IMMUNE : INTELLIGENT MONITORING AND MANAGING OF UNEXPECTED EVENTS

C. Döll[1] , A. Varga[2] , C. Kappenberger[3] , G. Hardier[1] , T. Ruprecht[3]
[1] ONERA - The French Aerospace Lab
Systems Control and Flight Dynamics Department DCSD
2, av. Ed. Belin, B.P. 4025, 31055 Toulouse Cedex 4, France
E-mail: carsten.doll@onera.fr
Tel.: +33 (0)5.62.25.29.20, Fax: +33 (0)5.62.25.25.64)
[2] DLR - German Aerospace Center
Institute of Robotics and Mechatronics
Münchner Straße 20 - 82234 Weßling, Germany
[3] DLR - German Aerospace Center
Institute of Flight Systems
Lilienthalplatz 7 - 38108 Braunschweig, Germany

Keywords: *fault modelling, fault detection, isolation and estimation FDIE, parameter identification, control law reconfiguration, adaptive control, supervisor.*

Abstract

The IMMUNE project (Intelligent Monitoring and Managing of UNexpected Events) focuses on basic research regarding event monitoring and event tolerant control. Event detection methods and control law reconfiguration approaches have been investigated for their applicability in the context of developing advanced aircraft flight control systems. For promising methods, prototype software tools have been implemented in a desktop simulation environment to serve as basis for evaluation studies.

1 Introduction

IMMUNE stands for Intelligent Monitoring and Managing of UNexpected Events. It was a three years project jointly run by ONERA and DLR teams within the DLR-ONERA Common Transport Aircraft Research Programme. The objective of this project is to demonstrate the capability and viability of intelligent techniques for monitoring and handling the Flight Control System (FCS) in real time, to improve aircraft safety and autonomy. The monitoring is based on several methods, including modern Fault Detection, Isolation and Estimation (FDIE) techniques and of course on-line identification. The handling of the detected events is contemplated by different reconfiguration or self-adapting techniques, based on Fault Tolerant Control (FTC) principles. Both actions are often strongly dependent and therefore linked via a supervisor architecture in charge of the decision making. See Fig. 1 for the possible interactions between FDIE and FTC.

The IMMUNE project focuses on the development of prototype software tools for the most promising methods concerning event monitoring and event tolerant control :

1. linear filter/observer based techniques
2. nonlinear filtering and estimation
3. residual signal generation
4. online parameter identification
5. risk quantification

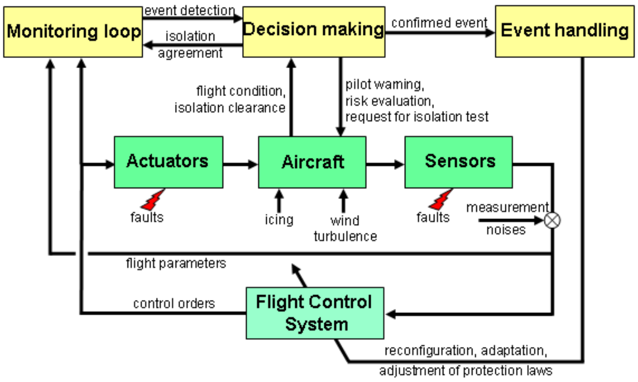

Fig. 1 Monitoring, decision making and handling of events

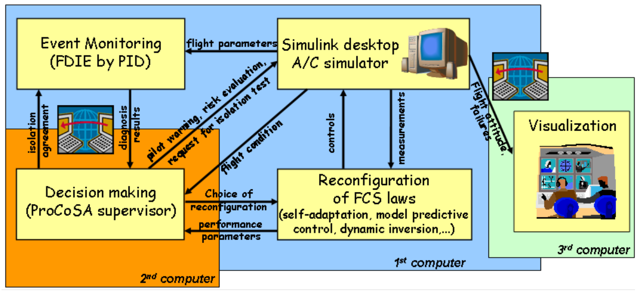

Fig. 2 The overall simulation environment including 3 computers

6. supervisor architecture and realization
7. adaptive dynamic inversion
8. self-adaptive control
9. model predictive control for actuator reallocation

Benchmark scenarii for relevant events like

- actuator faults (losses of efficiency, jams and runaways) for inner and outer ailerons, left and right elevators, stabilizers and the rudder
- sensor faults
- change of aerodynamics due to icing accretion

have been defined and implemented in a simulation environment to ease the evaluation and the comparison of design methods and tools.

2 The common simulation environment

A common simulation environment was defined and implemented at the beginning of the project. It is depicted in Fig. 2. The A/C model, the FDIE and the FTC are simulated under Matlab/Simulink© on one PC which communicates through a network with a second PC where the decision making process is implemented by a Petri-Net player ProCoSA©. The first PC is also linked to a third visualization PC controlling a 2-D cockpit animation with a failure panel for pilot information or a 3-D trajectory animation, for example using Flight Gear©.

The Matlab/Simulink© desktop simulator on the first PC is shown in Fig. 3. It has a very modular structure, which allows to integrate gradually new components, e.g. online Parameter IDentification (PID) and Model Predictive Control (MPC) as shown in the figure.

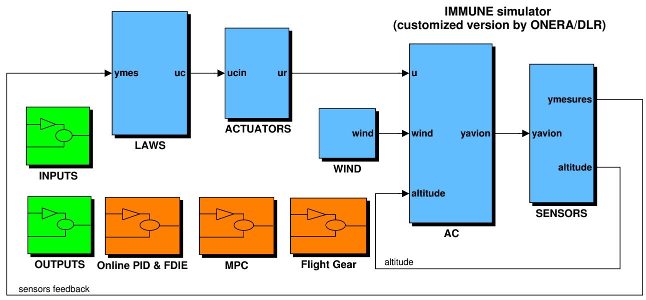

Fig. 3 The Simulink© desktop simulator including A/C model, FDIE and FTC as well as visualization blocks

As shown in Fig. 1 and 3, the aircraft modeling involves an actuator block (including actuator dynamics, rate and deflection limitations, fault emulation), a sensor block (including a transfer to the location of the real sensors, additive colored noises, fault emulation), and the A/C model by itself (including flight mechanics, aerodynamics, propulsion, a modeling of wind and turbulence, icing effects, etc.). The A/C model results from a generic model representative of the behavior of a long range commercial aircraft with characteristics given in Table 1.

Design feature	Value	Unit
Overall length	60	m
Wing span	60	m
Wing reference area	360	m^2
Wing sweep (25% chord)	30	^{o}
Fuselage diameter	5.6	m
Max engine thrust (sea level)	320	kN
Max operating Ma number	0.86	–
Max take-off weight	230	t
Max landing weight	180	t
Max zero-fuel weight	170	t

Table 1 Aircraft main design features

The nominal FCS involves three parts, one for the lateral/directional laws computing roll and yaw equivalent control orders, one for the longitudinal laws computing pitch and speedbrake equivalent control orders, and one for the autothrottle computing engine thrust orders. These high level FCS orders are then splitted into a set of real control deflections for the different actuators, namely one pair of elevators, one trimmable stabilizer, one rudder, one pair of inner ailerons, one pair of outer ailerons, one pair of engines, and 6 pairs of spoilers. In the fault free situation, the rudder is used for yaw control, the spoilers as speedbrakes, the stabilizer for low pass pitch control, the elevators for high pass pitch control, the inner ailerons for roll control in clean and high lift configurations, the outer ailerons for roll control in high lift configurations, and the engines for speed control.

In faulty situations the handling block can reallocate the orders in order to benefit from physical redundancies between some actuators. Typical faults can be simulated for each actuator (like loss of efficiency, runaway or jam) and for Air Data Reference ADR sensors (like bias, drift or stuck value).

Apart from the sensor/actuator faults, the desktop environment permits to simulate realistic flight conditions by integrating disturbances (wind, turbulence) and measurement noises into the scenario. Following [9], a Dryden spectral representation is used to generate turbulence by filtering band-limited white noise with appropriate forming filters (a Von Karman model could also be chosen). First and second-order filters are implemented depending on the velocity components. The colored noises added to the A/C measurements result from a first-order low-pass filtering, with cut-off frequencies and standard deviations tuned up for each subset of signals (angle of attack α/sideslip angle β, Euler angles $\Phi/\Theta/\Psi$, angular rates $p/q/r$, airspeed V, load factors $n_x/n_y/n_z$, etc.).

Concerning unknown events, the simulator enables also to model flight conditions with an iced aircraft. For that purpose, variations of the aerodynamic coefficients are computed by an additive icing model, which allows to represent a wide range of situations: no ice, icing with the deicing system on or off, level of ice more or less severe.

The supervisor is simulated on the second PC by a Petri-Net player ProCoSA©. See for example Fig. 4 for the encoding of the flight plan management.

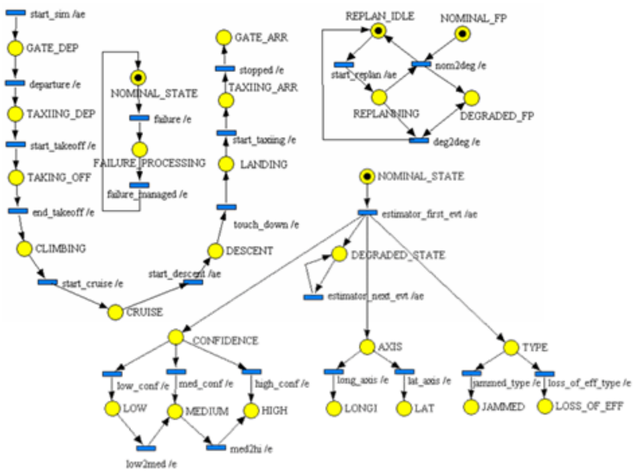

Fig. 4 Flight plan monitoring encoded via Petri-Net

The 3-D visualization of the A/C motion is performed using Flight Gear© on the third PC.

The A/C trajectory is shown with respect to the earth reference, here a flight over the sea. The control surface deflections are amplified and visualized at the same time. The fault status of the surfaces is indicated by a color code. A possible status is depicted in Fig. 5. Healthy control surfaces are plotted in green, here the rudder and the left elevator. A failure has recently occured on the right elevator which is not yet detected. It is indicated in red just for information. A problem has already been detected involving the ailerons, but the faulty aileron is not yet isolated. After this first detection stage, all 4 ailerons are hence plotted in orange. After failure isolation and estimation, the faulty aileron will be plotted in yellow, the healthy ones being plotted in green again.

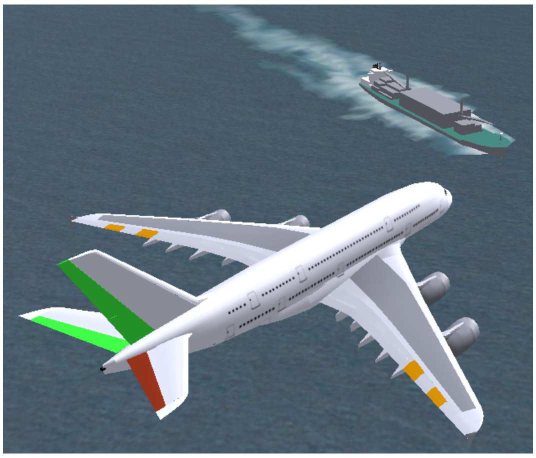

Fig. 5 3-D visualization of A/C motion and the fault status of its control surfaces

For the investigation of the pilot behavior in a fault situation, basic cockpit displays (Primary Flight Display (PFD), Horizontal Situation Indicator (HSI) and Engine Indication and Crew Alerting System (EICAS)) have been developed and enhanced with a fault monitoring display, serving for a better cueing of pilots (Fig. 6). See also [4, 10] for other possible visualizations.

3 Actuator and Control Surface Fault models

The IMMUNE model has been equipped with the possibility to simulate the effects of control sur-

Fig. 6 2-D visualization of the fault status of A/C control surfaces

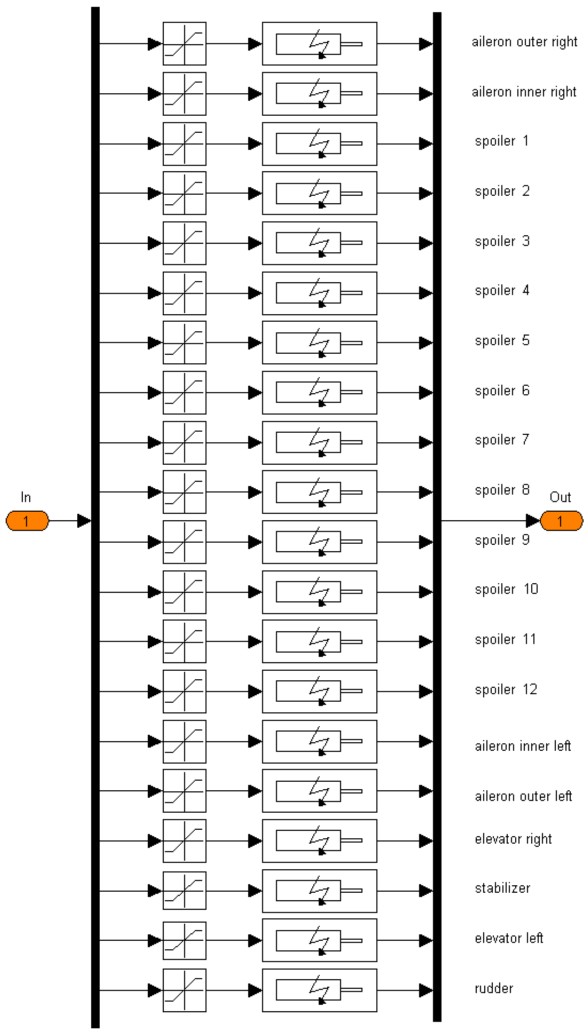

Fig. 7 Actuation Fault Model

face actuation and efficiency faults. The fault modelling was partly inspired by the crash of EL-AL flight 1862 in the vicinity of Amsterdam on October 4^{th}, 1992, following the separation of two engines and subsequent damage to wing and ailerons [13]. Faults can be simulated for all control surfaces. Specifically, the following fault types are available for each control surface:

- Actuator Jam
- Actuator Runaway with arbitrary end position
- Control Surface Loss of Efficiency

Simulation of the individual faults is achieved by altering the original control surface position signal. A dedicated actuator fault block was inserted in the signal path of each control surface position signal as can be seen in Fig. 7. Depending on the selected fault type these blocks change the received input signal to display the respective fault characteristic at the block output.

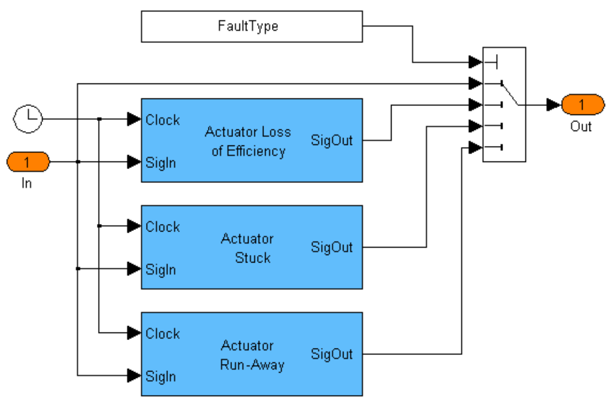

Fig. 8 Actuator Fault Simulation Block

Each actuator fault block can simulate all fault types as described above for each associated control surface. The fault block is depicted in Fig. 8. Each fault block can simulate one fault type at any given time. The activation of faults is time-triggered, giving the possibility to simulate the onset of faults at arbitrary simulation times. The implementation of the different fault types is described in the following sections.

3.1 Control Surface Loss of Efficiency

The efficiency of an aircraft control surface does, among other influences, directly depend on its physical shape. In case that this should be altered due to a mechanical damage or partial loss of the surface this also has a direct effect on the control surface aerodynamic efficiency. The contribution of each control surface to the total control momentum acting on the aircraft is linear dependant on the actuator (i.e. control surface) position according to the law

$$\Delta M_{cs} = C_{cs} \Delta\delta_{cs} l_{cs} \quad (1)$$

where ΔM_{cs} represents the generated control moment, C_{cs} the control surface aerodynamic efficiency, $\Delta\delta_{cs}$ the surface deflection and l_{cs} the distance from the aircraft center of mass to the aerodynamic center of the control surface. Due to this relation, a loss of efficiency can also be interpreted as a scaling of the control surface deflection.

Therefore, the effect of such a loss of aerodynamic efficiency can be modelled with good accuracy by limiting the possible actuator movement itself. In terms of simulation functionality, an efficiency loss of 50% after fault onset means that the actuator driving the control surface will only reach half of its commanded stroke. Adaption of the aerodynamic model properties during runtime of the model is therefore not required.

In order to implement this fault type a scaling factor is applied to the actuator position signal entering the fault block after the fault onset. The scaling factor has the effect of a simple gain inserted into the original actuator signal path. It should be noted that by using this modelling approach, the actuator position signal at the output can not be interpreted as the representation of the real actuator stroke any more. This is due to the fact that the block introduces a sharp jump from the current to the scaled actuator position value in the moment of fault onset. However, this modelling approach is sufficiently accurate for studying the effects of a control surface loss of efficiency fault and for validation of FDI algorithms.

3.2 Actuator Jam

This fault type means that the actuator and its associated control surface is jammed or stuck in its current position. The actuator signal entering the fault block will remain in the position it assumed in the moment of the fault onset. The position of the surface after the jam can therefore only be influenced by proper selection of the fault onset time but not by defining a specific fault position. In order to bring a control surface into a particular position the runaway fault can be used which will be introduced more detailed in the following section.

3.3 Actuator Runaway with arbitrary end position

Runaway of an actuator are among the most critical failure modes and are common to almost all types of servo hydraulic actuation systems. During a typical runaway fault, the servo valve controlling the actuator is blocked in the open position by a mechanical defect causing the actuator to extend to one of its end positions where it will remain if the actuator is not passivated by switching it into a bypass mode. However, in order to provide more flexibility for the IMMUNE fault simulation it was decided to include an optional and arbitrary end position for the runaway. Since the end position of the runaway is customizable this fault type can also be used to simulate the effects of a free floating control surface, i.e. an actuator disconnect from the control surface, by defining zero as the end point of the runaway. Due to the linear dependency of control surface moment and displacement as introduced above, nulling the control surface position permanently will effectively result in the inability to generate control moments with this control surface. Again, in this case the control surface deflection visible at the output of the fault block does not represent its real, physical behaviour. In reality the surface would not be fixed in its neutral position in such a case but always be oriented parallel to the local flow field.

As an alternative, this fault type is also well suited to bring the actuator and its associated control surface into a defined position that might be required for analysis of a particular scenario.

4 Developed FDIE methods

In the following, two studied methods and the corresponding results are briefly presented. The papers [3, 5, 10] present in a more detailed manner other FDIE methods developed within the IMMUNE framework.

4.1 The bank of linear residual filters for actuator FDIE

The FDD part usually includes the generation of residual signals [12], (norm-based) evaluation of residual signals and (thresholds-based) decision making.

The design of FDD components relies on recently developed numerically reliable algorithms and dedicated robust numerical software tools. Applications of these techniques and tools for the monitoring of primary actuator faults of large transport aircraft are reported in [11]. For the setup of different parts of the FDD system (i.e. residual generation, residual evaluation and decision making), but also for controller reconfiguration, a generic Simulink blockset allowing rapid prototyping has been developed.

Fig. 9 shows a typical actuator fault situation (stuck) and the resulting FDD process signals. Before the actuator fault occurs at $t_{f,act}$, the actuator output signal y_{act} follows the input signal u adequately. During the presence of the fault ($t_{f,act} < t \leq t_{f,dea}$) the actuator (and therefore the aileron) is stuck. As soon as the evaluated residual $\theta(t)$ exceeds its predefined threshold J_{th} the fault is detected ($t_{f,det}$). Notable is the rather short detection time, indicating satisfactory performance of the used method. The detection of the fault triggers the fault identification process, which is successfully accomplished at $t_{f,id}$. When the fault disappears ($t_{f,dea}$), still no inputs are commanded to the actuator, as the systems requires some time to 'recognize' that no fault is present any more. However, during this period ($t_{f,dea} < t < t_{norm}$) the actuator remains at its neutral position, not generating any undesired moments. Finally, the aileron is used again by the control system to generate the required moments ($t_{f,norm}$).

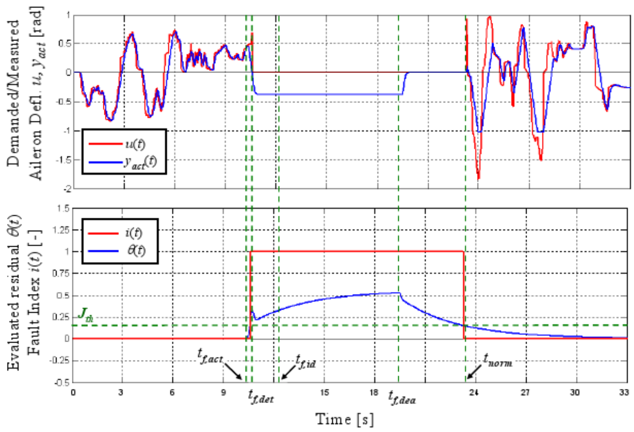

Fig. 9 Aileron actuator fault with FDD-process signals

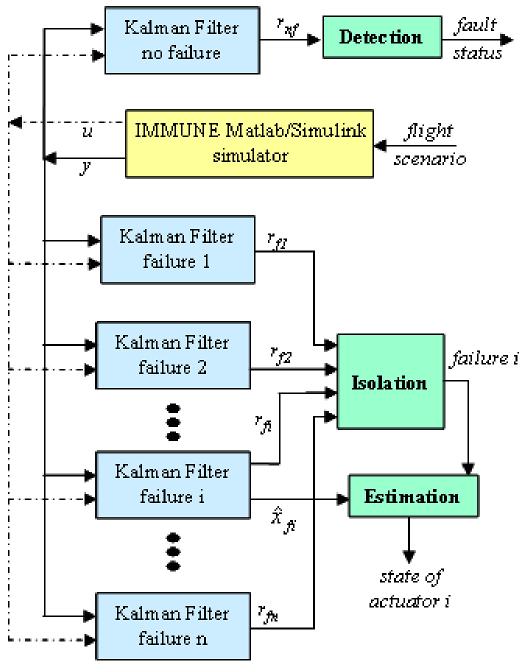

Fig. 10 General scheme of the method - detection step as well as isolation and estimation step

4.2 A non-linear filtering technique for actuator FDIE

Another actuator fault detection, isolation and estimation method, based on non-linear filtering techniques, has been developed. The approach is summarized in the scheme of Fig. 10. Detection is performed using an extended Kalman filtering technique based on a nonlinear representation of aerodynamics (on-board modeling). A χ^2-type detection criterion using the prediction error completes the algorithm.

Once a fault is detected, isolation is realized in a hierarchical approach via a bank of Kalman filters running in parallel, each one based on a nonlinear model representative of a particular fault, see e.g. [2], and by using an hypothesis testing algorithm combining probabilities of each hypothesis (computed from innovation).

Finally, estimation of the control surface effectiveness or stuck value is performed thanks to an augmented state vector. In order to tune the method, realistic flight scenarios have been run using the common desktop environment of §2, including realistic measurement noises, turbulence wind and parametric uncertainties in the non-linear model. This approach detects, isolates and estimates in a robust manner stucked actuators or runaways.

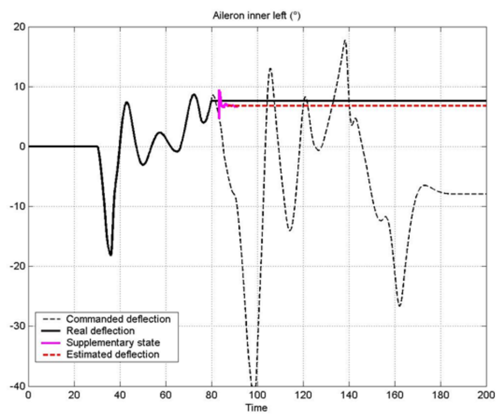

Fig. 11 Estimation of the stucked position

To illustrate this powerful approach, Fig. 11 shows the time history of a typical failure scenario. The left inner aileron jammed at $80s$ at its actual position (black line). The failure is first detected after $3s$ and isolated after another $5.8s$. The additional state (magenta line) allows at the same time the estimation of the stucked position. The estimated control surface deflection is then plotted in a dashed red line. The estimation error is of about 6.6% or 0.5^o which is completely sufficient for following FTC methods.

5 Developed FTC methods

In the following, two studied methods and the corresponding results are briefly presented. The papers [6, 10] present in a more detailed manner other FTC methods developed within the IMMUNE framework.

5.1 Off-line indirect adaptive control

The aim was to develop an off-line indirect adaptive control law for the longitudinal and lateral motions around a trim point. Firstly, a linearized model is estimated off-line using closed loop signals I/O signals obtained with the nonlinear simulator and the initial control law with a single experiment corresponding to a classical reference input signal (a series of steps in our application). Here, the main validation criteria is not to get an accurate estimated model, but to obtain satisfactory performance when designing the control law with the estimated model.

Secondly, a static feedback gain is synthesized using this estimated model, and validated on the nonlinear simulator. The gains are synthesized using a modal approach, see [8]. The main tuning parameters are the closed loop poles to be placed, and also a static feed-forward gain. Moreover, the choice of eigenvectors appears crucial in the lateral case. A reference performance is given by the nonlinear time-domain simulations obtained with the initial control law.

This off-line adaptive control scheme can be seen as a preliminary stage for an on-line adaptive control scheme, see [1] for a first application to a linearized longitudinal transport aircraft model.

5.2 Reconfiguration by adaptive dynamic inversion

Here, it is assumed that the aircraft parameter variations are well estimated by the methods proposed in §4. An adaptive dynamic inversion approach is used for the reconfiguration of the control laws following primary flight control surface failures, more precisely actuators stuck at trimmed/untrimmed deflection with or without missing entire or partial surfaces. The overall reconfiguration approach is depicted in Fig. 12.

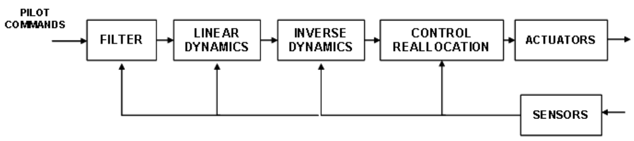

Fig. 12 Reconfiguration by Adaptive Dynamic Inversion

It consists of:

- redistributing the pilot's commands in order to give more control authorities to the remaining healthy actuators and effectors,
- optimizing the reallocation by inverting the instantaneous moment demand in the attainable moment subset (example shown in Fig. 13)
- adjusting the feedback and feed-forward gains to compensate for deviations of the aircraft state parameters from a reference model.

Fig. 14 presents the simulation of a reconfiguration following a hard-over failure of the stabilizer (ih) using the left and right elevators (dq). The transient response of the aircraft shows a good recovery from this failure scenario. A similar approach called Adaptive Nonlinear dynamic inversion (ANDI) was also proposed by [7].

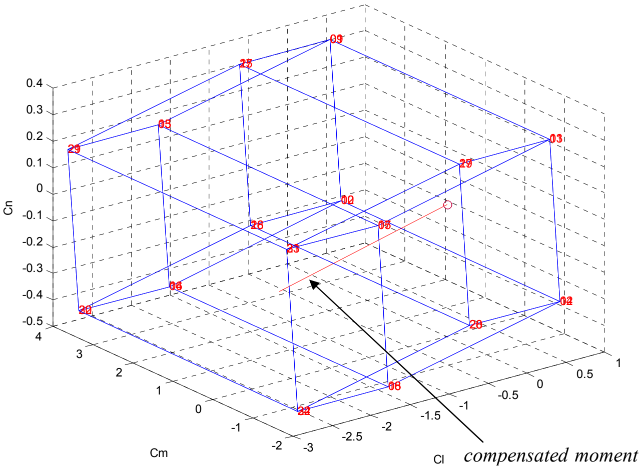

Fig. 13 Attainable moment subset

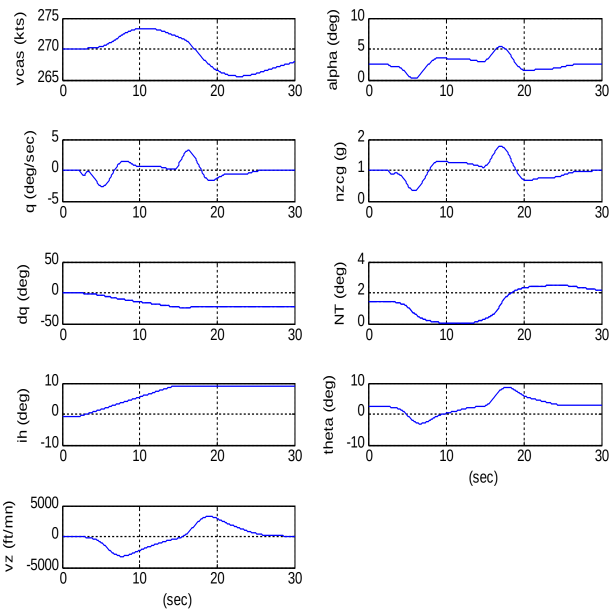

Fig. 14 Reconfiguration after stabilizer runaway

6 Conclusion

FDIE and FTC are at the moment very important research domains, especially for aeronautical applications, see for example NASA's Validation & Verification of Safety Critical Systems project. The IMMUNE (Intelligent Monitoring and Managing of UNexpected Events) project contributed to this research domains by focusing on basic research regarding event monitoring and event tolerant control.

In this paper, the common simulation environment was first presented in detail. Many event detection methods and event tolerant control approaches have been investigated. For promising methods prototype software tools have been implemented into the common simulation environment. Some FDIE and FTC methods studied during the project are briefly introduced within this paper. Another FDIE and another FTC method are presented in a more detailed manner in two other papers of the same session. Identification and control specialists worked for the first time jointly on a common application and learned a lot about the interconnections between their respective methods. ONERA and DLR teams will profit from this experience and continue to improve the interfaces between FDIE and FTC methods in upcoming projects.

7 Acknowledgements

We want to acknowledge the work of AIRBUS/EDYC team in the development of the initial version of the Matlab/Simulink© desktop environment used for the A/C simulation. We also want to thank B. Jacquier, G. Ferreres, J.F. Gabard and B. Dang-Vu (all ONERA) and S. Hecker (DLR) for their respective inputs.

8 Copyright Statement

References

[1] Antoinette P and Ferreres G. Robust gain scheduling techniques for adaptive control. *Proc European Control Conference*, Budapest, Hungary, August 2009.

[2] Ducard G and Geering H. A reconfigurable flight control system based on EMMAE method. *Proc American Control Conference*, Minneapolis, Minesota, USA, June 2006.

[3] Hardier G and Bucharles A. On-line parameter identification for in-flight aircraft monitoring. *Proc 27th ICAS Congress*, Nice, France, September 2010.

[4] Hecker S, Varga A, and Looye G. A desktop environment for assessment of fault diagnosis based fault tolerant flight control laws. *Proc IEEE International Symposium on Computer Aided Control Systems Design*, San Antonio, Texax, USA, September 2008.

[5] Kappenberger C and Christmann J. Using simulation-based filtering approaches for in-flight monitoring of control surfaces. *Proc 27th ICAS Congress*, Nice, France, September 2010.

[6] Lafourcade L, Cumer C, and Döll C. IMMUNE : Control reallocation after surface failures using model predictive control. *Proc 27th ICAS Congress*, Nice, France, September 2010.

[7] Looye G, Thümmel M, Kurze M, Otter M, and Bals J. Nonlinear inverse models for control. *Proc 3rd International Modelica Conference*, Hamburg, Germany, 2005.

[8] Magni J. F. *Robust Modal Control with a Toolbox for Use with Matlab*. Kluwer academic/plenum publishers edition, 2002.

[9] MIL. *US Military Handbook MIL-HDBK-1797*, 1997.

[10] Ossmann D, Hecker S, and Varga A. A versatile simulation environment of FTC architectures for large transport aircraft. *Proc 27th ICAS Congress*, Nice, France, September 2010.

[11] Varga A. Fault detection and isolation of actuator failures for a large transport aircraft. *Proc 1st CEAS European Air and Space Conference*, Berlin, Germany, 2007.

[12] Varga A. On designing least order residual generators for fault detection and isolation. *Proc Proc. of the 16th Int. Conf. on Control Systems and Computer Science*, pp 323–330, Bucharest, 2007.

[13] Wolleswinkel H. On the accident with EL-AL 1862 on October 4^{th}, 1992 at Amsterdam - Bijlmermeer. Final report, Netherlands Aviation Safety Board, 1994.