A VERSATILE SIMULATION ENVIRONMENT OF FTC ARCHITECTURES FOR LARGE TRANSPORT AIRCRAFT

Daniel Ossmann , Simon Hecker , Andras Varga
***German Aerospace Center (DLR), 82234 Wessling, Germany**

Keywords: *Fault Detection, Fault Tolerant Control, Flight Control, Aircraft Simulation*

Abstract

We present a simulation environment with 3-D stereo visualization facilities destined for an easy setup and versatile assessment of fault detection and diagnosis based fault tolerant control systems. This environment has been primarily developed as a technology demonstrator of advanced reconfigurable flight control systems and is based on a realistic six degree of freedom flexible aircraft model. The aircraft control system architecture includes a flexible fault detection and diagnosis system and a reconfigurable nonlinear dynamic inversion based controller, able to handle different fault situations.

1 Introduction

The complexity of technical systems has been continuously growing in the last years to face various challenges like highly optimized performance, increased safety demands, reduced design and operation costs. A typical challenging application is the design of advanced flight control systems (FCS) for large transport aircraft aiming to significantly reduce the workload of pilots, ensuring best handling qualities simultaneously with increased passenger comfort and safety. Despite a high system complexity, the fault tolerant operation of an aircraft has to be guaranteed over the whole flight envelope in presence of many possible unexpected events and inherent uncertainties in the operation environment. To ensure flight safety and increase flight autonomy, the aircraft industry traditionally uses (physical) actuator and sensor redundancy. However, this hardware-redundancy based *fault detection and diagnosis* (FDD) approach is becoming increasingly problematic when used in conjunction with the many innovative technical solutions being developed by the aeronautical sector to satisfy the *greener* and *safer* imperatives demanded by the society.

In the recent years, to alleviate this safety bottleneck, efforts have been invested to develop *fault tolerant control* (FTC) architectures which strongly rely on advanced model based FDD techniques. Developing high performance FTC is an interdisciplinary activity encompassing aspects of advanced control techniques (e.g., adaptive, reconfigurable, robust), fault detection and isolation, multi-objective optimization, as well as real-time implementation of FDD-based FTC schemes. Each of these disciplines is a field of intensive research itself. Therefore, the successful application of FTC is not a trivial task and requires an optimal combination of various aspects.

A typical FTC architecture based on FDD techniques is depicted in Fig. 1. The FDD part usually includes the generation of the residual signals [2], (norm-based) evaluation of residual signals and (thresholds-based) decision making [8]. The fault identification part determines the fault type and size, and triggers the appropriate reconfiguration of the controller, to ensure a safe system operation in the presence of faults.

In this paper we describe a dedicated desktop environment developed at DLR for the setup, simulation-based assessment and optimization-based tuning of FTC architectures as that pre-

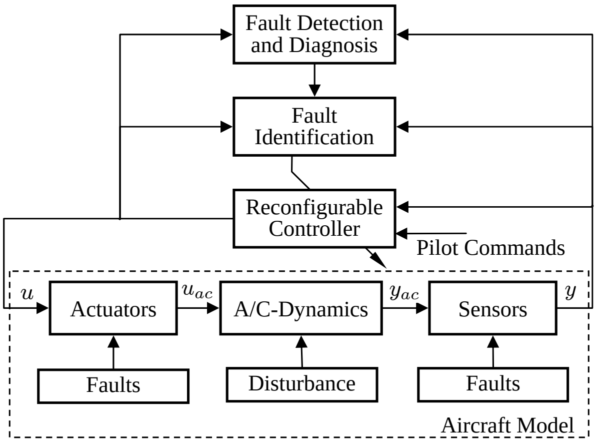


Fig. 1 Fault diagnosis based fault tolerant control

sented in Fig. 1. In section 2 the detection, isolation and identification algorithms used to fulfill different fault monitoring tasks are discussed in detail. Further, the corresponding generic Simulink blockset allowing rapid prototyping of the fault monitoring system is covered in this section. An overview of the simulation environment including a realistic six degree of freedom flexible aircraft model, together with its fault models, a complete fault monitoring system and a fault tolerant controller is presented in section 3. Section 4 covers some simulation results and finally, the outlook in section 5 briefly describes planned enhancements and improvements.

2 Fault Monitoring

Fault monitoring in a system can be done in different degrees of accuracy, depending on whichever information is required. Basic *fault detection* algorithms provide information of the occurrence of any fault in the system. Advanced fault detection and isolation algorithms are able to exactly localize the fault in the system. Fault detection together with fault isolation are often referred to as *fault diagnosis*. A further step is the so called *fault identification* providing qualitative and quantitative information on the occurred fault (e.g., position at which an actuator is stuck). In many cases fault tolerant control is only possible if the exact information on the nature and magnitude of the fault is available. This confers fault identification an essential role in any fault tolerant control system.

Fig. 2 presents an overview of the main tasks of a fault monitoring system. Typically, such a system processes the controlled input vector u and the measured output vector y and delivers estimations of the values of a fault signal f. The residual signal r is generated by a residual generator to indicate the presence or absence of a fault. For this purpose, a residual evaluation signal θ (e.g., an approximation of the norm of r) serves as basis for the decision making, where θ is compared to a pre-defined threshold. The resulting decision signal i (e.g., indicating the localization of the fault) may trigger fault identification processing to finally calculate the value f of the fault in the system.

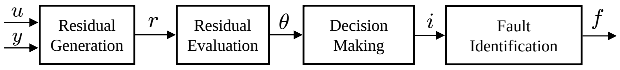


Fig. 2 Main tasks of fault monitoring

A detailed description of the algorithms used to accomplish theses steps and an overview of the corresponding generic Simulink blockset is given in the following subsections.

2.1 Residual Generation

Residual generators shall provide residual signals indicating the occurrence of faults in a system, using the systems input u and output y. In the presence of disturbances, noise and uncertainties the design of such residual generators can become a highly demanding task. In the ideal case the residuals are decoupled from these effects and are highly sensitive only to the faults. Depending on the purpose of the FDD system, a residual generator with a scalar output may comply with the need of fault detection, whereas a bank of residual generators may be required for fault detection and isolation. Various different approaches have been developed in recent years to solve the fault detection and isolation problem. For the design of the residual generators presented in this paper we rely on recently developed numerically reliably algorithms and dedicated robust numerical software tools developed within DLR [2], [4].

In general, residual generators can be designed to work either on a system-wide or on component level, both possibilities showing advantages and disadvantages in their implementation and application. System-wide approaches in aircraft fault monitoring show their strength especially in cases when certain control surfaces are damaged, i.e. their efficiency is degraded, as the position sensors within actuators are not able to provide information about the surface's efficiency. Also, the estimation of critical sensor faults can be done only on system-wide level. The main drawback of using system-wide residual generators is the large number of uncertainties (e.g., uncertain aerodynamics) and noise sources (gusts, measurements) they have to robustly cope with, which represents a real challenge for the design algorithms.

Uncertainties and noise are also existent for residual generators working on component level, however on a much lower level. Furthermore, models of single components are usually less complex than the overall (nonlinear) model of an aircraft, such that the complexity and design effort for component-level residual generators may be reduced. For example, they are well suited for monitoring some actuator faults (e.g., stuck, runaway) and are therefore used in the FDD system of the underlying simulation model presented in this paper. On the other hand, component level approaches assume the availability of perfect local sensor data, and therefore may have difficulties discerning between real component faults and sensor induced faults. For the simulation example presented later in this paper the focus lies on a component level FDD approach, where fault isolation can be achieved for each monitored component (actuator) for specific classes of faults.

2.2 Signal Evaluation and Decision Making

The evaluation of the residual signal often requires the computation of a measure of the residual signal energy, for which the 2-norm of the signal is usually an appropriate choice. Two approximations of the 2-norm are implemented in the fault monitoring system of the simulation model. In what follows we describe them for continuous signals. Similar evaluation signals can be computed for sampled signals as well.

The so called sliding-window evaluation signal

$$\theta(t) = \alpha r^2(t) + \beta \int_{t-T}^{t} r^2(\tau) d\tau, \qquad (1)$$

approximates the 2-norm on a finite sliding-window of width T, where $\alpha \geq 0$ and $\beta \geq 0$ are weights for instantaneous and long-term values, respectively. For a real time implementation, (1) can be expressed as the output of a first order delay-differential equation

$$\begin{array}{rcl} \dot{\rho} & = & r^2(t) \\ \theta(t) & = & \alpha r^2(t) + \beta(\rho(t) - \rho(t-T)). \end{array} \qquad (2)$$

This representation is used for the implementation of the evaluation algorithms in the Simulink blockset.

Alternatively, the so called Narendra signal evaluation can be used

$$\theta(t) = \alpha r^2(t) + \beta \int_{0}^{t} e^{-\gamma(t-\tau)} r^2(\tau) d\tau, \qquad (3)$$

which can be generated by a first order differential equation

$$\begin{array}{rcl} \dot{\rho}(t) & = & -\gamma\rho(t) + \beta r^2(t) \\ \theta(t) & = & \rho(t) + \alpha r^2(t), \end{array} \qquad (4)$$

where γ is the forgetting factor, playing a similar role as the time window width T in (1). The signal $\theta(t)$ is compared to a specific threshold J_{th} and is ideally equal to zero or sufficiently small in fault free situations, whereas it shall exceed the threshold J_{th} when a fault occurs in the system. Hence, the appropriate selection of the values of the free parameters α, β, and T or γ, together with an appropriate threshold J_{th} essentially influences the performance of the FDD system. Typical performance criteria in this context are the minimum detectable fault magnitude, the false alarm rate, the missed detection rate or the detection delay. Multi objective optimization may be used to tune these parameters and to optimize the above performance criteria.

The decision making process finally determines if a fault is present in the system or not by comparing the evaluation signal $\theta(t)$ to the threshold J_{th}, where the decision logic

$$\begin{array}{lcll} \theta(t) & < & J_{th} \Rightarrow i(t) = 0 \Rightarrow \text{no fault} \\ \theta(t) & \geq & J_{th} \Rightarrow i(t) = 1 \Rightarrow \text{fault} \end{array} \tag{5}$$

is used. If a residual generator with multiple outputs r_k, $k = 1,...,q$ is designed, each evaluation signal $\theta_k(t)$ is compared to its corresponding threshold $J_{th,k}$ and the components of the vector $i(t)$ are set according to the following logic

$$\begin{array}{lcl} \theta_k(t) & < & J_{th,k} \Rightarrow i_k(t) = 0 \\ \theta_k(t) & \geq & J_{th,k} \Rightarrow i_k(t) = 1 \end{array} \tag{6}$$

If so called *strong fault isolation* is provided by the residual generator, each component r_k of the residual vector r indicates the presence or absence of the corresponding fault f_k of the fault vector f (of dimension q) and is decoupled from all other faults. The resulting nonzero components of the vector $i(t)$ thus directly indicate which faults are present.

If only *weak fault isolation* is provided by the residual generator, the resulting decision vector $i(t)$ has to be additionally compared to the columns of a $q \times m_f$ fault signature matrix S, where m_f is the number of faults (or fault combinations) which are coded in S. For example, a matching with column k of S may indicate the presence of fault f_k (see [8] for further details). Multiple faults may not always be isolated with residual generators providing weak fault isolation because of the inherent coupling of the faults and residuals, due to the lack of sufficient output measurements. However, while residual generators providing strong fault isolation can cope with multiple faults, strong fault isolation is not always possible, as the achievable decoupling mainly depends on the available number of measurements.

The evaluation and decision methods presented above are included in a recently developed Simulink blockset [9] , which includes additional enhancements as decision schemes based on more sophisticated weak isolation algorithms (e.g., r_k are vectors instead of scalars) and discrete time processing counterparts. The rich functionality of this blockset allows to easily generate and implement FDD systems closely adapted to the characteristics of the underlying models. In addition, m-functions have been implemented, to automatically generate parts or even the whole Simulink model of the FDD system. Especially when dealing with several residual generators running in parallel (e.g., in a multi-model based approach), these functions have proven their usefulness.

2.3 Fault Identification

Providing complete information on the characteristics of occurred faults is of paramount importance for the ongoing safe system operations of fault tolerant control schemes. The task of fault identification is to gather such quantitative and qualitative information on faults. Fault identification can be performed using various techniques, as for example, on-line parameter estimation, feature extraction using digital signal processing (e.g., fast Fourier or wavelet transforms), signal detection methods, or even multi model based model detection [5]. Statistical signal processing methods based on mean and covariance computations are frequently employed to identify simple faults like stuck devices (actuator/sensor), runaways, or efficiency losses of actuators. The detection of oscillatory failure case of flight actuators can be addressed using specialized spectral analysis techniques [6].

3 Simulation Environment

Fig. 3 shows the block diagram of the implemented fault tolerant control system architecture built around a highly sophisticated model of a flexible aircraft. The aircraft model also includes actuator and sensor models with additional fault inputs, to allow the injection of various fault signals (see also Fig. 1). A complete FDD and fault identification system is part of the FTC architecture and serves for triggering controller reconfiguration actions. The reconfigurable controller is based on nonlinear dynamic inversion (NDI) and enables safe operations for several fault scenarios.

The aircraft behavior can be visualized in real-time with a 3-D stereo system. Besides the full autonomic modes the aircraft can be flown manually, hence, a high quality 2-D digital instrument panel has been developed including a fault monitoring display. The injection of fault signals can be activated from the fault control panel over a computer network connection. The single elements of the environment are described in more detail in the following.

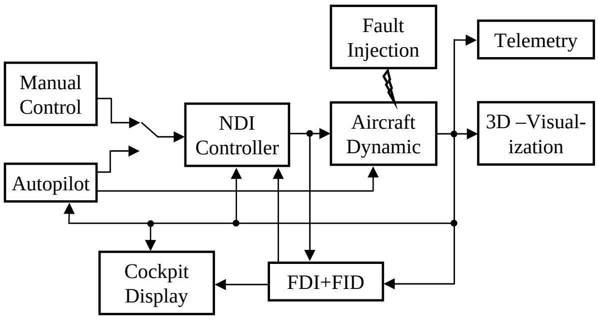


Fig. 3 Fault Tolerant Control Simulation Setup

3.1 Aircraft Model

The structure of the generic closed-loop aircraft simulation model is shown in Fig. 1. The aircraft input vector u of dimension 24 includes the demanded deflections of 2 outer ailerons (left/right wing), 2 inner ailerons (left/right wing), 12 spoilers (6 on the left wing/ 6 on the right wing), 2 elevators (left/right), one horizontal stabilizer, one rudder and four engine throttles positions (left/right). The aircraft model consists of the actuators block (including actuator dynamics, actuator saturations and the actuator fault models), the aircraft block (including flight mechanics, structural dynamics, aerodynamics, propulsion, environment) and the sensors block (including the sensor dynamics and the sensor fault models). The output vector y includes all measurable variables required to control and monitor the aircraft, as well as the measurements provided to the pilot via the cockpit display.

3.1.1 Actuators with Faults

The generic, nonlinear aircraft model includes several redundant control surfaces in all three axes enabling the accommodation of actuator and sensor faults. Each actuator model is a first order system with the commanded surface deflection u as input and the resulting surface deflection u_{ac} as output. Furthermore typical actuator faults like "stuck actuator", "actuator runaway" and "loss of efficiency" are modeled as additional inputs f to the model. The corresponding LTI model is given by

$$\begin{aligned} \dot{x}_{act} &= \tau^{-1}(-x_{act}+u+f) \\ u_{ac} &= x_{act} \end{aligned} \tag{7}$$

where the value of the time constant τ varies from actuator to actuator for different control surfaces. The transfer functions from u to u_{ac} and f to u_{ac} are thus equal to

$$G_u(s) = \frac{1}{\tau s + 1}$$

Furthermore, upper and lower position limitations u_{max} and respectively u_{min} on the actuator input $u(t)$ and a rate saturation $\dot{u}_{max}$ on $\dot{u}(t)$ are also parts of the actuator model, so that

$$\tilde{u} = F(u, u_{max}, u_{min}, \dot{u}_{max}) \tag{8}$$

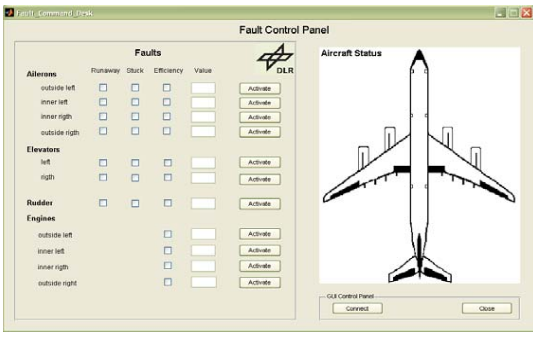
is the actual input applied instead u via a generally nonlinear mapping F.

To monitor actuator faults at component level, a simple linear residual generator as presented in (7) is used with the input-output form

$$\mathbf{r}(s) = \mathbf{u}_{ac}(s) - G_u(s)\tilde{\mathbf{u}}(s). \tag{9}$$

where $\mathbf{r}(s)$, $\mathbf{u}_{ac}(s)$ and $\tilde{\mathbf{u}}(s)$ are Laplace-transformed vectors. Notice that due to the presence of position- and rate limitations within the actuator model, the actual input $\tilde{u}$ is used instead u.

To inject fault signals into the actuator models, a fault control panel has been developed to define and activate external fault control inputs for actuators as shown in Fig. 4. This feature is also helpful when fulfilling pilot-in-the-loop missions, as the faults can be activated independently by a supervising person and the resulting pilot behavior can be investigated.



Fault Control Panel

Faults

	Runaway	Stuck	Efficiency	Value	
Ailerons					
outside left	□	□	□		Activate
inner left	□	□	□		Activate
inner rigth	□	□	□		Activate
outside rigth	□	□	□		Activate
Elevators					
left	□	□	□		Activate
rigth	□	□	□		Activate
Rudder	□	□	□		Activate
Engines					
outside left			□		Activate
inner left			□		Activate
inner rigth			□		Activate
outside right			□		Activate

Fig. 4 Fault Control Panel

3.1.2 Fault Tolerant Controller

The nominal controller has six outputs consisting of one roll, one pitch, one yaw command, two engine throttle commands and a speed brake command. The basic controller is based on nonlinear dynamic inversion (NDI) and is designed to ensure adequate flying and handling qualities in fault-free situations. A control distribution block has been implemented to distribute the six commands to the 24 actuator inputs. Control reconfiguration actions (as a results of fault occurrence) can be easily performed by changing the entries of the underlying control distribution matrix.

The control reconfiguration happens in two steps. As soon as the FDD algorithms detect and isolate a fault, the distribution block of the controller is reconfigured so that the affected control surface does not receive any commands from the controller any more. The required forces and moments are allocated to the remaining control devices. If the fault can be identified in a second step, the controller compensates the identified fault value in its feed-forward path and handles it like a measurable disturbance, further improving the aircraft performance in the fault situation [1]. While this reconfiguration approach leads usually to higher workloads of the fault-free surfaces and may even result in an asymmetric behavior, still it ensures a sufficient controllability of the aircraft by exploiting the available actuation redundancy.

3.2 Visualization

Besides a striking illustration of the aircraft status, the visualization must allow a better and more intuitive understanding of the controlled aircraft behavior. The aircraft and its environment is visualized with a highly sophisticated 3-D stereo visualization tool, enabling visualizations in real time, as depicted in Fig. 5. The visualization has been adjusted for fault monitoring purposes so that control surface with and without failures are clearly distinguishable. For example, the inner left aileron in Fig. 5 is faulty and therefore displayed in red.

To be able to manually fly the aircraft basic cockpit displays (Primary Flight Display (PFD), Horizontal Situation Indicator (HSI) and Engine Indication and Crew Alerting System (EICAS)) have been developed. For a better cueing of pilots and to allow monitoring of the pilot actions in fault situations the displays have been enhanced with a fault monitoring display as shown in Fig. 6 [7].

4 Simulation Results

Fig. 7 shows a typical actuator fault situation (stuck) and the resulting FDD signal processing. Before the actuator fault occurs at $t_{f,act}$, the actuator output signal u_{ac} follows the input signal u. During the presence of the fault ($t_{f,act} < t \leq t_{f,dea}$) the actuator (and therefore the corresponding control surface) is stuck. As soon as the residual evaluation signal $\theta(t)$ exceeds its predefined threshold J_{th}, the fault is detected ($t_{f,det}$) and the first stage of the controller reconfigura-

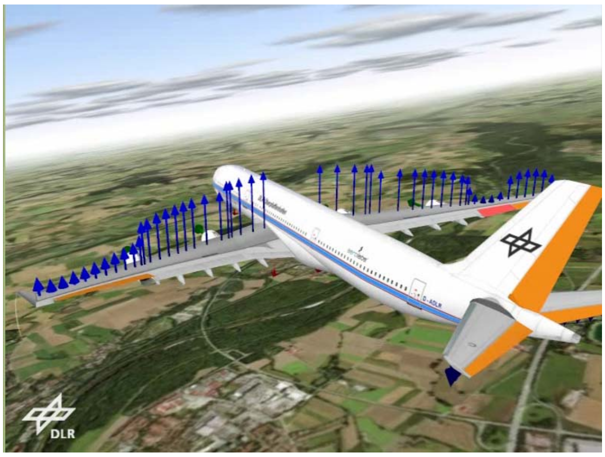


Fig. 5 3-D Aircraft Visualization

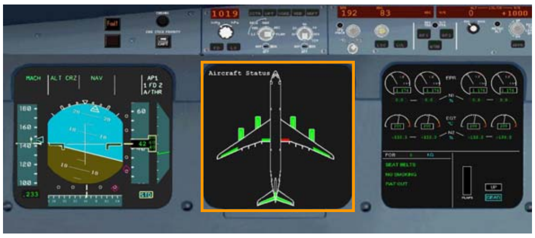


Fig. 6 Cockpit Visualization with Fault Display

tion (see section 3.1.2) is initiated, so that the isolated device is excluded from any control actions ($u = 0$). The rather short detection time indicates a satisfactory performance of the used detection method. The detection of a fault also triggers the fault identification process, which is successfully accomplished at $t_{f,id}$, when the second phase of the controller reconfiguration is initiated. In this phase, the fault is handled as a measurable disturbance and is canceled in the feed-forward path of the controller. 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 control system allocates the demanded deflections to the control surfaces, as fault free operation of the surfaces is possible again ($t_{f,norm}$).

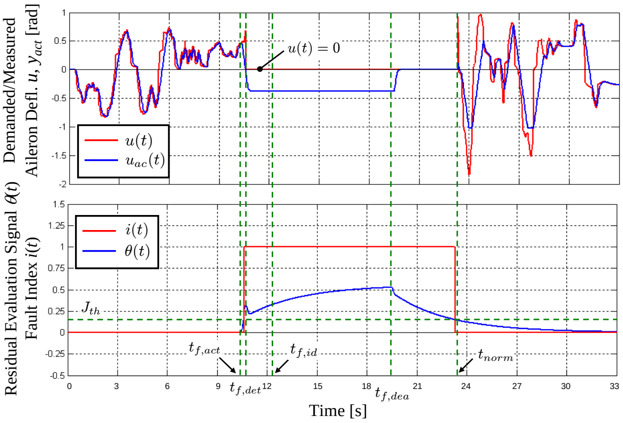


Fig. 7 Process and FDD signals for an actuator fault

5 Outlook

As certain faults are not detectable on a component level due to for example the lack of a sufficient number of sensors, future work will also investigate detection filters designed for a system wide approach. On this level robustness and model uncertainties play a bigger role than on the component level and the designed residual generators must be robust with respect to these uncertainties.

Furthermore, global optimization techniques will be used to find an optimal setting of the free parameters within the FDD system to improve the performance (e.g. false alarm fate, detection delay), which may involve worst case search and multi objective optimizations.

The environment can also serve for the clearance of an FTC-based flight control systems, which involves the robustness analysis of reconfigured system stability and performance, and of the reconfiguration performance in the presence of parametric and flight condition uncertainties.

6 Acknowledgment

The desktop simulation environment has been developed in the framework of the joint ONERA-DLR IMMUNE (Intelligent Monitoring and Management of Unexpected Events) project. We acknowledge the contributions by the teams of both companies during the different phases of the

project.

7 Copyright Statement

References

[1] G. Looye, M. Thümmel, M. Kurze, M. Otter and J. Bals. Nonlinear Inverse Models for Control. In *Proc. Third international Modelica conference*, *Hamburg*, 2005.

[2] A. Varga. Linear FDI-Techniques and Software Tools. FAULT DETECTION Toolbox V0.8 - Technical Documentation IB 515-08-18, German Aerospace Centnter (DLR), Institute of Robotics and Mechatronics, 2008.

[3] A. Varga. Monitoring acturator failures for a large transport aircraft - the nominal case. In *Proc. of IFAC Symp. SAFEPROCESS'2009*, *Barcelona*, 2009.

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

[5] A. Varga. Least order fault and model detection using multi-models. *Proc. of CDC'09, Shanghai, China*, 2009.

[6] P. Goupil. Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy. *Proc. of SAFEPROCESS'09, Barcelona, Spain*, 2009.

[7] S. Kuhlmann, I. Sturhan, A. Jaros and G. Sachs. Flexible and Efficient Tools for Cost-Effective Simulation Environment. *Proc. AIAA Conference*, *San Francisco*, 2005.

[8] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, 1998.

[9] S. Hecker, A. Varga and G. Looye. A desktop enviroment for assesment of fault diagnosis based fault tolerant control laws. In *Proc. IEEE CACSD Conference*, *San Antonio*, 2008.