Design of Active Inputs for Set-Based Fault Diagnosis*

Joseph K. Scott1, Rolf Findeisen2, Richard D. Braatz1, and Davide M. Raimondo3

Abstract—Effective fault diagnosis depends on the detectability of the faults in the measurements, which can be improved by a suitable input signal. This article presents a deterministic method for computing the set of inputs that guarantee fault diagnosis, referred to as separating inputs. The process of interest is described, under nominal and various faulty conditions, by linear discrete-time models subject to bounded process and measurement noise. It is shown that the set of separating inputs can be efficiently computed in terms of the complement of one or several zonotopes, depending on the number of fault models. In practice, it is essential to choose elements from this set that are minimally harmful with respect to other control objectives. It is shown that this can be done efficiently through the solution of a mixed-integer quadratic program. The method is demonstrated for a numerical example.

I. INTRODUCTION

High-performance systems such as chemical plants [1], [2] and aerospace systems [3] are becoming increasingly complex. Together with increasingly stringent performance requirements, this has made manual, human-supported detection of component malfunctions challenging, even for highly experienced operators [2]. At the same time, economic considerations have led to the use of inexpensive, less reliable components in many mass market applications. Despite this, control systems are required to maintain stable and reliable operation, which necessitates the early detection of faults that may lead to performance degradation and potentially critical situations.

Effective fault response requires fast and accurate determination of whether a fault has occurred (fault detection) and which component has failed (fault identification). These tasks are rendered difficult by the confounding effects of disturbances, measurement noise, and the compensatory actions of the control system. In response, either an active or a passive approach can be taken. Passive fault detection methods rely solely on the measurements accrued during normal operation, compared with model predictions or historical data, and are now quite mature; e.g., residual and observer-based methods [1], [4], set-based approaches [5], [6], [7], [8], [9], [10], and many more [11], [12], [13]. In contrast, the active approach involves injecting a signal into the system to improve detectability of the fault [14], [15], [16], [17]. Both passive and active approaches can further be categorized into model-based [2] and model-free methods [1].

This article presents a deterministic, model-based approach for active fault diagnosis. It is assumed that the process of interest, under nominal and various faulty conditions, is described by a set of linear discrete-time models subject to bounded disturbances and measurement noise. The first contribution is a method for computing the set of inputs that are guaranteed to generate outputs consistent with at most one model, thereby providing a complete fault diagnosis. Such inputs are referred to as separating inputs.

The set of separating inputs is nonconvex and can be empty. Provided that it is nonempty, the second contribution of the article is an efficient method for selecting a separating input by optimization. Broadly, the purpose of this optimization is to minimize the deleterious effects of the separating input with respect to other control objectives. While the best general formulation of this optimization problem is an interesting open question in fault-tolerant control, here we simply present a solution method suitable for formulations with quadratic costs and linear constraints.

A method for computing active inputs in a deterministic setting was given in [16]. Under the assumption that there is exactly one nominal and one faulty model, it was shown that the set of separating inputs can be computed as the complement of a projection of a high-dimensional polytope. While this characterization is interesting, it is well known that polytope projection is very computationally intensive and numerically unstable for polytopes of dimension greater than about 10. For the present application, this limit is easily exceeded.

The book [17] proposes a different approach based on bilevel optimization. The outer program searches for a minimum two-norm input, while the inner program restricts the feasible set to separating inputs. Overall, this optimization problem is nonconvex and is solved by a specialized algorithm based on two key assumptions: (i) there are exactly two models, one nominal and one faulty, and (ii) the disturbances and measurement noise are energy bounded rather than pointwise bounded. Various extensions of this approach have been investigated, including methods for continuous-time and nonlinear systems [17], [18], asymptotically optimal implementations [19], [20], and methods for systems under linear feedback control [21], [22]. A more general optimization formulation has also been proposed that permits multiple fault models and arbitrary objectives and control constraints [17], [23]. However, the structure of the two-model formulation is lost and the method instead relies on general purpose software to solve difficult optimal control problems.

*BP is acknowledged for financial support of this project.
1J. Scott and R. Braatz are with the Massachusetts Institute of Technology {jkscott,braatz}@mit.edu
2R. Findeisen is with the Institute for Automation Engineering, Laboratory for Systems Theory and Automatic Control, Otto-von-Guericke University Magdeburg rolf.findeisen@ovgu.de
3D. Raimondo is with the Identification and Control of Dynamic Systems Laboratory, University of Pavia davide.raimondo@unipv.it
constrained by two-point boundary-value problems.
In this work, we derive a characterization of the set of separating inputs that, in the case of only two models, is equivalent to that in [16]. However, the derivation here is different in its use of linear mappings and Minkowski sums of sets, rather than projections. This suggests an alternative numerical method based on zonotope computations, which are efficient and numerically robust even in high dimensions. When optimizing over the set of separating inputs, the use of zonotopes here leads to a nonconvex bilevel program. To address this, we present a reformulation in terms of a mixed-integer quadratic program (MIQP) where the number of integer variables has a favorable scaling that can be exploited. Compared to [20] and the related articles listed above, our formulation is different in that it treats the case where the disturbances and noise are pointwise bounded rather than energy bounded. Moreover, the resulting optimization problems are simple to implement and practically solvable, while being flexible with respect to the choice of objective, the presence of other control constraints, and the possibility of multiple fault models.

II. PROBLEM STATEMENT

Consider a system that, on each interval \([k, k+1]\), obeys one of \(m\) discrete-time linear time-invariant (LTI) models. One of these models is nominal, and the rest faulty. If model \(i \in M \equiv \{1, \ldots, m\}\) is obeyed on \([k, k+1]\), then the states and measurements of the system, denoted \(x_k \in \mathbb{R}^n\) and \(y_k \in \mathbb{R}^p\), respectively, evolve according to

\[
\begin{align*}
x_{k+1} &= A[i]x_k + B[i]u_k + r[i] + B_w[i]w_k, \\
y_{k+1} &= C[i]x_{k+1} + s[i] + D_v[i]v_{k+1}.
\end{align*}
\]

The initial state \(x_0\), disturbance \(w_k\), and measurement noise \(v_k\) are all assumed to lie within zonotopes \(Z_0 \subset \mathbb{R}^n\), \(W \subset \mathbb{R}^m\), and \(V \subset \mathbb{R}^p\), respectively.

Let \([0,N]\) be a time interval on which any sequence of faults may occur. More precisely, the model \(i_{k+1}\) that is obeyed on \([k, k+1]\) may vary arbitrarily with \(k\) (see Remark 1). We are interested in synthesizing an input sequence \((u_0, \ldots, u_{N-1})\) such that the sequence \((i_1, \ldots, i_N)\) is uniquely specified by the measurements \((y_1, \ldots, y_N)\). Note that the model active in \([k, k+1]\) determines \(y_{k+1}\) as opposed to \(y_k\). For example, when \(N = 1\), this implies that the mapping \((x_0, u_0, w_0, v_1) \rightarrow y_1\) is specified completely by the model \(i_1\), which we aim to identify.

Ideally, \(N\) should be chosen as small as possible. Here, we assume that \(N\) is specified and focus on the computation of the control sequence. This computation can be iterated with \(N\) increasing from 1 until the problem becomes feasible. The following section states the problem more precisely in the case where \(N = 1\). In §II-B, it is shown that we may analyze this simpler case and apply our results for any \(N > 1\) using an inflated system.

A. The 1-Step Problem

For every \(i \in M\), let \(\Omega[i]\) denote the set of points \((z_x, z_w, z_y) \in X_0 \times \mathbb{R}^n \times W \times \mathbb{R}^m \times V \times \mathbb{R}^p\) satisfying

\[
\begin{bmatrix}
A[i] & B[i] & B_w[i] & -I & 0 & 0 \\
0 & 0 & 0 & C[i] & D_v[i] & -I
\end{bmatrix}
\begin{bmatrix}
z_x \\
z_w \\
z_y \\
z_x^+ \\
z_w^+ \\
z_y^+
\end{bmatrix}
= \begin{bmatrix}
-s[i] \\
-s[i]
\end{bmatrix}.
\]

Definition 1: A pair \((u_0, y_1)\) is consistent with model \(i\) if

\[
\exists (x_0, w_0, x_1, y_1) : (x_0, u_0, w_0, x_1, y_1, y_1) \in \Omega[i].
\]

Otherwise, the pair is inconsistent with model \(i\).

Definition 2: An input \(u_0 \in \mathbb{R}^m\) separates models \(i\) and \(j\) if \((y_1, y_1) \in \Omega[i]\) such that \((u_0, y_1)\) is consistent with both \(i\) and \(j\). It is a separating input if it separates every \(i, j \in M\) with \(i \neq j\). The set of all separating inputs is denoted by \(\mathcal{J}\).

By definition, if a separating input \(u_0\) is applied to the real system, then the measurement \(y_1\) provides a guaranteed fault diagnosis, regardless of its value. It is possible, however, that an input that is not separating will nonetheless lead to a diagnosis for particular values of \(y_1\). The problems we address are now stated as:

1) Efficiently compute a simple representation of \(\mathcal{J}\).
2) Given a positive semidefinite matrix \(R \in \mathbb{R}^{n \times n}\) and a convex polytope \(U \subset \mathbb{R}^m\), solve

\[
\inf \{ u_0^T R u_0 : u_0 \in U, \ u_0 \in \mathcal{J} \}.
\]

B. The N-Step Problem

To denote sequences on \([0,N]\) associated with (1), let \(\sigma_1 \equiv (\sigma_1, \ldots, \sigma_N)\) when \(\sigma\) represents \(x, y, \) or \(v\), and \(\bar{\sigma}_1 \equiv (\bar{\sigma}_0, \ldots, \bar{\sigma}_N)\) when \(\sigma\) represents \(u\) or \(w\). Assuming that a single fault model is active on \([0,N]\) and dropping the superscript for convenience, \(N\) applications of (1) furnish matrices \(\bar{A}, \bar{B}\), etc., such that

\[
\begin{align*}
\bar{x}_1 &= \bar{A}_0 x_0 + \bar{B}_0 u_0 + \bar{r}_0 + \bar{B}_w w_0, \\
\bar{y}_1 &= \bar{C}_1 x_1 + \bar{s}_1 + \bar{D}_v v_1.
\end{align*}
\]

For example,

\[
\bar{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.
\]

The system matrices in (4) can easily be modified to represent a sequence of faults \((i_1, \ldots, i_N)\) on \([0,N]\) by assigning the appropriate \(i\) to each instance of the original system matrices; e.g. \(A^{N-1}B\) becomes \(A^{[i_1]} \cdots A^{[i_N]} B^{[i_1]}\). Thus, noting that (4) has exactly the same form as (1), the \(N > 1\) problem for (1) is analogous to the \(N = 1\) problem for (4), with two important caveats:

1) Each sequence of faults on \([0,N]\) is represented by an additional 1-step model of the form (4). If all
sequences of faults are permissible, then \( \tilde{n}_m = (n_m)^N \)
1-step models are required.

2) If two sequences of faults on \([0, N]\) lead to different 
\( A \) matrices on some \([k, k+1], k > 1\), then the corre-
sponding 1-step models (4) differ in \( (A, B, B_u, f) \).

In the remainder of the article, we will analyze exclusively 
the simpler 1-Step problem, with the understanding that all 
results can be applied directly to the general problem with 
\( N > 1 \) by forming appropriate inflated systems.

**Remark 1:** From Item 1 above, it is clear that allowing 
arbitrary sequences of faults on \([0, N]\) leads to a prohibitively 
large number of scenarios when \( n_m = N \) are large. How-
ever, practical considerations often exclude many scenarios, 
for example by limiting the frequency of faults (i.e. imposing 
a minimum number of repeats \( i_k = i_{k+1} = \ldots = i_{k+d} \) after 
a transition) or by restricting the temporal order in which 
faults may occur (i.e., rendering transitions between certain 
elements of \( \mathcal{M} \) impossible). While the method presented 
here is valid in the general case, we note that some such 
restrictions will be necessary to maintain efficiency.

**Remark 2:** Our formulation is appropriate only for design-
ning open-loop separating inputs. If one permits the use of 
measurements to adjust the input sequence \([16],[22]\), then 
the \( N \)-step problem would no longer be analogous to the 
1-step problem.

### III. Characterizing Separating Inputs

This section presents a characterization of the set of 
separating inputs and shows that this set can be computed 
efficiently using zonotopes. For any \( Q,P \subseteq \mathbb{R}^n \), define

\[
A Q \equiv \{Aq : q \in Q\}, \quad (6)
\]

\[
Q + P \equiv \{q + p : q \in Q, \ p \in P\}, \quad -Q \equiv \{-q : q \in Q\}.
\]

**Definition 3:** For every \( u_0 \in \mathbb{R}^n \), define the sets

\[
F[i]^j \equiv C[i] \left( A[i] X_0 + r[i]^j + B[i] W \right) + s[i] + D[i] V, \quad (7)
\]

\[
Y[i]^j(u_0) \equiv F[i]^j + C[i] B_u u_0. \quad (8)
\]

Clearly, \( Y[i]^j(u_0) \) is the set of all possible outputs of model \( i \) 
given \( u_0 \). Thus, we have the following:

**Lemma 1:** A pair \((u_0,y_1)\) is consistent with model \( i \) iff 
\( y_1 \in Y[i]^j(u_0) \), and \( u_0 \in U \) is a separating input iff

\[
Y[i]^j(u_0) \cap Y[j]^i(u_0) = \emptyset, \quad i,j \in \mathcal{M}, \ i \neq j. \quad (9)
\]

**Theorem 1:** The set of separating inputs is given by

\[
\mathcal{S} = \{u_0 : \Delta L[i]^j u_0 \notin \left(F[i]^j + (-F[j]^i)\right), \ \forall i,j \in \mathcal{M}, \ i \neq j\},
\]


**Proof:** Using (7), (9) can be expressed as

\[
F[i]^j \cap \left(F[i]^j + \Delta L[j]^i u_0\right) = \emptyset. \quad (10)
\]

By Lemma 2.1 in [24], this is equivalent to the requirement 
that \( \Delta L[j]^i u_0 \) is not in \( F[i]^j + (-F[j]^i) \). The theorem now 
follows from Lemma 1.

Theorem 1 provides the characterization of \( \mathcal{S} \) that we 
will use for optimization in §IV. Though it is straightforward,
this characterization has significant advantages over previous 
formulations. Most importantly, it allows \( \mathcal{S} \) to be computed 
efficiently using zonotopes, as described below. Furthermore, 
it suggests some interesting observations on the structure of the 
problem:

1) When attempting to achieve (9), the effect of \( u_0 \) is to 
translate \( Y[i]^j(u_0) \) with respect to \( Y[j]^i(u_0) \) by a vector 
in the subspace \( \mathcal{R}(\Delta L[j]^i) \) (see (10)).

2) Every minimum two-norm input separating \( i \) and \( j \) is 
an element of the orthocomplement of \( \mathcal{M}(\Delta L[j]^i) \). If \( i \) 
and \( j \) differ by few elements of \( B \) and \( C \), this subspace 
could be low dimensional. If \( AL[j] = 0 \), the condition 
for separating \( i \) and \( j \) is independent of \( u_0 \).

### A. Efficient Computation of \( \mathcal{S} \) using Zonotopes

Zonotopes are centrally symmetric convex polytopes that 
can be described as Minkowski sums of line segments [25].
In generator representation, a zonotope \( Z \) is prescribed by 
its center \( c \in \mathbb{R}^n \) and generators \( g_1, \ldots, g_m \in \mathbb{R}^n \) as follows:

\[
Z = \{c + \sum_{j=1}^m \xi_j g_j : \|\xi\|_\infty \leq 1\}. \quad (11)
\]

We use the notation \( Z = \{G,c\} \), where \( G \equiv [g_1 \ldots g_m] \), and 
note that \( Z \) can be equivalently defined as the image of the 
unit hypercube in \( \mathbb{R}^m \) under the affine mapping \( \xi \mapsto G \xi + c \).

Zonotopes obey the following identities, where \( Z = \{G,c\} \) and \( Y = \{G,Y,c\} \):

\[
Z + Y = \{G_1, G_2, c_1 + c_2\}, \quad (12)
\]

\[
Z - Y = \{G_1, -G_2, c_1 - c_2\}. \quad (13)
\]

In particular, the set of zonotopes is closed under the set 
operations defined in (6), and the results can be computed 
very efficiently.

The order of a zonotope is defined as \( n \times n \). A first-
order zonotope with linearly independent generators is a 
parallelotope. The results of the operations (12) can be 
higher order than the arguments. To deal with this, a number 
of techniques are available for enclosing a given zonotope 
within a zonotope of lower order. In the numerical studies 
in §V, Method C in [26] is used.

The key observation of this section is that, by the assump-
tion that \( V, W, \) and \( X_0 \) are zonotopes, the sets

\[
Z[i]^j \equiv \left(F[i]^j + (-F[j]^i)\right), \quad i,j \in \mathcal{M}, \ i \neq j, \quad (13)
\]

are also zonotopes. This follows directly from the definition, 
since each \( Z[i]^j \) is defined by a sequence of linear mappings 
and Minkowski sums applied to the sets \( V, W, \) and \( X_0 \). 
Although the set \( \mathcal{S} \) is neither a zonotope nor the compliment 
of zonotopes itself, Theorem 1 shows that it is fully specified 
by computing and storing the zonotopes \( Z[i]^j \). Thus, 
the identities (12) provide a simple, efficient, and numerically 
robust method for characterizing \( \mathcal{S} \). It is shown in the 
next section that this representation is sufficient for solving 
optimization problems constrained by \( \mathcal{S} \).
Remark 3: The analysis in [16] shows that $\mathcal{S}$ is the complement of a polytope (two-model case) when $X_0$, $W$, and $V$ are zonotopes. However, under the present assumption that $X_0$, $W$, and $V$ are zonotopes, it does not follow from that analysis that $\mathcal{S}$ can be written in terms of the complement of a zonotope, nor that it can be computed efficiently using the identities (12). This is because the computation of $\mathcal{S}$ involves intersections, and the intersection of zonotopes is no longer a zonotope.

IV. OPTIMIZATION OVER $\mathcal{S}$

This section considers the optimization problem (3). According to Theorem 1, this is equivalent to

$$\inf_{u_0 \in U} u_0^T R u_0 \text{ s.t. } \Delta [u_0^T u_0] \notin Z[i], \forall i, j \in \mathcal{M}, i \neq j. \tag{14}$$

Since each $Z[i]$ is closed, the feasible set above is not closed, and there may not exist $u_0^*$ attaining the infimum. For this reason, we will introduce a minimum separation threshold after some further reformulation of (14) (see (16)).

Let $q \in \{1, \ldots, Q\}$ index the combinations $i, j \in \mathcal{M}$ with $i \neq j$, and denote $\Delta [u_0^T u_0] = \Delta[q]$ and $Z[i] = Z[q] = \{g[q], G[q]\}$. Furthermore, let $n_q$ denote the number of columns in $G[q]$. The following assumption ensures that each $Z[q]$ is a full-dimensional polytope in $\mathbb{R}^n_q$, which is required for further reformulations.

Assumption 1: \text{rank}(G[q]) = n_q, \forall q \in \{1, \ldots, Q\}. From the developments in §III, it is simple to show that Assumption 1 holds whenever $X_0$ is a full-dimensional zonotope in $\mathbb{R}^n$ and \text{rank}(G[q] \mathcal{A}[q] = n_q, \forall q \in \mathcal{M}.

Lemma 2: For each $u_0 \in U$ and $q \in \{1, \ldots, Q\}$, let

$$\tilde{\delta}[q](u_0) \equiv \min_{\delta[q]} \tilde{\delta}[q] \text{ s.t. } \Delta[q][u_0] = G[q][\xi[q] + c[q]], \|\xi[q]\|_{\infty} \leq 1 + \delta[q]. \tag{15}$$

Then $\Delta[q]u_0 \notin Z[q]$ iff $\tilde{\delta}[q](u_0) > 0$.

Proof: Choose any $u_0 \in U$ and $q \in \{1, \ldots, Q\}$. Since $G[q]$ is full row rank, (15) is feasible. If $\Delta[q][u_0] \notin Z[q]$, then $H_\xi[q]$ such that $\|\xi[q]\|_{\infty} \equiv 1$ and the equality constraint holds. Thus, (15) does not have a feasible point with $\tilde{\delta}[q] \geq 0$, and it follows that $\tilde{\delta}[q](u_0) > 0$. Conversely, if $\tilde{\delta}[q](u_0) > 0$, then (15) does not have a feasible point with $\delta[q] \leq 0$. This implies that $H_\xi[q]$ such that $\|\xi[q]\|_{\infty} \leq 1$ and the equality constraint in (15) holds, and hence $\Delta[q]u_0 \notin Z[q]$.

We may now rewrite (14), with the addition of a minimum separation threshold $\varepsilon > 0$, as the bilvel program

$$\min_{u_0 \in U} u_0^T R u_0 \text{ s.t. } \varepsilon \leq \tilde{\delta}[q](u_0), \forall q \in \{1, \ldots, Q\}. \tag{16}$$

The inner programs (15) are linear, and hence they can be replaced by their necessary and sufficient conditions of optimality to obtain a single level program. Before doing so, it is shown that we may introduce an upper bound on each $\tilde{\delta}[q](u_0)$ in (16), which is useful for further reformulations.

Lemma 3: There exists $\tilde{\delta}[q] > 1$ large enough that

$$\Delta[q][u_0] \subset \{g[q][\xi[q] + c[q]] : \|\xi[q]\|_{\infty} \leq 1 + \tilde{\delta}[q]\}. \tag{17}$$

Furthermore, (17) implies $\tilde{\delta}[q](u_0) \leq \tilde{\delta}[q], \forall u_0 \in U$.

Proof: Existence of $\tilde{\delta}[q]$ follows from compactness of $\Delta[q][u_0]$ and the fact that $\mathcal{S}(G[q]) = \mathbb{R}^n_q$ (Assumption 1). Choose $u_0 \in U$, let $\delta[q] \equiv \delta[q](u_0)$ and let $\xi[q]$, satisfy

$$\Delta[q][u_0] = G[q][\xi[q] + c[q]], \quad \|\xi[q]\|_{\infty} \leq 1 + \delta[q]. \tag{18}$$

By optimality of $\delta[q]$, $\delta[q] \leq \delta[q]$ for any $(\delta[q], \xi[q])$ satisfying (18). Clearly, $\Delta[q][u_0]$ is an element of the left-hand side of (17), and hence also of the right. Thus, $\exists \xi[q] : (\delta[q], \xi[q])$ satisfies (18). But this implies $\delta[q] \leq \tilde{\delta}[q]$.

We now replace the inner programs (15) with their necessary and sufficient conditions of optimality. For each $q \in \{1, \ldots, Q\}$, these are [27]:

$$\Delta[q][u_0] = G[q][\xi[q] + c[q]], \tag{19}$$

$$\|\xi[q]\|_{\infty} \leq 1 + \delta[q], \tag{20}$$

$$(G[q])^T \lambda[q] = \mu_1[q] - \mu_2[q], \tag{21}$$

$$1 = (\mu_1[q] + \mu_2[q])^T 1, \tag{22}$$

$$\mu_1[q], \mu_2[q] \geq 0, \tag{23}$$

$$0 = \mu_1[q](\xi[k] - 1 - \delta[q]), \quad \forall k = 1, \ldots, n_k, \tag{24}$$

$$0 = \mu_2[q](\xi[k] + 1 + \delta[q]), \quad \forall k = 1, \ldots, n_k. \tag{25}$$

We may now reformulate (16) as

$$\min_{u_0, \delta[q], \xi[q], \lambda[q], \mu_1[q], \mu_2[q]} u_0^T R u_0 \tag{26}$$

s.t. $u_0 \in U,$

$$\varepsilon \leq \delta[q] \leq \tilde{\delta}[q], \quad (19) - (25) \text{ hold } \quad \forall q \in \{1, \ldots, Q\}. \tag{19}$$

Unfortunately, this reformulation is not directly useful because the complementarity constraints (24)–(25) are nonconvex. Moreover, they are known to violate constraint qualifications, causing serious numerical problems. Thus, we propose to reformulate these constraints following the approach in [28]. We introduce the binary variables $p_1[q], p_2[q] \in \{0, 1\}$ and replace (24)–(25) with the implications:

$$p_1[q] = 1 \implies \mu_1[q] \text{ free, } (\xi[k] - 1 - \delta[q]) = 0, \tag{27}$$

$$p_1[q] = 0 \implies \mu_1[q] = 0, (\xi[k] - 1 - \delta[q]) \text{ free,} \tag{28}$$

$$p_2[k] = 1 \implies \mu_2[q] \text{ free, } (\xi[k] + 1 + \delta[q]) = 0, \tag{29}$$

$$p_2[k] = 0 \implies \mu_2[q] = 0, (\xi[k] + 1 + \delta[q]) \text{ free.} \tag{30}$$

These implications can be enforced through linear constraints as follows. First, note that the constraints (20), (22)–(23), and $\delta[q] \leq \tilde{\delta}[q]$ together imply the following bounds for all $k = 1, \ldots, n_k$:

$$\mu_1[q], \mu_2[q] \in [0, 1], \tag{28}$$

$$(\xi[k] - 1 - \delta[q]) \in [-2(1 + \tilde{\delta}[q]), 0], \tag{29}$$

$$(\xi[k] + 1 + \delta[q]) \in [0, 2(1 + \tilde{\delta}[q])]. \tag{30}$$
Using these bounds, (24)–(25) are equivalent to (see [28])
\[
\mu[q]_{1,k} \leq p[q]_{1,k}, \quad \mu[q]_{2,k} \leq p[q]_{2,k},
\]
\[
(\xi^q_{1,k} - 1 - \delta^q) \in [-2(1 + \delta^k)(1 - p[q]_{1,k}), 0),
\]
\[
(\xi^q_{2,k} + 1 + \delta^q) \in [0, 2(1 + \delta^k)(1 - p[q]_{2,k})].
\]
The program (26) is thus equivalent to
\[
\min_{\mathbf{u}_0, \delta^q, (\xi^q_{1,k}, \xi^q_{2,k}), \mu[q]_1, \mu[q]_2, p[q]_1, p[q]_2} \mathbf{u}_0^T \mathbf{R} \mathbf{u}_0
\]
subject to
\[
\begin{align*}
\mathbf{u}_0 &\in \mathcal{U}, \\
\epsilon &\leq \delta^q \leq \delta^q_{0,0}, & (19)–(23), \\
p[q]_{1,k}, p[q]_{2,k} &\in \{0, 1\}, & (31)–(33), \forall k &\in \{1, \ldots, n_k^q\}, \\
\forall q &\in \{1, \ldots, Q\}.
\end{align*}
\]
This is a mixed-integer quadratic program (MIQP) and can be solved efficiently using, for example, CPLEX [29]. The number of binary variables in (34) is twice the total number of generators in the zonotopes \(Z[i]_1, 2Y_{\text{nom}}^q [i]_2\). This is favorable for several reasons. First, the generator representation of a zonotope is often much more compact than its halfspace or vertex representation. Second, it is simple and efficient to compute a reduced order zonotope containing the original [26]. It is shown in the following section that such a reduction can greatly simplify (34) with only a small impact on the optimal solution due to conservatism in \(\mathcal{F}\). If zonotopes of a fixed order \(\ell\) are used, then the number of binary variables in (34) is \(2Qn_{\ell}n_f\), which scales linearly with \(n_f\) for fixed \(Q\) and \(N\).

V. NUMERICAL EXAMPLE

Consider the nominal model defined by
\[
\begin{align*}
\mathbf{A}[1] &= \begin{bmatrix} 0.6 & 0.2 \\ -0.2 & 0.7 \end{bmatrix}, \\
\mathbf{B}[1] &= \begin{bmatrix} -0.3861 & 0.1994 \\ -0.1994 & 0.3661 \end{bmatrix}, \\
\mathbf{C}[1] &= \begin{bmatrix} 0.7 & 0 \\ 0 & 0.3 \end{bmatrix}, \\
\mathbf{B}_e[1] &= \begin{bmatrix} 0.125 \times 0.0598 \\ 0.0598 \times 0.125 \end{bmatrix}, \\
\mathbf{D}[1] &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
\mathbf{r}[1] &= \mathbf{s}[1] = \begin{bmatrix} 0 \\ 1 \end{bmatrix},
\end{align*}
\]
and define the four fault models \(i = 2, \ldots, 5\) by the following modifications to the nominal model:
\[
\begin{align*}
\mathbf{B}[2] &= \begin{bmatrix} -0.3861 & 0 \\ -0.1994 & 0.3661 \end{bmatrix}, \\
\mathbf{B}[3] &= \begin{bmatrix} 0.1994 & 0.3661 \\ 0 & 0.3 \end{bmatrix}, \\
\mathbf{A}[4] &= \begin{bmatrix} 0.6 & 0 \\ -0.2 & 0.7 \end{bmatrix}, \\
\mathbf{A}[5] &= \begin{bmatrix} 0.6 & 0.2 \\ 0 & 0.7 \end{bmatrix}.
\end{align*}
\]
Models 2 and 3 have faulty actuators, and models 4 and 5 have system faults. In generator notation, define \(\mathbf{x}_0 = \{1_{2 \times 2}, [2, 1]^T\}\) and \(\mathbf{w} = \mathbf{V} = \{0.1 \mathbf{1}_{2 \times 2}, 0\}\) (e.g., \(\mathbf{w} = \{\mathbf{w} \in \mathbb{R}^2 : \|\mathbf{w}\|_\infty \leq 1\}\)) where it is specified that \(\mathbf{x}_0 \in \mathcal{X}_0\) and \((\mathbf{w}_k, \mathbf{x}_k) \in \mathcal{W} \times \mathcal{X}\), \(\forall k \in \mathbb{N}\). The objective is to synthesize an input that will distinguish between these five models after \(N = 4\) steps. It is assumed that one model is active on all of \([0, N]\); i.e., any fault can occur at \(k = 0\), but no additional faults occur in \([0, N]\). For the optimization over separating inputs, specify \(\mathbf{R} = I_{2 \times 2}\) and
\[
U = \{[\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}] : \|\mathbf{u}_k\|_\infty \leq 2.5, \ k = 1, \ldots, N\}. \quad (36)
\]
To limit the number of binary variables, the number of generators describing each \(Z[i]_1 (i, j \in \{1, \ldots, 5\}, i \neq j)\) was limited to 4 using the conservative order reduction suggested in [30]. The proposed method found the minimum two-norm separating input in 38.62 s, with norm 5.4293. \(^1\) The circles in Figure 1 show Monte Carlo samples of the outputs of each of the five models at \(k = 1, \ldots, 4\) when the optimal separating input is injected. The zonotopes in the lower right panel show the sets of output variables consistent with each of the five models at \(k = 4\), which are clearly disjoint. This implies that any sequence of measurements taken on \([0, N]\) will be consistent with at most one model, thus providing the desired fault diagnosis.

As mentioned previously, limiting the number of generators describing each \(Z[i]_1\) reduces the required computational effort, but leads to more conservative results. For this example, it is interesting to note that reducing the number of generators to 2 gives the same optimal solution after only 0.52 s. This is generally not the case. For example, consider the problem of distinguishing between models 1 and 2 in only \(N = 2\) steps, with the relaxed control constraint
\[
U = \{[\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}] : \|\mathbf{u}_k\|_\infty \leq 3.5, \ k = 1, \ldots, N\}. \quad (37)
\]
In this case, reducing the number of generators from 4 to 2 causes an increase in the norm of the optimal separating input from 3.50 to 4.90. The additional conservatism in the two-generator case leads to an unnecessarily large separation of the output sets for the two models.

As a second numerical example, consider two models that are significantly more difficult to distinguish, which are the above Models 1 and 2 but with the modifications
\[
\begin{align*}
\mathbf{A}[1] &= \begin{bmatrix} 0.8820 \quad 0.2940 \\ -0.2940 \quad 1.0290 \end{bmatrix}, \\
\mathbf{A}[2] &= \begin{bmatrix} 0.6587 \quad 0.2940 \\ -0.2940 \quad 0.7030 \end{bmatrix}.
\end{align*}
\]
\(^1\)Desktop PC (Intel i7, 3.4GHz, 8 GB RAM) running Ubuntu 11.10 and using a single core; Optimization using CPLEX 12.2 [29].
and \( X_0 = \{ 2 \mathbf{1}_{2 \times 2}, [2 \ 1]^T \} \). Again, it is assumed that one of these two models is active on all of \([0, N]\). The minimum horizon for which a separating input exists (with 4 generators) is found to be \( N = 65 \). Despite this long horizon, the proposed method locates the optimal separating input on \([0, N]\) in 0.15 s.

Finally, two methods for the computation of \( \mathcal{J} \) have been compared: the method discussed in [16] and the zonotope-based approach proposed in this paper (right).

VI. CONCLUSIONS

A deterministic method is proposed for computing the set of inputs that are guaranteed to lead to a fault diagnosis in a specified period of time. It is shown that this set can be characterized in terms of the complement of a finite number of zonotopes, and can be computed efficiently and reliably. Furthermore, a computationally practical optimization formulation has been derived for choosing an optimal separating input in a flexible way. Using this formulation, a separating input of minimum norm can be computed, or the set of separating inputs can be used within a more complex predictive control calculation.

REFERENCES