Guaranteed Active Fault Diagnosis for Uncertain Nonlinear Systems

Joel A. Paulson², Davide M. Raimondo³, Rolf Findeisen¹, Richard D. Braatz², and Stefan Streif¹

Abstract—An input design method is presented to actively isolate faults for polynomial or rational systems in the presence of unknown-but-bounded uncertainties. For active fault isolation, the input is required to lead to outputs consistent with at most one fault model despite disturbances, measurement noise, and parametric uncertainty. This task is posed in terms of a bilevel optimization problem where the inner program verifies, for a given input, that the outputs are consistent with at most one model, while the outer program determines the minimally harmful input. Because of the nonlinear dynamics, we propose to replace the inner program with a convex relaxation that can be efficiently solved while still guaranteeing fault detection and isolation. The approach is numerically demonstrated on a two-tank system with three fault models.

I. INTRODUCTION

Fault detection and isolation (FDI) has become increasingly important in maintaining stable, reliable, and profitable operations in the presence of component malfunctions, drifting parameters, and other abnormal events. Process disturbances, measurement noise, model nonlinearities, and other sources of uncertainty make fault diagnosis a challenging task that is further complicated by the steadily increasing complexity of industrial systems. By now, many methods have been proposed to address these challenges, such as residual- and observer-based methods [1]–[4], set-based approaches [5], [6], and data-based methods [1]. The majority of these methods are passive, meaning that the inputs are not actively changed and that the fault status of the system is deduced only on the basis of measurements obtained during standard operation, compared with model predictions or historical data. However, faults may not be detectable or isolable at the current operating conditions, or when faults are obscured by the corrective action of the control system itself. Then it is necessary to inject a signal into the system to improve fault detectability and isolability, which is an approach known as active fault diagnosis [7]. Although active fault diagnosis can significantly improve fault isolation, the required excitations can have adverse effects on the process that must be minimized.

The active input design problem has been addressed using deterministic [8]–[13], stochastic [7], [14]–[16], and hybrid [17]–[19] approaches. With few exceptions, the work is restricted to linear systems whereas, in reality, almost all systems exhibit nonlinear dynamics.

This paper presents a deterministic model-based approach for active fault diagnosis of nonlinear systems with polynomial or rational dynamics subject to unknown-but-bounded uncertainties and disturbances. This work builds on a framework for set-based analysis of nonlinear systems (cf. [18]) and allows for a deterministic formulation of the fault diagnosis problems. To decide whether a fault has occurred with certainty, the measured outputs are compared with the set of outputs from the nominal and faulty models [19].

The main contribution in this work is a method to determine optimal inputs for guaranteed active fault diagnosis despite process nonlinearities and uncertainties. To achieve this, the input design problem is formulated as a bilevel optimization problem in which the outer program minimizes the two-norm input, while the inner program guarantees that only inputs that separate the output sets are selected (i.e., inputs that guarantee the output measurements are consistent with at most one model). In order to solve this nonconvex bilevel problem, we propose a method that takes advantage of the fact that the relaxed inner program is convex at a fixed input (cf., [19]). As suboptimality of this input is directly related to the tightness of the convex relaxation (i.e., how close the relaxation approximates the true set), different ways to tighten the relaxation are presented and discussed. Moreover, to provide a measure of output set separation, the inner convex program is reformulated in terms of an optimization-based consistency measure (cf. [18]).

II. PROBLEM STATEMENT

Given a process subject to \(n_f\) possible faults, consider discrete-time models of the form

\[
\begin{align*}
  x_{k+1} &= g(x_k, u_k, w_k, p) \\
  y_k &= h(x_k, u_k, v_k, p)
\end{align*}
\]

representing the nominal and all the possible faulty dynamics. The superscript \(i \in \mathcal{J} := \{0, 1, \ldots, n_f\}\), denotes the various fault scenarios \(\mathcal{F} := \{f_0, f_1, \ldots, f_{n_f}\}\) considered, with \(f_0\) corresponding to the nominal model.

In (1), \(x_k \in \mathbb{R}^{n_x}\), \(y_k \in \mathbb{R}^{n_y}\), \(u_k \in \mathbb{R}^{n_u}\), \(w_k \in \mathbb{R}^{n_w}\), \(v_k \in \mathbb{R}^{n_v}\) denote the system states, inputs, outputs, process noise, and measurement noise at time-point \(k\), respectively. The possibly uncertain model parameters are denoted by \(p \in \mathbb{R}^{n_p}\). The functions \(g\) and \(h\) are assumed to be polynomial or rational. In general, each fault model \(f_i\) has its own set of variables \(x_k^{[i]}\), \(w_k^{[i]}\), \(u_k^{[i]}\), \(y_k^{[i]}\), \(p^{[i]}\) and functions \(g^{[i]}\) and \(h^{[i]}\) with possibly different dimensions. However, to shorten and...
simplify the notation, the superscript \([i]\) is omitted for all
variables and functions appearing in (1).

The model parameters are assumed to be unknown-but-bounded. Furthermore, the states as well as the process and measurement noise are assumed bounded, leading to the overall uncertainty description:

\[ p \in \mathcal{P}, \ x_k \in \mathcal{X}_k, \ w_k \in \mathcal{W}_k, \ v_k \in \mathcal{V}_k, \ \forall k \in \mathcal{T} \] (2)

where \( \mathcal{T} = \{0, 1, \ldots, n_t\} \) is the collection of time instances considered. To simplify the presentation, only one model is assumed to be active during \( \mathcal{T} \). The framework can easily be extended to include fault sequences.

### A. Set-based Fault Diagnosis

This work employs the notion of consistency for the design of separating inputs. The core idea is to check consistency by means of a feasibility problem that takes into account the model and the uncertainty description. The feasibility problem is derived next.

Denote sequences on \( \mathcal{T} \) by \( \overline{\sigma} := [\sigma_0, \sigma_1, \ldots, \sigma_{n_t}]^T \). Then, given a sequence of inputs and outputs, \((\overline{u}, \overline{y})\), combining (1) with the uncertainty description in (2) into a single feasibility problem (FP) for each model \( i \in \mathcal{I} \) gives

\[
\begin{align*}
\text{FP}^{[i]}(\overline{u}) : \quad & \text{find } \xi_{\text{FP}} \\
& \text{s.t. } x_{k+1} = g(x_k, w_k, x_k, p), \quad \forall k \in \mathcal{T} \setminus {n_t} \\
& \quad \text{and } x_k = h(x_k, w_k, v_k, p), \quad \forall k \in \mathcal{T} \\
& \quad x_k \in \mathcal{X}_k, \quad w_k \in \mathcal{W}_k, \quad v_k \in \mathcal{V}_k, \quad \forall k \in \mathcal{T} \\
& \quad p \in \mathcal{P}
\end{align*}
\]

where \( \xi_{\text{FP}} = [x_0, \ldots, x_{n_t}, w_0, \ldots, w_{n_t}, y_0, \ldots, y_{n_t}, v_0, \ldots, v_{n_t}, p]^T \) lumps all the variables except the input into a single vector. The superscript \([i]\) on the variables in \( \text{FP}^{[i]}(\overline{u}) \) has been omitted to simplify notation.

**Definition 1** (Consistency). Input and output sequences \((\overline{u}, \overline{y})\) are said to be consistent with a fault candidate \( f^{[i]} \) if \( \text{FP}^{[i]}(\overline{u}) \) admits a solution. Otherwise, the sequence \((\overline{u}, \overline{y})\) is called inconsistent.

### B. Optimal Input Design for Active Fault Diagnosis

Passive set-based fault detection and isolation based on \( \text{FP}^{[i]}(\overline{u}) \) has been proposed in [6] considering fixed or unknown-but-bounded inputs. The main focus of this work is the design of separating inputs for active fault detection and isolation in the presence of uncertainties, which is formally stated below.

**Definition 2** (Robust Separating Input). An input \( \overline{u} \) separates models \( f^{[i]} \) and \( f^{[j]} \) at time \( n_t \), if \( \overline{y} \) such that \((\overline{u}, \overline{y})\) is consistent with both \( f^{[i]} \) and \( f^{[j]} \). If \( \overline{u} \) separates all model output sets subject to output, process, and parametric uncertainties, then \( \overline{u} \) is a robust separating input.

The following combined feasibility problem (cf. [19]) is defined as

\[
\begin{align*}
\text{FP}^{[i,j]}(\overline{u}) : \quad & \text{find } \xi^{[i]}_{\text{FP}}, \xi^{[j]}_{\text{FP}} \\
& \text{s.t. } \text{constraints in } \text{FP}^{[i]}(\overline{u}) \cup \text{constraints in } \text{FP}^{[j]}(\overline{u}) \\
& \quad y^{[i]}_{n_t} = y^{[j]}_{n_t}
\end{align*}
\]

which checks if there exists \((\xi^{[i]}_{\text{FP}}, \xi^{[j]}_{\text{FP}})\) such that the measured outputs of models \( f^{[i]}, f^{[j]} \in \mathcal{F} \) intersect at time \( n_t \). Both models have their own distinct set of states, process and measurement noise, parameters, and outputs that appear as free variables in the program. This problem provides a robustness certificate when the program is shown to be infeasible for all possible model combinations [19]. The set of all robust separating inputs is defined as \( \mathcal{U}^* := \{ \overline{u} : \text{FP}^{[i,j]}(\overline{u}) \text{ is infeasible, } \forall (i, j) \in \mathcal{I}, i > j \} \).

Note that \( \text{FP}^{[i,j]}(\overline{u}) \) is defined using only the output at the final time meaning any \( \overline{u} \in \mathcal{U}^* \) applied to the system guarantees that the output sets at the final time do not intersect. This can be replaced with requirements in which the output sets either do not intersect either at all, or at least at one time instance of \( \mathcal{T} \). The first requirement can be straightforwardly obtained by replacing the condition \( y^{[i]}_{n_t} = y^{[j]}_{n_t} \) with \( y^{[i]}_{k} = y^{[j]}_{k} \), \( \forall k \in \mathcal{T} \), and the second requirement is considered in [19].

Note that for unique fault diagnosis, every possible pairwise comparison of the models of interest must have separated output sets [19], which requires a check of \( \binom{n_t+1}{2} \) distinct combinations. This number can be reduced by applying a hierarchical approach that first selects inputs that separate a number of the models and then refines the input to improve separability for closely related models.

In this work, we not only want to separate the models, we also want to derive an input that is optimal with respect to a certain performance index. The overall problem considered in this paper is stated as:

**Problem 1.** (Optimal Separating Input \( \overline{u} \)). Find a separating input \( \overline{u} \) that solves

\[
\begin{align*}
\text{min } \quad & \overline{u}^\top R \overline{u} \\
\text{s.t. } & \overline{u} \in \mathcal{U} \cap \mathcal{U}^* \quad (3)
\end{align*}
\]

where \( R \) is a positive-semidefinite weighting matrix and \( \mathcal{U} \) is a convex set that represents the input constraints.

The solution \( \overline{u} \), assuming that it exists, is guaranteed to provide robust fault diagnosis within \( n_t \) time steps. However, due to the nonlinearities and uncertainties, the \( \text{FP}^{[i,j]}(\overline{u}) \) is generally nonconvex, making \( \mathcal{U}^* \) very difficult to characterize. Next we efficiently tackle these problems using convex relaxations (Sec. III) and bilevel optimization (Sec. IV).

### III. CONVEX RELAXATIONS

Providing a robustness certificate is not a trivial task for nonlinear and uncertain systems due to the general nonconvexity of \( \text{FP}^{[i,j]}(\overline{u}) \). Although difficult, this is a necessary task for characterizing the set of robust separating inputs \( \mathcal{U}^* \) that appears in the optimization (3).

For polynomial systems, we can convexly relax the single feasibility problems, \( \text{FP}^{[i]}(\overline{u}) \), into a semidefinite or linear program that provides provable inconsistency certificates for the existence of solutions [6], [19]. These convex outer approximations of the original, generally nonconvex, feasible sets can be used in \( \text{FP}^{[i,j]}(\overline{u}) \) so that solutions can efficiently be computed [19]. The drawback of such an approach is that the outer approximations introduce conservatism into
the problem, so they only characterize a subset of $U^*$. Nevertheless, robustness certificates derived from these convex relaxations allow the computation of an optimal and robust separating input (Sec. IV), where the tightness of the relaxation controls the degree of conservatism in the solution.

A. Relaxation Method

The first step is to transform the $Fp[i] (\bar{u})$ into a quadratically constrained program (QCP) by expressing all the dynamic and output equations as $\xi^T A \xi$, where $\xi \in \mathbb{R}^{n_{\xi}}$ is a minimal basis of monomials for the equations of model $f[i]$ and $A \in \mathbb{R}^{n_{\xi} \times n_{\xi}}$ is a symmetric matrix. The vector $\xi$ contains the elements of $\xi_{FP}$, the constant term 1, and any additional monomials greater than degree two necessary to represent all equations. Such a quadratic decomposition can always be found; however, it is in general not convex. By introducing a symmetric matrix $X = \xi \xi^T$ and replacing the resulting trace$(X) \geq 1$ and rank$(X) = 1$ constraints with the weaker positive-semidefinite constraint $X \succeq 0$, the $Fp[i] (\bar{u})$ is relaxed into a semidefinite program (SDP) denoted by $SDP[i] (\bar{u})$. It is important to note that this relaxation only increases the solution set. See [18] for further details and references regarding this relaxation method.

Using the index set $L = \{1, \ldots, n_{eq}\}$ where $n_{eq}$ represents the total number of equality constraints in model $f[i]$ and assuming that (2) can be written as $n_{ineq}$ linear inequality constraints in terms of the uncertain variables (i.e., $B \xi \leq 0$ where $B \in \mathbb{R}^{n_{ineq} \times n_{\xi}}$) results in

$$SDP[i] (\bar{u}) : \begin{cases} \text{find } X \\ \text{s.t. } & \text{trace}(A(l)(\bar{u})X) = 0, \forall l \in L \\ & \text{trace}(ee^T X) = 1 \\ & B(\bar{u})Xe \leq 0 \\ & X \succeq 0 \end{cases}$$

where $e = [1, 0, \ldots, 0]^T \in \mathbb{R}^{n_{\xi}}$. Note that the matrices $B(\bar{u})$ and $A(\bar{u})$ depend on the input $\bar{u}$; however, at this level, the input is treated as a constant (the input is minimized in an outer loop optimization explained in Sec. IV).

To deal with larger problems that involve many constraints and variables, the $SDP[i] (\bar{u})$ can be relaxed to a linear program (LP) denoted by $LP[i] (\bar{u})$, which is done by simply dropping the constraint $X \succeq 0$.

Solving $SDP[i] (\bar{u})$ instead of the original $Fp[i] (\bar{u})$ usually leads to the inclusion of false solutions. As a result, the input set $U^*$ for which separation is guaranteed gets overly restricted. To alleviate this effect, constraints can be added that are redundant in the $\xi_{FP}$ basis, but are not necessarily redundant in the higher dimensional basis $X$. Such redundant constraints can be constructed by [18]:

$$\text{trace}(B(\bar{u})^T B(\bar{u})_J X) \geq 0, \forall (i, j) \in \{1, \ldots, n_{ineq}\}, i \geq j$$

where $B(\bar{u})_i \in \mathbb{R}^{1 \times n_{\xi}}$ represents the $i$th row of $B(\bar{u})$. Eq. (4) includes the McCormick relaxations for bilinear monomials. Including these redundant constraints in the $SDP[i] (\bar{u})$ adds $n_{ineq}(n_{ineq} - 1)/2$ constraints that make the solution more demanding to compute but can significantly tighten the solution set [18]. The tradeoff between speed and conservatism is discussed in Sec. V in the context of a numerical example.

B. Robustness Certificates Using Convex Relaxations

This section shows how to efficiently determine robustness certificates, i.e., guaranteed output set separation, using the convexly relaxed feasibility problems.

Define the problems $SDP[i,j] (\bar{u})$ and $LP[i,j] (\bar{u})$ similarly as $Fp[i,j] (\bar{u})$ (see Sec. II). Since the convexly relaxed problems must contain the original output sets (i.e., provide infeasibility certificates for $Fp[i,j] (\bar{u})$), they can also be used to check if a given $\bar{u}$ is a robust separating input (see [19]). We can now similarly define the sets $U_{SDP} := \{ \bar{u} : SDP[i,j] (\bar{u}) \text{ is infeasible}, \forall (i, j) \in J, i > j \}$ and $U_{LP} := \{ \bar{u} : LP[i,j] (\bar{u}) \text{ is infeasible}, \forall (i, j) \in J, i > j \}$. Since the relaxed output sets must contain the actual set, but not vice versa, it directly follows that $U_{LP} \subseteq U_{SDP} \subseteq U^*$. Note that $U_{LP} \subseteq U_{SDP}$ holds only when the same set of constraints are included in both relaxations.

IV. OPTIMAL ROBUST SEPARATING INPUT

In this section, Problem 1 is tackled by employing robustness certificates and bilevel optimization. The inner program of the bilevel optimization verifies, for a given input, that the output sets do not overlap and are therefore consistent with at most one model, while the outer program determines the minimally harmful input. $Fp[i,j] (\bar{u})$ could be directly solved to determine if the sets overlap at a given input. However, the result from the feasibility test (i.e., infeasibility or feasibility) provides no suitable measure for the outer program to determine a direction that will improve the objective function value. This situation can be avoided by reformulating the $Fp[i,j] (\bar{u})$ in terms of an optimization-based consistency measure $\delta$ whose value provides a direct measure of output set separation for a fixed input. This measure provides useful information for the outer solver to guide the input toward a minimum, as discussed below.

Section IV-A formally presents the measure for output set separation. Section IV-B elaborates on the bilevel optimization strategy. The presented approach to determine output set separation does not require the explicit computation of reachable sets as, for example, done in [20]. Also note that a similar consistency measure has been used within the context of outlier detection in [18].

A. Measure of Output Set Separation

The determination of the output set separation requires a reformulation of $Fp[i,j] (\bar{u})$ in terms of a scaling parameter $\delta$. Depending on $\delta$, the bounds will be either inflated ($\delta > 0$) or shrunk ($\delta < 0$) until the output sets intersect. Intersection of the output sets is checked by solving $Fp[i,j] (\bar{u})$ for a given $\delta$ and input $\bar{u}$. Since many $\delta$ values will satisfy this relationship, $\delta$ is minimized under the constraints that the output sets still overlap. $\delta$ then provides a measure of separation in the sense that the larger the distance between
the output sets, the larger the minimum value of $\delta$. If an inflation of the bounds, i.e. $\delta > 0$, was required for output set separation, then the provided input separates the output sets despite all bounded uncertainties.

The inflated bounds (i.e., inequality constraints) can be written as $B_{in} \xi_{FP} \leq b_{in}(\tilde{u}) + \delta$, where $B_{in}$ is a sparse matrix that relates each of the variables in $\xi_{FP}$ to their lower and upper bounds in the vector $b_{in}(\tilde{u})$, which might depend on the input. Using this definition, $FP[i,j](\tilde{u})$ can be reformulated in terms of the minimum inflation parameter $\delta$: $\hat{\delta}[i,j](\tilde{u}) := \min_{\delta} \delta$

s.t. equality constraints in $FP[i](\tilde{u})$

equality constraints in $FP[j](\tilde{u})$

(5)

Theorem 1: $FP[i,j](\tilde{u})$ is infeasible if $\hat{\delta}[i,j](\tilde{u}) > 0$.

Proof: Choose any $i, j \in J$ and any $\tilde{u} \in U$. If $\hat{\delta}[i,j](\tilde{u}) > 0$, then (5) does not have a feasible point with $\delta \leq 0$. Therefore, $\hat{\mathcal{H}}[i,j](\xi_{FP}, \xi_{FP})$ such that the constraints stated in $FP[i,j](\tilde{u})$ hold. Thus, $FP[i,j](\tilde{u})$ is infeasible. Conversely, assume that $FP[i,j](\tilde{u})$ is infeasible. Then $\hat{\mathcal{H}}[i,j](\xi_{FP}, \xi_{FP})$ such that the constraints in $FP[i,j](\tilde{u})$ hold, which implies that (5) does not have a feasible solution with $\delta \leq 0$. Thus, $\hat{\delta}[i,j](\tilde{u}) > 0$.

From Theorem 1, we can express the set of separating inputs as $U^* = \{ \tilde{u} : \hat{\delta}[i,j](\tilde{u}) > 0, \forall (i, j) \in J, i > j \}$. It directly follows from Sec. III-B that:

$U^*_{SDP} = \{ \tilde{u} : \hat{\delta}[i,j](\tilde{u}) > 0, \forall (i, j) \in J, i > j \}

U^*_{LP} = \{ \tilde{u} : \hat{\delta}[i,j](\tilde{u}) > 0, \forall (i, j) \in J, i > j \}$

and $\hat{\delta}[i,j](\tilde{u}) \leq \hat{\delta}[i,j](\tilde{u}) \leq \hat{\delta}[i,j](\tilde{u})$ at a fixed $\tilde{u}$ where $\hat{\delta}[i,j](\tilde{u})$ and $\hat{\delta}[i,j](\tilde{u})$ are defined in the same manner as (5) with the convexly relaxed constraints used in place of the actual constraints.

B. Determining the Optimal Input

The previous section derived a method for characterizing the set of separating inputs with the measure of output set separation $\hat{\delta}[i,j](\tilde{u})$. Using this framework, (3) can be redefined as:

$$\inf_{\tilde{u} \in \ell} \tilde{u}^\top R\tilde{u}$$

s.t. $\hat{\delta}[i,j](\tilde{u}) > 0$, $\forall (i, j) \in J$, $i > j$.

(7)

$\hat{\delta}[i,j](\tilde{u})$ is defined as the solution to a nonconvex optimization which clearly makes (7) a nonlinear nonconvex bilevel program (BLP). Only very few results for BLPs with nonconvex inner programs exist [21]. The computational complexity can be reduced by convexly relaxing the $\hat{\delta}[i,j](\tilde{u})$ constraint in (7), using the relaxations in Sec. III for example, to form a convex BLP that can be solved using existing algorithms [22]. However, we solve this problem differently as explained below.

The resulting convexly relaxed problem is

$$\min_{\tilde{u} \in \ell} \tilde{u}^\top R\tilde{u}$$

s.t. $\hat{\delta}_{CR}[i,j](\tilde{u}) \geq \epsilon$, $\forall (i, j) \in J$, $i > j$

(8)

where the subscript CR stands for convex relaxation (e.g., SDP or LP). In (8), a minimum separation threshold $\epsilon > 0$ is introduced to ensure that there exists a $\tilde{u}^*$ that attains the minimum. The program for $\hat{\delta}_{CR}[i,j]$ has the same structure as (5) with the constraints defined in terms of the particular relaxation. Note that $\hat{\delta}_{CR}[i,j](\tilde{u})$ (i.e., the inner program) is now convex for a fixed $\tilde{u}$.

Replacing $\hat{\delta}[i,j](\tilde{u})$ with its equivalent Karush-Kuhn-Tucker (KKT) conditions leads to a single program with, first, a large number of nonconvex complementary constraints equal to the number of inequalities in $\hat{\delta}_{CR}[i,j](\tilde{u})$ and, second, variable bounds that are generally complicated and highly nonlinear functions of the input due to its propagation through the system dynamics. As this procedure is used by many of the aforementioned algorithms to solve BLPs, computing the solution can be very computationally expensive, which is compounded by a combinatorial growth in the number of inner programs with the number of models to be separated and a polynomial growth in the size of each inner program with the number of time points.

Although standard algorithms can be used to solve (8), we propose a simple alternative that is easy to implement regardless of model complexity. The idea is to use a deterministic local nonlinear solver to supply input values to the convex inner program, $\hat{\delta}_{CR}[i,j](\tilde{u})$, that can be efficiently solved using a variety of solvers, such as CPLEX [23] or SeDuMi [24]. Because $\hat{\delta}_{CR}[i,j](\tilde{u})$ provides a measure of distance between the set of outputs, the outer solver can easily compute a gradient of this constraint with respect to the input using finite differencing. Combined with the gradient/subgradients of the outer objective function, a feasible descent direction can be estimated to update $\tilde{u}$. This process can be repeated until optimality is achieved. This approach is used for the example of a two-tank system in the next section.

V. EXAMPLE

Consider the two-tank system in Fig. 1.

![Fig. 1. Sequential two-tank system.](image-url)
system input and can vary with time. The states (outputs) of the system are the true (measured) heights of the tanks, which are denoted by \( x_1(y_1) \) and \( x_2(y_2) \) for tanks 1 and 2, respectively. This example considers operating conditions in which the system satisfies \( x_1 \geq x_2 \) under all fault scenarios.

Three fault scenarios are considered. First \( (f_1) \), a leakage in tank 1 represented by the flow \( q_{1,\text{L}} \). Second \( (f_2) \), the valve \( V_1 \) becomes clogged and its throughput is reduced by 50%. Third \( (f_3) \), a large leak occurs in tank 2 that increases the outflow to 5 times its nominal value. These scenarios can all be represented with the same model structure. Therefore, we will first derive a general description of the system and then present the different sets of parameters in the nominal and faulty models.

Under the aforementioned assumptions, the nonlinear discrete-time model is given by

\[
\begin{align*}
x_{1,k+1} &= x_{1,k} + \frac{\Delta t}{A} (q_{0,k} - q_{1,\text{L}} - q_{12,k}) \\
x_{2,k+1} &= x_{2,k} + \frac{\Delta t}{A} (q_{12,k} - q_{23,k})
\end{align*}
\]

where \( \Delta t \) is the sampling time and the flowrates are

\[
\begin{align*}
q_{0,k} &= u_k, \quad q_{1,\text{L}} = c_{1L} \sqrt{x_{1,k}} \\
q_{12,k} &= c_{12} \sqrt{x_{1,k} - x_{2,k}} \\
q_{23,k} &= c_{23} \sqrt{x_{2,k}}
\end{align*}
\]

The parameters \( c_{1L} \), \( c_{12} \), and \( c_{23} \) are valve coefficients. The measured heights are corrupted with measurement noise:

\[
\begin{align*}
y_{1,k} &= x_{1,k} + v_{1,k} \\
y_{2,k} &= x_{2,k} + v_{2,k}
\end{align*}
\]

The non-polynomial parts of (9) can be reformulated by introducing additional variables and constraints [6]:

\[
\begin{align*}
(dw_{12,k})^2 &= x_{1,k} - x_{2,k} \\
(Sq_{1k})^2 &= x_{1,k} \\
(Sq_{2k})^2 &= x_{2,k}
\end{align*}
\]

Placing \( dw_{12,k}, Sq_{1k}, \) and \( Sq_{2k} \) in (9) instead of the appropriate square root terms results in a polynomial model. Defining the parameter vector \( p = [A, c_{1L}, c_{12}, c_{23}]^\top \), the fault models can be classified as

\[
\begin{align*}
f^{[0]} &= \{(9)-(12), \quad p^{[0]} = [A, 0, c_{12}, c_{23}]^\top \} \\
f^{[1]} &= \{(9)-(12), \quad p^{[1]} = [A, c_{1L}, c_{12}, c_{23}]^\top \} \\
f^{[2]} &= \{(9)-(12), \quad p^{[2]} = [A, 0, 0.5c_{12}, c_{23}]^\top \} \\
f^{[3]} &= \{(9)-(12), \quad p^{[3]} = [A, 0, c_{12}, 5c_{23}]^\top \}
\end{align*}
\]

### B. Simulation

Eq. (8) was solved to compute the optimal separating input \( \tilde{u}^* \) for the nominal and three fault models summarized in (13) using the method in Sec. IV. The convex relaxations in Sec. III were used. The implementation was done in Matlab using fmincon as the outer solver and CPLEX [23] (respectively, CVX [25] with SeDuMi [24]) as the inner solver for the linear (respectively, semidefinite) relaxations.

All simulations used the parameter values \( \Delta t = 5 \text{ s}, A = 1.54 \times 10^{-2} \text{ m}^2, c_{1L} = c_{12} = c_{23} = 1.2 \times 10^{-4} \text{ m}^5/\text{s}, R = I_{n_u \times n_u} \), and \( n_t = 4 \). The uncertain initial tank levels were chosen to be \( x_{1,0} \in [0.95, 1.05] \) and \( x_{2,0} \in [0.475, 0.525] \) with bounded measurement noise \( v_{1,k}, v_{2,k} \in [-0.05, 0.05], \forall k \in T \) all in units of meters. Bounds on the remaining states and outputs were calculated using the model equations, the uncertain initial condition interval, and the bounded measurement noise using the Matlab Intlab toolbox [26]. The initial guess for \( \tilde{u} \) in the outer program was chosen to be 0.1 for all time points in all simulations. Different initial guess values resulted in the exact same optimal solution for this example. Note that our proposed method easily handles process noise and parametric uncertainty; however, they were excluded from this particular example for simplicity.

### C. Results

Table I compares our method for different levels of convex relaxations. The linear (respectively, semidefinite) relaxation without (4) is denoted as LP (respectively, SDP). The letter “t” stands for “tight” and precedes the abbreviation when the additional constraints in (4) are included in the relaxation. The LP relaxation is the fastest method by far, but is also the most conservative (i.e., requires a larger input to separate the fault models). The LP relaxation found a separating input 40 to 50 times faster than the t-LP relaxation, but with a norm that is 35 to 50% larger.

Another interesting observation is that the SDP relaxation was more conservative than the t-LP case. This example highlights the importance of the additional constraints (4), which include the McCormick relaxations, in directly affecting the tightness of the relaxation. Furthermore, optimization time for the t-LP relaxation scaled much more favorably with the number of models in the simulation than for the SDP relaxation. Note that, in order to reduce the online computational burden, it is always possible to compute an approximate explicit solution to the input separation problem in the same way as described in [27].

Fig. 2 shows Monte Carlo samples of the outputs for each of the four models when the optimal separating input \( \tilde{u}^* \), computed using the t-LP relaxation, is injected. In the lower right panel, we can clearly see that all output sets are completely separated at the chosen final time point \( k = n_t = 4 \), which indicates that any sequence of these measurements taken on the interval \( T \) are consistent with at most one model. Thus, a complete fault diagnosis of the system has been achieved.

### VI. CONCLUSIONS

A deterministic method is proposed for computing a guaranteed separating input for fault isolation of nonlinear polynomial and rational uncertain systems based on convex relaxations and bilevel optimization. The derived bilevel program has a convex inner program and could be solved to global optimality using standard methods, e.g., branch and bound. We propose an alternative solution method that uses a nonlinear outer solver to supply inputs to the convex inner program and iteratively step towards a minimum. Although global optimality is not guaranteed \textit{a priori}, the proposed
TABLE I
Comparison of the proposed method at different levels of convex relaxation. By including the additional constraints in (4), t-LP is a tighter relaxation than LP. Computations performed on a desktop PC (Intel i7, 2.7 GHz, 8 GB RAM) running Windows 7 (64-bit) using a single core.

<table>
<thead>
<tr>
<th>Models, $f^{(i)}$</th>
<th>Relaxation</th>
<th>$|\hat{\alpha}^*| \times 10^4$</th>
<th>CPU time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i = {0, 1}$</td>
<td>LP</td>
<td>39</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>t-LP</td>
<td>26</td>
<td>468</td>
</tr>
<tr>
<td></td>
<td>SDP</td>
<td>34</td>
<td>270</td>
</tr>
<tr>
<td>$i = {0, 1, 2}$</td>
<td>LP</td>
<td>166</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>t-LP</td>
<td>123</td>
<td>882</td>
</tr>
<tr>
<td></td>
<td>SDP</td>
<td>147</td>
<td>834</td>
</tr>
<tr>
<td>$i = {0, 1, 2, 3}$</td>
<td>LP</td>
<td>166</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>t-LP</td>
<td>123</td>
<td>1014</td>
</tr>
<tr>
<td></td>
<td>SDP</td>
<td>147</td>
<td>1710</td>
</tr>
</tbody>
</table>

method is much more computationally efficient and showed promising results when applied to an example problem. Furthermore, the general formulation is flexible with respect to the solution method and choice of objective, constraints, and number of possible fault models.

Further work in analyzing the degree of conservatism added to the solution from the convex relaxation methods is recommended. Of particular interest would be an adaptive approach that heuristically adds constraints that are most likely to tighten the relaxation while warm starting the algorithm with the previously computed solution.

REFERENCES


