stable matrix. Typically A_f will take the form of a diagonal matrix with positive entries. Substituting for $y_{p,2}(t)$ from (5) into (6) yields

$$\dot{z}_f(t) = -A_f z_f(t) + A_f C_{p,2} x_p(t) + A_f f_o(t) + A_f d_{p,2}(t) \tag{7}$$

Augmenting the systems in (1) and (7) gives a system of order $(n_n + n_q)$ of the form

$$\underbrace{\begin{bmatrix} \dot{x}_p(t) \\ \dot{z}_f(t) \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} A_p(\rho) & 0 \\ A_f C_{p,2} & -A_f \end{bmatrix}}_{A(\rho)} \underbrace{\begin{bmatrix} x_p(t) \\ z_f(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} B_p(\rho) \\ 0 \end{bmatrix}}_{B(\rho)} u(t) + \underbrace{\begin{bmatrix} 0 \\ A_f \end{bmatrix}}_{F} f_0(t) + \underbrace{\begin{bmatrix} M_p & 0 \\ 0 & A_f \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} \xi_p(t) \\ d_{p,2}(t) \end{bmatrix}}_{\xi(t)} \tag{8}$$

Create a new 'output' from the augmented system in (8) of the form

$$\underbrace{\begin{bmatrix} y_{p,1}(t) \\ z_f(t) \end{bmatrix}}_{y_m(t)} = \underbrace{\begin{bmatrix} C_{p,1} & 0 \\ 0 & I_{n_q} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x_p(t) \\ z_f(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} d_{p,1}(t) \\ 0 \end{bmatrix}}_{d(t)} \tag{9}$$

In the above, the outputs y_m are a combination of the actual and the filtered outputs, and represents perfect measurements of the *augmented states* $x(t)$ (save for the ever present disturbance $d(t)$). The system represented in (9) and (8) is in the form of an actuator fault reconstruction problem. Also, note that by construction, $\mathrm{rank}(CF) = n_q$ where C and F are defined in (9) and (8). This will allow first order sliding mode methods to be adopted.

As argued in Chapter 6 of [1], the augmented system (8)-(9) can be transformed into an output canonical form $(A(\rho), F, C) \mapsto (\tilde{A}(\rho), \tilde{F}, \tilde{C})$ such that the augmented outputs and the fault distribution matrix has the following structure

$$\tilde{C} = \begin{bmatrix} 0 & I_{n_p} \end{bmatrix} \quad \text{and} \quad \tilde{F} = \begin{bmatrix} 0 \\ \hline 0 \\ A_f \end{bmatrix} \left.\vphantom{\begin{matrix} 0 \\ A_f \end{matrix}}\right\} F_2 \tag{10}$$

where $F_2 \in \mathbb{R}^{n_p \times n_q}$. In the coordinate system associated with $(\tilde{A}(\rho), \tilde{F}, \tilde{C})$ the observer for system (8)-(9) considered in this paper has the structure

$$\dot{\hat{x}}(t) \;=\; \tilde{A}(\rho)\hat{x}(t) + \tilde{B}(\rho)u(t) - \tilde{G}_l(\rho)e_y(t) + \tilde{G}_n\nu(t) \tag{11}$$

$$\hat{y}(t) \;=\; \tilde{C}\hat{x}(t) \tag{12}$$

where the state estimate $\hat{x} \in \mathbb{R}^{n_n+n_q}$ and the gain matrices $\tilde{G}_l(\rho)$ and $\tilde{G}_n \in \mathbb{R}^{(n_n+n_q)\times n_p}$. The output estimation error $e_y(t) = \hat{y}(t) - y_m(t)$. The term $\nu(t) \in \mathbb{R}^{n_p}$ is a nonlinear injection signal, used to induce a sliding motion ([28]), is given by

$$\nu = -\mathcal{K}(t, y, u, \rho)\frac{P_o e_y}{\|P_o e_y\|} \tag{13}$$

where $P_o \in \mathbb{R}^{n_p \times n_p}$ is a symmetric positive definite (s.p.d) matrix which satisfies

$$\tilde{A}_{22}^{\mathrm{T}}P_o + P_o\tilde{A}_{22} = -I_{n_p} \tag{14}$$

In the Lyapunov equation (14) the matrix $\tilde{A}_{22} \in \mathbb{R}^{n_p \times n_p}$ is chosen as fixed and Hurwitz. The design freedom associated with the observer are the gains $G_l(\rho), G_n$ the scalar $\mathcal{K}(t, y, y, \rho)$ and the Hurwitz matrix $\tilde{A}_{22}$. The modulation scalar $\mathcal{K}(t, y, u, \rho)$ is any function chosen so that

$$\mathcal{K}(t, y, u, \rho) > \|A_f\|a(t) + \eta_0 \tag{15}$$

where η_0 is a positive scalar and $a(t)$ is the bound on the fault from (3). From (9) and (12) the output estimation error

$$e_y(t) = \hat{y}(t) - y_m(t) = Ce(t) - d(t) \tag{16}$$

where the state estimation error $e(t) = \hat{x}(t) - x(t)$. Subtracting (8) from (11) yields

$$\dot{e}(t) = \tilde{A}(\rho)e(t) - \tilde{G}_l(\rho)e_y(t) + \tilde{G}_n\nu(t) - \tilde{F}f_o(t) - \tilde{M}\xi(t) \tag{17}$$

The design freedom in the observer will be selected to force the output estimation error $e_y(t)$ to zero in finite time, to induce a stable sliding motion on the surface

$$\mathcal{S} = \{e \in \mathbb{R}^{n_n+n_q} : e_y = 0\} \tag{18}$$

If the measurement corruption term $d_p(t) \equiv 0$, and if the estimation error is constrained to the surface in (18), the output of the observer exactly follows the plant output. The idea is that, during sliding, the signal $f_o(t)$ will be estimated using the concept of 'equivalent output injection' which represents the signal $\nu(t)$ must take on average to maintain sliding [29]. As argued in [17] the fixed gain $\tilde{G}_n$ can be parameterized in terms of a design matrix $L \in \mathbb{R}^{(n_n+n_q-n_p)\times n_p}$ given by

$$\tilde{G}_n = \begin{bmatrix} -L \\ I \end{bmatrix} \tag{19}$$

Furthermore L has the special structure

$$L = \begin{bmatrix} L_1 & 0 \end{bmatrix} \tag{20}$$

with $L_1 \in \mathbb{R}^{(n_n+n_q-n_p)\times(n_p-n_q)}$. Once L_1 and hence L is determined, the fixed gain $\tilde{G}_n$ from (19) is determined. The gain $\tilde{G}_l(\rho)$ from (11) (which also depends on L) is parameterized by the fixed Hurwitz matrix $\tilde{A}_{22}$. Once these parameters are chosen, the matrix $G_l(\rho)$ can described explicitly in terms of a formula involving L, $\tilde{A}_{22}$ and sub-matrices extracted from $A(\rho)$. For details see [17]. The remaining design freedom is associated with the reconstruction signal and is given by

$$\hat{f}_o = W\nu_{eq} \tag{21}$$

where the fixed matrix $W \in \mathbb{R}^{n_q \times n_p}$. The philosophy used in this paper is to synthesize the gains L_1 (and thus $\tilde{G}_n$, $\tilde{G}_l(\rho)$) and W so that the $\mathcal{L}_2$ gain between the uncertainty (ξ, d_p) and the fault estimation error $\hat{f}_o - f_o$ is less than a scalar γ. As argued in [17], this can be posed as an LMI optimization problem. For design purposes, it is assumed that $d(t)$ from (9) emerges from the exogenous model

$$\dot{d}(t) = -a_f d(t) + a_f \phi(t) \tag{22}$$

where $a_f \in \mathbb{R}^+$ and the unknown signal $\phi(t)$ is assumed to be bounded. The dynamics in (8)-(9) are augmented with (22) and integrated into the LMI optimization framework (as unobservable but detectable dynamics). To provide additional design freedom and to provide a 'tuning knob' to tradeoff robustness with respect to $\xi(t)$ and amplification of the output disturbance $d(t)$, a weighting matrix

$$\Delta = \text{diag}(\delta_1, \delta_2) \tag{23}$$

is introduced where δ_1=diag$(\delta_{1,1} \ldots \delta_{1,n_k+n_q})$ and δ_2=diag$(\delta_{2,1} \ldots \delta_{2,n_p})$. These parameters introduce a balance between the requirement for the fault estimation to be insensitive to uncertainty in the plant $\xi(t, y, u)$, and insensitive to the imperfect measurements $\phi(t)$. For details see [17].

As in [10], it will be assumed that (8) and (9) can be represented as a polytopic system where the range of admissible ρ corresponds to a polytope $\mathcal{P}$ with vertices $\omega_1, \omega_2, \ldots \omega_{n_\omega}$ where $n_\omega = 2^{n_r}$. As discussed in [30], the affine LPV system matrices $(A(\rho), B, C, F)$ in (8) and (9) can be replaced by $(A(\omega_i), B, C, F)$ and the LMIs can be solved for all the vertices of the polytopic system.

4.1. Design Summary

A summary of the design method for sensor fault reconstruction is as follows:

1. Re-order the system states such that the sensors considered prone to faults are in the lower half of the output vector as given in (5).
2. Augment the plant model from (1) with the filtered version of the sensors prone to faults to create the system in (8). This involves the selection of the matrix A_f as in (9). Now the system order of the augmented system becomes $n_n + n_q$.
3. Create the augmented plant outputs from the 'new' filtered outputs, together with the plant measurements considered to be fault free.
4. Change the coordinates of the augmented system so that the output distribution matrix is as given in (10).

5. Design L in (20) using the LMI optimization scheme described in [17]. The user inputs which help specify the LMIs which will be solved, are the LPV plant model, the uncertainty matrix M and the design parameters Δ and a_f which are all defined in (22)-(23). The output of the LMI optimization process is the gain L_1 in (20) and the weight W from (21) which will be used to provide the estimate of the fault.
6. Once L has been obtained, the nonlinear gain $\tilde{G}_n$ is given in (19).
7. Choose the stable design matrix $\tilde{A}_{22}$ and solve the Lyapunov equation (14) to obtain P_o.
8. Once L and $\tilde{A}_{22}$ have been selected, the observer gain $\tilde{G}_l(\rho)$ can be explicitly computed using the formula given in [17].
9. The gains in the original coordinates are given by reversing the transformation used to achieve the canonical form (10).
10. The modulation gain $\mathcal{K}(t, y, u, \rho)$ must be chosen to satisfy (15).
11. The reconstruction signal as defined in (21) is then given by

$$\hat{f}_o = W\nu_{eq} \tag{24}$$

12. The 'equivalent output injection' signal $\nu_{eq}(t)$ is given by (13).

Remark: In this paper a unit vector approach (13) is employed to create the sliding motion. In implementation, this will be approximated by a continuous sigmoidal function [1]. However, at the expense of greater complication, a smooth super-twisting structure can be employed [18, 31]. Here the simplicity of the sigmoid is preferred to minimise the computational load.

5. ADDSAFE Benchmark Design

5.1. ADDSAFE benchmark model

The ADDSAFE aircraft benchmark model, provided by AIRBUS [27, 20, 23, 22] (Figure 1) represents a high fidelity nonlinear model of a rigid body twin engine, civil commercial aircraft, with highly detailed nonlinear actuator and sensor models, together with a complex aerodynamic database. The model (which includes pilot side stick and pedals inputs, and a flight control law to provide closed-loop nonlinear aircraft autopilot capabilities) is highly representative of the actual aircraft flight physics with realistic handling qualities [27].

5.2. ADDSAFE LPV model

For design purposes, an LPV model approximating the nonlinear benchmark system will be considered. The LPV plant used as the basis of the observer in this paper is the lateral model developed in [32], which has been derived from the full nonlinear benchmark model as part of the ADDSAFE project. Here, the lateral states are taken as

$$x_p = [\phi \ V_y \ p \ r]^{\mathrm{T}} \tag{25}$$

which represents roll angle, velocity in the y-axis, roll rate and yaw rate, while the measured outputs are

$$y_p = [\phi \ \beta \ p \ r]^{\mathrm{T}} \tag{26}$$

which represents roll angle, side-slip angle, roll rate and yaw rate. The lateral control surfaces are the rudder, 4 ailerons and 8 spoilers. Note that since the state $V_y(t)$ is not one of the measured outputs of the aircraft and in order to obtain a fixed output distribution matrix C (as in equation (2)), $V_y(t)$ must be created (algebraically) from the measurements. Here $V_y(t)$ is approximated in real-time from the measurements of $\beta(t)$ and $V_{tas}(t)$ using the expression

$$V_y(t) = V_{tas}(t) \sin(\beta(t)) \tag{27}$$

Figure 1: ADDSAFE benchmark model

The LPV system matrices associated with the models from [32] are given by

$$A_p(\rho) = A_{p,0} + A_{p,1}\rho_1 + A_{p,2}\rho_2 + A_{p,3}\rho_3 + A_{p,4}\rho_4 \tag{28}$$

$$B_p(\rho) = B_{p,0} + B_{p,1}\rho_1 + B_{p,2}\rho_2 + B_{p,3}\rho_3 + B_{p,4}\rho_4 \tag{29}$$

See the Appendix for details of the entries of these matrices.
In [32] the LPV parameters ρ, chosen to describe the variation of the aircraft dynamics, are mass (m), centre of gravity in the x-direction (X_{cg}), altitude h, and conventional airspeed V_{cas}. Theoretically the LPV model is only valid for variations of ±10% for mass m and position of centre of gravity X_{cg}, ±25% for altitude h and ±19% for conventional airspeed V_{cas} from the trim point

$$\left[\begin{array}{cccc} m(0) & X_{cg}(0) & h(0) & V_{cas}(0) \end{array}\right]^{\mathrm{T}} = \left[\begin{array}{cccc} 200,000(kg) & 30\%(\text{mac}) & 20,000(ft) & 290(kt) \end{array}\right]^{\mathrm{T}}$$

As employed in the observer design process, the LPV parameters ρ are

$$\left[\begin{array}{cccc} \rho_1 & \rho_2 & \rho_3 & \rho_4 \end{array}\right] := \left[\begin{array}{cccc} \bar{m} & \bar{X}_{cg} & \bar{h} & \bar{V}_{cas} \end{array}\right] \tag{30}$$

which represent normalized parameters varying in the interval $\left[\begin{array}{cc} -1 & 1 \end{array}\right]$ created directly from the substitution

$$\bar{m} = \frac{m - 200,000}{20,000}, \ \bar{X}_{cg} = \frac{X_{cg} - 0.3}{0.03}, \ \bar{h} = \frac{h - 20,000}{5000}, \ \bar{V}_{cas} = \frac{V_{cas} - 290}{30}$$

where m, X_{cg}, h, V_{cas} vary in the range of

$$m(kg) \in \left[180,000 \;\; 220,000\right], X_{cg} \in \left[0.27 \;\; 0.33\right], h(ft) \in \left[15,000 \;\; 25,000\right], V_{cas}(kt) \in \left[260 \;\; 320\right] \tag{31}$$

5.3. LPV Observer Design

For the faulty yaw rate $r(t)$ sensor scenario considered in this paper, the original lateral states of the LPV model $x_p(t)$ presented in (25) are already in the canonical form of (5) since the sensor which will be monitored for potential faults $r(t)$ is at the bottom of the output vector. In the notation of (5)

$$x_p(t) \;\; = \;\; \left[\begin{array}{c} y_{p,1}(t) \\ \hline y_{p,2}(t) \end{array}\right] = \left[\begin{array}{c} \phi(t) \\ V_y(t) \\ p(t) \\ \hline r(t) \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}\phi\\V\\p\end{array}}\right\}\text{fault free} \\ \}\text{prone to fault} \end{array} \tag{32}$$

and in this specific problem $n_n = 4, n_p = 4, n_q = 1$. The scalar variable A_f (in this case) which defines the output filter shown in (8), has been chosen as $A_f = 0.01$. The new augmented system output in (9) is

$$\underbrace{y_m(t) = \begin{bmatrix} y_{p,1}(t) \\ z_f(t) \end{bmatrix}}_{y_m(t)} = \underbrace{\left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \end{array}\right]}_{C} \underbrace{\begin{bmatrix} x_p(t) \\ z_f(t) \end{bmatrix}}_{x(t)} \tag{33}$$

In the output canonical form described in (10), the augmented states become

$$\tilde{x}(t) = \left[\begin{array}{c|cccc} r(t) & \phi(t) & V_y(t) & p(t) & z_f(t) \end{array}\right]^{\mathrm{T}} = \left[\begin{array}{c} x_1(t) \\ \hline x_2(t) \end{array}\right] = \left[\begin{array}{c|c} x_1(t) & y^{\mathrm{T}}(t) \end{array}\right]^{\mathrm{T}} \tag{34}$$

where $y_m(t) = \tilde{C}\tilde{x}(t)$ and $\tilde{C} = \begin{bmatrix} 0 & I_4 \end{bmatrix}$. Here it is assumed that the uncertainty is dominant in the roll rate and yaw rate channels and therefore

$$M = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \end{bmatrix}^{\mathrm{T}} \tag{35}$$

The 'design scalar' associated with the sensor measurement a_f from (22) has been chosen as $a_f = 25$. It is assumed that the effect of uncertainty is small compared to the effect of the disturbance and the design scalars δ_1 have been chosen smaller than δ_2 from (23). In this particular design $\delta_1 = diag(0.01, 0.0001)$. Other than the fault f_o, it is assumed that the corruption to the state z_f is negligible i.e. $d_{p,2}(t) = 0$. The parameter δ_2 has been chosen as $\delta_2 = diag(1, 60, 1, 0)$. Note that the second parameter in δ_2 corresponds to V_y which has magnitude (in the unit kts) significantly larger than the other states (whose units are rad) and is therefore weighted more heavily. The last parameter in δ_2 corresponds to the zero in $d(t)$ from (9). Using these parameters, solving the LMIs described in [17], yields $L = \begin{bmatrix} 0.0355 & 0.0277 & -0.0000 & 0 \end{bmatrix}$ from (20) and $W = \begin{bmatrix} -0.0413 & 0.0005 & 0 & 100.0000 \end{bmatrix}$ from (21). The stable design matrix $\tilde{A}_{22}$ has been chosen as $\tilde{A}_{22} = diag(-3.1, -3.2, -3.3, -3.4)$, which yields as a consequence of (14) the matrix $P_o = diag(0.1613, 0.1563, 0.1515, 0.1471)$. In the original coordinates, the fixed gain G_n is given by

$$G_n = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ -0.0710 & -0.0554 & 0.0001 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

and the varying gain $G_l(\rho)$ is given by

$$G_l(\rho) = G_{l_0} + G_{l_1}\rho_1 + G_{l_2}\rho_2 + G_{l_3}\rho_3 + G_{l_4}\rho_4 \tag{36}$$

Details of the individual components G_{l_i} are given in the Appendix. For implementation purposes, the 'equivalent output injection' signal $\nu_{eq}(t)$ can be approximated online to any degree of accuracy by replacing (13) with the smooth approximation

$$\nu_{eq}(t) = -\mathcal{K}(t, y, u, \rho)\frac{P_o e_y}{\|P_o e_y\| + \delta} \tag{37}$$

where δ is a small (design) scalar. The scalar δ is used to smooth the sign function and provide a degree of freedom in terms of the tradeoff between robustness and chattering [33]. Here, δ has been chosen as 0.01 and the modulation gain $\mathcal{K}(t, y, u, \rho)$ was chosen as $\mathcal{K}(t, y, u, \rho) = diag(10, 10, 10, 13)$.

Note that during the industrial evaluation, the fault estimate/reconstruction signal (21) is filtered by a simple first order low pass filter with a time constant of 0.05. The filter is required to remove high frequency signal components in order to provide reliable fault detection and isolation in the presence of system noise.

6. Overall FDD configuration

Figure 2 shows the overall structure of the proposed FDD scheme. The idea is to complement the existing consistency (voting) scheme by adding a further analytical redundancy check (without requiring any extra hardware redundancy), which exploits the sliding mode fault reconstruction signal to determine if the voted signal ($r_{triplex}$) is fault free, and if not, then to isolate the faulty sensor.

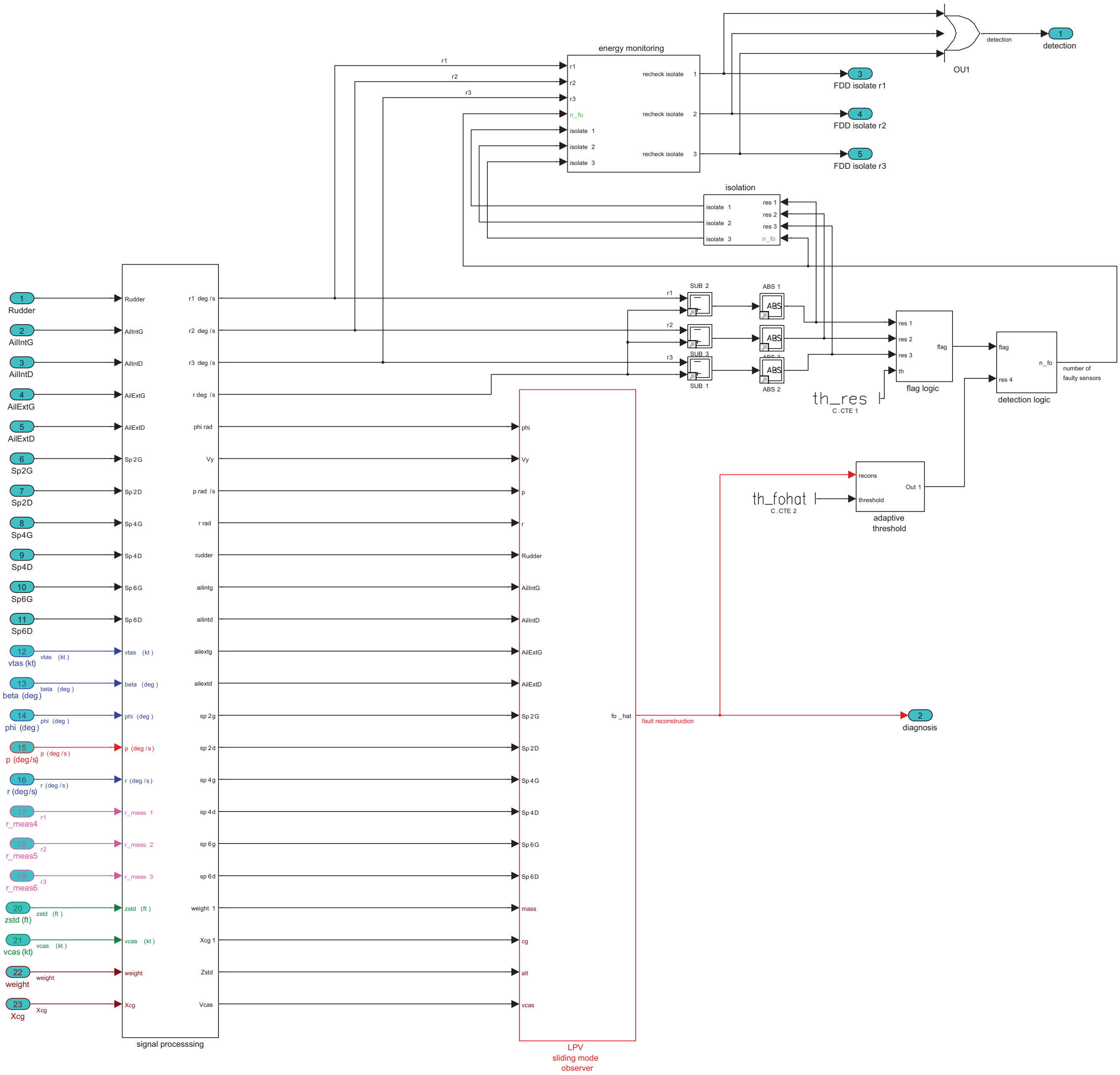


Figure 2: Overall FDD scheme

6.1. *Signal Processing*

The signal processing block consists of filters to ensure that the input signals are smooth. The block also contains unit conversions from degrees to radians and other signal conversions (e.g. V_y is calculated from V_{tas} and β). The block also computes the LPV varying parameters.

6.2. *Core Sliding Mode Observer*

The core sliding mode observer consists of a model of the plant in LPV form, together with the time-varying observer gain $G_l(\rho)$ and the fixed gain G_n given in (11). The observer block also contains a set of 5 integrators for estimating the states to provide the output error signals $e_y(t)$ used in the observer 'feedback' calculations. The discontinuous signum function in the sliding mode observer is replaced by the sigmoid function in (37) to produce smooth signals used to create the 'output error injection signal'. This signal is also used for the reconstruction of the faults present in the system.

6.3. Fault Detection and Isolation Logic

The other processing blocks, outside of the core sliding mode observer, implement the fault detection and isolation logic. The idea is to use the fault reconstruction signal $\hat{f}_o(t)$ provided by the core sliding mode observer to decide if any sensors are faulty. In the case when a fault is detected, the logic also decides if faults are present in one sensor or two sensors (Case A or Case B).

6.3.1. Flag Logic

The primary function of the flag logic is to detect if a sensor fault is present. It is primarily based on the 'consolidation' process, currently used in service [25]. The 'triplex' signal from the voting/selection process is compared with the three independent individual sensor measurements creating three error signals. Any deviation of these error signals above a small fix threshold will trigger a boolean logic signal called 'flag' to take the value 1 to indicate the presence of a fault. Although a fault can be detected, there is no information about which of the three sensor measurements is faulty (i.e isolation): this will be handled by a separate dedicated block.

6.3.2. Detection Logic

The detection logic block uses the boolean 'flag' signal and the boolean signal from the adaptive threshold to decide if there is one or two faulty sensors (the default is zero). A series of logic checks are used to confirm if one or two sensors are faulty, and the integer output signal, called n_{fo}, represents the number of faulty sensors (0,1 or 2). The n_{fo} signal is used to isolate any faulty sensors. A small confirmation time of 1 sec is exploited when $n_{fo} = 1$, and one of 0.1 sec when $n_{fo} = 2$.

The implemented detection and diagnosis logic is given in Table 3. The voted signal ($r_{triplex}$) is compared with all the three redundant signals from the ADIRS to generate three residual signals. Any nonzero residual is used as a fault detection flag, while the fault reconstruction signal is used to determine if one or two sensors are faulty – therefore confirming whether the $r_{triplex}$ signal is fault free. The benefit of using this approach is that only one (rather than three) observer needs to be implemented in order to keep the computational load (on the flight control computer) as low as possible. From an implementation point of view, this configuration has a modular structure, *and only complements the existing validated FDD scheme.* Therefore retrofitting the proposed scheme into the existing FDD architecture is possible without requiring any major changes. From a certification point of view, this modular configuration is also beneficial as it can be validated independently from the existing FDD.

residual	SMO reconstruction	overall detection	estimated number of faulty sensors n_{fo}
≈ 0	≈ 0	0	0
$\neq 0$	≈ 0	1	1
$\neq 0$	$\neq 0$	1	2

Table 3: Detection logic and diagnosis

6.3.3. Adaptive Threshold

For added robustness and correct isolation of faulty signals (a decision between one or two faulty sensors) especially in the presence of aggressive manoeuvres, an adaptive threshold is used. The output of this block is used by the detection and isolation logic to provide the correct isolation of the faulty sensor. The adaptive scheme provides a varying threshold level based on the maximum value of any of the (scaled) control surface deflections. The dynamics of the adaptive threshold is represented by a first order filter given by

$$\dot{T}_{h_a} = -0.6T_{h_a} + 1.2max(max(\frac{1}{12}|(\delta_{ail}, \delta_{sp})|, \frac{1}{2}|\delta_{rud}|)) \tag{38}$$

where T_{h_a} is the adaptive threshold level and $\delta_{ail}, \delta_{sp}, \delta_{rud}$ are the aileron, spoilers and rudder deflections respectively. A boolean signal is created by comparing the adaptive threshold to the (absolute) reconstruction signal from the observer. In the event of a fault occurring, and if the reconstruction signal is above the threshold, then it can be concluded that 2 sensors are faulty and a boolean signal is set to 1, otherwise only one sensor is faulty (and the boolean is set to 0).

6.3.4. Isolation Logic

The isolation logic utilizes the error signal (obtained from comparing the triplex signals and the three measurements in the flag logic block) and the n_{fo} signal from the detection logic block. The idea is that if the estimated number of faulty measurements $n_{fo} = 1$, then the existing 'consolidation' process has selected the two correct sensors to create the triplex signal sent to the controller, and therefore the one faulty sensor has been correctly isolated by the existing mechanism. However, when two sensors are faulty the existing 'consolidation' process may select the wrong/faulty signal and create an erroneous triplex signal. In this situation, the observer reconstruction signal will be above the adaptive threshold and this should indicate that two sensors are faulty ($n_{fo} = 2$) and that the 'consolidation' process has selected an incorrect choice for the triplex signal. Boolean signals are then used to correctly identify the faulty sensor.

6.3.5. Energy Monitoring

A dedicated separate isolation block for the case of a noisy sensor fault is required for added robustness. This is due to the fact that the existing 'consolidation' process may not produce a clear and fast isolation when two sensors are faulty. However the new isolation logic described above also cannot provide a fast definite isolation of the correct faulty sensor. Instead, the idea here is to independently check if any of the sensor measurements have an abnormal noise signature, and then override the isolation boolean signals from the isolation logic block. A 'moving window' of 30 samples created from the (numerical) derivative of each of the yaw rate sensor measurements is used together with a fixed threshold to 'trip' a boolean signal from 0 to 1 (and therefore to isolate the correct faulty noisy sensor). If any one of the three boolean signals from the energy monitoring block is 1, then it will override the isolation boolean flag from the isolation logic block.

7. ADDSAFE FES

For validation purposes, the ADDSAFE nonlinear aircraft model has been embedded within the so-called functional engineering simulator (FES). This is a software based validation tool developed by DEIMOS [21, 23]. The ADDSAFE-FES simulation environment runs under MATLAB/SIMULINK and XML and has been developed specifically for the assessment of the FDD algorithms developed within the ADDSAFE project. The tools provide the necessary 'infrastructure' to run various kinds of simulation tests (including parametric runs and intensive Monte-Carlo campaigns to cover a wide range of the flight envelope, conditions and manoeuvres).

Two distinct versions of the ADDSAFE-FES have been developed for use in the project. The simplest version is intended to provide simulation and verification tools to facilitate the tuning of the FDD methods. The second more sophisticated version is the industrial benchmarking and validation FES which has been used to thoroughly validate and evaluate the selected FDD designs. The differences between the two versions of the ADDSAFE-FES are given in Table 4 [21]. The key difference is that the Monte Carlo approach is only available in the industrial FES.

Features	simulation and verification FES	Industrial Benchmarking and Validation FES
FDD benchmark model	✓	✓
pre-defined scenarios	✓	✓
visualization of simulation outputs	✓	✓
failure detection metrics	✓	✓
Monte Carlo simulation	×	✓
Simulation campaign mode	✓	✓
Automatic batch simulations	✓	✓

Table 4: simplified vs. Industrial FES comparison [21]

7.1. Simulation and Verification FES (Parametric FES)

The simplified simulation and verification FES employs parametric sweeps of the flight conditions based on fixed grid parameter dispersions instead of random Monte Carlo dispersions [21, 23]. The results are obtained for particular flight manoeuvres (depending on the fault type [25, 23]) and include variations in the operating conditions (altitude (h), speed (V_{cas}), mass (m), center of gravity (X_{cg})), uncertainty in the aerodynamic coefficients of the roll pitch and yaw moments (C_l, C_m, C_n) and the coefficients of forces in the X, Y, Z axes (C_X, C_Y, C_Z), together with imprecise knowledge of altitude h, speed V_{cas}, mass m, and center of gravity X_{cg}. The simulation campaign sweeps the entire flight envelope using a combination of grid parameters given in Table 5 (which gives $3 \times 3 \times 4 \times 3 = 108$ possible combinations). For each of these combinations, three additional perturbations based on the minimum, nominal and maximum levels of uncertainties in the aerodynamic coefficients and sensor measurements have also been tested. This results in a total of 324 simulation runs for each type of fault (including fault free runs).

grid	dispersion			
param	1	2	3	4
$m(\times 1000Kg)$	120	180	233	
$X_{cg}(\%)$	0.17	0.3	0.41	
$h(\times 1000ft)$	8	18	28	38
$V_{cas}(kts)$	160	220	300	

sequence	dispersion		
parameters	min	nom	max
$\delta C_X, \delta C_Y, \delta C_Z(\%)$	-5	0	5
$\delta C_l, \delta C_m, \delta C_n(\%)$	-5	0	5
$\delta m(\%)$	-10	0	-10
$\delta X_{cg}(\%)$	-10	0	-10
$\delta h(\%)$	-10	0	-10
$\delta V_{cas}(\%)$	-10	0	-10

Table 5: Grid and sequence for parameter dispersion [21]

7.2. Industrial Benchmarking and Validation FES (Industrial FES - Monte-Carlo)

For the industrial FES, a broadly similar setup was considered involving variations in operating conditions, perturbations and uncertainty. The significant difference compared to the simplified FES is, here, the variations are associated with a Monte-Carlo setup which ensures a random sweep of the entire flight envelope to assess the robustness and performance. A total of 2200 runs are employed comprising 200 runs for six different fault free manoeuvres (6 × 200 =1200 runs), and 1000 runs for the faulty case. This assessment constitutes the industrial evaluation for the ADDSAFE benchmark problems. It should be noted that although the simpler FES was used for tuning purposes once a specific design was submitted for Monte Carlo based assessment, no further modification of the design, could be made.

Remark: Note that due to the large amount of data generated during the simulations, only the most significant plots will be presented in this paper. All the fault types considered are from the scenarios associated with the ADDSAFE benchmark problem as defined in [25, 23]. Further details about the FES can be found in [23, 21].

7.3. Design, testing, evaluation and validation process

In this paper different sets of results will be presented (see Table 6) to demonstrate the evolution of the design from initial checking (using the ADDSAFE model) to tuning (in the simplified FES) and finally the industrial benchmarking evaluation supervised by the industrial partners within the ADDSAFE consortium (using the industrial FES).

Block type	Single run	Simulation and Verification FES (parametric)	Industrial Benchmarking and Validation FES (Monte-Carlo)	'Maturation' tests
SIMULINK	✓	x	x	x
SAO	x	✓	✓	✓

Table 6: Level of tests conducted and block used

The single run tests provide a basic first level evaluation and do not include any uncertainties or random variations in the flight conditions. This controlled single run test incorporating known faults, has been used to check that the observer was working correctly. The second level of tests, employing the Simulation and Verification FES with grid parametric dispersion, was used for tuning purposes to ensure that the design under investigation provides good performance over a wide range of the flight envelope, as a precursor to the industrial benchmarking and validation. The final definitive tests are conducted on the Industrial Benchmarking and Validation FES with full Monte-Carlo dispersion.

This paper also presents 'maturation' tests, which represent additional tests which were not considered during the industrial evaluation process. These tests have been carried out in order to ensure that the proposed FDD design is robust and its performance is maintained throughout the flight envelope and for various manoeuvres, different levels of faults (in terms of magnitude and rate) occurring at different times.

7.4. SAO graphical symbol library

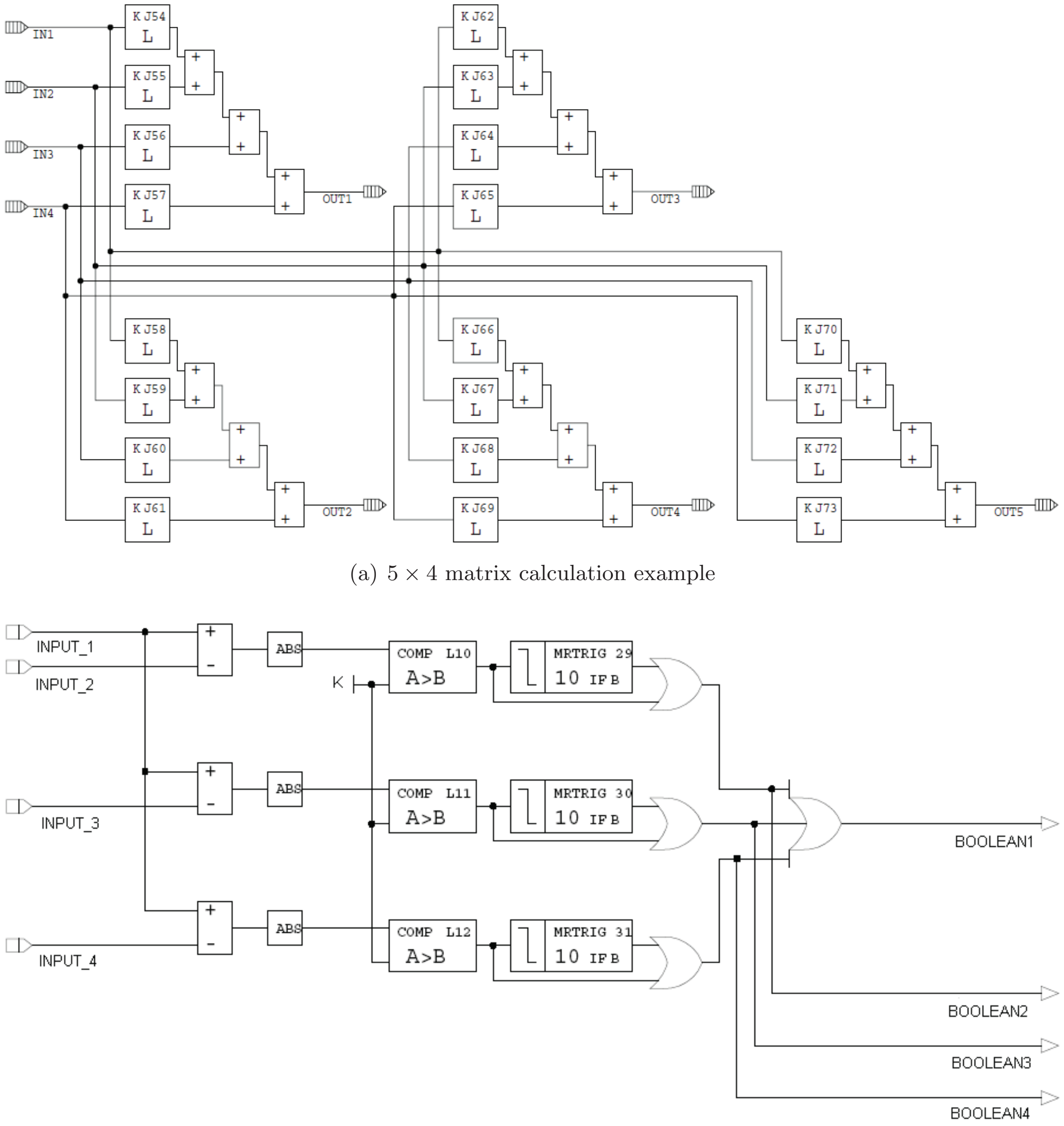
As demonstrated in Table 6, for the industrial FES, the FDD scheme has been implemented using industrially approved 'SAO' blocks provided by AIRBUS [23]. The SAO blocks represent a graphical symbol library (very much in the manner of SIMULINK blocks). However, they represent a very much more limited set of mathematical operations which are compliant with industrial coding standards. These blocks allow software codes to be generated automatically, in a form which can be implemented on the flight control computers used by AIRBUS. The limitation imposed by using SAO blocks guarantees that the schemes have a complexity level which allows them to be implementable on the actual flight control computer. An example of these blocks is presented in Figure 3, where for example, it can be seen that matrix multiplication, which can be represented by a single block in SIMULINK, must be broken down to a series of scalar multiplication and addition operations. This gives some insight into the way in which the FDD schemes have been implemented for the industrial validation.

8. Simulation Results

Remark: Note that due to industrial confidentiality constraints, some of the plots which follow are expressed in terms of percentage of the admissible range. Also, for the same reason, certain specific values of the aircraft parameters are not given.

8.1. Single run simulation results

The results in this section present the initial single run tests which provide a preliminary first check of the observer implementation. In these tests there are no uncertainties or variations in the flight conditions. Many different tests have been conducted using these single run setups in order to tune the observer, and to ensure the best performance possible using the nonlinear ADDSAFE model. Figures 4 and 5 represent some of the results which have been obtained during the single run tests



(a) 5 × 4 matrix calculation example

(b) Detection and isolation logic example

Figure 3: Example of SAO blocks

for the case in which two sensors are faulty. (The case when one sensor is faulty is not presented here for brevity as the triplex signal used in the controller is the correct one and therefore no fault is reconstructed.)

8.1.1. Two faulty sensors

Figure 4 shows the results for an oscillation fault. The simulations are initiated at a trim point inside the range where the LPV model is valid (as specified in equation (31) and [32]) and a turn coordination manoeuvre is conducted at an altitude of 23000 ft, speed of 300kts, an aircraft weight of 210 tonnes and centre of gravity at 32% MAC. Figure 4(a) shows the variation in terms of the LPV parameters (speed and altitude) while Figure 4(b) shows the states of the aircraft. In this simulation, sensors 1 and 2 contain an oscillatory fault signal from 20sec onwards. Figure 4(c) shows that the yaw rate sensors r_1 and r_2 (overlap dashed blue lines) contain the fault while r_3 (the solid red line) is the correct measurement. Figure 4(c) shows that the voting scheme has selected the faulty measurement ($r_{triplex}$) to be sent to the controller. As a consequence there is a significant effect on the performance of the aircraft as seen in Figure 4(b). Figure 4(d) shows that sliding is maintained ($\|e_y\|$ is close to zero) and good fault reconstruction is obtained (the reconstructed fault overlaps the actual fault).

Figure 5 shows the results for an additive fault on sensors r_1 and r_2 at 15 sec during a cruise condition. The simulations are conducted at a trim point technically outside the LPV range specified in [32] at an altitude of 37000 ft, a speed of 267kts, a weight of 185 tonnes and the centre of gravity at 28% MAC. This is presented to show the efficacy of the proposed method over a wide flight envelope compared to the restricted envelope where the LPV model is valid. Figure 5(a) shows the LPV parameter variations in terms of speed and altitude. Figure 5(b) shows the effect of the sensor fault on the states of the aircraft (in nominal fault free cruise conditions all theses lateral states should be close to zero). Figure 5(c) shows that the yaw rate sensors r_1 and r_2 contain faults and that the voting scheme has selected one of the faulty measurements ($r_{triplex}$). Figure 5(d) shows that sliding is maintained ($\|e_y\|$ is close to zero) and good fault reconstruction has been obtained.

8.2. Parametric Simulation Results

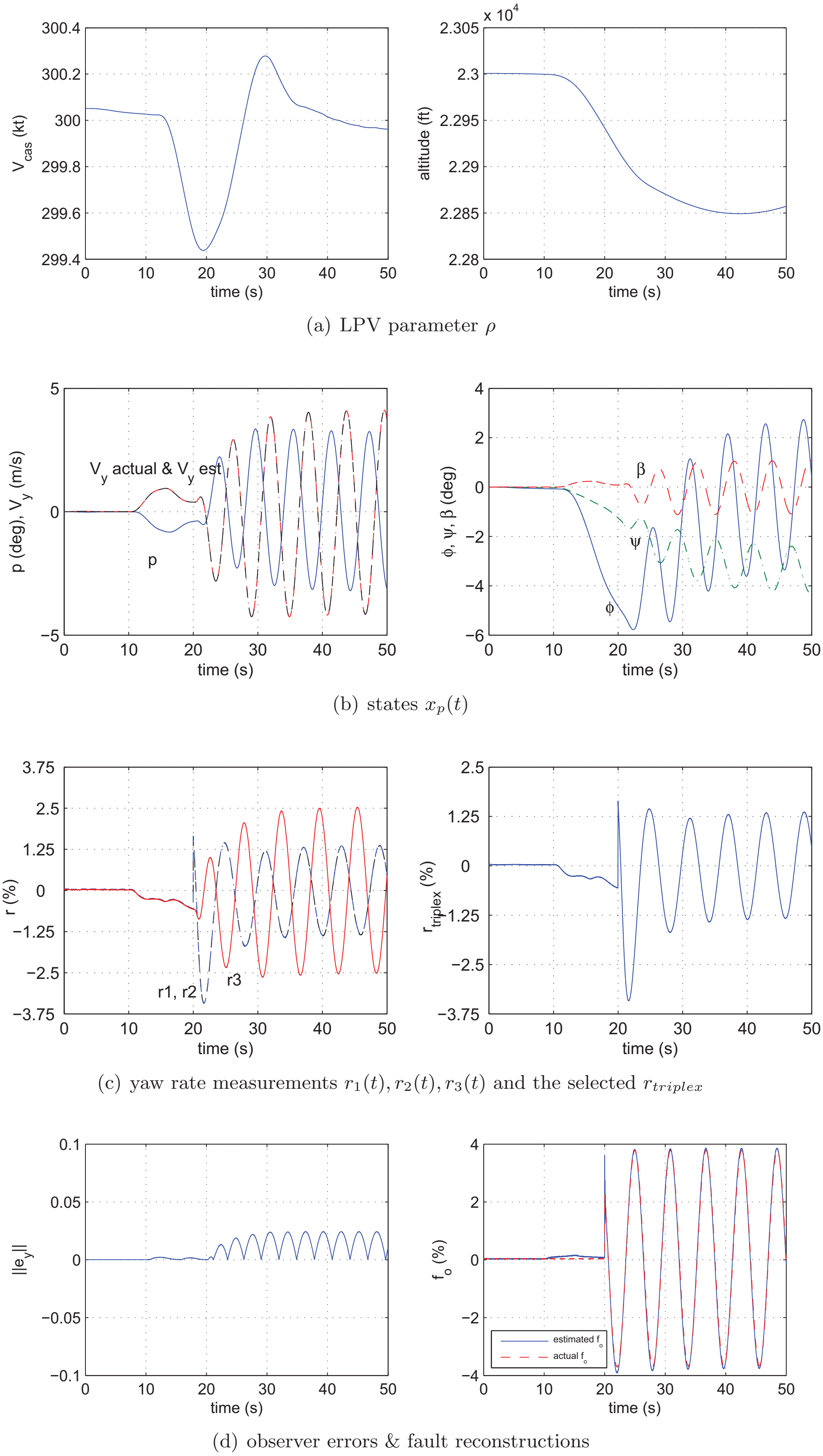
All the parametric FES results were conducted based on a cruise flight manoeuvre with variations in the operating conditions and uncertainty as specified in Table 5. The simulation campaign run on the FES, sweeps the entire flight envelope using a combination of grid parameters which results in a total of 324 simulation runs for each type of fault. For consistency, and to allow performance evaluation, various types of faults (with different magnitudes and frequencies) were all set to occur at 2 sec from the start of the simulation as indicated in Table 5. This also means the total run time for all the 324 runs is completed within a reasonable amount of time to allow fast assessment and possible re-tuning of the observer. All the fault types considered in the ADDSAFE benchmark problem as defined in Table 2 have been carried out. However, due to the large amount of data generated during the simulations, only the most significant plots are presented in this section. Here, only oscillatory sensor faults have been plotted, involving both one and two faulty sources. The other parametric simulation results are summarized in Table 7.

8.2.1. Fault Free

Figure 6 shows the variation of the LPV parameters (altitude, V_{cas}, mass and X_{cg}) during fault free cruise conditions on the full nonlinear ADDSAFE model for all 324 parametric runs. Figure 6 shows that the variation of the LPV parameters is well beyond the range in which the LPV model is valid theoretically. This does however provide a more challenging test of the proposed scheme.

8.2.2. Jam Fault cases

As mentioned earlier, all the type of faults considered in the ADDSAFE benchmark problem as defined in Table 2 have been tested. However, for brevity, only jam fault cases will be shown here (a one sensor fault (Jam-A) scenario and two faulty sensors (Jam-B) scenario), to highlight the capabilities of the proposed scheme.



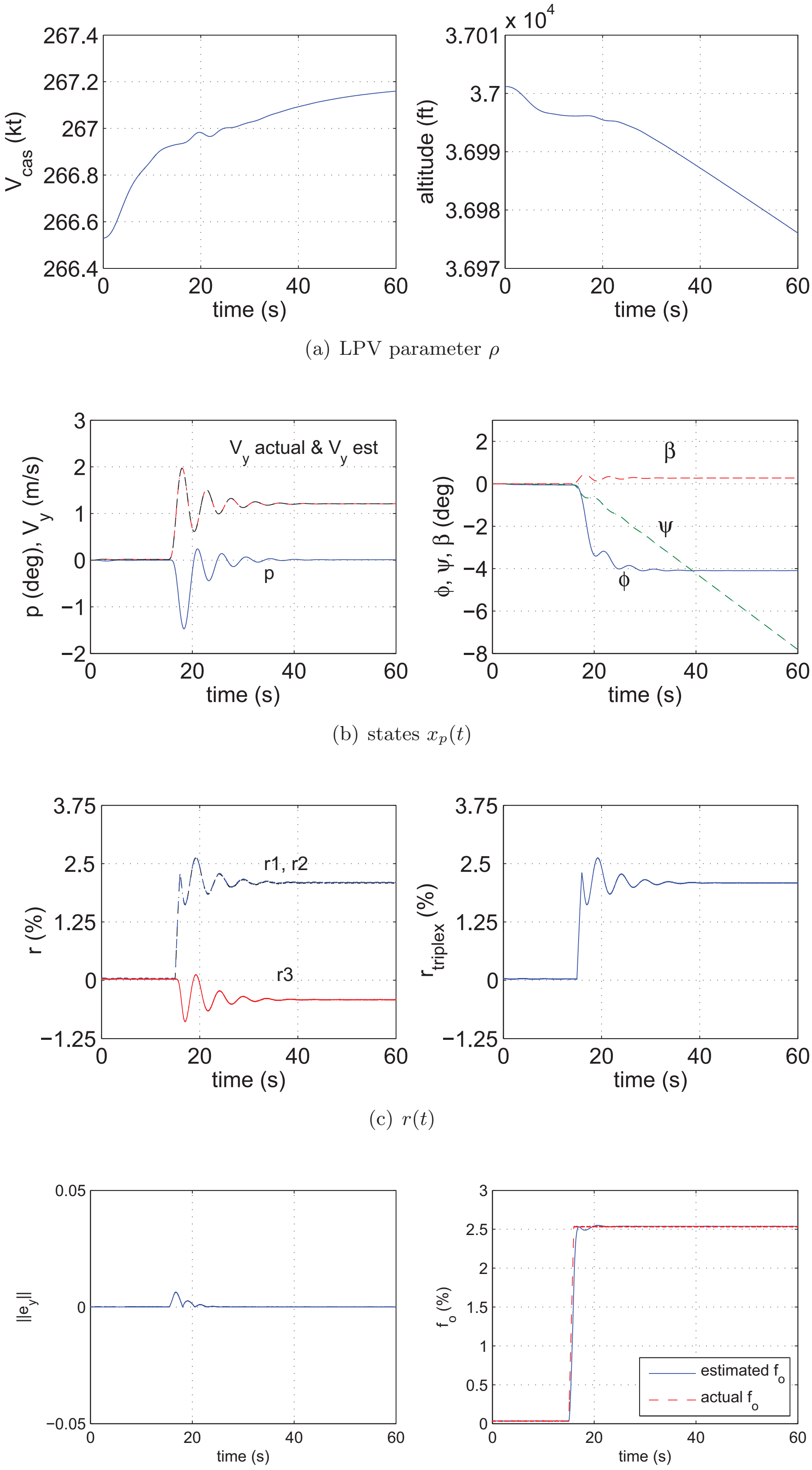
(a) LPV parameter ρ

(b) states $x_p(t)$

(c) yaw rate measurements $r_1(t), r_2(t), r_3(t)$ and the selected $r_{triplex}$

(d) observer errors & fault reconstructions

Figure 4: Inside LPV range - turn coordination: Oscillation Fault



(a) LPV parameter ρ

(b) states $x_p(t)$

(c) $r(t)$

(d) observer errors & fault reconstructions

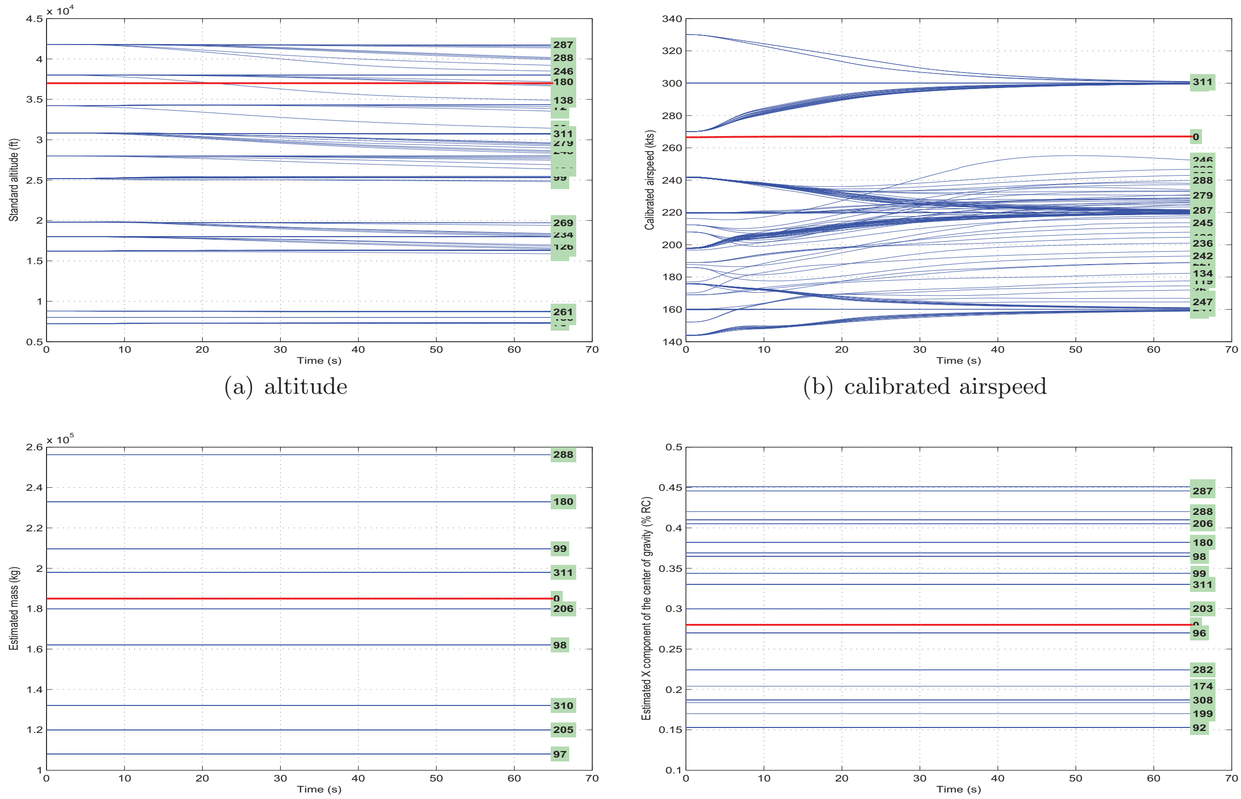
Figure 5: Outside LPV range - cruise: Additive Fault

8.2.3. Sensor 1 Jam (Jam-A)

Figure 7 shows the results when a jam occurs in one of the yaw rate sensors. Figures 7(a)-7(c) show all three redundant sensor measurements from the ADIRS, while Figure 7(d) shows the triplex signal sent to the controller. It can be clearly seen that at 2 sec, sensor 1 contains an offset jam fault signal while sensors 2 and 3 continue to provide the correct measurements (a yaw rate close to zero for a cruise condition). As shown by the triplex signal in Figure 7(d), the existing FDD scheme is able to select the correct fault free signals from sensors 2 and 3. Figure 7(f) shows the fault reconstruction from the sliding mode observer, while Figure 7(f) shows the fault detection signal from the proposed scheme. The fault reconstruction signals are close to zero, indicating that the triplex signal is fault free and therefore the existing FDD has selected the correct measurement, thus providing isolation of the faulty signal. A small threshold for the reconstruction and residual signals is able to confirm the presence of a fault.

8.2.4. Sensor 1 and 2 Jam (Jam-B)

Figure 8 shows the results when jam faults occur in two of the yaw rate sensors. As before Figures 8(a)-8(c) show all three ADIRS yaw rate measurements. It can be seen that sensors 1 (Figure 8(a)) and 2 (Figure 8(c)) contain the jam fault starting at 2 sec, and that the current FDD scheme has voted for one of the faulty signals as the triplex measurement (as seen in Figure 8(d)). The actual yaw rate of the aircraft is given by the measurement from sensor number 3 shown in Figure 8(c). Note that the oscillations seen in Figure 8(c) are due to the behavior of the aircraft which is operating in closed loop with a faulty measurement. The fault reconstruction signal in Figure 8(e) shows a nonzero reconstruction signal confirming that the triplex signal contains a fault. Figure 8(f) shows the detection signal immediately after the fault occurs, providing fast detection.



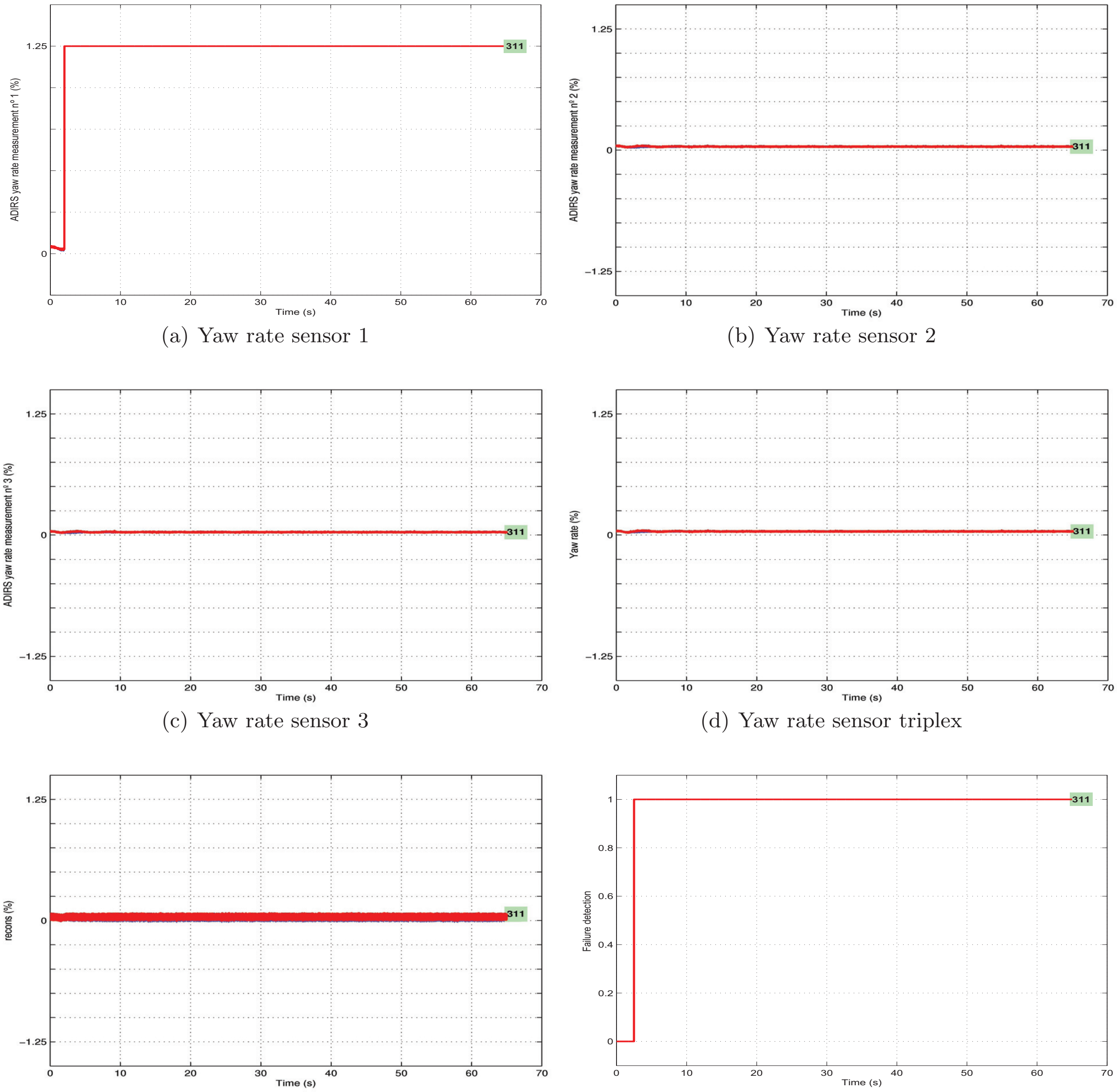
(a) altitude

(b) calibrated airspeed

(c) mass

(d) Xcg

Figure 6: Parametric: fault free – LPV parameter variations



(a) Yaw rate sensor 1

(b) Yaw rate sensor 2

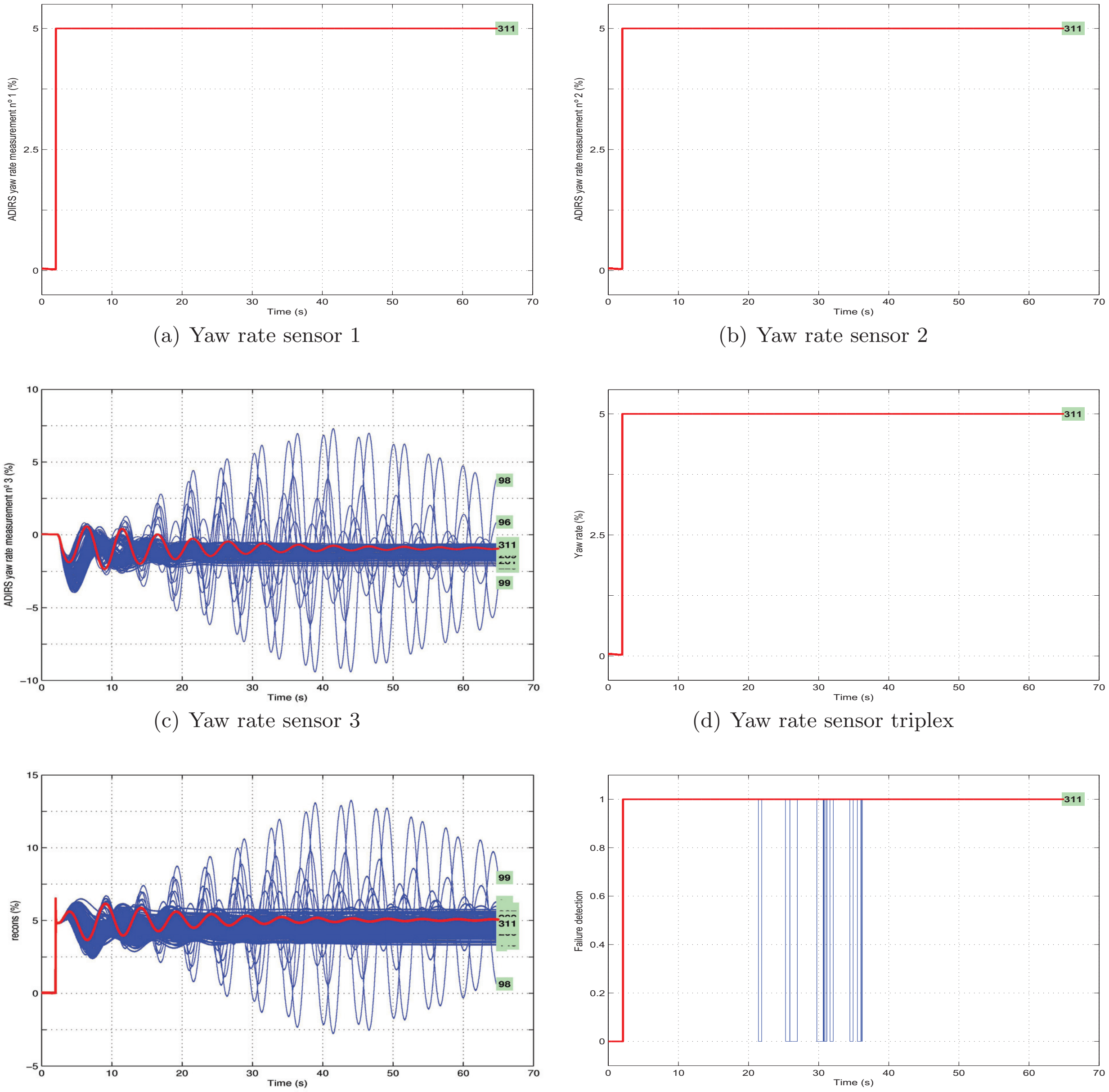
(c) Yaw rate sensor 3

(d) Yaw rate sensor triplex

(e) fault reconstruction

(f) fault detection

Figure 7: Parametric: yaw rate sensor jam A (1 faulty sensor)



(a) Yaw rate sensor 1

(b) Yaw rate sensor 2

(c) Yaw rate sensor 3

(d) Yaw rate sensor triplex

(e) fault reconstruction

(f) fault detection

Figure 8: Parametric: yaw rate sensor jam B (2 faulty sensor)

8.2.5. Parametric FES statistics

Table 7 shows the statistical results (as specified in Table 2). Table 7 shows no false alarms during fault free conditions, and no missed detections for the faulty cases. Table 7 also shows that the FDD scheme has detected the faults well before the maximum allowed detection time ($t_{D_{fdd}} max \ll 1$) which is highly desirable. The slowest detection time is associated with the fast runaway (Runaway-B) fault case. However the maximum detection time is still only 36.11% of the maximum allowable detection time defined in [25].

fault type	n_{f_o}	amplitude or rate* (%)	t_f	true det (%)	$t_{D_{fdd}}$ max†
fault free: cruise ADIRS	0	0	N/A	100	N/A
fault free: pitch protection 1	0	0	N/A	100	N/A
fault free: yaw angle mode	0	0	N/A	100	N/A
fault free: AoA protection 1	0	0	N/A	100	N/A
fault free: turn coordination 2	0	0	N/A	100	N/A
fault free: nose up	0	0	N/A	100	N/A
Oscill-A	1	1.25%	2	100	0.0342
Oscill-B	2	3.75%	2	100	0.3056
Jam-A	1	1.25%	2	100	0.0083
Jam-B	2	5%	2	100	0.3056
Runaway-A	2	2.5%	2	100	0.2733
Runaway-B	2	50%	2	100	0.3611
nrz-A	1	320%	2	100	0.0083
nrz-B	2	320%	2	100	0.3056
Noise-A	1	$\sigma^2 = 1.25\%$	2	100	0.0085
Noise-B	2	$\sigma^2 = 50\%$	2	100	0.3611

Table 7: Parametric FES results

t_f = failure time

$t_{D_{fdd}}$ max = maximum detection time

Remark: Note that, throughout the paper, the magnitude or rate of the faults are expressed as a percentage of the maximum admissible range. This is due to industrial confidentiality restrictions. Also the maximum FDD detection times $t_{D_{fdd}} max$ are given as normalized values. A value of 1 indicates that the fault is detected at the required detection time, while any value between 0 and 1 indicates a faster than the required detection time.

8.3. Industrial Evaluation Results

The results presented here were obtained from the industrial benchmarking and validation FES supervised by the industrial partners within ADDSAFE. A set of plots from the industrial evaluation campaign will be presented first, followed by statistical results, and finally other industrial evaluations (including tests not originally specified in the benchmark problem). Again, as in the previous section, only the most significant plots are presented. However, all the fault types considered in the ADDSAFE benchmark problem as defined in Table 2 have been tested during the industrial evaluation.

8.3.1. Monte-Carlo Simulation Results

Fault Free: Figures 9 - 10 show variations in the lateral states for the fault free case during the yaw angle mode and coordinated turn manoeuvres respectively. These variations highlight the changes in the operating conditions for different manoeuvres, and demonstrate wide coverage of the flight envelope.

Oscillatory faults: Figure 11 shows the fault reconstruction and detection signals for scenarios involving one (Oscill-A) and two (Oscill-B) sensors with oscillatory faults. Figure 11(a) shows (as expected) no fault is reconstructed for the one faulty sensor case, as the triplex signal is fault free. Figure 11(b) shows that the one sensor fault case has been detected correctly.

Figures 11(c)-11(d) show the fault reconstruction and detection signals in a two faulty sensors scenario. Figure 11(c) clearly shows an oscillatory sensor fault is present and Figure 11(d) shows the fault detection signal.

Jam faults: Figure 12 shows the fault reconstruction and detection signals for a one (Jam-A) and two (Jam-B) sensors jam situation. As in the previous cases, in the event when one sensor is faulty, Figure 12(a) shows no fault reconstruction indicating that the triplex signal is fault free.

In the case when two sensors jam, Figure 12(c) shows nonzero fault reconstructions, indicating that two sensors are faulty and the triplex measurements are faulty.

Runaway faults: Figure 13 shows the results for the case when two sensors contain a slow (Run-A) and fast (Run-B) runaway. These fault conditions result in faults of high magnitude as seen in Figures 13(a)-13(c). These figures also show clearly the reconstruction of the slow and fast runaway faults. The non-zero reconstruction signals indicate that the triplex signal used by the controller is faulty, and therefore indicates that two sensors in the ADIRS are faulty. Figure 13(b)-13(d) shows that the faults are detected quickly after the fault occurs.

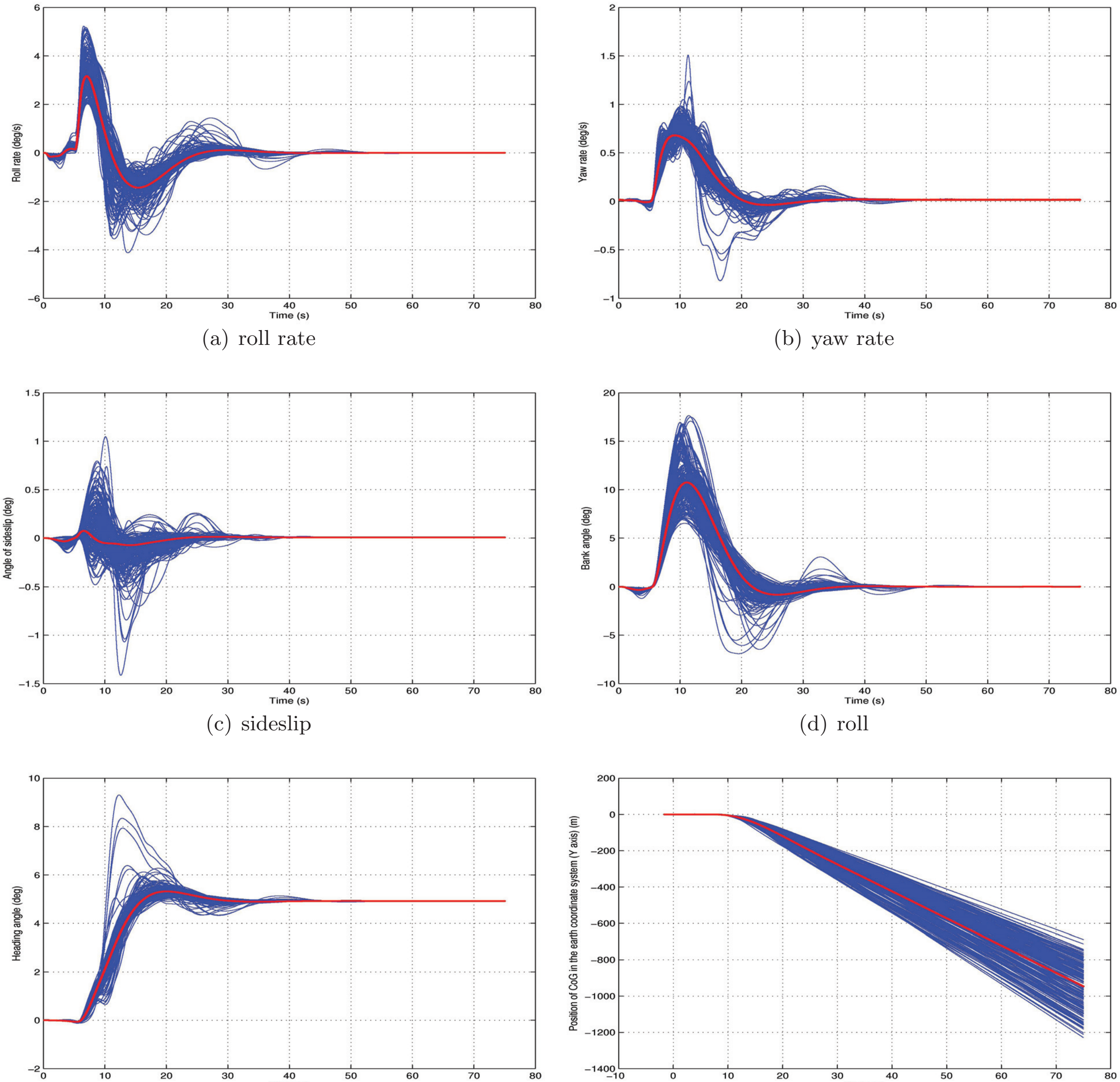
NRZ faults: Figure 14 shows the results when one and two sensors present a non-return to zero (NRZ) fault. Note that 100 Monte-Carlo runs are presented in these figures, although all of them lie on top of each other. Figures 14(a)-14(b) show that one faulty sensor is present, since the reconstruction signal is close to zero for all the Monte-Carlo runs.

Figures 14(c)-14(d) show a ‘zoomed-in’ plot (0-10sec) of the NRZ fault on two sensors. A ‘zoomed-in’ plot is shown here due to the fast, high frequency step-like signals associated with the NRZ type faults. Figure 14(c) shows nonzero fault reconstructions indicating the two sensors are faulty. Figure 14(c) also indicates that despite the varying flight conditions and uncertainties during the Monte-Carlo evaluations, the reconstruction signals are almost the same for all the runs as the signals overlap each other. The detection in Figures 14(d) also show consistent detection times despite the varying flight conditions and uncertainties.

Noise faults: Figure 15 shows the results for one (Noise-A) and two (Noise-B) sensors with noise type faults. Figures 15(a)-15(b) show the results in the one faulty sensor case. Here, the fault reconstruction signals are close to zero, indicating that the voted triplex signal is fault free and therefore the existing FDD scheme is isolating correctly the faulty measurement and excluding it from being used by the controller.

Figures 15(c)-15(d) show the Monte-Carlo evaluation results in the event when two sensors contain noise type faults. Zoomed-in plots from 0-10sec are shown due to the high frequency nature of the noise type faults. Figure 15(d) shows a very fast detection time almost immediately after the fault occurs at 2 sec. The non-zero fault reconstruction signals in Figure 15(c), indicate that there are two faulty sensors, and the triplex signals sent to the controller contain faulty measurements.

Plots Summary: In summary, for the cruise condition scenario considered for the industrial evaluation, with varying flight conditions and uncertainties, the detection plots for each type of fault are consistent throughout the Monte-Carlo evaluation (the detection signals overlap each other). This indicates the ability of the reconstruction and evaluation logic to provide consistent and reliable fault detection over a wide range of the flight envelope.



(a) roll rate

(b) yaw rate

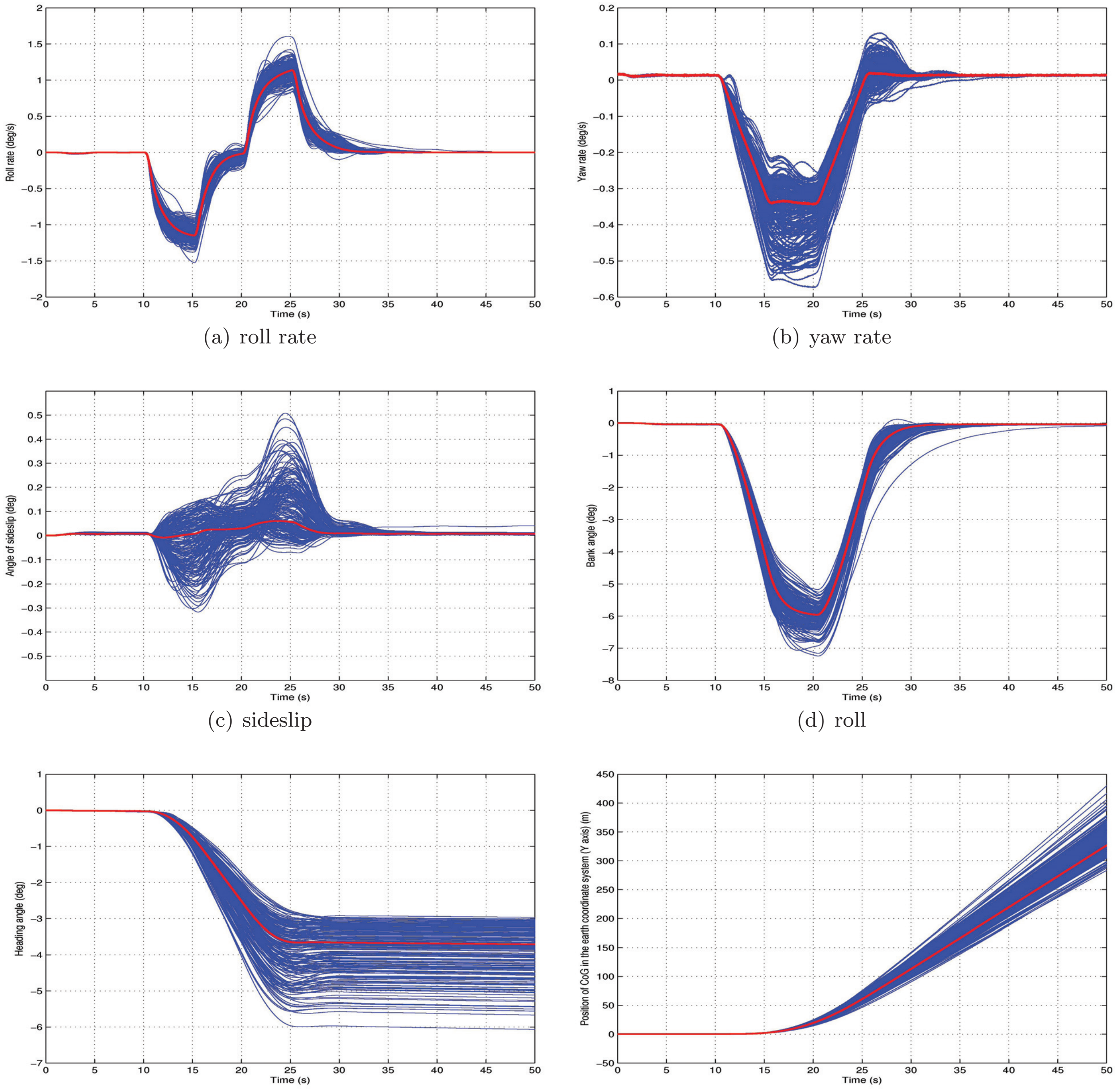
(c) sideslip

(d) roll

(e) yaw

(f) y_{earth} position

Figure 9: Monte-Carlo: yaw angle mode – fault free states



(a) roll rate

(b) yaw rate

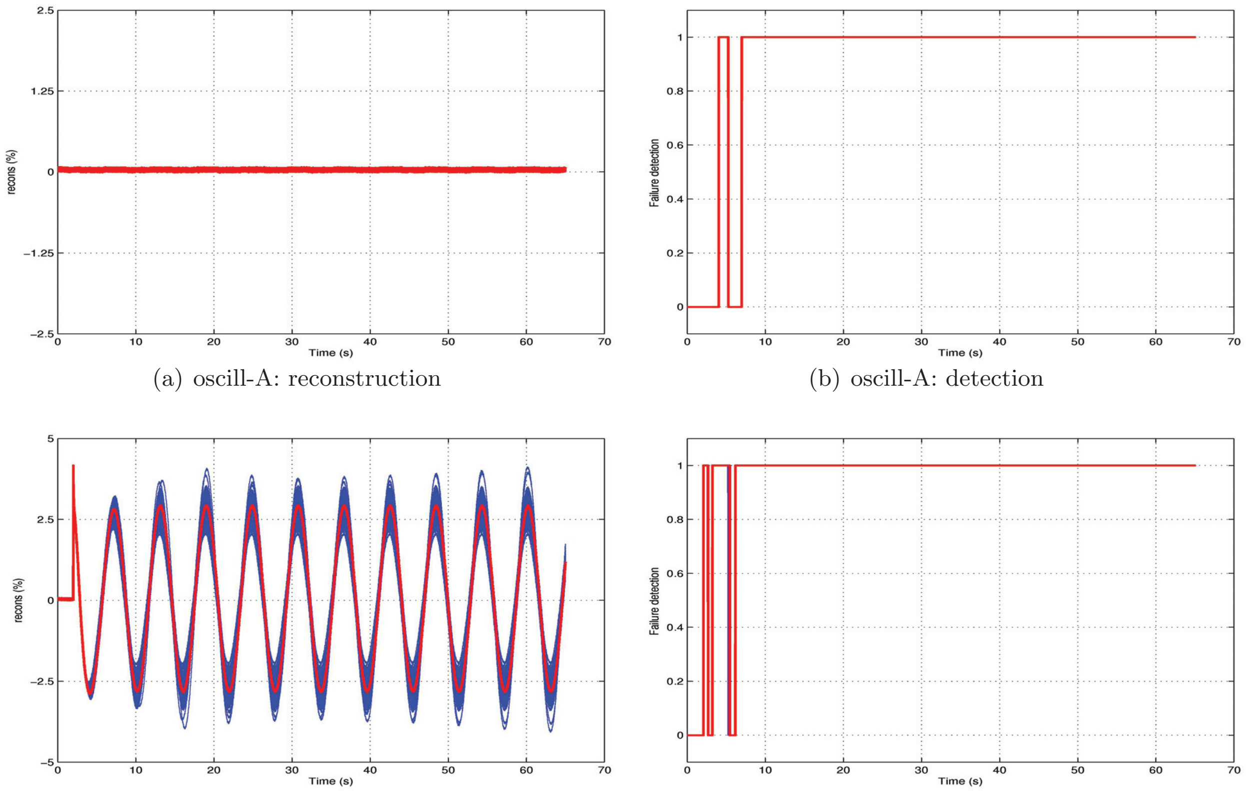
(c) sideslip

(d) roll

(e) yaw

(f) y_{earth} position

Figure 10: Monte-Carlo: coordinated turn – fault free states



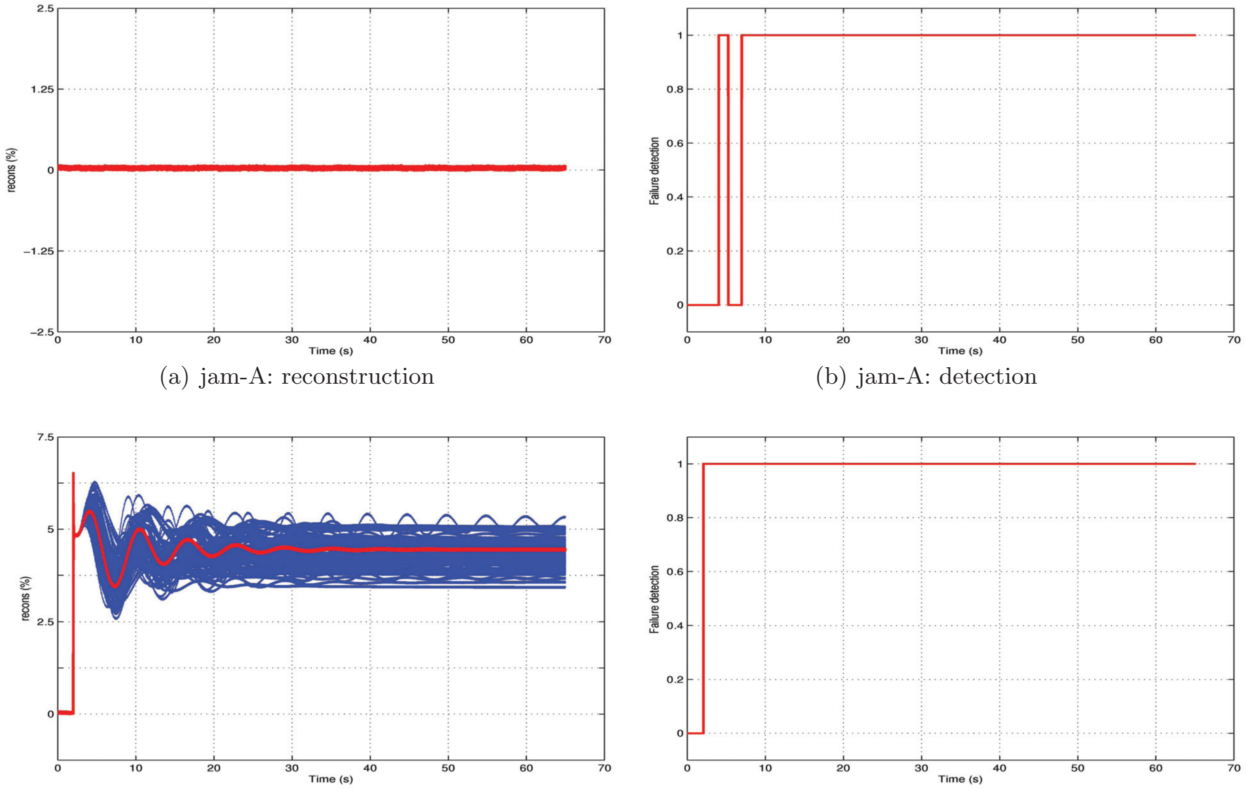
(a) oscill-A: reconstruction

(b) oscill-A: detection

(c) oscill-B: reconstruction

(d) oscill-B detection

Figure 11: Monte-Carlo: oscillatory faults (1 faulty sensor (Oscill-A) and 2 faulty sensors (Oscill-B))



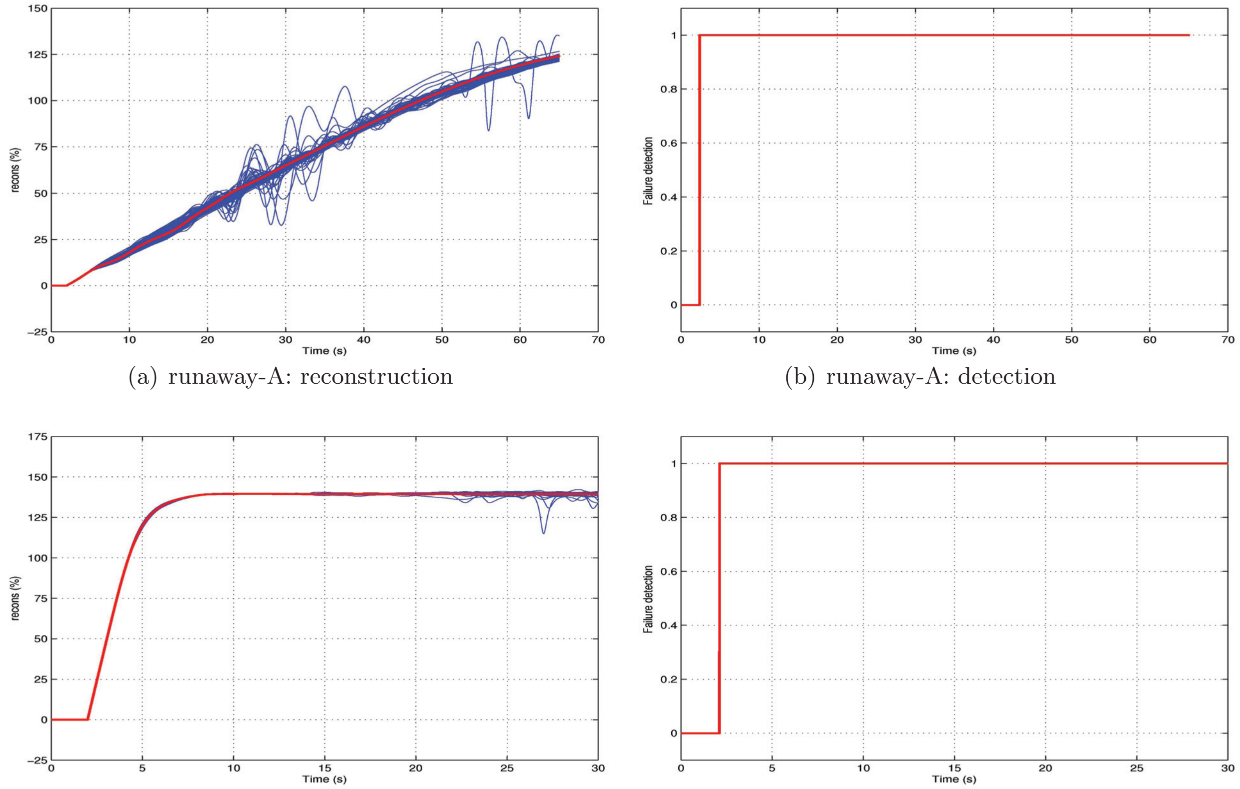
(a) jam-A: reconstruction

(b) jam-A: detection

(c) jam-B: reconstruction

(d) jam-B: detection

Figure 12: Monte-Carlo: jam faults (1 sensor jam (Jam-A) and 2 sensors jam (Jam-B))



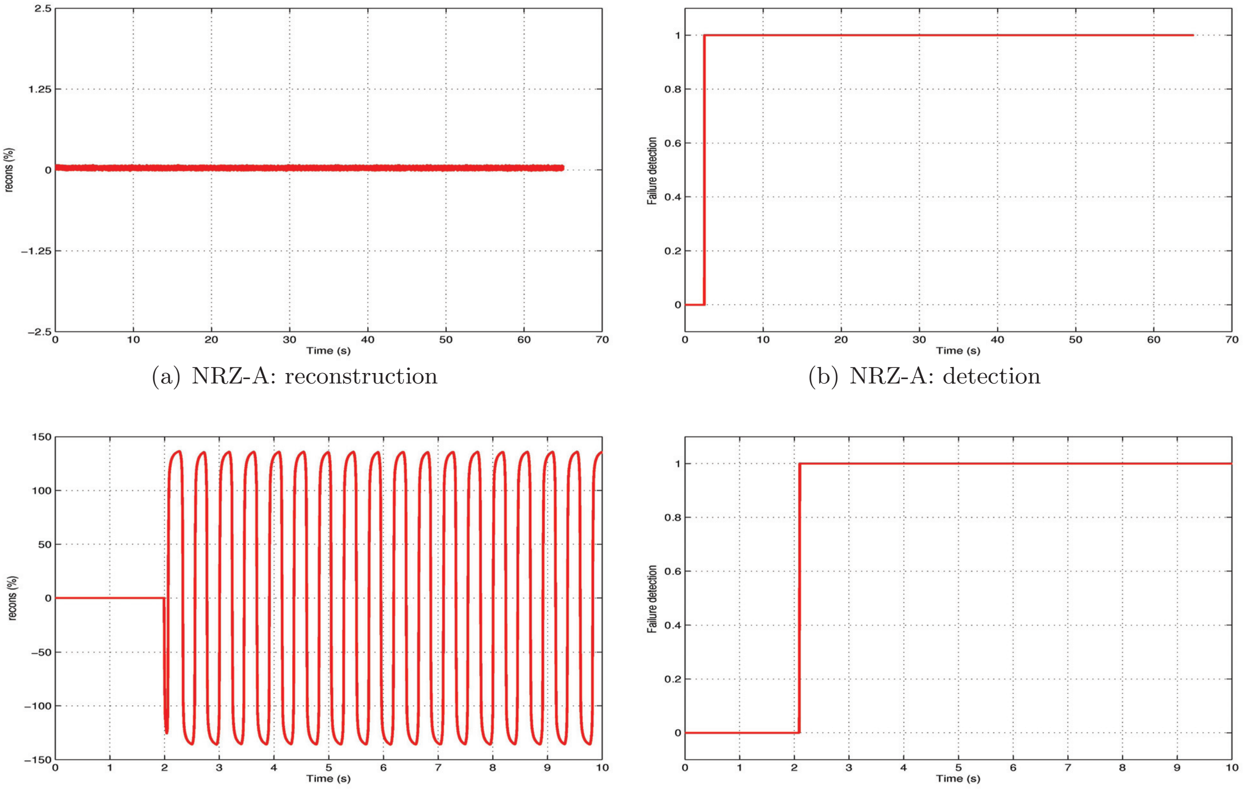
(a) runaway-A: reconstruction

(b) runaway-A: detection

(c) runaway-B: reconstruction

(d) runaway-B: detection

Figure 13: Monte-Carlo: runaway faults (2 sensors slow runaway (Run-A) and 2 sensors fast runaway (Run-B))



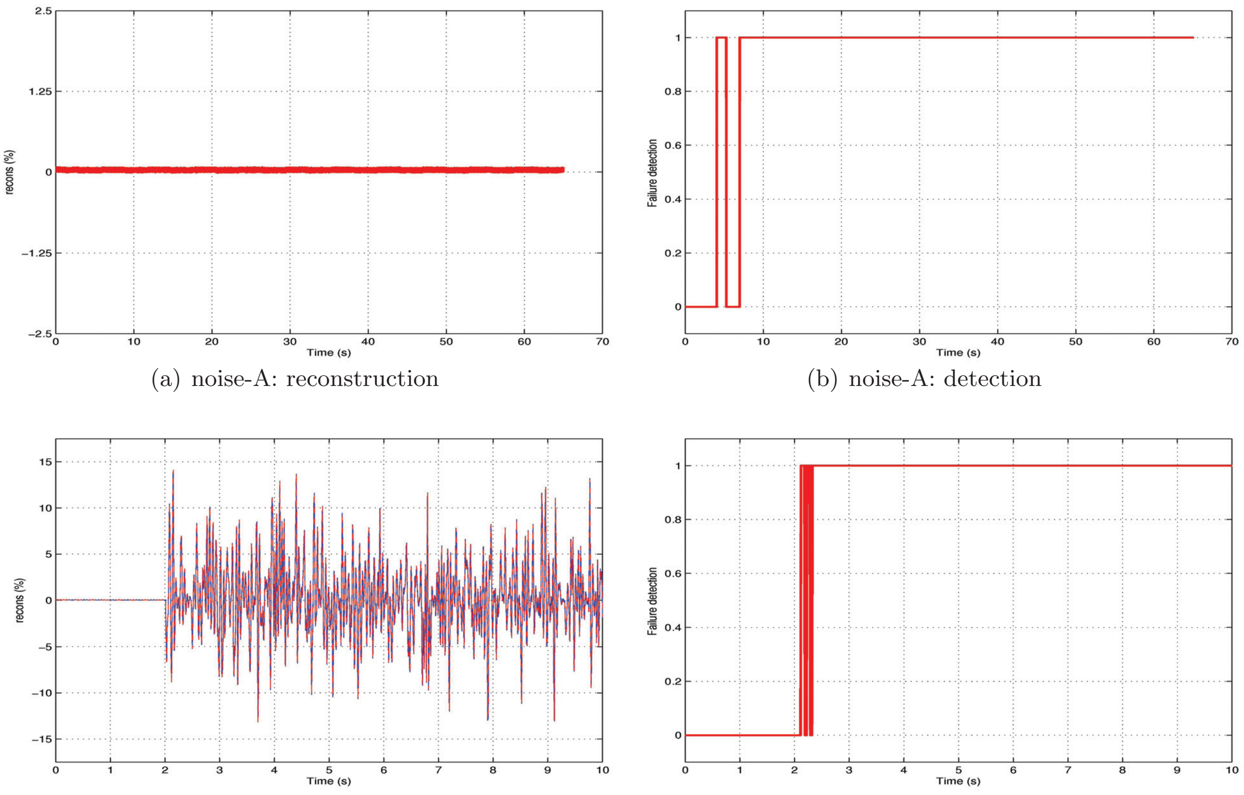
(a) NRZ-A: reconstruction

(b) NRZ-A: detection

(c) NRZ-B: reconstruction

(d) NRZ-B: detection

Figure 14: Monte-Carlo: non return to zero faults (1 sensor fault (NRZ-A) and 2 sensors fault (NRZ-B))



(a) noise-A: reconstruction

(b) noise-A: detection

(c) noise-B: reconstruction

(d) noise-B: detection

Figure 15: Monte-Carlo: noise faults (1 faulty sensor (Noise-A) and 2 faulty sensors(Noise-B))

8.3.2. Monte-Carlo FES statistics

Table 8 shows a summary of the performance of the proposed FDD scheme obtained from the industrial tests. Two sets of tests have been conducted: the first set is based on the SIMULINK block version, while the second set of evaluation results is based on the AIRBUS SAO blocks. Table 8 shows no false alarms during fault free conditions with various manoeuvres (under varying flight conditions and uncertainties setup in a random Monte-Carlo fashion) for both setups (SIMULINK and SAO blocks). Table 8 also shows 100% detection for all types of faults considered on both setups. Tests on the SIMULINK version show a maximum detection time of less than 37% from the maximum allowable, while tests on the SAO block version give a maximum detection time of less than 34%. These results on various types of faults with different magnitudes and rates show consistent performance, with an overall maximum detection time of less than 40% of the maximum allowable.

8.3.3. AIRBUS Evaluation Results

Table 9 shows the results based on the fault free case, with different manoeuvres not considered in the ADDSAFE benchmark tests specified in Table 2. Table 9 shows that no false alarms have occurred. This highlights the robustness and the potential of the proposed scheme to deal with other flight conditions not formally explored in the ADDSAFE benchmark problem.

8.4. Other Tests (SAO 'maturation' tests)

The results in this section are from the 'maturation' tests which consist of scenarios not considered in the original ADDSAFE benchmark problem. These tests have been carried out in order to ensure that the proposed FDD design is robust, and its performance is maintained throughout the flight envelope. All the test results are presented in Tables 10 - 11. These tests include different failure times, different fault levels and also different flight manoeuvres.

fault type	n_{f_o}	amplitude or rate* (%)	t_f	true det (%)	$t_{D_{fdd}}$ max†
using SIMULINK blocks					
fault free: cruise ADIRS	0	0	N/A	100	N/A
fault free: pitch protection 1	0	0	N/A	100	N/A
fault free: yaw angle mode	0	0	N/A	100	N/A
fault free: AoA protection 1	0	0	N/A	100	N/A
fault free: turn coordination 2	0	0	N/A	100	N/A
fault free: nose up	0	0	N/A	100	N/A
Oscill-A	1	1.25%	2	100	0.0342
Oscill-B	2	3.75%	2	100	0.3056
Jam-A	1	1.25%	2	100	0.0083
Jam-B	2	5.00%	2	100	0.3056
Runaway-A	2	2.50%	2	100	0.2733
Runaway-B	2	50.00%	2	100	0.3611
nrz-A	1	320.00%	2	100	0.0083
nrz-B	2	320.00%	2	100	0.3056
Noise-A	1	$\sigma^2 = 1.25\%$	2	100	0.0085
Noise-B	2	$\sigma^2 = 50.00\%$	2	100	0.3611
using SAO blocks					
fault free: cruise ADIRS	0	0	N/A	100	N/A
fault free: pitch protection 1	0	0	N/A	100	N/A
fault free: yaw angle mode	0	0	N/A	100	N/A
fault free: AoA protection 1	0	0	N/A	100	N/A
fault free: turn coordination 2	0	0	N/A	100	N/A
fault free: nose up	0	0	N/A	100	N/A
Oscill-A	1	1.25%	2	100	0.0342
Oscill-B	2	3.75%	2	100	0.2778
Jam-A	1	1.25%	2	100	0.0083
Jam-B	2	5.00%	2	100	0.2778
Runaway-A	2	2.50%	2	100	0.2733
Runaway-B	2	50.00%	2	100	0.3333
nrz-A	1	320.00%	2	100	0.0083
nrz-B	2	320.00%	2	100	0.2778
Noise-A	1	$\sigma^2 = 1.25\%$	2	100	0.0085
Noise-B	2	$\sigma^2 = 50.00\%$	2	100	0.3333

Table 8: Industrial simulation results - Monte-Carlo

8.4.1. Different failure time occurrences - cruise condition

The first set of results in Table 10 shows the results for different failure times. The original failure time in the benchmark problem is 2 sec. Here the failure is set to occur at 4 sec. The results show good detection times, well below 34% of the maximum allowed detection time.

8.4.2. Different fault amplitudes - cruise condition

The second set of results in Table 10 are associated with different levels of fault magnitude for all the 10 different categories of faults considered. The levels of the faults shown here are approximately half of the original ones considered in the industrial tests. This represents a much more challenging situation as the faults have smaller magnitude compared to the original specifications. Again the results show good detection times (well below 47% of the maximum allowed detection time).

fault type	n_{f_o}	amplitude or rate (%)	t_f	true det (%)	$t_{D_{fdd}}$ max
fault free: DQM	0	0	N/A	100	N/A
fault free: DPM	0	0	N/A	100	N/A
fault free: DRM	0	0	N/A	100	N/A
fault free: All trim mode	0	0	N/A	100	N/A
fault free: Side slip mode	0	0	N/A	100	N/A
fault free: Speed Change	0	0	N/A	100	N/A
fault free: Nose Down	0	0	N/A	100	N/A
fault free: Nz Limitation 1	0	0	N/A	100	N/A
fault free: Speed Mach protection	0	0	N/A	100	N/A
fault free: Roll protection	0	0	N/A	100	N/A
fault free: Nose up 2	0	0	N/A	100	N/A

Table 9: AIRBUS evaluation - SAO blocks: tests on an additional 11 manoeuvres

8.4.3. Fault cases under different flight manoeuvres

These tests have been conducted using 5 different flight manoeuvres associated with all the different types of faults considered in the benchmark problem (a total of 50 different test cases). Table 11 only shows the result which involve two faulty sensors on different lateral manoeuvres (namely the yaw angle mode and turn coordination). The results in Table 11 again show detection times well below 34% of the maximum allowed.

fault type	n_{f_o}	amplitude or rate	t_f	true det (%)	$t_{D_{fdd}}$ max
different failure time (SAO blocks)					
Oscill-A	1	1.25%	4	100	0.0083
Oscill-B	2	3.75%	4	100	0.2778
Jam-A	1	1.25%	4	100	0.0083
Jam-B	2	5.00%	4	100	0.2778
Runaway-A	2	2.50%	4	100	0.2667
Runaway-B	2	50.00%	4	100	**0.3333**
nrz-A	1	320.00%	4	100	0.0083
nrz-B	2	320.00%	4	100	0.2778
Noise-A	1	$\sigma^2 = 1.25\%$	4	100	0.0083
Noise-B	2	$\sigma^2 = 50.00\%$	4	100	0.2778
different failure amplitude (SAO blocks)					
Oscill-A	1	1.00%	2	100	0.0373
Oscill-B	2	1.875%	2	100	0.2778
Jam-A	1	1.00%	2	100	0.0083
Jam-B	2	2.50%	2	100	0.2778
Runaway-A	2	1.25%	2	100	**0.4667**
Runaway-B	2	25.00%	2	100	0.3611
nrz-A	1	160.00%	2	100	0.0083
nrz-B	2	160.00%	2	100	0.2778
Noise-A	1	$\sigma^2 = 0.625\%$	2	100	0.0085
Noise-B	2	$\sigma^2 = 25.00\%$	2	100	0.3333

Table 10: Parametric FES (different failure time and different failure amplitide/rate) - SAO blocks

fault type	n_{f_o}	amplitude or rate	t_f	true det (%)	$t_{D_{fdd}}$ max
different manoeuvres (SAO blocks)					
Oscill-B: yaw angle mode	2	3.75%	2	100	0.2778
Oscill-B: turn coordination 2	2	3.75%	2	100	0.2778
Jam-B: yaw angle mode	2	5.00%	2	100	0.2778
Jam-B: turn coordination 2	2	5.00%	2	100	0.2778
Runaway-A: yaw angle mode	2	2.50%	2	100	0.2800
Runaway-A: turn coordination 2	2	2.50%	2	100	0.2733
Runaway-B: yaw angle mode	2	50.00%	2	100	**0.3333**
Runaway-B: turn coordination 2	2	50.00%	2	100	**0.3333**
nrz-B: yaw angle mode	2	320.00%	2	100	0.2778
nrz-B: turn coordination 2	2	320.00%	2	100	0.2778
Noise-B: yaw angle mode	2	$\sigma^2 = 50.00\%$	2	100	0.3333
Noise-B: turn coordination 2	2	$\sigma^2 = 50.00\%$	2	100	0.3333

Table 11: Parametric FES, other manoeuvres - SAO blocks

9. Conclusion

This paper has described the development and application of a robust sliding mode observer scheme to the yaw rate sensor fault ADDSAFE benchmark problems. An LPV observer has been synthesised using LMIs, where the objective is to minimize the effect of uncertainty on the sensor fault reconstruction. The approach first converts the sensor fault reconstruction problem into an actuator one by augmenting the plant states with the filtered outputs of those measurements which are to be monitored. The observer gain associated with the nonlinear injection is fixed, while the gain associated with the linear injection is parameterized by the LPV variable. The fault reconstruction signal is obtained from the output error injection signal associated with the underlying sliding mode observer. The results from various stages of the scheme's development, from the initial design, through tuning and finally rigorous industrial assessment using Monte-Carlo simulations have been presented. The industrial evaluation and assessment have been conducted using a realization of the scheme using AIRBUS's so-called SAO library which allows the automatic generation of flight certifiable code which can be implemented on the actual flight control computer.

A set of maturation tests, based on various manoeuvres, flight conditions and fault settings not considered in the ADDSAFE benchmark specifications have also been conducted. All the results show good and consistent performance throughout various stages of the design, testing and assessment process, and therefore offers potential for further industrial tests and implementations.

References

[1] H. Alwi, C. Edwards, C. P. Tan, Fault Detection and Fault–Tolerant Control Using Sliding Modes, Advances in Industrial Control, Springer-Verlag, 2011.

[2] C. Edwards, T. Lombaerts, H. Smaili, (Eds.), Fault Tolerant Flight Control: A Benchmark Challenge, Vol. 399, Springer-Verlag: Lecture Notes in Control and Information Sciences, 2010.

[3] S. Armeni, A. Casavola, E. Mosca, Robust fault detection and isolation for LPV systems under a sensitivity constraint, International Journal Of Adaptive Control And Signal Processing 23 (2009) 55–72.

[4] J. Bokor, G. Balas, Detection filter design for LPV systems - a geometric approach, Automatica 40 (2004) 511–518.

[5] J. Bokor, Z. Szabo, G. Stikkel, Failure detection for quasi LPV systems, in: IEEE Conference on Decision and Control, 2002.

[6] A. Casavola, D. Famularo, G. Franz, M. Sorbara, A fault-detection, filter-design method for linear parameter-varying systems, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 221 (6) (2007) 865–874.

[7] S. Grenaille, D. Henry, A. Zolghadri, A method for designing fault diagnosis filters for LPV polytopic systems, Journal of Control Science and Engineering 2008 (doi:10.1155/2008/231697) (2008) 1–11.

[8] D. Henry, A. Falcoz, A. Zolghadri, Structured H_∞/H_- LPV filters for fault diagnosis: Some new results, in: Proceedings of the IFAC Symposium SAFEPROCESS '09, Barcelona, 2009, pp. 420–425.

[9] M. Sato, Filter design for LPV systems using quadratically parameter-dependent lyapunov functions, Automatica 42 (11) (2006) 2017–2023.

[10] X. Wei, M. Verhaegen, Mixed $\mathcal{H}_-/\mathcal{H}_\infty$ fault detection observer design for LPV systems, in: IEEE Conference on Decision and Control, 2008, pp. 1073–1078.

[11] G. J. Balas, Linear, parameter–varying control and its application to a turbofan engine, International Journal of Robust and Nonlinear Control 12 (2002) 763–796.

[12] H. Pfifer, S. Hecker, Generation of optimal linear parametric models for LFT-based robust stability analysis and control design, in: Proceedings of the 47th IEEE Conference on Decision and Control, CDC, 2008, pp. 3866–3871.

[13] K. Nonami, S. Sivrioglu, Sliding mode control with gain scheduled hyperplane for LPV plant, in: K. D. Young, U. Özgüner (Eds.), Variable structure systems, sliding mode and nonlinear control, Vol. 247 of Lecture Notes in Control and Information Sciences, Springer Berlin / Heidelberg, 1999, pp. 263–279.

[14] S. Sivrioglu, K. Nonami, Sliding mode control with time-varying hyperplane for AMB systems, IEEE/ASME Transactions On Mechatronics 3 (1) (1998) 51–59.

[15] D. V. Efimov, L. M. Fridman, T. Raissi, A. Zolghadri, R. Seydou, Interval estimation for LPV systems applying high order sliding mode techniques, Automatica 48 (9) (2012) 2365–2371.

[16] H. Alwi, C. Edwards, A. Marcos, Actuator and sensor fault reconstruction using an LPV sliding mode observer, in: AIAA Guidance, Navigation and Control Conference and Exhibit, no. AIAA-2010-8157, 2010.

[17] H. Alwi, C. Edwards, Robust actuator fault reconstruction for LPV systems using sliding mode observers, in: Proceedings of the 49th IEEE Conference on Decision and Control, CDC, 2010.

[18] H. Alwi, C. Edwards, Validation of sliding mode observer FDI schemes on the ADDSAFE functional engineering simulator, in: Proceedings of the IFAC Symposium SAFEPROCESS '12, Mexico City, 2012.

[19] C. P. Tan, C. Edwards, Sliding mode observers for robust detection and reconstruction of actuator and sensor faults, International Journal of Robust and Nonlinear Control 13 (2003) 443–463.

[20] P. Goupil, A. Marcos, Advanced Diagnosis for Sustainable Flight Guidance and Control: The European ADDSAFE Project, in: SAE AeroTech Congress & Exhibition, 2011.

[21] V. Fernández, J. M. Ramón, FES software users manual, ADDSAFE Technical Note D1.2.1, DEIMOS (2011).

[22] P. Goupil, G. Puyou, A high fidelity AIRBUS benchmark for system fault detection and isolation and flight control law clearance, in: European Conference for AeroSpace Sciences (EUCASS'11), 2011.

[23] P. Goupil, A. Marcos, Industrial benchmarking and evaluation of ADDSAFE FDD designs, in: Proceedings of the IFAC Symposium SAFEPROCESS '12, Mexico City, 2012.

[24] A. Marcos, Advanced fault diagnosis for sustainable flight guidance and control, in: 6th European Aeronautics Days, AERODAYS, Madrid, Spain, 2011.

[25] P. Goupil, FDD problem definition, Report D1.1.2, ADDSAFE (2010).

[26] P. Goupil, AIRBUS state of the art and practices on FDI and FTC in flight control system, Control Engineering Practice 19 (6) (2011) 524–539.

[27] P. Goupil, AIRBUS benchmark description, Tech. Rep. ADDSAFE D1.1.3, AIRBUS OPERATIONS SAS (2010).

[28] C. Edwards, S. Spurgeon, R. Patton, Sliding mode observers for fault detection, Automatica 36 (2000) 541–553.

[29] V. I. Utkin, Sliding Modes in Control Optimization, Springer-Verlag, Berlin, 1992.

[30] P. Apkarian, P. Gahinets, G. Becker, Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example, Automatica 31 (9) (1995) 1251–1261.

[31] A. Levant, Robust exact differentiation via sliding mode technique, Automatica 34 (3) (1998) 379–84.

[32] S. Hecker, Nominal and faulty LFT/LPV models, ADDSAFE report D1.3.2-3, DLR (2010).

[33] C. Edwards, S. K. Spurgeon, Sliding Mode Control: Theory and Applications, Taylor & Francis, 1998.

Appendix: The system matrices $\boldsymbol{A(\rho)}, \boldsymbol{B(\rho)}$ and gains $\boldsymbol{G_l(\rho)}$

$A(\rho) = A_0 + A_1\rho_1 + A_2\rho_2 + A_3\rho_3 + A_4\rho_4$

$$A_0 = \begin{bmatrix} 0 & 0 & 1.0000 & 0.0465 \\ 9.7981 & -0.1170 & 8.3767 & -197.8845 \\ 0 & -0.0285 & -1.7405 & 0.2744 \\ 0 & 0.0044 & -0.1349 & -0.2246 \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 0 & 0 & 0.0082 \\ 0 & 0.0123 & 1.6744 & -0.0821 \\ 0 & 0.0012 & 0.1096 & 0.0021 \\ 0 & -0.0003 & -0.0017 & 0.0115 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & -0.0005 \\ 0 & -0.0001 & -0.0963 & 0 \\ 0 & 0.0000 & -0.0005 & -0.0009 \\ 0 & -0.0002 & -0.0023 & 0.0041 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 0 & 0 & 0.0017 \\ 0 & 0.0075 & 1.0838 & -15.0464 \\ 0 & 0.0008 & 0.1242 & 0.0018 \\ 0 & -0.0005 & 0.0027 & 0.0274 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0 & 0 & 0 & -0.0146 \\ 0 & -0.0142 & -2.1456 & -19.5283 \\ 0 & -0.0031 & -0.1236 & -0.0195 \\ 0 & 0.0005 & 0.0025 & -0.0161 \end{bmatrix}$$

$B(\rho) = B_0 + B_1\rho_1 + B_2\rho_2 + B_3\rho_3 + B_4\rho_4$

$$B_0 = \left[\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0863 & 0.0059 & -0.0059 & 0.0051 & -0.0051 & -0.0012 & 0.0012 \\ 0.0045 & 0.0113 & -0.0113 & 0.0054 & -0.0054 & -0.0074 & 0.0074 \\ -0.0148 & 0.0003 & -0.0003 & -0.0001 & 0.0001 & -0.0006 & 0.0006 \end{matrix} \right.$$

$$\left. \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0012 & 0.0012 & -0.0012 & 0.0012 & -0.0012 & 0.0012 \\ -0.0074 & 0.0074 & -0.0074 & 0.0074 & -0.0074 & 0.0074 \\ -0.0006 & 0.0006 & -0.0006 & 0.0006 & -0.0006 & 0.0006 \end{matrix} \right]$$

$$B_1 = \left[\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0089 & -0.0006 & 0.0006 & -0.0005 & 0.0005 & 0 & 0 \\ -0.0002 & -0.0008 & 0.0008 & -0.0004 & 0.0004 & 0.0003 & -0.0003 \\ 0.0008 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 \end{matrix} \right.$$

$$\left. \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0003 & -0.0003 & 0.0003 & -0.0003 & 0.0003 & -0.0003 \\ 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 \end{matrix} \right]$$

$$B_2 = \left[\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7200 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6500 & -0.6500 & 0.0500 & -0.0500 & 0.2300 & -0.2300 \\ 0.9500 & 0.1100 & -0.1100 & 0.0800 & -0.0800 & -0.0400 & 0.0400 \end{array}\right.$$

$$\left.\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2300 & -0.2300 & 0.2300 & -0.2300 & 0.2300 & -0.2300 \\ -0.0400 & 0.0400 & -0.0400 & 0.0400 & -0.0400 & 0.0400 \end{array}\right] \times 10^{-4}$$

$$B_3 = \left[\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0015 & 0.0002 & -0.0002 & 0.0003 & -0.0003 & 0 & 0 \\ 0.0005 & -0.0013 & 0.0013 & -0.0003 & 0.0003 & -0.0001 & 0.0001 \\ 0.0001 & -0.0000 & 0.0000 & 0.0001 & -0.0001 & -0.0001 & 0.0001 \end{array}\right.$$

$$\left.\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0001 & 0.0001 & -0.0001 & 0.0001 & -0.0001 & 0.0001 \\ -0.0001 & 0.0001 & -0.0001 & 0.0001 & -0.0001 & 0.0001 \end{array}\right]$$

$$B_4 = \left[\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0156 & 0.0013 & -0.0013 & 0.0011 & -0.0011 & 0 & 0 \\ 0.0009 & 0.0012 & -0.0012 & 0.0004 & -0.0004 & -0.0010 & 0.0010 \\ -0.0028 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0002 \end{array}\right.$$

$$\left.\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0010 & 0.0010 & -0.0010 & 0.0010 & -0.0010 & 0.0010 \\ -0.0002 & 0.0002 & -0.0002 & 0.0002 & -0.0002 & 0.0002 \end{array}\right]$$

$G_l(\rho) = G_{l_0} + G_{l_1}\rho_1 + G_{l_2}\rho_2 + G_{l_3}\rho_3 + G_{l_4}\rho_4$

$$G_{l_0} = \begin{bmatrix} 3.0983 & -0.0013 & 1.0000 & 0 \\ 16.8272 & 8.5679 & 8.3669 & 0 \\ -0.0097 & -0.0361 & 1.5595 & 0 \\ -0.1021 & -0.0781 & -0.1348 & 0 \\ -0.0004 & -0.0003 & 0 & 3.3900 \end{bmatrix}$$

$$G_{l_1} = \begin{bmatrix} -0.0003 & -0.0002 & 0 & 0 \\ 0.0029 & 0.0145 & 1.6744 & 0 \\ -0.0001 & 0.0012 & 0.1096 & 0 \\ -0.0004 & -0.0006 & -0.0017 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_{l_2} = \begin{bmatrix} 0.0000 & 0.0000 & 0 & 0 \\ 0 & -0.0001 & -0.0963 & 0 \\ 0.0000 & 0.0000 & -0.0005 & 0 \\ -0.0001 & -0.0003 & -0.0023 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$G_{l_3} = \begin{bmatrix} -0.0001 & -0.0000 & 0 & 0 \\ 0.5345 & 0.4245 & 1.0831 & 0 \\ -0.0001 & 0.0008 & 0.1242 & 0 \\ -0.0010 & -0.0012 & 0.0027 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G_{l_4} \quad = \quad \begin{bmatrix} 0.0005 & 0.0004 & -0.0000 & 0 \\ 0.6937 & 0.5271 & -2.1466 & 0 \\ 0.0007 & -0.0026 & -0.1236 & 0 \\ 0.0006 & 0.0010 & 0.0025 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$