A Desktop Environment for Assessment of Fault Diagnosis Based Fault Tolerant Flight Control Laws

Simon Hecker, Andras Varga and Gertjan Looye

***Abstract*— We present a simulation based software environment conceived to allow an easy assessment of fault diagnosis based fault tolerant control techniques. The new tool is primary intended for the development of advanced flight control applications with fault accommodation abilities, where the requirements for increased autonomy and safety play a premier role.**

I. INTRODUCTION

Fault tolerant control (FTC) based on *fault detection and diagnosis* (FDD) is a complex task including aspects of several disciplines like [1]

- fault detection, isolation and identification
- optimal, adaptive and robust control
- reconfigurable/restructurable control
- computing, communication, real-time implementation, simulation and display techniques.

Each of above disciplines is itself a field of active research. Therefore, the successful application of FTC is not a trivial task and requires an optimal combination of these disciplines.

The most frequent applications of FTC are in flight control, where stringent safety requirements have to be fulfilled before certifying an aircraft. For instance, the detection and accommodation of single faults is of critical importance since it can be seen as part of any civil aircraft specification according to the safety requirements of the main international flight regulation boards (FAA/FAR and EASA/CS). Of relevance for FTC is the requirement for the modern aircraft design that no single failure must lead to a catastrophic consequence.

In this paper we present a dedicated desktop environment which has been developed for the setup, simulation, optimization and evaluation of typical FDD based FTC architectures for flight control systems. In what follows we enumerate the main goals of the developed software environment:

1) Setup of a realistic benchmark model: A six degree of freedom generic, nonlinear rigid body aircraft model has been chosen, which provides a number of redundant control surfaces (ailerons, spoilers, elevators, a trimmable horizontal stabilizer, rudder, differential thrust) in all axis to be used for accommodating failures in actuator/surfaces or sensors. This model, augmented with actuator and sensor fault models, can serve for both real-time simulations/visualization as well as for various analysis task to be detailed in what follows.

Simon Hecker, Andras Varga and Gertjan Looye are with the Institute of Robotics and Mechatronics, German Aerospace Center, 82234 Wessling, Germany simon.hecker@dlr.de, andras.varga@dlr.de, gertjan.looye@dlr.de

2) Evaluation of the effective flight envelope: The flight envelope is usually restricted after the occurrence of failures. To ensure a satisfactory level of pilot situation awareness, the knowledge of the effective flight envelope is very desirable to start reconfiguration actions or to adjust protection laws. The developed desktop environment allows to perform systematic studies of aircraft trimmability in faulty situations.

3) Flexible and easy setup of FDD based FTC architectures: This is one of the main goals of the developed software environment. For this purpose, a comprehensive Simulink library has been developed which provides generic blocks for the main FDD and FTC functional components: residual generation, residual evaluation, decision making, fault identification, controller reconfiguration or switching.

4) Real time simulation and visualization: The aircraft model without failures or with some activated failures can be simulated in real-time, and controlled by a ”pilot” via a joystick or by an autopilot (to perform some desired manoeuvres). A 3D-visualization system of the flying aircraft allows to directly follow and quantify the aircraft behavior in faulty situations. A realistic 2D visualization of cockpit instrumentation is complemented by informative graphical information indicating the exact location of faults.

The paper is organized as follows. In section II the closed-loop aircraft model is presented and descriptions of available actuators, sensors and their corresponding faults are given. Trimming and linearization tools have been developed to allow generation of linearized models for the needs of linear fault detection and isolation techniques. Another application of the developed trimming tools is described in Section III in determining the effective flight envelopes for various failures. Section IV covers the visualization related aspects. The fault monitoring aspects based on linear system techniques for residual generation together with related topics like residual evaluation and decision making are briefly described in section V. They form the basis for implementing a generic Simulink blockset to facilitate building FDD based FTC architectures. The new modelling tool are presented in section VI. Finally, three main envisaged extensions of the developed environment are mentioned in a short outlook.

II. THE BENCHMARK MODEL

A. Fault Tolerant Control Architecture

The new environment is intended to support analysis and simulation studies of a typical fault tolerant control architecture based on fault diagnosis techniques as that presented in Fig. 1. Here the block for fault diagnosis provides information on the presence of faults, their locations

(isolation) as well as estimations of fault size and fault mode to allow appropriate reconfiguration actions. Of particular importance for FTC are multi-model approaches to fault detection, where several models corresponding to various fault modes are employed (e.g., via multiple Kalman filters or bank of fault detectors) to detect the fault mode. Controller reconfiguration actions may involve simple parameter change (e.g., gain scheduling in the case of employing passive robustness enhancing methods), control reallocation (e.g., in the case of redundant actuation for partial or total actuator failures) or switching among several predesigned controllers designed for particular fault modes (e.g., in the case of employing multi-model adaptive control techniques).

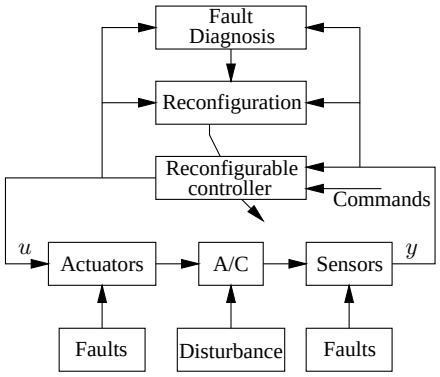

Fig. 1. Fault diagnosis based fault tolerant control

B. Closed-loop aircraft model

The structure of the generic closed-loop aircraft simulation model is shown in Fig. 1. The aircraft input vector u has dimension 22 including the deflections (in [deg]) 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 trimmable horizontal stabilizer, one rudder and two engine throttles (left/right). The aircraft model consists of the actuators block (including actuator dynamics, actuator saturations and the actuator fault modelling), the A/C block (including flight mechanics, aerodynamics, propulsion, environment) and a sensors block (including the sensor fault modelling). The generic model may be representative for a commercial aircraft with the data in Table I. The aircraft model provides a very modular structure, such that faults and also split control surfaces could be easily included into the model. Therefore this model was preferred compared to other models like the public available B747 [6]. The nominal controller actually has only six outputs consisting of one roll, one pitch, one yaw command, two engine throttle commands and a speed brake command. Therefore the overall reconfigurable controller also includes a control distribution block, that splits these commands to the 22 actuators. The standard splitting in the fault free case uses the ailerons for roll control, the elevators and the trimmable horizontal stabilizer for pitch control, the rudder for yaw control and the spoilers are used as speedbrakes.

TABLE I
AIRCRAFT DATA

Wingspan	60m
Overall length	65m
Height	20m
Airspeed range	150-550 kts
Maximum operating Mach number	0.86
Operating weight empty	120000kg
Maximum takeoff weight	220000kg
Engines	2

In the case of actuator faults one may reconfigure/replace the distribution block (or also the nominal controller) by exploiting the existing redundancy in the available control surfaces (e.g., one may use asymmetric spoiler deflections for roll control in case of aileron faults).

In the actuators block, first order linear systems of the form

$$g(s) = \frac{K}{s+K}$$

are used to describe the actuator dynamics. Limitations describing the minimum and maximum deflections of the actuators/surfaces are also included. Furthermore typical actuator faults like ”stuck actuator”, ”actuator runaway” and ”loss of efficiency” are modelled for each actuator. In the sensors block typical faults like ”bias”, ”drift” or ”stuck sensor” are modelled. Faults can be triggered by time events.

The primary purpose of the benchmark model is to support simulation based studies for FTC applications. Moreover, this model also serves for trimming and linearization purposes, to support linear system technique based methods for designing residual generators (see section V) or trimmability studies for determining effective flight envelopes.

C. Trimming and linearization

A highly versatile trimming function `trimex.m` has been developed for MATLAB by the second author to find equilibrium points for a nonlinear system given a set of trim conditions on the state, input, output or state derivative vectors. This function has a similar functionality as the standard MATLAB trimming function `trim.m` available in Simulink. However, there are two main differences between `trim` and `trimex`. While `trim` relies on an optimization based trimming, `trimex` relies on efficient nonlinear system solvers available via the *mex*-function interfaces to nonlinear system solvers and least-squares routines from the subroutine libraries MINPACK [7] and PORT [3]. A very useful feature implemented in `trimex` is the optional trimming with simple bounds on the trim variables. The superiority of the new trimming tool `trimex` over `trim` in what concerns speed (factor of 10 faster) and reliability (accuracy and feasibility) of the results has been demonstrated in many trimmability studies.

The second main difference concerns handling of underdetermined systems, a typical case which arises in flight control applications with redundant control surfaces. Such systems are handled directly by `trim` via its optimization based setting. This approach does not generally guarantee

physically meaningful trim results (e.g., symmetric controls when trimming a symmetric aircraft). In the case of `trimex`, a flexible mechanism has been devised to allow to eliminate the indeterminacy, by allowing to work, instead the full control vector u, with a smaller size control input $\widetilde{u}$ such that $u = B_d\widetilde{u}$, where B_d is a so-called control distribution matrix. This matrix can be used to allocate a few control actions to many control surfaces, but also can be used to deactivate a set of control surfaces during trim or for scaling purposes.

The new trimming tools underly dedicated linearization tools of the aircraft model. The primary use of linear models is in solving fault detection problems using linear system techniques. Moreover, these models can also serve to develop linear parametric varying (LPV) models which can be used for robust controller design or in robustness analysis. The linearization tool builds the linearized aircraft model (see Fig. 1) by coupling the linearized aircraft model with the linear actuator and sensor models. For linearization of the nonlinear model the standard MATLAB function `linmod` is used.

III. EVALUATION OF THE EFFECTIVE FLIGHT ENVELOPE

The flight envelope is a closed area in the speed-altitude diagram that specifies the capabilities of an aircraft design in terms of speed and altitude in a straight and level flight. The inspection of the flight envelope allows pilots to easily read out information like the take-off speed, stalling speed, achievable ceiling, maximum level speed, maximum speed at given altitude, or maximum ground speed. In general, the operation of an aircraft outside of its flight envelope is considered as dangerous and must be avoided.

A basic procedure for building a flight envelope is to perform first a systematic trimmability study over all attainable equilibrium points. However, generally the domain of trimmable equilibrium points is larger than the actual (or *effective*) flight envelope due to various limitations imposed on flight operation in terms of passenger comfort, safety considerations, aircraft maneuverability, or available thrust. In the case of faults, additional restrictions arise because of reduced manoeuvrability of the aircraft or reduced thrust capabilities. This leads automatically to further reductions of the effective flight envelope.

In Fig. 2 we present the aircraft trimmability results in the case of a clean configuration for normal operation and for a faulty case with one engine-out. The blue/green domain delimits the attainable equilibrium points in normal operation, while the red/black domain corresponds to the stall region for which the corresponding equilibrium values of the *angle of attack* (AoA) exceed 9°. The blue region represents the attainable equilibrium points for the faulty aircraft, where the AoA does not exceed 9°. The red region corresponds to AoA values larger than 9°. As can be seen, the attainable equilibrium set for the faulty aircraft is significantly smaller than for the normal operation. Similar restrictions occur in the case of actuator/surface faults. The benchmark setup

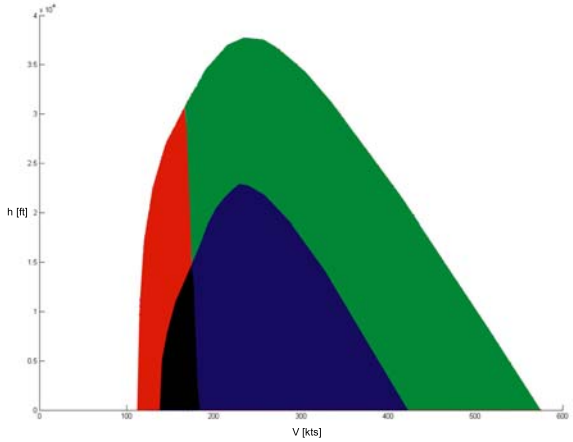

Fig. 2. Attainable equilibrium sets for normal and faulty (engine out) cases

allows to perform extensive trimmability studies for all relevant fault situations using the accompanying fast and accurate trimming tools. A challenging task for such analyses is to formulate meaningful trim conditions for various fault cases. For example, in the case of actuator faults, the underlying trim computations tacitly involve appropriate control reconfiguration actions (e.g., compensating a stuck surface). In this way, a representative set of precomputed effective flight envelopes can be generated to increase pilots situation awareness by providing crucial information on the aircraft maneuverability and achievable performance in case of faults.

In addition, the determination of the effective flight envelope yields important information for the reconfiguration of the flight control system. Especially, the protection laws should be adapted to cope with the new restrictions caused by the reduced maneuverability of the aircraft. In the case of severe failures, the most important issue is to guarantee a safe landing of the aircraft. Therefore, it is important to determine additional information on maneuverability and achievable performance of the faulty aircraft, by considering besides the straight and level flight, also co-ordinated turns or prespecified climb/sink rates in order to plan and perform a complete landing maneuver.

IV. VISUALIZATION

High-quality desktop visualization becomes more an more important in the flight control law design process. This helps the engineer to better understand the dynamics of the aircraft and allows to interactively fly the aircraft to qualitatively assess control law performance and to find weaknesses before an implementation in a full flight simulator. End 2004 the engineering company AeroLabs AG developed, in co-operation with the Technische Universität München (Chair of Flight Mechanics and Flight Control) and based on specifications form the DLR Institute of Robotics and Mechatronics, a new visualization tool for use with real-time desktop simulation of aircraft models. As exclusive feature for DLR, the capability of visualizing the aircraft and its

environment using 3-D stereo projection was implemented.

To allow an interactive flight of the aircraft a standard cockpit instrumentation including a Primary Flight Display (PFD), a Horizontal Situation Indicator (HSI) and a Engine Indication and Crew Alerting System (EICAS) has been developed and is available during flight (see Fig. 3). The platform independent Simulation Control and Management System (SCAMSY) [5] was used to implement the instrumentation. SCAMSY uses a C-like syntax to use Open-GL for 3D-visualization. A UDP interface allows to exchange data via network and therefore it is possible to run the simulation and the visualization on different computers. Furthermore, the cockpit instrumentation can be interactively used to control the simulator (e.g., to activate faults, to change the aircraft configuration or to change environment variables). To inform the pilot about the current status of

Fig. 3. Arrangement of Cockpit instruments

the aircraft, a 2D plot of the aircraft showing all monitored actuators can be included in the cockpit instruments (see Fig. 4). Control surfaces and engines that work properly have green color and all actuators and engines with successfully detected faults are marked red. This gives the pilot a simple but important overview and is much more intuitive than traditional text based warnings. Furthermore, the same coloring schema can be employed for the 3D visualization shown in Fig. 5. Here, the control surfaces and engines of the 3D flying aircraft can be colored in green and red depending on their actual status.

Fig. 4. Fault monitoring 2D

V. FAULT MONITORING

Fault detection (or monitoring) concerns with detection of any fault which may occur in the monitored system. *Fault*

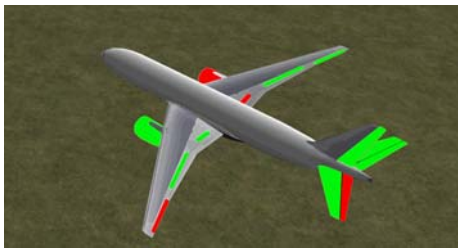

Fig. 5. Fault monitoring 3D

isolation localizes the faults or group of faults which can occur in the system. Here, the occurrence of faults is an aspect which is important in formulating the fault isolation problems. The two extreme situations are the *simultaneous occurrence*, when all faults acting on a system can occur simultaneously at a given time and *one-at-a-time occurrence*, when each fault occurs only by itself (i.e., never occurs simultaneously with other faults). This latter case is often assumed designing monitoring systems providing weak fault isolation. Fault detection and isolation is also known in fault detection literature as *fault diagnosis*. *Fault estimation* performs the quantitative approximation of additive fault inputs and often can be interpreted as a stronger form of fault isolation. *Fault identification* provides quantitative and qualitative information on the detected fault modes and can be seen as an important aspect for the applicability of FTC techniques. Fault identification can be performed independently by using specific approaches (e.g., on-line parameter estimation or active fault detection at component level), or can be assimilated with fault estimation. Multi-model based fault detection approaches implicitly provide fault identification by identifying the model which best describe the current behavior of the plant.

For a system with inputs u and outputs y, a typical fault diagnosis structure is presented in Fig. 6, where the main components are: the residual generator block which generate the residual signal r, the residual evaluation block to compute an evaluation signal θ (e.g., approximate norm) and the decision block which based on thresholds, provides indication on the presence or absence of faults. These components are

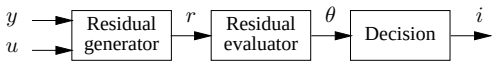

Fig. 6. Fault Diagnosis

described in the following subsections.

A. Residual generation

The function of the residual generator is to deliver on basis of control inputs u and measured outputs y a residual signal r which indicates the presence or absence of faults. For a linear system with additional disturbance inputs d and fault inputs f, the residual generators are usually linear filters (e.g., Kalman filters, observer-like devices), whose output r,

in the ideal case, is decoupled of any influence of d, but is sensitive to all components of f. While a scalar output residual generator is usually sufficient to perform the fault detection task, a bank of residual generators or a vector of residual signals is necessary to achieve fault isolation.

For the design of linear residual generators for fault detection and isolation (exact or approximate) we rely on recently developed general and numerically reliable design methods [8], [9], [11] and associated software tools [10] developed by the second author. The existing design tools allow to easily perform feasibility and performance evaluation studies, via a flexible user interface. The software explicitly supports multi-model based detection techniques, which are popular for FTC applications [2]. An important feature of the new generation of algorithms is that the resulting residual generators have least dynamical orders, which allow to consider systems of relatively high orders even for the multi-model based approaches, where a large number of residual generators must usually run in parallel.

For fault isolation, structured residuals sets are used for fault isolation. To each fault or combination of faults a unique fault signature can be assigned in a unique way. The design of residual generator can be performed to achieve a desired specification to achieve weak or strong fault isolation. For a comprehensive presentation of this aspect see [4].

B. Residual evaluation

The function of the residual evaluator is to generate an evaluation signal θ as an approximate measure of the residual signal energy (e.g., of the 2-norm). The evaluation signal in fault-free situations must be zero (or sufficiently "small"), while in a faulty situation this signal must exhibit a sufficiently large magnitude in order to allow a reliable decision making. With such a signal, a decision on the existence or absence of a fault is simple, relying on a comparison with a suitable threshold value. The following evaluation signals can be generated

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

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

where the parameters α and β allow a desired weighting between instantaneous and long-term measures, T defines an observation window, and $\gamma < 0$ is an appropriate forgetting factor. For discrete-time processing with time increment Δ, equivalent definitions are

$$\theta(t) = \alpha r^2(t) + \beta \sum_{i=1}^{M} r^2(t - T + (i-1)\Delta) \tag{3}$$

$$\theta(t) = \alpha r^2(t) + \beta \sum_{i=0}^{k} \overline{\gamma}^{k-i} r^2(i\Delta) \tag{4}$$

where $M = T/\Delta$, $k = t/\Delta$ and $0 < \overline{\gamma} < 1$ is a forgetting factor. Note that in (1)-(4) $r(t)$ is assumed to be scalar and one may either substitute $r^2(t)$ by $r^T(t)r(t)$ or apply (1)-(4) to each component of $r(t)$ in the case that $r(t)$ is a vector valued signal.

C. Decision making

The role of decision making is to determine if a fault occurred or not using the generated evaluation signal. For this purpose, $\theta(t)$ is compared with a threshold value J_{th} to decide for the presence or absence of a fault using the following simple decision logic

$$\begin{array}{lcl} \theta(t) \geq J_{th} & \Rightarrow & \text{alarm for fault} \\ \theta(t) < J_{th} & \Rightarrow & \text{no fault} \end{array} \tag{5}$$

In the case of a residual vector corresponding to a structured residual set, the above comparison is performed for each component of the residual vector r to decide which component fired (being above the threshold) or not fired (being below the threshold). The resulting fault signature is then compared with the signature underlying the design of the residual generator to obtain the information which faults occurred and which ones not.

The main aim of a reliable decision making is to prevent false alarms and eliminate missed detections. These conflicting requirements often are complemented with even harder requirements as the prompt detection of faults and the reliable detection of incipient faults. Therefore, choosing the parameters intervening in the evaluation signal generators and the value of the threshold J_{th} is important for the overall performance of the fault diagnosis system.

VI. SIMULINK BLOCKSET

A generic Simulink blockset (see Fig. 7) for fault detection, isolation and controller reconfiguration has been developed and is part of the desktop environment. The blockset contains four main categories of blocks:

1) Residual generator blocks: Generic linear filter blocks for fault detection, and fault detection and isolation are provided to allow an easy integration of the linear residual generators calculated with the tools available in [10] into the Simulink environment

2) Residual evaluation blocks: Blocks are provided for both continuous-time and discrete-time residual signal evaluators. In both cases, one may choose between implementations based on windowing or forgetting factors (1)– (4). For example, in Fig. 8 and Fig. 9 the parameter setting menu and the block diagram of a continuous-time signal evaluator are shown, where the user only has to enter the values of free parameters α, β and T.

3) Decision making blocks: Several thresholds based decision blocks have been implemented for various standard fault detection problems (fault detection, weak FDI, strong FDI). A special block supports a least-distance based decision for the multi-model based fault detection approach.

4) Reconfiguration blocks: Two generic controller reconfiguration blocks have been implemented, which allow a smooth blending between two different controllers. Several new reconfiguration blocks are planned as extensions. Various blocks of the Simulink blockset contain parameters to be set by user (e.g., window width, weighting coefficients, threshold values). The aim is to find values which allow fast detection rates, without false alarms or missing detections.

Therefore, choosing the best values of these parameters is not a trivial task and presently is entirely done using extensive simulations. One of the main directions for the intended further development of the desktop environment is to automate the selection of the free parameters by using optimization driven search.

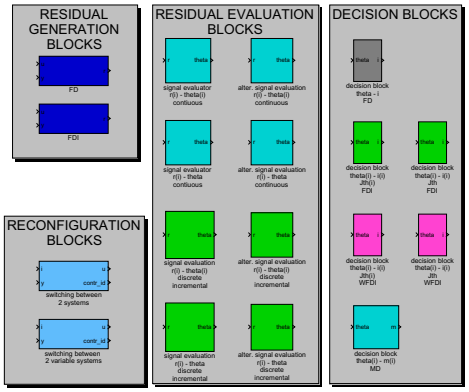

Fig. 7. Simulink blockset

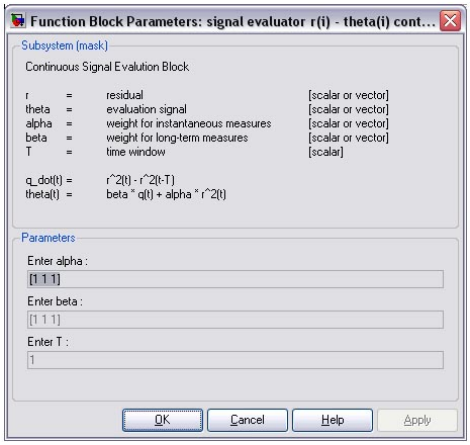

Fig. 8. User menu of a signal evaluation block

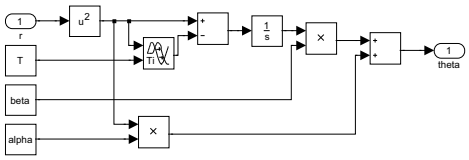

Fig. 9. Block diagram of a signal evaluation block

VII. OUTLOOK

The desktop environment will be extended with additional functionality to perform advanced analysis and tuning task:

1) Assessment and optimization of model-based FDD performance: The assessment of capabilities of model-based fault detection and diagnosis for FTC involves the feasibility of fault isolability and fault identifiability in presence of uncertainties, analysis of the detection robustness, detection speed, and detection sensitivity for given thresholds. Furthermore, the free parameters in the FDD part of the FTC architecture (e.g., weights in residual evaluation blocks or thresholds in decision making blocks) can then be globally tuned using multi-objective optimization and finally the desktop environment may serve to maximize fault detection sensitivity and minimize detection speed, missing detections and false alarm rates.

2) Assessment of FTC based on controller reconfiguration techniques: Using controller reconfiguration strategies like switching and reallocation is part of the existing FTC strategies. Here, an important role is played by the analysis of stability and performance of the reconfigured system and the employed controller switching methods (e.g., bumpless transfer).

3) Clearance of FTC-based flight control systems: The environment can 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.

VIII. ACKNOWLEDGMENT

The reported work has been partially performed in the framework of an DLR-ONERA joint research project IMMUNE (Intelligent Monitoring and Management of Unexpected Events). We acknowledge the contributions of Timo Ruprecht (DLR, Institute of Flight Systems) in implementing the actuator and sensor fault models and of Heinz Schabreiter in preparing a first prototype version of the Simulink blockset.

REFERENCES

[1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, Berlin, 2003.

[2] J.D. Boškovic and R.K. Mehra. A multiple model-based reconfigurable flight control system design. In *Proc. IEEE CDC, Tampa, Florida*. IEEE, December 1998.

[3] P. A. Fox, A. P. Hall, and N. L. Schryer. The PORT mathematical subroutine library. *ACM Trans. Math. Softw.*, 4:104–126, 1978.

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

[5] S. Kuhlmann, I. Sturhan, A. Jaros, and G.Sachs. Flexible and efficient tools for cost-effective simulation environment. In *Proc. AIAA Conference*, 2005.

[6] A. Marcos and G. Balas. Development of Linear-Parameter-Varying Models for Aircraft. *Journal of Guidance, Control and Dynamics*, 27(3), 2004.

[7] J. J. Moré. User's Guide for MINPACK-1. Applied Mathematics Division Report ANL-80-74, Argonne National Laboratory, Argone,IL, 1980.

[8] A. Varga. New computational approach for the design of fault detection and isolation filters. In M. Voicu, editor, *Advances in Automatic Control*, volume 754 of *The Kluwer International Series in Engineering and Computer Science*, pages 367–381. Kluwer Academic Publishers, 2004.

[9] A. Varga. Numerically reliable methods for optimal design of fault detection filters. In *Proc. of CDC'05, Seville, Spain*, 2005.

[10] A. Varga. A fault detection toolbox for Matlab. In *Proc. of CASCD*, Munich, 2006.

[11] A. Varga. On designing least order residual generators for fault detection and isolation. In *Proc. 16th International Conference on Control Systems and Computer Science*, pages 323 – 330, Bucharest, Romania, 2007.