Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Software for Auxiliary Signal Design

S. L. Campbell[1] R. Nikoukhah[2]

***Abstract*— Recently an approach for multi-model identification and failure detection in the presence of model uncertainty and bounded energy noise over finite time intervals has been introduced. This approach involved offline computation of an auxiliary signal and online application of a hyperplane test. This paper discusses progress in developing a software package to carry out this procedure.**

I. Introduction

Failure detection has been the subject of many studies. Most of this work concerned *passive failure detection*. In the passive approach, for material or security reasons, the detector monitors the system but has no way of acting upon it. A major drawback with the passive approach is that failures can be masked by the operation of the system. This is true, in particular, for controlled systems where the desirable robustness of control systems tends to mask abnormal behaviors of the systems. In contrast, *active detection* consists in acting upon the system using a test signal in order to detect abnormal behaviors which would otherwise remain undetected during normal operation. The use of extra input signals specifically in the context of failure detection has been introduced by Zhang [11] and later developed by [5], [6]. We consider robustness in a deterministic setting.

Here the normal and failed behaviors of a process are modeled by two or more linear uncertain systems. In this paper we restrict ourselves to two models. Failure detectability in linear systems thus becomes a linear multi-model identification problem. In most cases, there is no guarantee that one of the models can be ruled out by simply observing the inputs and outputs of the system. For this reason in some cases a test signal, usually referred to as auxiliary signal, is injected into the system to expose its behavior and facilitate the detection (identification) of the failure.

Let v be the inputs taken over by the failure detector mechanism, u the rest of the inputs, if any, and y the outputs of the system. An auxiliary signal v guarantees failure detection if and only if $\mathcal{A}_0(v) \cap \mathcal{A}_1(v) = \emptyset$ where $\mathcal{A}_i(v)$ is the set of input-outputs $\{u, y\}$ consistent with Model i, $i = 0, 1$, for a given input v. We call such a v a *proper auxiliary signal*. Unreasonably "large" signals are often proper, but cannot be applied in practice. There are many requirements on a test signal during the test period including the desire that the system should continue to operate in a reasonable manner, the test period $[0\ T]$ should be short, and the effect of the auxiliary signal on the system minimal. In a series of papers [7], [8], [3] we have developed such an approach. The full mathematical description can be found in the monograph [2] which will appear in 2004. Here we discuss the numerical implementation of this approach. All material in Section 3 and beyond has not been published before.

II. Summary of Approach: Continuous Case

Both continuous and discrete systems are of importance. Space limitations force us to restrict the summary to the continuous case. We first summarize the general procedure. The linear model (1) represents normal and failed systems. It can be considered as a generalization of the model used in Chapter 4 of [10].

$$\dot{x}_i = A_i x_i + B_i v + M_i \nu_i, \tag{1a}$$
$$E_i y = C_i x_i + D_i v + N_i \nu_i. \tag{1b}$$

Here $i = 0, 1$ correspond to the normal and failed system models respectively, y is the measured output, and ν_i and x_i are model noises and states. We assume here for simplicity that the systems have no measured inputs besides the auxiliary signal v. System matrices have arbitrary but consistent dimensions; the only conditions are that N_i's have full row rank and E_i's have full column rank.

The constraint (or noise measure) on the initial condition and uncertainties is

$$\mathcal{S}_i(v, s) = x_i(0)^T P_{i0}^{-1} x_i(0) + \int_0^s \nu_i^T J_i \nu_i \, dt < 1, \quad \forall s \in [0, T], \tag{2}$$

where J_i's are signature matrices. The bound (2) allows for both additive and model uncertainty. With only additive uncertainty we have $J_i = I$ and need only consider $s = T$.

The assumption is that for failure detection, we have access to y, given a v, consistent with one of the models. The problem is to find an optimal v for which observation of y provides enough information to decide from which model y has been generated. That is, there exist no solution to (1a), (1b) and (2) for $i = 0$ and 1 simultaneously. We consider cost functions on v of the form:

$$\delta(v) = \xi(T)^T W \xi(T) + \int_0^T |v|^2 + \xi^T U \xi \, dt, \tag{3a}$$

[1] Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. USA. e-mail: `slc@math.ncsu.edu` Research supported in part by the National Science Foundation under DMS-0101802, DMS-020695, and ECS-0114095.

[2] INRIA, Rocquencourt BP 105, 78153 Le Chesnay Cedex, France.

where W and U are positive semi-definite matrices and

$$\dot{\xi} = F\xi + Gv, \quad \xi(0) = 0. \tag{3b}$$

Here F and G are matrices, with appropriate dimensions, chosen by design considerations. In many applications, one takes $F = A_0$ and $G = B_0$ so that (3b) represents the normal behavior of the system without the uncertainties. Then $\xi(t)$ is an a-priori estimate of $x_0(t)$. Thus penalizing ξ amounts to penalizing the perturbation of the system during the test period assuming no failure has occurred. It is also possible to consider ξ to contain both an estimate of x_0 and x_1 in order to reduce the effect of the auxiliary signal on the behavior of the system whether or not a failure has occurred.

Since the N_i's are full row rank, we have that for any L^2 functions v and y, there exist L^2 functions ν_i satisfying (1a)-(1b). Thus the non-existence of a solution to (1a), (1b) and (2) is equivalent to $\sigma(v,s) \geq 1$ for some s where $\sigma(v,s) = \inf_{\substack{\nu_0,\nu_1,y\\ x_0,x_1}} \max(\mathcal{S}_0(v,s), \mathcal{S}_1(v,s))$ subject to (1a)-(1b), for $i = 0, 1$. Then

$$\sigma(v,s) = \max_{\beta\in[0,1]} \phi_\beta(v,s), \tag{4a}$$

where

$$\phi_\beta(v,s) = \inf_{\substack{\nu_0,\nu_1,y\\ x_0,x_1}} \beta\mathcal{S}_0(v,s) + (1-\beta)\mathcal{S}_1(v,s), \tag{4b}$$

subject to (1a)-(1b), $i = 0, 1$. Let $X^\perp$ and $X_\perp$ denote maximal rank right and left annihilators of matrix X. Let

$$\begin{pmatrix} F_0 & F_1 \end{pmatrix} = \begin{pmatrix} E_0 \\ E_1 \end{pmatrix}^\perp,$$

$$x = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \nu = \begin{pmatrix} \nu_0 \\ \nu_1 \end{pmatrix}, A = \begin{pmatrix} A_0 & 0 \\ 0 & A_1 \end{pmatrix},$$

$$B = \begin{pmatrix} B_0 \\ B_1 \end{pmatrix}, M = \begin{pmatrix} M_0 & 0 \\ 0 & M_1 \end{pmatrix}, \; N = \begin{pmatrix} F_0N_0 & F_1N_1 \end{pmatrix},$$

$$D = F_0D_0 + F_1D_1, \; C = \begin{pmatrix} F_0C_0 & F_1C_1 \end{pmatrix},$$

$$P_\beta^{-1} = \begin{pmatrix} \beta P_{0,0}^{-1} & 0 \\ 0 & (1-\beta)P_{1,0}^{-1} \end{pmatrix}, J_\beta = \begin{pmatrix} \beta J_0 & 0 \\ 0 & (1-\beta)J_1 \end{pmatrix}.$$

We reformulate Problem (4) as

$$\phi_\beta(v,s) = \inf_{\nu,x} x(0)^T P_\beta^{-1} x(0) + \int_0^s \nu^T J_\beta \nu \, dt \tag{5}$$

subject to

$$\begin{aligned} \dot{x} &= Ax + Bv + M\nu, & \text{(6a)} \\ 0 &= Cx + Dv + N\nu. & \text{(6b)} \end{aligned}$$

We assume that for some $\beta \in [0,1]$, that $N_\perp^T J_\beta N_\perp > 0$, and the Riccati equation

$$\begin{aligned} \dot{P} = (A - S_\beta R_\beta^{-1}C)P + P(A - S_\beta R_\beta^{-1}C)^T - \\ PC^T R_\beta^{-1} CP + Q_\beta - S_\beta R_\beta^{-1} S_\beta^T, \end{aligned} \tag{7}$$

where $P(0) = P_\beta$ and $\begin{pmatrix} Q_\beta & S_\beta \\ S_\beta^T & R_\beta \end{pmatrix} = \begin{pmatrix} M \\ N \end{pmatrix} J_\beta^{-1} \begin{pmatrix} M \\ N \end{pmatrix}^T$, has a solution on [0,T]. Then the problem is: $\min_v \delta(v)$ subject to $\max_{\substack{\beta\in\mathcal{B}\\ s\in[0,T]}} \phi_\beta(v,s) \geq 1$. Define

$$\lambda_{\beta,s} = \max_{v\neq 0} \frac{\phi_\beta(v,s)}{\xi(s)^T W \xi(s) + \int_0^s |v|^2 + \xi^T U \xi \, dt},$$

$$\lambda_\beta = \max_{s\leq T} \lambda_{\beta,s}, \tag{8}$$

so that we end up having to solve the following problem

$$\begin{aligned} \max_v \inf_{\nu,x} x(0)^T P_\beta^{-1} x(0) - \lambda \xi(T)^T W \xi(T) + \\ \int_0^s \nu^T J_\beta \nu - \lambda(\xi^T U \xi + |v|^2) \, dt \end{aligned} \tag{9}$$

subject to (3b) and (6). Rewriting these constraints with $\mu^T = \begin{pmatrix} x^T & \xi^T \end{pmatrix}^T$, the cost in (9) becomes

$$\begin{aligned} \mu(0)^T \begin{pmatrix} P_\beta^{-1} & 0 \\ 0 & 0 \end{pmatrix} \mu(0) + \mu(s)^T \begin{pmatrix} 0 & 0 \\ 0 & -\lambda W \end{pmatrix} \mu(s) \\ + \int_0^s \nu^T J_\beta \nu - \lambda |v|^2 - \mu^T \mathcal{Q}_\lambda \mu \, dt, \end{aligned}$$

where $\mathcal{Q}_\lambda = \begin{pmatrix} 0 & 0 \\ 0 & \lambda U \end{pmatrix}$. A necessary condition for this minimization problem can be expressed as the following two-point boundary-value problem (TPBVP)

$$\frac{d}{dt} \begin{pmatrix} \mu \\ \zeta \end{pmatrix} = \begin{pmatrix} \bar{\Omega}_{11} & \bar{\Omega}_{12} \\ \bar{\Omega}_{21} & \bar{\Omega}_{22} \end{pmatrix} \begin{pmatrix} \mu \\ \zeta \end{pmatrix} = H \begin{pmatrix} \mu \\ \zeta \end{pmatrix}, \tag{10a}$$

$$V_0 \begin{pmatrix} \mu(0) \\ \zeta(0) \end{pmatrix} + V_s \begin{pmatrix} \mu(s) \\ \zeta(s) \end{pmatrix} = 0. \tag{10b}$$

Here

$$V_0 = \begin{pmatrix} I & \begin{pmatrix} -P_\beta & 0 \\ 0 & 0 \end{pmatrix} \\ 0 & 0 \end{pmatrix}, V_s = \begin{pmatrix} 0 & 0 \\ \begin{pmatrix} 0 & 0 \\ 0 & -\lambda W \end{pmatrix} & I \end{pmatrix}$$

$$\begin{aligned} \begin{pmatrix} \bar{Q}_{\lambda,\beta} & \bar{S}_{\lambda,\beta} \\ \bar{S}_{\lambda,\beta}^T & \bar{R}_{\lambda,\beta} \end{pmatrix} &= \begin{pmatrix} \bar{B} \\ \bar{D} \end{pmatrix} \Gamma_{\lambda,\beta}^{-1} \begin{pmatrix} \bar{B} \\ \bar{D} \end{pmatrix}^T, \\ \Gamma_{\lambda,\beta} &= \begin{pmatrix} J_\beta & 0 \\ 0 & -\lambda I \end{pmatrix}, \end{aligned}$$

$$\bar{A} = \begin{pmatrix} A & 0 \\ 0 & F \end{pmatrix}, \qquad \bar{B} = \begin{pmatrix} M & B \\ 0 & G \end{pmatrix},$$

$$\bar{C} = \begin{pmatrix} C & 0 \end{pmatrix}, \qquad \bar{D} = \begin{pmatrix} N & D \end{pmatrix}.$$

$$\begin{aligned} \bar{\Omega}_{11} &= -\bar{\Omega}_{22}^T = \bar{A} - \bar{S}_{\lambda,\beta} \bar{R}_{\lambda,\beta}^{-1} \bar{C}, \\ \bar{\Omega}_{12} &= \bar{Q}_{\lambda,\beta} - \bar{S}_{\lambda,\beta} \bar{R}_{\lambda,\beta}^{-1} \bar{S}_{\lambda,\beta}^T, \\ \bar{\Omega}_{21} &= \bar{C}^T R_{\lambda,\beta}^{-1} \bar{C} + \mathcal{Q}_\lambda. \end{aligned}$$

The optimal v and ν satisfy

$$\begin{pmatrix} \nu \\ v \end{pmatrix} = \alpha \Gamma_{\lambda,\beta}^{-1} \begin{pmatrix} \bar{D}^T \bar{R}_{\lambda,\beta}^{-1} \bar{C} & \bar{D}^T \bar{R}_{\lambda,\beta}^{-1} \bar{S}_{\lambda,\beta}^T - \bar{B}^T \end{pmatrix} \begin{pmatrix} \mu \\ \xi \end{pmatrix}, \tag{11}$$

where α is a to be determined scalar. We need to compute first λ_β which is the largest value of λ for which the TPBVP (10) is not well-posed for some $s \in [0, T]$.

Lemma 2.1: The TPBVP

$$\begin{aligned} \dot{x} &= Hx, & (12a)\\ 0 &= V_0 x(0) + V_s x(s), & (12b) \end{aligned}$$

is well-posed if and only if $V_0 + V_s\Phi(s)$ is invertible where

$$\dot{\Phi} = H\Phi, \quad \Phi(0) = I. \tag{13}$$

Computation of Φ based on (13) is in general not practical, except on short intervals, since H is a Hamiltonian matrix and hence is not stable. When H is time-invariant, a simple and numerically efficient test of the well-posedness of (12) can be done by block diagonalizing H,

$$SHS^{-1} = \begin{pmatrix} A_f & 0 \\ 0 & -A_b \end{pmatrix}, \tag{14}$$

where A_f and A_b do not have any eigenvalues with strictly positive real parts. A calculation shows that invertibility of $V_0 + V_s\Phi(s)$ is equivalent to the invertibility of the better conditioned

$$\Psi(s) \triangleq V_0 S^{-1} \begin{pmatrix} I & 0 \\ 0 & e^{A_b s} \end{pmatrix} + V_s S^{-1} \begin{pmatrix} e^{A_f s} & 0 \\ 0 & I \end{pmatrix}.$$

Lemma 2.2: Under our assumptions, $\lambda > \lambda_\beta$ if and only if $\Psi(s)$ is invertible for all $s \in [0, T]$.

A λ-iteration scheme can now be implemented using any standard ordinary differential equation solver with a root finder option. In particular, we have to solve:

$$\begin{aligned} \dot{\Psi}_f &= A_f \Psi_f, \quad \Psi_f(0) = I, & (15a)\\ \dot{\Psi}_b &= A_b \Psi_b, \quad \Psi_b(0) = I, & (15b) \end{aligned}$$

and test to see if the surface

$$0 = \det(\Psi(s)) \tag{16}$$

is crossed. Then λ_β is the infinum over the set of λ's for which the above system can be solved over the interval $[0, T]$ without any surface crossing. Then optimal λ^* and β^* are obtained by $\lambda^* = \max_{\beta \in \mathcal{B}} \lambda_\beta$. Note that for $\lambda = \lambda_\beta$, the surface crossing may happen inside the interval $[0, T]$, say at T^*. This simply means that the optimal proper auxiliary signal can be defined over the interval $[0, T^*]$. Nothing is gained by increasing the test period and s we can let $T = T^*$.

When $\lambda = \lambda^*$ and $\beta = \beta^*$, (10) has a non-zero solution. This solution allows us to compute the optimal proper auxiliary signal from (11). By computing a vector in the nullspace of the matrix

$$\begin{pmatrix} \begin{pmatrix} -\Psi_f(s) & 0 \\ 0 & I \end{pmatrix} & \begin{pmatrix} I & 0 \\ 0 & -\Psi_b(s) \end{pmatrix} \\ V_0 S^{-1} & V_s S^{-1} \end{pmatrix},$$

we can find consistent values of $x_f(0)$ and $x_b(T)$. But x_f and x_b satisfy

$$\begin{aligned} \dot{x}_f &= A_f x_f, & (17a)\\ \dot{x}_b &= -A_b x_b, & (17b) \end{aligned}$$

with A_f and A_b which do not have eigenvalues with positive real parts. Thus x_f and x_b can be computed from (17a) and (17b) respectively by forward and backward integration. Finally, the solution of the boundary value problem is obtained from

$$x = \begin{pmatrix} \mu \\ \xi \end{pmatrix} = S^{-1} \begin{pmatrix} x_f \\ x_b \end{pmatrix}. \tag{18}$$

The optimal auxiliary signal is computed from (11) by choosing α such that $\|v\|^{-1} = \sqrt{\lambda^*}$.

III. Software

We have implemented these procedures for the continuous case in Scilab programs. Scilab is a software environment developed at INRIA [1], [4]. It is used at a number of industries and has a large user base. It has a very similar syntax to MATLAB so that a Scilab program can be easily converted to MATLAB. However, Scilab has the advantage that it is publicly available and can be downloaded to a number of different operating systems. Scilab may be downloaded form the French research center INRIA (Institut National de Recherche en Informatique et en Automatique). The Scilab website is at `http://www.scilab.org/`.

These are not industrial grade programs. Rather they are given to illustrate the methodology and to solve simple examples. They should, however, be useful for constructing the solution to reasonable sized, well-posed problems. We have tested them on a number of examples. Some large or close to singular problems may require adjusting certain parameters. Page limitations prohibit a full listing of programs. The programs described here can be downloaded from the web site `http://www.math.ncsu.edu/~slc/www/BOOKS/AuxSig.html`. Other software will be placed there as it becomes available.

Matrices A, B, C, D, M, N used in problem formulation (6) are obtained from the two candidate models. A Scilab function `datas` generates all the matrices needed for the algorithms. `datas` also returns matrices F, E_0 and E_1 to be used for the computation of the separating hyperplane. To use the functions defined below, the user must have defined the sizes of x and v in the current Scilab environment as `nx` and `nv`, respectively. Two distinct methods have been implemented.

A. Riccati-based solution

One method, not discussed in the earlier sections, is based on the computation of a solution to a parameterized Riccati equation. This is done using the built-in Scilab function `ode` which solves the ODE defined by Scilab function `Ricci`. Function `rio` computes the coefficients of the Riccati equation. These coefficients depend on λ and β so they are evaluated frequently. Function `lbcalc` computes λ_β for a given β.

B. The block diagonalization approach

The block diagonalization approach is simpler than the Riccati based solution and is particularly well suited to the case where the cost is in the form (3).

The function `lbcalc2` is similar to `lbcalc`, but `lbcalc2` uses a different approach to compute λ_β. It also takes F, G, U and W as additional arguments. The logic of the program is very similar to `lbcalc`, but the test used for the bisection method is based on Lemma 2.2 and uses the function `tpbvs` to find the shortest time interval over which the BVP is singular. `tpbvs` uses the built-in Scilab ODE solver `ode` with option `root` (with root finder). The function being integrated is defined in `sys` corresponding to (15) and the zero crossing surface is in `sysr` corresponding to (16). The function `matbdiag` block diagonalizes H as indicated in (14),

C. Suspension example

To illustrate we consider an example of a vehicle suspension. The model is somewhat idealized but serves to illustrate the application of the software presented in this paper. It is modeled as a mass M attached to a rolling wheel through a spring mass system. The control is a torque applied to the wheel. Friction in the wheel and axle are ignored. The model is:

$$M\ddot{y} - Mr\sin(b)\ddot{\theta} + a\dot{y} + ky = 0, \quad (19a)$$
$$-Mr\sin(b)\ddot{y} + ((m+M)r^2 + J)\ddot{\theta} = w, \quad (19b)$$

where a and k are suspension parameters, J, m and r are wheel parameters (rotational inertia, mass and radius), b is the angle the suspension makes with the vertical (assumed constant), w is the torque variation on the wheel, θ measures the rotation of the wheel and y is the variation in the length of the suspension from its rest length. Note that $y = 0$, $\theta = \theta_0 + \alpha t$ is a solution of (19) when $w = 0$. This corresponds to traveling down a level road at a constant speed with no disturbances. We replace θ in (19) by $\theta - \theta_0 - \alpha t$ so that θ is now the deviation from this reference angle and disturbances in the initial conditions are disturbances from $y = 0$, $\theta = \theta_0 + \alpha t$. If wheel friction is included, then w becomes the change in torque from w_o where w_o is the constant torque needed to maintain the speed α. The outputs are (noisy) measurements of the suspension length y and the rotational rate $\dot{\theta}$ so that the output equations, without the measurement noise, are

$$y_1 = y, \quad (20a)$$
$$y_2 = \dot{\theta}. \quad (20b)$$

1) Additive noise formulation: System (19) is already linear and can be put into first order form by by defining the state as $y, \dot{y}, \theta$ and $\dot{\theta}$. Even if both equations in (19) have additive noise, $x_1 = \dot{y}, x_3 = \dot{\theta}$ would not have noise. This situation is common in applications and illustrates why we never assumed the matrices M_i were full row rank. We will assume that the noise in the suspension itself is negligible and that there is a noise variable added to (19b) which one could think of as noise (in force units) arising from irregularities in the road surface. In addition, we assume an additive noise on each of the output channels to represent measurement noise. The failure to be modeled is a complete failure of the sensor measuring y. That is, $y_1 = 0$. Let

$$\mathcal{P} = \begin{pmatrix} M & -Mr\sin(b) \\ -Mr\sin(b) & (m+M)r^2 + J \end{pmatrix}, \quad (21)$$
$$\mathcal{R}^T = \begin{pmatrix} k & 0 & 0 & 0 \end{pmatrix}, \quad \mathcal{Q} = \begin{pmatrix} a & 0 \\ 0 & 0 \end{pmatrix}. \quad (22)$$

The normal system can be modeled as in (1a) where

$$A_0 = \begin{pmatrix} 0 & I \\ -\mathcal{P}^{-1}\mathcal{R} & -\mathcal{P}^{-1}\mathcal{Q} \end{pmatrix}, \qquad D_0 = 0,$$
$$B_0 = \begin{pmatrix} 0 \\ \mathcal{P}^{-1}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix}, \qquad C_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$M_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ .001 & 0 & 0 \end{pmatrix}, \; N_0 = \begin{pmatrix} 0 & .01 & 0 \\ 0 & 0 & .01 \end{pmatrix}. \quad (23)$$

The failed system is identical to the normal system except the first row of C_1 is set to zero because the failed first sensor now only outputs noise:

$$C_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (24)$$

The objective is to find a torque signal $w(t)$ of minimum energy (optimal proper auxiliary signal v of smallest L^2 norm) so that the failure of the sensor can be detected, and to construct the associated hyperplane test over the interval $[0, 36]$. The optimal auxiliary signal is illustrated in Fig. 1. The h vector of the hyperplane test in this case has two components. They are illustrated in Figure 2. Note the difference in scale. The hyperplane test greatly emphasizes the output of the first sensor.

We set the noise bound at 10 and used the minimal proper auxiliary signal computed using a bound of only one in ten simulations using white noise. White noise is far from being worse case noise. The amplitude of the white noise yields results for which it is difficult to detect failure by visual inspection. The hyperplane test, however, provided correct detection in every simulation. A typical simulation result is given in Figures 3 and 4.

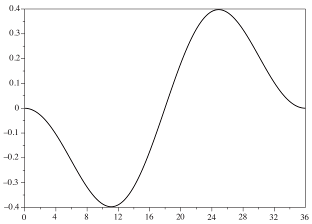

Fig. 1. Optimal auxiliary signal for the suspension problem: additive noise case.

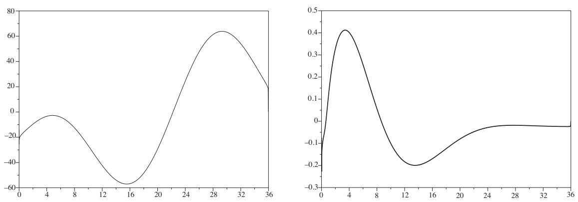

Fig. 2. The first (left) and second (right) component's of the h vector of the hyperplane test.

2) Model uncertainty formulation: Let us now consider again the same suspension problem but with the following modification in (19b):

$$-Mr\sin(b)\ddot{y} + ((m+M)r^2 + J)\ddot{\theta} = (1+\delta)w, \quad (25)$$

where $|\delta| < \bar{\delta}$ is a possibly time varying model uncertainty. This corresponds to uncertainty in the gain of the torque control input w. The resulting models are the same as those of the previous example except for the M_i matrices which become

$$M_i = \begin{pmatrix} 0 & 0 \\ \mathcal{P}^{-1}\begin{pmatrix} 0 \\ 1 \end{pmatrix} & 0 \end{pmatrix}, \quad (26)$$

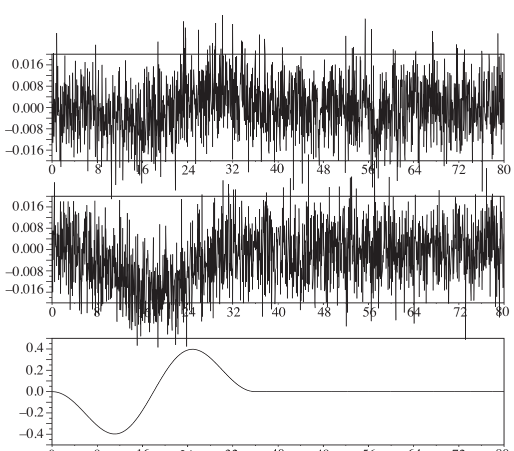

Fig. 3. Normal system. The outputs of the sensors and v.

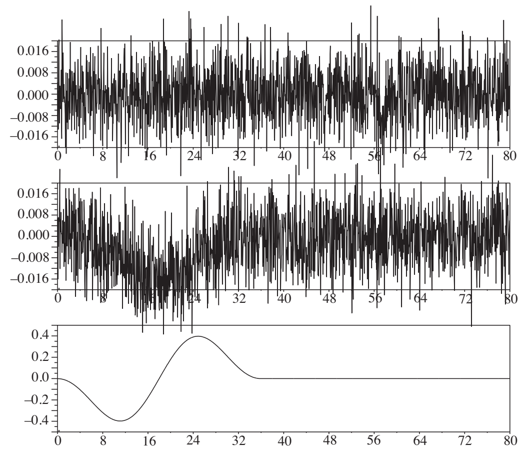

Fig. 4. Failed system. The outputs of the sensors and v.

$i = 0, 1$, and the presence of G and H matrices $G_i = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$, $H_i = \bar{\delta}$. For this particular problem, the shape of the resulting optimal auxiliary signal is not that much different from the one obtained in the previous case where added uncertainty was considered. However, since the model uncertainty is in the B_i matrices, a stronger input signal is needed to overcome the possibly lower norm of the input to output function.

One advantage of the current formulation is that it is easy to consider a number of different problems. For example, it is easy to consider the situation where instead of measuring $\dot{\theta}$, we measure θ. For that, we simply need to modify C_0 and C_1 by exchanging their 2nd and fourth columns.

3) General cost formulation: Suppose that we not only want to have the test signal small but, if a failure has not occurred, we also want the system to have returned close to its original steady operating point at the end of the test period. This can be done by using a nominal model of the normal system for (3b), setting $U = 0$ in (3) and using W in (3) to put a weight on the final value of the nominal system. By nominal system, we mean the model of the unfailed system with zero initial condition and no noise. This gives the effect of the test signal on the unfailed system if there had been no noise and amounts to taking $F = A_0$ and $G = B_0$ in (3).

The choice of the weight to be placed on the final condition, however, should be made with care. Suppose we simply set $W = cI$ where c is a scalar. Figure 5 shows the results using the nominal model and different values of c. The top is $c = 0$, then a small c, and finally at the bottom, large c. It is clear that the final weighting affects primarily one of the components of the state which is θ. Indeed, since the auxiliary signal starts by speeding up the vehicle before slowing it down, if $c = 0$, then the distance traveled by the car during the test period is more than what would have been if no auxiliary signal were used. When $c > 0$, the extra distance traveled is reduced. This is done by larger

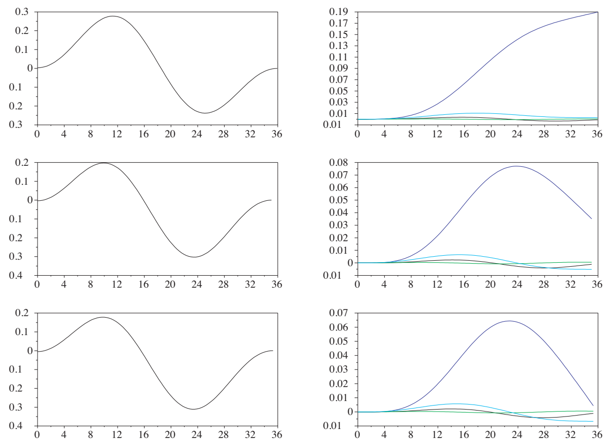

Fig. 5. The auxiliary signal (left) and the state trajectory of the nominal model (right) for three values of c

negative torque in the second half of the test period. The L^2 norm of v increases as c is increased. In addition, in the lowest graph we see that the effort to make $\theta(T)$ small has actually made the final values of the other variables worse then they were with $c = 0$. But the actual distance covered has no importance in this problem. What matters is the vertical position of the suspension, its speed, and the speed of the vehicle. So not only is the $W = cI$ weight useless as far as the final value of θ is concerned, it made things worse.

The correct way to choose W would be to put a weight on y and $\dot{y}$. It would also be reasonable to put a weight on $\dot{\theta}$ so that the test does not affect too much the speed of the vehicle. Thus W should be in the form $W = \mathrm{Diag}(w_1, w_2, 0, w_4)$where w_1, w_2 and w_4 are positive numbers to be chosen properly based on design considerations.

4) Program for suspension problem: A Scilab program for solving the suspension problem is given below. In order to run these programs the functions described above are supposed to be defined in the file `function.sci`. The script assumes that the function `tpbmodel`, used for integrating the complete TPBVP, the function `bakeq`, used for integrating the backward system, and the function `intrp` used for the computation of $P(t)$ by interpolation, are also included in the file `function.sci`. A user only has to specify the problem in the lines up to `end of user input`. The rest of the program uses default values of the parameters including Ng and n. For sensitive problems they can be reset. n sets the mesh used to find β and Ng is the number of points used to find the Riccati equation solution. Note that since we are seeking to estimate where the Riccati equation solutions fails it is important to use a fixed grid or else the grid size would grow too large as we approached the desired point.

IV. Discrete Systems

The theory for the discrete case is carried out similarly to the continuous case. More recently, since [2] was written, we have considered also sampled data systems [9].

With discrete systems there is always the option to just address the problem as a large optimization problem. We refer to this as the static case since the special structure due to the dynamic equations is not exploited. However, these static problems can become very large so it is often better to use discrete versions of Riccati equations and similar techniques. A suite of Scilab programs to carry out the design of discrete auxiliary signals is currently under development for both the static and the discrete time cases. They should be complete by late fall and will be available on the same web site as the continuous codes.

V. Conclusion

We have outlined a procedure for active failure detection over finite time horizons in the presence of bounded uncertainty. We have introduced software to carry out this procedure, and illustrated the software's use on the example of a vehicle suspension. The software can handle both additive and model uncertainty and allows for a variety of ways to measure the size of the auxiliary signal. Several of these possibilities are demonstrated on the suspension example.

References

[1] C. Bunks, J. P. Chancelier, F. Delebecque, C. Gomez (ed.), M. Goursat, R. Nikoukhah, and S. Steer, *Engineering and Scientific Computing with Scilab*, 1999, Birkhauser, Basel.

[2] S. L. Campbell and R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*, Princeton University Press, to appear in 2004.

[3] S. Campbell and R. Nikoukhah, *Auxiliary signal design for robust active failure detection: the general cost case*, Proc. Safeprocess 2003, Washington, DC, 2003, 259-264.

[4] J. P. Chancelier, F. Delebecque, C. Gomez, M. Goursat, R. Nikoukhah, and S. Steer", *Introduction à Scilab*, Springer-Verlag, France, 2002.

[5] F. Kerestecioğlu, *Change Detection and Input Design in Dynamical Systems*, Research Studies Press, Taunton, U.K., 1993.

[6] F. Kerestecioğlu and M. B. Zarrop, *Input design for detection of abrupt changes in dynamical systems*, Int. J. Control, 59 (1994), 1063-1084.

[7] R. Nikoukhah, S. L. Campbell, K. Horton, and F. Delebecque, *Auxiliary signal design for robust multi-model identification*, IEEE Transactions Automatic Control, 47 (2002), 158–163.

[8] R. Nikoukhah and S. L. Campbell, *Active failure detection: Auxiliary signal design and on-line detection*, Proc. IEEE Med. Conf. Control and Automation, 2002.

[9] R. Nikoukhah and S. L. Campbell, *Auxiliary signal design for failure detection in uncertain sampled-data systems*, Proc. 2003 European Control Conference, 2003, to appear.

[10] I.R. Petersen, A.V. Savkin, *Robust Kalman Filtering for Signals and Systems with Large Uncertainties*, Birkhauser, 1999.

[11] X. J. Zhang, *Auxiliary Signal Design in Fault Detection and Diagnosis*, Springer-Verlag, Heidelberg, 1989.