Input-Dependent Threshold Function for an Actuator Fault Detection Filter

Héctor Rotstein [b] Ryan Ingvalson [a] Tamas Keviczky [a] Gary Balas [a]

[a] *Dept. of Aerospace Engineering, University of Minnesota, MN55455, USA*
[b] *Rafael and the Dept. of Electrical Engineering, The Technion, Haifa 3200, Israel*

Abstract

The flight control system is a safety-critical component of an UAV and hence should include a level of fault detection (FD) and controller reconfiguration to be used in case a faulty condition is declared. This paper reports the design of a FD filter for the lateral channel of an UAV. The main theoretical contribution is a new threshold function that enhances the fault detection capabilities in the face of substantial model uncertainty. Simulation result shows that the new threshold function in combination with a fault detection filter can detect a fault in the lateral channel of an UAV flight control.

Key words: Fault detection, H-infinity control, Aerospace computer control, Threshold functions, Uncertainty

1. Introduction

The flight control system is a safety-critical component of an UAV and hence should include a level of fault detection (FD) and controller reconfiguration to be used in case a faulty condition is declared. This paper reports the design of a FD filter for the lateral channel of an UAV, performed in the framework of the *Software Enabled Control* (SEC) program (Samad and Balas, 2003.) SEC was a research initiative undertaken by DARPA and the U.S. Air Force Research Laboratory (AFRL) to exploit recent developments in software and computing technologies for applications to control systems. The program culminated with a flight test during June 2004, where the main technologies developed during the project were demonstrated in a "simulated" UAV.

To understand the challenge involved in designing an FD filter for the SEC flight test, consider the block diagram illustrated in Figure 1. The signals available for control and FD were the control signal u and the measurement signal y. Hence the *system* of interest was an input/output *black-box* for which the accurate nonlinear model *DemoSim* was developed by Boeing and provided to the groups involved in the SEC. As could be expected, the resulting model and system involve complexities that cannot be ignored when

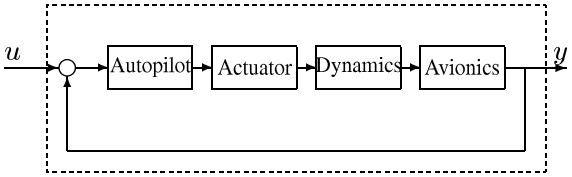

Fig. 1. Setup for Fault Detection in the SEC experiment. Notice that only the signals u and y are available for processing

doing control and FD design: unknown time-delays, non-linear behavior in the form of various limiters at the auto-pilot level, plant variations over the flight-envelope, etc.

The signals internal to the closed-loop system formed by the auto-pilot, actuators, aircraft dynamics and avionics were not available for design or use neither for *DemoSim* nor for the actual flying platform. The reason for this limited information was a combination of proprietary and confidential information and, needless to say, greatly limits the FD options. Indeed, almost every approach to FD assumes that either a possibly inexact model of the internal dynamics of the system of interest exists or that learning from examples is possible. In the present case, a model was indeed available but it was too limited to allow, for instance, the evaluation of the effect of an actuator fault. Moreover, faults due to changes in internal dynamics or variables could not be simulated since the possible dynamics/variables were unaccessible for experiments. The challenge was then to develop an FD system that could be designed and tested both in a hardware-in-the-loop and actual flight in the face of the limited information and actuation available. As reported in this paper, the challenge was addressed by using a number of tools, including H_∞ fault detection, a full nonlinear aircraft simulator for fault models estimation and the development of a new threshold function as detection tool.

An Input-Dependent Threshold Function

Fault detection is understood as the ability to recognize unexpected changes in the functioning of a system, usually resulting from physical failures or breakdowns. An FD scheme usually consists of two stages: an FD filter for generating residuals and a decision stage for analyzing the residuals and deciding if a fault has actually occurred. Relatively little has been done in combining robust FD filters with the synthesis of a *robust* threshold strategy. For example, in (Stoustrup et al., 2003) the optimal threshold function is investigated, where optimality is understood in terms of false-alarm and miss-detection rates. This approach provides a practical solution when the basic trade-off is with measurement noise, but becomes less convenient when measurement noise is small as compared to model uncertainty.

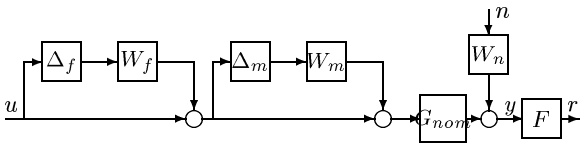

Fig. 2. Fault Detection Setup

In the presence of large model uncertainty the residuals generated by any FD filter are usually not small and hence a threshold function must be designed capable of separating the effect of tolerable model mismatch from a fault. The present paper presents an example of such a design using energy-motivated arguments. The approach here has some points of contact with (Shim and Sznaier, 2003), here a *model invalidation* argument is used to decide *off-line* whether a fault has occurred or not. Exploiting the special structure of the SEC problem, it is shown in this paper that an alternative criterion may be formulated which dramatically reduces the computation cost and hence can be implemented in real-time.

2. The H_∞ Fault Detection Filter

Figure 2 shows the setup for the FD problem under consideration. The block G_{nom} represents the nominal model of the system. As seen in the figure there are two *uncertain* blocks that one could consider. The first block Δ_f is associated with the fault. The second block Δ_m models the uncertainty involved in the modeling. The weighting functions W_f and W_m allow the introduction of a priori knowledge on the nature of the fault and the plant uncertainty. As mentioned in the introduction, the system object of the fault detection involves a closed-loop configuration, and the two uncertainty blocks represent an input-open effort at modeling plant uncertainty and faulty dynamics. Hence also the reason for having two cascaded *multiplicative uncertainty* blocks included in the configuration. This setup clearly highlights that if there is no frequency separation between faults and uncertainty, then the two cannot be distinguished using input-output signals.

Following the idea of H_∞ norm-based fault detection (see, e.g., Chen and Patton, 1999), the configuration in Fig. 2 is re-drawn as shown in Fig. 3. In this figure, the uncertainty blocks Δ_f and Δ_m have been

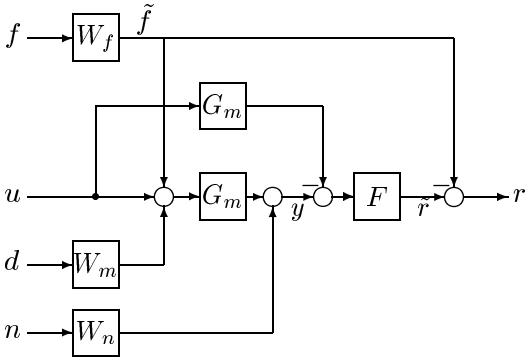

Fig. 3. Setup for the H_∞ Fault Detection Filter design. If the transfer function from f to r is "small," then the residual $\tilde{r}$ tracks the fault signal $\tilde{f}$

removed and two new fictitious signals f and d have been included. The objective of the H_∞design is to make the transfer matrix T_{rw} between the *extended* input $w = [f \quad u \quad d]^T$ and the *residual* r small in an H_∞ sense. This would imply, in particular, that the *residual error* $\tilde{r}$ of the FD filter tracks the fault $\tilde{f} = W_f f$. Notice that the particular uncertainty model selected also covers disturbances at the plant input.

One of the advantages of this formulation is that standard tools for H_∞ control design (e.g., Balas et al., 2001) can be used to design the filter. Notice, however, that other alternative approaches to H_∞ fault detection are possible, as discussed in (Chen and Patton, 1999, Ch. IX). For example, one may one to maximize (in some sense) the minimum transmission channel between faults and residuals.

From Fig. 3, the residual is related to the inputs by

$$\begin{aligned} r &= F\left[G_m\left(W_f f + W_m d\right) + W_n n\right] - W_f f \\ &= \left(FG_m - I\right) W_f f + FG_m W_m d + FW_n n \end{aligned} \tag{1}$$

Equation 1 shows the trade-off involved in this fault detection problem. In order to track the fault, the filter F should invert the plant G_m in the bandwidth of the fault as determined by W_f. At the same time, the filter should be small enough to attenuate the effect of noise $W_n n$ and of plant uncertainty (and input disturbance) $G_m W_m d$. If there is a good frequency separation for these two, then a good solution can be obtained by approximating G_m^{-1} in the frequency band of interest and then "rolling-off" to prevent disturbance and noise affect the residuals.

Notice that if the passing-bands of W_f and $G_m W_m$ or W_n largely overlap, then no FD filter will be able to isolate a fault adequately. This should be a major concern in the design of a FD filter for a closed-loop system. In many cases of practical importance, however, separation will hold approximately and H_∞ filtering may be used to produces the best compromise. This observation motivates the topic of the next section: the design of an adequate decision criterion.

3. Is there a Fault?

In the absence of a clear frequency separation, plant uncertainty will result in a non-negligible residual even if no fault is present. The use of a simple thresholding strategy will hence give rise to a large number of false alarms or, if the threshold value is increased, miss-detections. This section describes an input-dependent threshold function that exploits the additional information assumed for the system. From Fig. 2,

$$\begin{aligned} r = F\left(\left[G_m\left(I + \Delta_m W_m\right)\left(I + \Delta_f W_f\right) - G_m\right] + \right. \\ \left. W u_n n\right) \end{aligned} \tag{2}$$

Following the model invalidation paradigm (Smith and Doyle, 1992, Smith et al., 1997) a fault will not be declared if there exist Δ_m stable and n such that $\|\Delta_m\|_\infty \leq 1$, $\|n\|_2 \leq 1$, and:

$$r = F\left(G_m \Delta_m W_m u + W_n n\right). \tag{3}$$

Namely, there exists an uncertainty and a noise consistent with the problem that can "explain" the observed data.

As shown in (Shim and Sznaier, 2003) in the context of fault detection, given $(u(\tau), r(\tau))$ for $\tau = 0, \cdots, t$, the problem of verifying the existence of a norm bounded uncertainty Δ_m subject to (3) can be transformed into an optimization problem with a linear matrix inequality constraint. This fact has interesting consequences for *off-line* fault detection, but involves the solution of a monotonously increasing optimization problem and hence cannot be used for on-line computations. The objective of this section is to present an alternative, albeit weaker, criterion suitable for real-time applications. Consider the projection operator:

$$(T_{t_1}^{t_2}u)(\tau) = \begin{cases} u(\tau)\ t_1 \leq \tau < t \\ 0 \qquad \text{elsewhere} \end{cases} \tag{4}$$

with the simplifying notation $T^t = T_0^t$. Then (3) can be replace by the stronger condition:

$$\left\|T^t r\right\|_2^2 \leq \left\|T^t F G_m \Delta_m W_m u\right\|_2^2 + \left\|T^t F W_n n\right\|_2^2 \tag{5}$$

for each time instant t. Given a causal operator G, one has $\|T^t G u\|_2 \leq \|G T^t u\|_2$, and so (5) implies

$$\left\|T^t r\right\|_2^2 \leq \left\|F G_m \Delta_m W_m T^t u\right\|_2^2 + \left\|F W_n T^t n\right\|_2^2 \tag{6}$$

Given the assumptions $\|\Delta_m\|_\infty \leq 1$, $\|n\|_2 \leq 1$,

$$\left\|T^t r\right\|_2^2 \leq \|F G_m W_m\|_\infty^2 \left\|T^t u\right\|_2^2 + \|F W_n\|_\infty^2 \tag{7}$$

Since for a given design the transfer matrices above are constant, (7) can be re-written as:

$$\left\|T^t r\right\|_2^2 \leq \alpha^2 \left\|T^t u\right\|_2^2 + \beta^2 \tag{8}$$

where:

$\alpha \doteq \|F G_m W_m\|_\infty$
$\beta \doteq \|F W_n\|_\infty$.

The condition (8) can be used for fault detection in real time applications. Indeed, one can compute the *threshold signal*:

$$f(t) = \left\|T^t r\right\|_2^2 - \alpha^2 \left\|T^t u\right\|_2^2 - \beta^2 \tag{9}$$

for each time t and declare a fault if $f(t) > 0$ at some time instant t.

Notice that if (5) holds for each time t, then (3) will also hold true. The opposite is not necessarily true for time-invariant uncertainties Δ_m, and in general the former condition will be much more restrictive. The condition becomes necessary and sufficient if the uncertainty is allowed to be time-varying. In addition to the above, the use of the triangular inequality and the norm-bounding properties makes (8) a sufficient, but in general far from necessary condition for (3). Hence (9) must be relaxed to make it useful in practice.

3.1. *Relaxing the Condition*

In addition to gap between (3) and (6), there are good reasons to relax the latter by introducing new design parameters. Indeed, during the fault detection filter design stage, the weighting functions W_m, W_n may be modified to achieve desirable behaviors of the filter not necessarily captured by the H_∞ formulation (e.g., a roll-off rate). This is especially true for the noise signal n, which more often than not is of a stochastic nature and hence can only be approximately modeled within an H_∞ design. Moreover, $H_\infty$is a *worst case* criteria and hence the threshold strategy defined above cannot deal directly with false-alarm and miss-detection rates, which are usually given in terms of probability of occurrence and are central in any fault detection design. In addition, the threshold strategy (9) may give rise to fault miss-detection since it trends to become insensitive to faults if W_m is relatively large as compared to the actual plant/model mismatch observed in practice. Finally, at some points during the operation of the system, one may want to allow for relatively large plant/model mismatch, not captured by the uncertainty bound W_m. In these instances, (9) may result in a *false-alarm* since the unduly mismatch effectively behaves as a fault.

In order to relax the condition in equation (8), two design parameters are introduced. First, the *running-norm* is modified by introducing a *forgetting factor* $\kappa < 1$:

$$S^t u(t)^2 \doteq \sum_{\tau=0}^{t-1} \|\kappa^{t-1-\tau} u(\tau)\|^2$$

This exponential decay of the influence of "old" data can be used for both norms in eqn. (9). The usage of the forgetting factor has two main consequences: Second, the noise level β is replaced by a tunning parameter $\overline{\beta}$ that can be used to reduce the false-alarm rate. This parameter can be tuned by analyzing $(u(t), r(t))$ data records under *benign* conditions, e.g., operating points where model/plant mismatch is small.

4. Fault Detection for the SEC Program

The fault detection design for the SEC program was based on the simulation *DemoSim* of the T-33/UCAV aircraft provided by Boeing to the SEC research groups. *DemoSim* is a black-box simulator of the T-33/UCAV. As mentioned in the introduction, although one can provide inputs, modify a few functioning parameters, and observe output logs there is

no access to the internal signals, dynamics, or logic of the simulator. As a consequence and in addition to standard model uncertainty, one needs to address the fact that *DemoSim* 's auto-pilot implementation and internal discrete logic (saturation levels, limiters) are unknown. To make things even harder from a fault detection viewpoint, the inputs to *DemoSim* are actually autopilot commands and hence only guidance-level (i.e. kinematic) control of the vehicle was possible.

To demonstrate the capability of designing a fault detection scheme using an H_∞ filter and the thresholding strategy discussed above, we concentrated on a single-input, single-output sub-system, namely the lateral-directional dynamics from $\dot{\chi}_{cmd}$ to χ_{meas}, subsequently referred to as the χ-channel. The reason for this selection was that it was expected that χ_{meas} would be essentially decoupled from the other two inputs when control commands are restricted to lie within tolerable limits. Thus, the FD problem could be simplified to a single-input, single-output (SISO) problem.

The following linear model was identified from input/output data obtained from the *DemoSim* 's χ-channel:

$$G_m = \frac{2.48 \cdot 10^{-3} z^3}{(z - 0.98)(z^2 - 1.89z + 0.90)}.$$

4.1. *Fault Model*

As mentioned above, *DemoSim* does not provide a way of internally simulating a fault and hence it was necessary to simulate a fault by either corrupting the input or output channel of *DemoSim* in such a way that the resulting output resembled a faulty system. There are many ways in which this can be done, and the for this project a multiplicative input fault was chosen, as shown in Figure 4. In this figure, u is the actual or *true* command to be fed to *DemoSim* , and $\hat{u}$ is the "corrupted" command which will produce the "faulty" output. The no fault scenario corresponds to the case when the "Fault On" switch is open, $\hat{u} = u$. Since only the lateral motion was being considered in this FD problem, it was necessary only to look at faults which would, in reality, have strong coupling to this channel. One such fault would be an aileron actuator fault. Hence, W_f was designed such that the overall

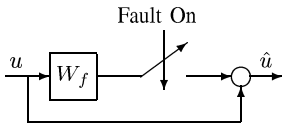

Fig. 4. Multiplicative Fault Input

system (i.e. with the multiplicative fault input) would behave as if a true aileron actuator fault occurred.

Physically, an aileron actuator fault may commonly result in changing the dynamics of the actuator, e.g. a change in the damping or natural frequency of the actuator. These changes could result from actual damage, i.e. faults, to the physical system—such as a loss of hydraulic pressure or damage to the aileron control surface.

W_f was estimated by using another aircraft simulation in which the parameters of the actuators could be changed and then performing frequency scaling to take into account the differences with the T-33/UCAV.

4.1.1. *Generation of Nominal and Faulty Responses*

As mentioned earlier, the fault being simulated is an aileron actuator fault. In the simulation, this actuator is modeled as a second-order system with a nominal natural frequency of $\omega_m = 16.4$ rad/sec and damping ratio $\zeta_m = 0.67$. Using these parameters, the *nominal* response y was generated with the simulation.

The collection of the faulty cases was chosen with natural frequency $\omega_f \in [15, 3]$ rad/sec and damping ratios from ranges $\zeta_f \in [0.7, 1.5]$. This was done to cause the actuator to exhibit a slower response; and, thus, be more characteristic of a truly faulty or degraded actuator. A number of these fault cases were simulated to generate faulty responses $\hat{y}$, which were used to generate a frequency response for each fault case. The upper bound fit $\overline{W}_f$ to these responses was the weight used in design. Future reference to W_f will be understood to refer to this upper bound. The resulting transfer function was:

$$W_f = \frac{2.6 \cdot 10^{-3} s(s + 10)(s + 5)(s + 0.3)}{(s + 0.9)(s + 0.2)(s^2 + 0.416s + 0.64)}.$$

5. Results

This section will contain a summary of the simulation results obtained from integrating the FD filter

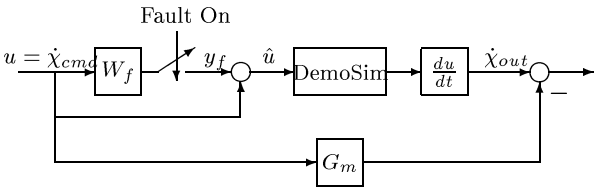

Fig. 5. Simulation Diagram

with *DemoSim* under a specific trajectory and fault scenario. Based upon the interconnection used in the $\mathcal{H}_\infty$ design, the simulation environment illustrated in Figure 5 was used to test the fault detection filter. For lack of space, the procedure for tunning the filter is not discussed here. It is important to stress, though, that the tunning was performed on a different scenario than the one considered in the simulation discussed next.

In this simulation, 0.5 deg/sec $\dot{\chi}$ step commands were issued to *DemoSim* —a positive step was followed by a negative step. The fault was turned on 70 seconds after the positive step command. The results are shown in Figure 6. The upper plot shows the behavior of the residual r and the "fault" signal f. As explained above this latter is zero until the fault is turned on. Notice that, due to model mismatch, the residual is not negligible also when the fault is not present. In the case under study, there is a noticeable difference in the behaviour of the residual with and without fault; this difference, though, is not enough to allow for a thresholding policy since the response depends on the size of the input. The lower plot shows the behavior of the threshold function computed as discussed above. The function is still sensitive to model mismatch in the absence of fault but in a much lesser degree. The function becomes larger than zero, i.e., a fault is declared, at $t = 247sec.$, meaning that it takes the FD system 17 sec. to detect the fault. As an illustration, compare with the "running" energy, namely, the energy over a running interval, of thte residual function as shown in Figure 7. Again, this running energy shows an increase when the fault is present, but this difference is not enough to guarantee appropriate levels of fault detection, since it scales badly with the input signal.

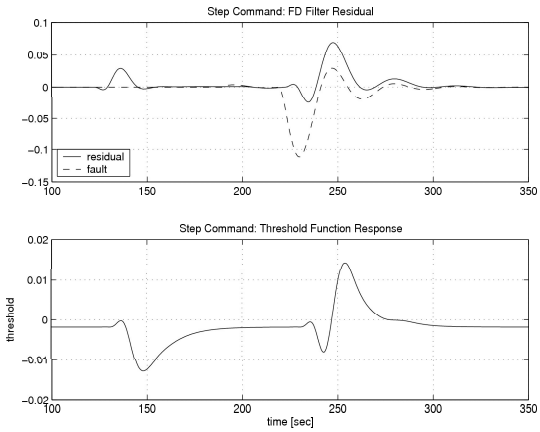

Fig. 6. Simulation Results. The upper figure shows the residual and the fault, while the lower shows the threshold function

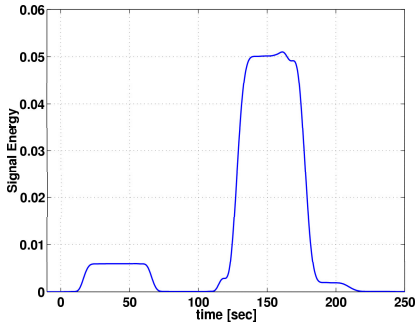

Fig. 7. Energy of the residual function

References

[1] Balas, Gary, John Doyle, Keith Glover, Andrew Packard and Roy Smith (2001). *μ-Analysis and Synthesis Toolbox*. The Mathworks Inc.

[2] Chen, Jie and Ron J. Patton (1999). *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers.

[3] Samad, Tariq and Balas, Gary, Eds.) (2003). *Software-Enabled Control*. Wiley Interscience.

[4] Shim, Duk-Sun and Mario Sznaier (2003). A caratheodory-fejer approach to simultaneous fault detection and isolation. In: *acc*. pp. 2979–2984.

[5] Smith, Roy and John Doyle (1992). Model validation: A connection between robust control and identification. *IEEE Trans. on Automatic Control* **37**, 942–952.

[6] Smith, Roy, G. E. Dullerud, S. Rangan and K. Polla (1997). Model validation for dynamically uncertain systems. *Mathematical Modelling of Systems* **3**(1), 43–58.

[7] Stoustrup, Jakob, H. Niemann and Anders la Cour Harbo (2003). Optimal threshold functions for fault detection and isolation. In: *ACC Proceedings, Denver, Colorado, USA*.