2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Direct Optimization Determination of Auxiliary Test Signals for Linear Problems with Model Uncertainty

Ivan Andjelkovic and Stephen L. Campbell

***Abstract*— Recently there has been increased interest in active approaches for fault detection which use auxiliary test signals. Theory and algorithms have been presented in the literature for the design of fault detection signals for linear systems with model uncertainty. These approaches cannot solve many problems with constraints. This paper gives the first direct optimization formulation of the more general constrained problem. The use of a direct optimization formulation allows the solution of problems not possible by the original algorithms such as problems with input and state constraints. Computational examples are given both to illustrate the theory and to discuss computational issues.**

I. INTRODUCTION

The traditional fault detection approach uses passive fault detection which is well established and widely used. The idea is to observe or estimate functions of parameters or states of the system, and if these deviate enough from the expected values, the fault would be declared. Thresholds are often chosen using a probabilistic approach. There are numerous references on research and implementation of passive approaches [9], [10], [11], [12], [16] many of which are model based. While these approaches are very useful, controller action can sometimes mask a fault until it becomes critical. In addition it may be desirable to detect incipient faults at an earlier stage.

The basic idea of active fault detection is that instead of just observing the system we act on it using an additional, "auxiliary or test" input signal. The test signal is applied over a short horizon. The purpose of this signal is to disturb the system just enough so we can conclude, based on the output, if part of the system is at fault. Such signals are called proper. During the testing, we want to avoid causing too large a disruption in systems operation. Hence "the best" criterion is usually focused on keeping the effect of the test signal small in some sense and returning to normal operation as quickly as possible by the end of the test. The best proper test signal is optimal. The actual criteria depends on the system and design considerations. Active detection allows the detection of faults before they become serious and can often detect faults before a passive approach can. Active fault detection can be used during testing of system design, as well as during the scheduled checkups and maintenance of the system. Our focus is on the applications of active fault detection during regular operation of the system. In certain safety critical processes, only a passive approach may be permitted. We are not talking here about the situation in which a fault is detected by passive means and then the system accommodates the fault [13]. Such systems are also sometimes called active [18]. The use of test signals, of course, is not new but dates to [14], [19]. Recent work on active fault detection includes that of [15], [18]. However, that work is either statistical in nature or limited to linear time invariant systems without constraints.

One active approach has been developed in [6]. There are two computational approaches toward computing the test signal defined by this method. One is based on control ideas such as Riccati equations. We refer to this as the CB or control based approach. The other makes use of advanced optimization software and will be referred to as DO or direct optimization. CB is easier to implement. DO has the potential to handle more difficult problems such as those with control constraints or operational constraints. Previous work has considered the DO solution of linear problems with additive uncertainty [6], [8] and some delay problems [6]. However, that DO formulation could not handle problems with model uncertainty. In this paper we give the first DO solution of problems with both model and additive uncertainty. We also show that this approach can solve problems that the CB approach and earlier DO formulations could not. This paper can be considered a sequel to [8]. Some of the results were briefly announced in the short survey paper [2]. Additional computational examples, discussion, and software implementation details are in [1]. In our approach while we talk of noise or uncertainty there are no statistical assumptions unlike [3]. The only assumptions on the uncertainty are that certain bounds hold. Thus the noise could include unmodeled nonlinearities and other effects.

We first quickly review the model uncertainty approach from [6]. We then formulate the optimization problem that we must solve. Next we discuss some of the needed software capabilities. We give several computational examples that illustrate both theoretical and computational issues. Although we use SOCS for the examples in this paper, our discussion is relevant to similar direct transcription software packages such as SOCX (Sparse Optimal Control Software Extended) from Access Analytics International.

II. PROBLEM FORMULATION

We assume there are two models, a faulty one and a nonfaulty one. There are bounds on the uncertainties. A test signal v is proper if the sets of possible outputs from the two models over all possible admissible noises are disjoint. Otherwise it is not proper. There is a cost associated to the

This work supported in part by ECS-0620986 and DMS-0907832
Cisco Systems, Cary, North Carolina, USA.
Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA. slc@math.ncsu.edu

test signal and we want the smallest proper test signal. For linear systems with additive uncertainty, the set of not proper v form an open convex set and we seek the smallest v on the boundary of this set. The characterization of a proper test signal as having non overlapping output sets is not computationally useful. Instead we characterize proper by the fact that if we assume that we get the same output from both models, then too much noise is required to observe the same output from both models and the noise bound is violated at least once. In particular, the smallest noises that can produce the same outputs from both models violates the noise bound. The model uncertainty case discussed here has a number of extra technical difficulties compared to the additive uncertainty case. For one, the size of the output set can grow with increasing v since more input can create more uncertainty. Another difficulty is that there are conditions that must be tested internal to the optimization interval.

We use an uncertainty formulation derived from [17]. We assume two models ($i = 0, 1$) of the form:

$$\dot{x}_i = (A_i + K_i\Delta_i G_i)x_i + (B_i + K_i\Delta_i H_i)v + M_i\nu_i \quad (1a)$$
$$y_i = (C_i + L_i\Delta_i G_i)x_i + (D_i + L_i\Delta_i H_i)v + N_i\nu_i. \quad (1b)$$

Model 0 is the normally operating system and model 1 is the faulty system. Here $x_i \in R^{n_i}$ are the n_i states, $y_i \in R^m$ are the m observed outputs, $v \in R^l$ are the l control test signals, and $\nu_i \in R^{a_i}$ are the additive noise components. x_0 and x_1 can have different dimensions and so can the ν_i. Matrices M_i and N_i are weights on the additive noise. Matrices L_i, K_i, G_i, H_i and Δ_i represent multiplicative (model) uncertainties. The following reformulation is standard. The model uncertainty is bounded by $\overline{\sigma}(\Delta_i(t)) \leq 1$ where $\overline{\sigma}$ is the largest singular value. Scaling of K_i, G_i, H_i, L_i includes bounds other than one. The test period is $[0,\ T]$. The additive uncertainty is bounded by

$$x_i(0)^T P_i x_i(0) + \int_0^s \nu_i^T \bar{\Gamma} \nu_i \ dt < 1, \quad \forall s \in [0, T], \quad (2)$$

where P_i is positive semidefinite and $\bar{\Gamma} = I$. Again scaling allows the consideration of bounds other than one. The notation $\bar{\Gamma}$ is used so that (2) and (6) can be analyzed at the same time [6]. We then consider the modified system:

$$\dot{x}_i = A_i x_i + B_i v + K_i\mu_i + M_i\nu_i \quad (3a)$$
$$y_i = C_i x_i + D_i v + L_i\mu_i + N_i\nu_i \quad (3b)$$
$$\xi_i = G_i x_i + H_i v \quad (3c)$$

with $\mu_i = \Delta_i\xi_i$. From the model uncertainty bound $\overline{\sigma}(\Delta_i(t)) \leq 1$, we get $|\mu_i| - |\xi_i| \leq 0,\ \forall t \in [0, T]$, which in turn implies

$$\int_0^s |\mu_i|^2 - |\xi_i|^2 \ dt < 0,\ \forall s \in [0, T]. \quad (4)$$

(4) allows for more noise since (4) allows $|\mu_i| \geq |\xi_i|$ on a subinterval. Adding (2) and (4) the overall noise bound is:

$$S_i = x_i(0)^T P_i x_i(0) + \int_0^s |\nu_i|^2 + |\mu_i|^2 - |\xi_i|^2 \ dt < 1, \quad \forall s \in [0, T]. \quad (5)$$

Bound (5) allows for even more noise than (4) and introduces more conservatism in our solution. This conservatism is not in the form of missed faults. The auxiliary signal v obtained this way will be proper. However, it may be suboptimal in terms of the original uncertainty. Bound (5) can be rewritten

$$x_i(0)^T P_i x_i(0) + \int_0^s \varphi_i^T \Gamma \varphi_i \ dt < 1, \forall s \in [0, T] \quad (6)$$

with $\varphi_i = \text{Stack}(\nu_i, \mu_i\, \xi_i)$ and $\Gamma = \text{Diag}(I, I, -I)$. Stack places the vector or matrices into a column vector or matrix.

Unlike in the additive uncertainty case, Γ is not positive definite in the model uncertainty case. As a consequence, for the model uncertainty case we need to make sure that (4) is not violated $\forall s \in [0, T]$ when deciding if v is proper and not just for $s = T$. We assume N_i have full row rank. This is not restrictive since it means we allow noise into all dynamic and output equations. A test signal that is proper under this assumption will be proper if N_i is not full row rank. We now derive formulas for the DO method.

We take the cost of the test signal v to be

$$\delta^2(v) = \psi(T)^T W \psi(T) + \int_0^T |v|^2 + \psi^T U \psi \ dt \quad (7a)$$
$$\dot{\psi} = F_0\psi + F_1 v,\ \psi(0) = 0. \quad (7b)$$

Matrices F_0, F_1 depend on design considerations. For example, ψ could be the amount v perturbs the nonfaulty model, the faulty model, or both. The end point term with weight W penalizes the change in the system at the end of the test period. W, U are positive semidefinite. v is proper if $y = y_1 = y_0$ means too much noise is required. This is,

$$\min_{s,y,x_{i0},\phi_i} \max_{i=0,1} \{S_0(x_0(0), \varphi_0, s), S_1(x_1(0), \varphi_1, s)\} \geq 1. \quad (8)$$

We replace the inner max using a parameter β between 0 and 1. Let $\phi_\beta(v, s)$ be

$$\inf_{x_i,\varphi_i,y,s} \beta S_0(x_0(0), \varphi_0, s) + (1 - \beta) S_1(x_1(0), \varphi_1, s). \quad (9)$$

Then v is proper if there is a $\{\beta, s\}$ so that $\phi_\beta(v, s) \geq 1$.

The differential-algebraic path constraints for our optimization problem are the equations (3) for $i = 0, 1$ and the condition that $y_0 = y_1$,

$$\dot{x} = Ax + Bv + M\nu \quad (10a)$$
$$0 = Cx + Dv + N\nu \quad (10b)$$
$$0 = Gx + Hv + \xi, \quad (10c)$$

where $A = \text{Diag}(A_0, A_1)$, $B = \text{Stack}(B_0, B_1)$, $M = \text{Diag}([K_0, M_0], [K_1, M_1])$, $x = \text{Stack}(x_0, x_1)$, $C = [C_0\ -C_1]$, $N = [L_0, N_0, -L_1, -N_1]$, $G = \text{Diag}(G_0, G_1)$, $\nu = \text{Stack}(\mu_0, \nu_0, \mu_1, \nu_1)$, $H = \text{Stack}(H_0, H_1)$, and $D = \text{Stack}(D_0, D_1)$.

It is beneficial to reduce the size of the problem by eliminating some of the algebraic constraints in (10). The D and H terms have no effect in how the problem is approached although they can impact on the solution. To simplify the discussion we will temporarily assume that $H = 0$ and

$D = 0$. If they are not zero, the matrices that we compute would have additional easily determined terms.

Since N_i have full row rank, we can perform a constant orthogonal change of additive noise components ν_i in ν resulting in $N_i = [\overline{N}_i, 0]$ with $\overline{N}_i$ being invertible. Using the same decomposition, we will have $M_i = [\overline{M}_i, \widetilde{M}_i]$ and $\nu_i = \text{Stack}(\overline{\nu}_i, \widetilde{\nu}_i)$. Then we use (10b) to solve for $\overline{\nu}_0$ and eliminate those variables from the other equations (We could choose to eliminate $\overline{\nu}_1$ instead). Solving for $\overline{\nu}_0$ will involve redefining the noise vector, and having all matrices involved multiplied by $\overline{N}_0^{-1}$. To keep our notation simple we will call $\overline{N}_0^{-1}C$ just C, and the same with the remaining parts of the matrix N. After substituting, our system becomes $\dot{x} = \hat{A}x + Bv + \hat{M}\nu$ with

$$\begin{aligned} \hat{A} &= \begin{bmatrix} A_0 - \overline{M}_0 C_0 & \overline{M}_0 C_1 \\ 0 & A_1 \end{bmatrix}, B = \begin{bmatrix} B_0 \\ B_1 \end{bmatrix}, \\ \hat{M} &= \begin{bmatrix} K_0 - \overline{M}_0 L_0 & \widetilde{M}_0 & \overline{M}_0 L_1 & \overline{M}_0 \overline{N}_1 & 0 \\ 0 & 0 & K_1 & \overline{M}_1 & \widetilde{M}_1 \end{bmatrix}, \end{aligned}$$

and a new noise vector $\nu = \text{Stack}(\mu_0, \widetilde{\nu}_0, \mu_1, \overline{\nu}_1, \widetilde{\nu}_1)$. Equation (10c) is used to solve for ξ and substitute into (5). The cost in the inner minimization problem (9) has the form:

$$x(0)^T P_\beta x(0) + \tfrac{1}{2} \int_0^s x^T Q x + x^T E \nu + \nu^T R \nu \, dt, \quad (11)$$

where $P_\beta = \beta P_0 + (1 - \beta) P_1$. Explicit formulas for the Q, E, R are in [1]. Finally, after introducing a new variable z to account for noise, and using Lagrange multipliers λ with Hamiltonian $\overline{H}(t) = \frac{1}{2}(x^T Q x + x^T E \nu + \nu^T R \nu) + \lambda^T(Ax + Bv + M)$ to replace the inner minimization problem with its necessary conditions, we obtain our DO problem:

$$\min \delta^2(v) = \psi(T)^T W \psi(T) + \int_0^T |v|^2 + \psi^T U \psi \, dt \quad (12a)$$

subject to path constraints

$$\begin{aligned} \dot{x} &= \hat{A}x + Bv + \hat{M}\nu && (12b) \\ \dot{\psi} &= A_0 \psi + B_0 v && (12c) \\ \dot{\lambda} &= -Qx - \tfrac{1}{2} E \nu - A^T \lambda && (12d) \\ \dot{z} &= \tfrac{1}{2}(x^T Q x + x^T E \nu + \nu^T R \nu) && (12e) \\ 0 &= R\nu + \tfrac{1}{2} E^T x + M^T \lambda && (12f) \end{aligned}$$

and boundary conditions and parameter bounds

$$\begin{aligned} \lambda(0) + 2P_\beta x(0) &= 0, \quad \psi(0) = 0 && (13a) \\ z(0) &= x(0)^T P_\beta x(0) && (13b) \\ 0 &< \beta < 1, \quad 0 \le s \le T && (13c) \\ z(s) &\ge 1, \quad \lambda(s) = 0. && (13d) \end{aligned}$$

(13d) says that (8) holds for some β, s. The correct handling of the s parameter was a major factor in the difficulty of setting up the DO solution of the model uncertainty problem. Working with (13d) requires some care since it is talking about a problem on a subinterval with unknown s.

III. Software Requirements

To solve the optimization problems we used the commercially available Sparse Optimal Control Software (SOCS) package [4]. SOCS uses a direct transcription method (collocation method) to convert the continuous system of differential-algebraic equations and bounds into its discretized approximation. That way, SOCS deals with a sparse, finite dimensional nonlinear programming problem and solves it as such. Our inner optimization problem stays the same since that characterizes proper. However, any additional constraints, such as restrictions on v are handled by the software which simplifies problem solution.

Several features of the chosen software were very important. They include; the ability to import an initial guess which is useful for problem nonlinearities or multiple minimums, allowing the time interval to vary which becomes extremely handy for the model uncertainty problem, allowing for multiple phases of varying length with different equations on each phase, allowing outside routines to define desired functions, ability to works with both equality and inequality path constraints, a default grid and initialization option, and control over the permitted numerical error. Our code was verified by comparing the solution of some simpler test problems to solutions obtained by codes from [6] written in Scilab [7], and run by K. Sweetingham.

The integrand in the cost function on v is defined over the testing interval $[0, T]$ while our condition for the noise bound depends on the parameter $s \in [0, T]$. For additive noise we did not have this issue since we were able to set $s = T$. For model uncertainty problems that is not the case since Γ is not positive definite anymore. When $U = W = 0$ we can set the end of the interval to be $T = s$, where s is the value for which one of the models causes the combined noise bound to be exceeded. We can do that since for $t > s$, we already know which model we are dealing with, so $v = 0$ is the optimal solution for $t > s$. Though this modification is simple, due to the nonlinearities in s, we can encounter local minimums depending on the initial guess for s. When either U or W is not 0, the problem is more difficult since the cost function now depends on ψ. Hence, even for $t > s$, v can help reduce the size of ψ. Thus $v = 0$ after $t = s$ is not necessarily an optimal solution anymore. Therefore we have to account in our code for both T and s. We solved that problem by splitting the problem into 2 phases. The final time of phase 1 is s. We use the system of equations derived in this section as constraints for this phase with one modification. We need the value of the cost function on interval $0 \le t \le s$ to be passed to phase 2. This is done by adding another differential equation to accumulate the cost. Phase 2 is defined on the time interval $s \le t \le T$. Since we know that the noise bound is already broken, we only have to account for the influence of v on ψ and how the cost is effected. Thus Phase 2 is much simpler. A phase must have nonzero length, so (13c) is implemented as $L_b \le s \le L_u$ where $L_b > 0$ but is close to zero and $L_u < T$ but is close to T.

IV. Computational Examples

Example 1: (Example 3.3.1[6].) The system of differential-algebraic constraints with model uncertainty is

$$\begin{aligned} \dot{x_0} &= \begin{bmatrix} 0.4\delta_1 & 1 \\ -1+0.4\delta_2 & 0 \end{bmatrix} x_0 + v \\ &\quad + \kappa \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \nu_0 \quad (14a) \\ \dot{x_1} &= \begin{bmatrix} 0.4\delta_3 & 3 \\ -3+0.4\delta_4 & 0 \end{bmatrix} x_1 + v \\ &\quad + \kappa \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \nu_1 \quad (14b) \\ 0 &= [1\ 2](x_0 - x_1) + [0\ 0\ 1](\nu_0 - \nu_1), \quad (14c) \end{aligned}$$

with $\kappa = 10^{-4}$ so that (3) is

$$\begin{aligned} \dot{x_0} &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_0 + v + \begin{bmatrix} 1 & 0 & \kappa & 0 & 0 \\ 0 & 1 & 0 & \kappa & 0 \end{bmatrix} \nu_0 \quad (15a) \\ \dot{x_1} &= \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix} x_1 + v + \begin{bmatrix} 1 & 0 & \kappa & 0 & 0 \\ 0 & 1 & 0 & \kappa & 0 \end{bmatrix} \nu_1 \quad (15b) \\ z_0 &= \begin{bmatrix} 0.4 & 0 \end{bmatrix} x_0 \quad (15c) \\ z_1 &= \begin{bmatrix} 0.4 & 0 \end{bmatrix} x_1 \quad (15d) \\ 0 &= \begin{bmatrix} 1 & 2 \end{bmatrix} (x_0 - x_1) \\ &\quad + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} (\nu_0 - \nu_1). \quad (15e) \end{aligned}$$

We take $P_i = I$ and $T = 10$.

If $U = W = 0$, then DO and CB both found that $s = 10$ gives the optimal proper signal v and computed essentially the same v. The cost is slightly improved using DO. This is probably due to the CB Scilab routine that uses β optimization on a fixed grid, while in SOCS β is a continuous parameter. For the computationally more difficult case where $U = W = I$, $T = 10$, the DO and CB approaches again produced the same v. Here s had only an upper bound of $s \leq 9.99$.

When using DO we sometimes have an issue with choosing initial guesses and local minimums showing up due to the nonlinearity in s. This is shown in the next computation which is similar to the first except that we take $\delta_2 = \delta_4 = 0$. Our DO default is to generate an initial guess by interpolating between any given bounds. Since the dependence on s is nonlinear, we can have local minimums. We will consider only when $U = W = 0$ and $T = 10$ here. Depending on the lower bound constraint L_b on s we obtained different values for s and hence different signals v as shown in Figures 1 and 2. The performance parameters of these 3 calculated signals are in Table I. Suboptimal signal 2 is intermediate in shape between Figure 1 and Figure 2.

TABLE I

Different solutions for v in Example 2.

Parameter	β	Cost	Time(s)	L_b	s
The best v	0.773	0.725	144.8	7	10
Suboptimal 1	.631	0.998	32.5	0.1	4.56
Suboptimal 2	.750	.809	32.8	3	7.65

The initial guess of s was L_b. DO got stuck twice in a local minimum as a consequence of the different initial guesses.

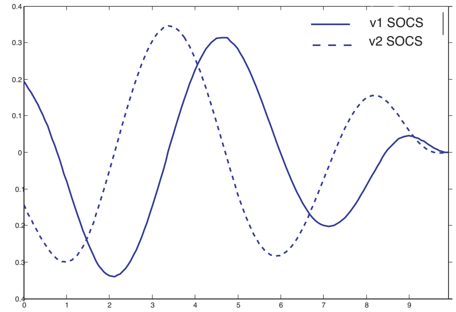

Fig. 1. Best v found for Example 1, case 2 using DO.

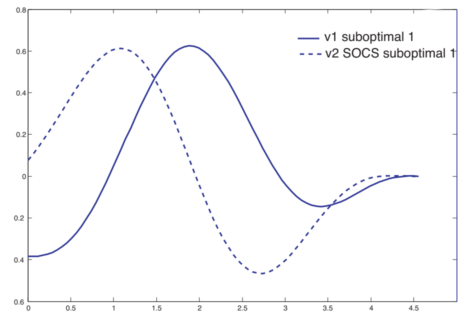

Fig. 2. Suboptimal signal 1 found using DO for Example 1.

While $L_b = 5, 7, 8$ or 9 gave us the best answer, $L_b = 6$ gave us the suboptimal 2 result. All v found were proper.

Example 2: The second example is a modification of Example 4.2.3 in [6]. Here

$$\delta^2(v) = \int_0^1 |v|^2 \, dt \quad (16a)$$

$$\begin{aligned} \dot{x_0} &= (-2 + g_0\delta_1)x_0 + v + \begin{bmatrix} 1 & 0 \end{bmatrix} \nu_0 \quad (16b) \\ \dot{x_1} &= (-1 + g_1\delta_2)x_1 + v + \begin{bmatrix} 1 & 0 \end{bmatrix} \nu_1 \quad (16c) \\ 0 &= x_0 - x_1 + \begin{bmatrix} 0 & 1 \end{bmatrix} \nu_0 - \begin{bmatrix} 0 & 1 \end{bmatrix} \nu_1. \quad (16d) \end{aligned}$$

The g_i parameterize the uncertainty level. Our test interval is $[0, 1]$. We now consider cases where the CB approach does not work, and the DO approach presented here does. We use Problem 2 with $P = 0$. The auxiliary signal v obtained for various values of g is in Figure 3 . $P = 0$ takes more

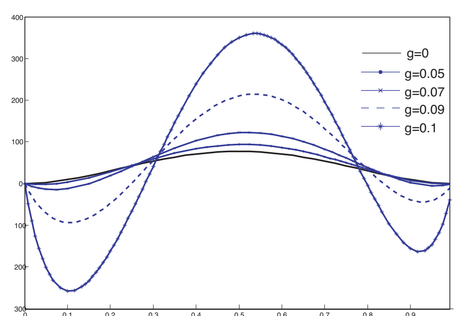

Fig. 3. Auxiliary signal v for various values of g, and $P = 0$ in Example 2.

CPU time than $P = I$. Having $P \neq 0$ and positive definite, no matter how small will guarantee that our output sets are bounded and convex [6]. We can not guarantee boundedness for $P = 0$, which in return increases the complexity of the problem for numerical solvers.

A. Hard Bound on the Auxiliary Signal v

Many problems have practical constraints on the auxiliary signal v. Here we consider hard bounds, $v_{min} \leq v \leq v_{max}$ or $v \leq v_{max}$. These problems cannot be solved by CB or any of the other published methods of finding test signals. DO can find many of these test signals and the DO implementation is straight forward given what has been done earlier in this paper. First, we consider $g = 0$ in Example 2. Figure 4 shows the result for no bound, $|v| \leq 60$, and $v < 50$. For the case studied here if v is proper, then so is $-v$. The unconstrained $v*$ was positive. For the case when $v \leq 50$, DO correctly found $-v*$. Here if we set $|v| \leq 50$, then there are no proper test signals.

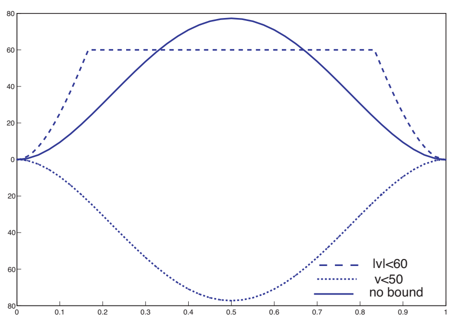

Fig. 4. DO solution with hard bound on auxiliary signal v, with $g = 0$ in Example 2.

Next we allow for a small amount of model uncertainty with $g = 0.05$. We use the $v*$ without v constraints as an initial guess, and we leave the ending time s free. The results are in Figure 5. Notice that both upper and lower bounds play a role in the $|v| \leq 71.2$ case. That is not surprising since the shape of the auxiliary signal when model uncertainty is present is more complex . Also, probably due to the increased difficulty, this time SOCS was not able to find that $-v*$ is the solution of $v \leq v_{max}$ type of bound when $v*$ was an initial guess, since it was not able to find a feasible point. We needed to use a negative initial guess.

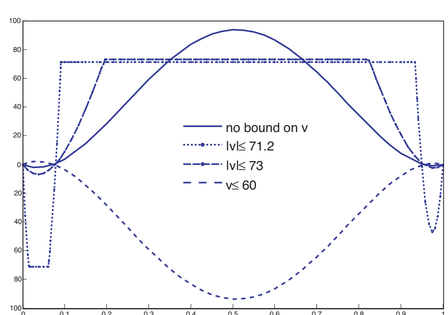

Fig. 5. DO solution with hard bound on v, with $g = 0.05$ in Example 2.

B. Soft Bound on the Auxiliary Signal v

Sometimes it is desirable to discourage v from approaching certain values $r(t)$. That can be achieved by increasing the cost function if v approaches the specific values $r(t)$. To illustrate the use of soft bounds, and point out some computational considerations, we define two new cost functions

$$\text{Soft Bound: } \delta^2(v) = \int_0^1 \frac{c_s}{(r^2 - v^2)^2}\, dt \qquad (17)$$

$$\text{Mixed bound: } \delta^2(v) = \int_0^1 \frac{c_s}{(r^2 - v^2)^2} + c_v v^2\, dt, \qquad (18)$$

where c_s and c_v are positive weight coefficients. We use $r^2 - v^2$ instead of just $r - v$ to assure that v is not close to $\pm r$. The other power of 2 is to assure that the integrand is positive.

If we are using soft bounds and we want v to be continuous, we may need to include hard bounds $v_{min} \leq v \leq v_{max}$. The reason is that DO is using a continuous v approximation on each phase. It can approximate piecewise continuous functions very closely in the L^2 sense by putting grid points close together. For example, without hard bounds we can end up with the solution in Figure 6 obtained for the mixed cost (18) case. Note that v jumps across the value to be avoided. The very short interval of chatter at the second discontinuity often occurs when computing piecewise smooth controls and is a numerical artifact which could be removed by adding additional phases incorporating the jumps.

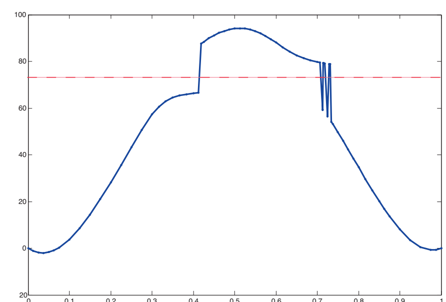

Fig. 6. Noncontinuous v found by DO with only soft bounds in Example 2.

The calculated auxiliary signals for $P = 0$ and $g = 0.05$ are in Figure 7. We used $c_s = 1$, $r = 73$ and $c_v = 10^{-8}$. We also used $v_{max} = -v_{min} = 72.99$ as hard bounds to insure that v was continuous.

C. Bound on the States

Often it is desirable to directly limit the state during the test. This leads to problems with state constraints. There are two ways to approach this. The first is to add the state constraint to the inner optimization problem. In that case the necessary conditions used in the inner optimization problem must be changed. The other, and simpler conceptually, is to just add the state constraint to the outer optimization so that they become an extra feasibility restriction on the proper v. The numerical solution of state constrained optimal control problems is often technically difficult and a number

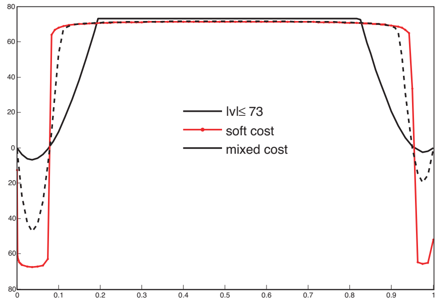

Fig. 7. v found by DO with soft and hard bounds on auxiliary signal in Example 2 with $g = 0.05$.

of subtleties exist when working with inequality constraints [5]. Here we examine putting constraints outside of the inner optimization problem and focus on additive noise only, $g = 0$ and constraints of the type $|x_i| \leq B_x$. By observing the values of x_0 and x_1 from the case without state constraints, we chose values $B_x = 20, 18, 16$ for our bound since they were not overly tight. Found proper signals v are in Figure 8 while states are in Figure 9. For $B_x = 16$, for some

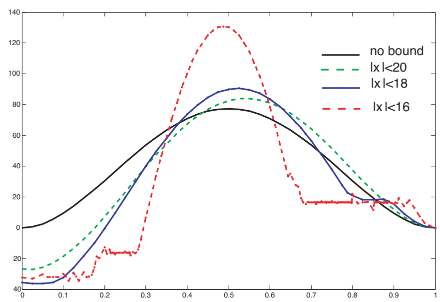

Fig. 8. Comparison of DO v for bounds on states x_i in Example 2.

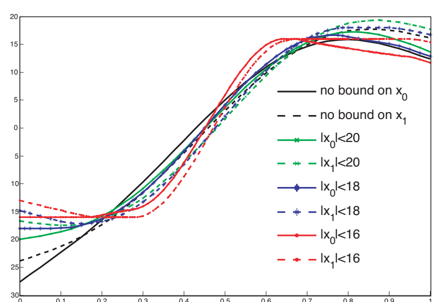

Fig. 9. Comparison of DO x_i for bounds on states x_i in Example 2.

time intervals, the computed auxiliary signal v is very noisy. The same, but not as obvious, is present for $B_x = 18$. This happens while the state constraint is active. In this case we actually have a differential algebraic equation (dae). From (16b) and (16c) we would expect v to be constant when the constraint is active and that is what is observed in the computed v. The result of the constraint activation can be chatter near the constraint and where the constraints change.

V. Conclusion

This paper has given the first DO formulation of the auxiliary test signal design problem for linear systems with both model and additive uncertainty. A DO formulation solves some additional important problems, such as when there are constraints on v and states x_i. These problems cannot be solved by other methods. We illustrated this with several examples. Computational issues were also discussed. Many systems are nonlinear. The CB approach does not readily extend to nonlinear systems. The DO approach given here has greater potential for being extended to nonlinear systems [1]. The CB approach is limited to two models. The DO approach is capable of working with more than two models and a test signal can be constructed to test for several faults by making it proper in several problems simultaneously [6]. The same type of approach can be used here for designing test signals for multiple faults in linear systems with model uncertainty.

References

[1] I. Andjelkovic, *Auxiliary Signal Design for Fault Detection for Nonlinear Systems: Direct Approach*, Ph.D. Thesis, North Carolina State University, 2008.

[2] I. Andjelkovic, K. Sweetingham and S. L. Campbell, "Active fault detection in nonlinear systems using auxiliary signals", *in Proc. American Control Conference (ACC)*, Seattle, WA, 2008.

[3] A. Benveniste, M. Basseville, R. Nikoukhah, and A. S. Willsky, "On the use of descriptor systems for failure detection and isolation", *in Proc. IFAC World Congress*, 7, Sidney, Australia, 1993, pp. 497–500.

[4] J. T. Betts, *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*, SIAM, PA; 2010.

[5] J. T. Betts, S. L. Campbell, and A. Engelsone, "Direct transcription solution of optimal control problems with higher order state constraints: theory vs practice", *Optim. & Eng.*, 8 (2007), pp. 1–19.

[6] S. L. Campbell, and R. Nikoukhah, *Auxiliary signal design for failure detection*, 1st ed. Princeton, NJ; 2004.

[7] S. L. Campbell, J.-P. Chancelier, and R. Nikoukhah, *Modeling and simulation in Scilab/Scicos with ScicosLab 4.4*, Springer, NY; 2005.

[8] S. L. Campbell, K. Horton, R. Nikoukhah, and F. Delebecque, "Optimization formulations of auxiliary signal design for rapid multi-model identification", *Automatica*, 38 (2002), pp. 1313–1325.

[9] J. Chen and R. J. Patton, *Robust Model-Based Fault Detection Diagnosis for Dynamic Systems*, Kluwer, Dordrecht, 1999.

[10] L. H. Chiang, E. L. Russell and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, Springer, NY; 2001.

[11] J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, NY; 1998.

[12] R. Isermann, *Fault-diagnosis systems*, Springer, Heidelberg, 2006.

[13] Mahmoud, M. M., Jiang, J., Zhang, Y., *Active fault tolerant control systems*, Vol. 287 of Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 2003.

[14] F. Kerestecioglu and M. B. Zarrop, "Input design for detection of abrupt changes in dynamical systems", *International Journal of Control*, 59 (1994), pp. 1063–1084.

[15] H. H. Niemann, "A setup for active fault diagnosis", *IEEE Transactions on Automatic Control*, 51 (2006), pp. 1572–1578.

[16] R. J. Patton, P. M. Frank and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*, Springer, Berlin, 2000.

[17] A. V. Savkin and I. R. Petersen, "A new approach to model validation and fault diagnosis", *J. Optimization Theory and Application*, 94 (1997), pp. 241–250.

[18] M. Simandl and H. Niemann, "Active fault detection and control: Unified formulation and optimal design", *Automatica*, 45 (2009), pp. 2052–2059.

[19] X. J. Zhang, *Auxiliary signal design in fault detection and diagnosis*, vol. 134 of Lecture Notes in Control and Information Sciences. Berlin: Springer-Verlag, 1989.