Multi-objective Complexity Reduction for Set-based Fault Diagnosis

Anton Savchenko\textsuperscript{1,2}, Petar Andonov\textsuperscript{1,2}, Philipp Rumschinski\textsuperscript{1} and Rolf Findeisen\textsuperscript{1}

Abstract—Fault diagnosis methods ensure safe operation of industrial plants. Steadily increasing appearance of larger and interconnected systems and the necessity to take process uncertainties into account drives the need for reliable diagnosis procedures. Set-based frameworks for model-based fault diagnosis allow to handle these challenges, albeit at a high cost of computations. We propose a method to reduce the complexity of polynomial discrete-time models that retain the guarantee of fault detection. The relaxation-based method substitutes uncertain parts of model dynamics which are not relevant to diagnosing the fault. The method is illustrated with a fault detection example for an automatic air conditioning system of a building.

Index Terms—Fault detection and isolation; Set-based methods; Complexity reduction; Process monitoring.

I. INTRODUCTION

Ensuring safe operation of industrial plants is an important task. To prevent component degradation as well as failures one has to be able to diagnose occurrences of faults in a timely manner [6]. As plants grow in size and complexity, the presence of faults can become apparent only from considering multiple plant components and corresponding interconnections. This, in turn, increases the complexity of the models one employs for diagnosis.

The task of fault detection is furthermore hindered by inherent disturbances in the process, as well as limited knowledge of process parameters and measurement data leading to a model-plant mismatch that can be misinterpreted as a fault. Distinguishing between the occurrence of a fault and the discrepancies between the plant and its model is, thus, a major challenge. Under the assumption that the acceptable noise and parametric uncertainties can be bounded to known sets we proposed in [11, 12] a set-based fault diagnosis approach. It provides guaranteed results, however, at a high computational load, making online operation challenging.

The aim of this work is to improve the computational complexity of the employed set-based framework for fault diagnosis. Starting with a full polynomial model we aim at determining a simpler approximation, that shows a better trade-off between the performance and the quality of the resulting diagnosis procedure. In the literature one distinguishes two common approaches for model reduction. One is based on constructing projections from controllability and observability criteria, e.g. principal component analysis [10], singular perturbation approximations [7, 9], or in the nonlinear case Karhunen-Loève expansion [15] and proper orthogonal decomposition [8]. A second set of methods relies on representing the transfer function via a power series and exploiting moment matching, Padé approximation or Krylov subspaces [3, 5]. To obtain a reduced model both approaches truncate the component basis, further increasing the discrepancy between the actual plant and the new model. Even if the methods provide error bounds, such as Hankel norm approximations [4], one has to apply it iteratively for the considered time horizon, which can decrease the quality of approximations.

This work proposes an approach that preserves the guarantee of set-based fault diagnosis, expanding the results of our previous work [14]. Instead of truncating the unimportant constraint elements, the presented method lifts the variable constraint elements, the presented method lifts the variable space and generates a relaxed model as its projection. This approach allows to remove unimportant for the diagnosis plant elements and interconnections, leading to an outer-approximation of the feasible fault scenarios while guaranteeing fault diagnosis.

II. FAULT DIAGNOSIS

We begin by outlining the general form of the set-based feasibility formulation [11], that is employed for the complexity reduction in the later sections.

The faultless plant is modeled via the following implicit discrete-time polynomial functions:

\begin{align*}
g_0(x(k+1), x(k), u(k), p) &= 0,
g_0(y(k), x(k), u(k), p) &= 0,
\end{align*}

where \( x(k) \in \mathbb{R}^{n_x} \times \mathbb{Z}^{d_x}, p \in \mathbb{R}^{n_p} \times \mathbb{Z}^{d_p}, u(k) \in \mathbb{R}^{n_u} \times \mathbb{Z}^{d_u} \)
and \( y(k) \in \mathbb{R}^{n_y} \times \mathbb{Z}^{d_y} \) are the system states, parameters, inputs and outputs. For the diagnosis we consider a finite time window of length \( n_k \) \( \in \mathbb{N} \), denoting with \( k \in \mathcal{K} = \{1, \ldots, n_k\} \) the set of time indices.

The faults are modeled via the modified dynamics \( g_f(x(k), u(k), p) \) and \( h_f(y(k), x(k), u(k), p) \), that represent the change compared to the faultless scenario. Thus both cases can be modeled using the following dynamics constraints:

\begin{align*}
M(s) : \begin{cases} 
g_0(\cdot) + s \cdot g_f(\cdot) = 0, 
g_0(\cdot) + s \cdot h_f(\cdot) = 0,
\end{cases}
\end{align*}

where \( s \in \mathcal{S} = \{0, 1\} \) denotes whether the fault has occurred or not. We refer to \( M(s) \) as an aggregated model, and the instances \( M(0) \) and \( M(1) \) are denoted as possible fault scenarios. We assume that \( M(s) \) does not contain decoupled dynamics. Although in this contribution we focus on a single
fault scenario, an aggregated model can incorporate any number of anticipated faults, expanding the domain of $s$ (see [12] for a more detailed explanation).

In reality every measurement device has limited accuracy, where an actual value lies inside an interval, e.g., $p \in [p, \overline{p}]$ and $y \in [y, \overline{y}]$. The notation of $p$ and $\overline{p}$ refers to the minimal and maximal admissible bounds. We refer to this type of uncertainty as unknown-but-bounded (UBB).

The uncertainties in the parameters $p$ and output measurements $y$ propagate to $x$ and $u$ and therefore they are also considered UBB. We formalize the description of $p, x, y, u$ as:

\[
p \in \mathcal{P} \subseteq \mathbb{R}^{n_p} \times \mathbb{Z}^{d_p},
\]

\[
y \in \mathcal{Y} = \{y_k \in \mathbb{R}^{n_y} \times \mathbb{Z}^{d_y}, k \in K\},
\]

\[
x \in \mathcal{X} = \{x_k \in \mathbb{R}^{n_x} \times \mathbb{Z}^{d_x}, k \in K\},
\]

\[
u \in \mathcal{U} = \{u_k \in \mathbb{R}^{n_u} \times \mathbb{Z}^{d_u}, k \in K\}.
\]

The states $\mathcal{X}$, outputs $\mathcal{Y}$ and inputs $\mathcal{U}$ are spanned across the time horizon $K$, consisting of the time instances of the measurements. Additionally, we denote the time horizon without the last instance as $K^- = K \setminus \{n_k\}$. To further simplify the notation we write $x(k)$ (resp. $y(k), u(k)$) instead of $x(k) \in \mathcal{X}_k$ (resp. $y(k) \in \mathcal{Y}_k, u(k) \in \mathcal{U}_k$), where $k \in K$.

Given this setup we are able to state definitions for consistent models, as well as fault diagnosis:

**Definition 1 (Consistency):** For a fault scenario $s \in \mathcal{S}$ the model $M(s)$ is said to be consistent with the input signals $\mathcal{U}$ and the output measurements $\mathcal{Y}$ if there exist $p \in \mathcal{P}, x \in \mathcal{X}, u \in \mathcal{U}$ such that $y \in \mathcal{Y}$.

**Definition 2 (Fault candidate):** For a given $s \in \mathcal{S}$ the considered fault scenario is said to be a fault candidate if $M(s)$ is consistent with the measurement data.

In essence, model-based fault diagnosis determines if the measurement data can be reproduced by a faulty plant model.

**Problem 1 (Fault diagnosis):** Determine $s \in \mathcal{S}$, such that $M(s)$ is a fault candidate.

Providing a general solution for Problem 1 is not a trivial task, especially when uncertainties have to be taken into account. For the considered set-based approach it leads to a mixed-integer nonlinear programming problem. Next we present a brief overview of the key aspects of the employed method. For the full description of the approach we refer to [11] and [12].

**Guaranteed set-based fault diagnosis via relaxation**

As shown in [11] and [12], Problem 1 can be solved by determining if the following feasibility problem, composed of semi-algebraic equations, has a solution for $s$:

\[
\begin{align*}
\text{find} & \quad s \\
\text{s.t.} & \quad g(x(k+1), x(k), u(k), p, s) = 0, \quad k \in K^-, \\
& \quad h(y(k), x(k), u(k), p, s) = 0, \quad k \in K, \\
& \quad (x, y, u, p) \in (\mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{P}), \\
& \quad s \in \mathcal{S}.
\end{align*}
\]

The feasibility problem $FP$ describes the set of values of $s$, that satisfy all of the dynamic constraints for the variables within their UBB sets. The feasible set of $FP$ is, therefore, a solution of Problem 1. Solving this feasibility problem directly, however, is not always possible. Following [11, 12] we propose a relaxation technique, transforming the general polynomial formulation into a mixed-integer linear program. This relaxation is conservative, meaning that no admissible solution of the original problem is excluded from the relaxed solution set.

However, even for the relaxed problem the size of the model can be prohibitively large to be solved in a reasonable time. Therefore, we introduce an approach to relax the model formulation, aiming at reducing the complexity of the resulting feasibility problem. We formalize the complexity reduction for fault diagnosis as follows:

**Problem 2 (Complexity reduction):** Find a feasibility formulation, which approximates $FP$ and allows solving Problem 1 with less computational effort.

We propose next a solution to Problem 2 through lifting of the variable space.

**III. REDUCED FEASIBILITY FORMULATION**

In this section we introduce a general concept of complexity reduction based on the lifting of the variable space [14]. The main idea is to simplify the constraints of the feasibility problem $FP$ without significantly changing its projection onto the subspace of $s$, i.e., without affecting the fault diagnosis result. We propose a general method of determining the lifting function based on the equality constraints in the formulation $FP$. It employs an arbitrary number of independent criteria to generate a totally ordered set of monomials ranked by their perceived importance to the task of fault diagnosis.

**Lifting and projecting**

To simplify the notation, all original variables of the problem formulation $FP$ are collected in

\[
\xi(k) = (x(k + 1), x(k), y(k), u(k), p) \in \mathbb{R}^{n_\xi} \times \mathbb{Z}^{d_\xi},
\]

where $n_\xi = 2n_x + n_y + n_u + n_p, d_\xi = 2d_x + d_y + d_u + d_p$ and $k \in K$. With slight abuse of notation, the constraints containing $x(n_k + 1)$ are omitted.

The problem $FP$ can be rewritten using $\xi$ as

\[
\begin{align*}
\text{find} & \quad s \\
\text{s.t.} & \quad Q_\xi(\xi(k), s) = 0, \quad k \in K, \\
& \quad \xi(k) \in \Xi_k, \quad k \in K, \\
& \quad s \in \mathcal{S},
\end{align*}
\]

where the functions $Q_\xi$ reformulate $g$ and $h$ in terms of the variables $\xi$. The set $\Xi_k$ is the Cartesian product of the sets $\mathcal{X}_k, \mathcal{Y}_k, \mathcal{U}_k$ and $\mathcal{P}$, bounding every component of $\xi(k)$.

We introduce next a set of lifting variables $\mu$, that substitute parts of the constraints of $FP_\xi$ as follows

\[
\begin{align*}
\text{find} & \quad s \\
\text{s.t.} & \quad Q_\mu(\mu(k), s) = 0, \quad k \in K, \\
& \quad \mu(k) = l(\xi(k)), \quad k \in K, \\
& \quad \xi(k) \in \Xi_k, \quad k \in K, \\
& \quad s \in \mathcal{S}.
\end{align*}
\]
The functions $l$ are called lifting functions, and their role is to separate those parts of the constraints that are unimportant to diagnosing the fault. Before explaining how to generate these lifting functions, we present the general concept of complexity reduction, that employs such a formulation.

For $FP_{\text{lf}}$ to be equivalent to $FP_{\text{LF}}$ the functions $l$ and $Q_{\text{lf}}$ cannot be chosen arbitrarily. We employ the following sufficient condition for their equivalence:

$$Q_{\text{lf}}(\xi(k), l(\xi(k)), s) \equiv Q_{\xi}(\xi(k), s).$$

The relaxation of the initial problem now can be achieved by splitting $FP_{\text{lf}}$ into two separate subproblems. The first one, that we denote as lifting problem, will determine the feasible set of $\mu$. The second, named projection problem, will then be employed to derive the feasible set of $s$, without a direct relation to those parts of the original dynamics that were lifted.

**Remark 1:** Since the proposed lifting strategy replaces parts of the original dynamics, as a result, the problem can become decoupled. In this case, the states and dynamic constraints of $Q$ that are not coupled with the fault switch $s$ can also be placed in the lifting problem, further reducing the size of the projection problem, as will be demonstrated in Section IV.

Overall, these two subproblems can be written in the following form

$$FP_{\text{lf}}(\Xi_k) : \begin{cases} \text{find } \mu \\ \text{s.t. } \mu(k) = l(\xi(k)), \quad k \in \mathcal{K}, \\
Q_2(\xi(k), \mu(k)) = 0, \quad k \in \mathcal{K}, \\
\xi(k) \in \Xi_k, \quad k \in \mathcal{K}, \end{cases}$$

$$FP_{\text{Proj}}(\Xi_k, \mathcal{W}_k) : \begin{cases} \text{find } s \\ \text{s.t. } Q_1(\xi(k), \mu(k), s) = 0, k \in \mathcal{K}, \\
\mu(k) \in \mathcal{W}_k, \quad k \in \mathcal{K}, \\
\xi(k) \in \Xi_k, \quad k \in \mathcal{K}, \\
s \in \mathcal{S}, \end{cases}$$

where $Q_{\text{lf}} = [Q_1, Q_2]$ separates the decoupled state dynamics, and $\mathcal{W}$ denotes the approximation of the feasible set of the lifting problem, i.e., $FP_{\text{lf}} \subseteq \mathcal{W}$. Solving these problems sequentially will result in an outer approximation of the feasible set $FP$ (see [14]). For the proof of

$$FP_{\text{lf}} \subseteq FP_{\text{Proj}}(\Xi_k, FP_{\text{lf}}(\Xi_k))$$

we refer to Theorem 1 in [14].

The computational performance of this formulation will be improved compared to $FP$, if $FP_{\text{lf}}$ is further simplified. Choosing the lifting functions in a way such that $s$ is not sensitive to perturbations in variables $\mu$, a more conservative approximation set $\mathcal{W}$ can be employed without significantly decreasing the diagnosis quality. One can, for example, completely omit the constraints $Q_2$, create a number of lifting subproblems for each time point $k$ individually, or approximate the feasible set with an axis-aligned box [14].

This means, the computational effort for outer-approximating $FP_{\text{lf}}$ can be considered as negligible compared to the projection subproblem. Thus, the overall improvement of computational efficiency can be assessed in terms of how much easier the problem $FP_{\text{Proj}}$ is to solve compared to $FP$. In practice it depends on the structure of the constraints, and in general we can expect a bigger reduction if the lifted parts contain high order monomials.

The rest of this section presents methods of determining the lifting functions $l$. With these choices we aim at satisfying the above conditions without greatly decreasing the overall quality of the estimates compared to the original formulation.

**Causal reasoning**

To determine which parts of the dynamics constraints are deemed important for fault diagnosis we analyze the structure of $FP$. To do so, one can employ the concept of causal reasoning, see i.a. [14] and references therein. This concept provides a measure on how strongly two variables are connected. Since fault diagnosis corresponds in our setup to the estimation of the fault switch $s$, it is sufficient to study the connection between the states and $s$.

**Definition 3 (Causal order measure):** A variable $\xi_i(k)$ is said to be of first causal order w.r.t. the fault switch $s$, if there exists an equality in $FP$ that contains $\xi_i(k)$ and $s$. A variable $\xi_j(k)$ is said to be of $n$th causal order, if there exists an equality in $FP$ that contains $\xi_j(k)$ and a state of $(n-1)$th causal order.

The strategy proposed in [14] suggests to pick a threshold value $\beta \in \mathbb{N}_+$ and lift every variable of causal order that is higher than $\beta$. This strategy works well for loosely connected systems, where the causal order can be rather high. However, for a system with a low maximal causal order $n$ one can only generate $n-1$ different relaxed models. This limits the ability to find the best trade-off between quality of the resulting estimates and the computational speed improvement. Therefore, we propose next additional criteria and a way to generate a total order of monomials, appearing in the formulation.

**Multiple objective problem reduction**

To enhance the decision flexibility for deciding which parts of the problem to be substituted we introduce two additional measures. Before explaining the proposed measures, we present a generalized structure of a monomial $m$:

$$m = c \cdot \xi(k)^\nu = c \cdot \prod_{i=1}^{n_c+d_c} \xi_i(k)^{\nu_i},$$

where $\xi$ is the vector of variables in the monomial $m$, the vector $\nu \in \mathbb{N}^{n_c+d_c}$ contains the powers of each variable in $\xi$ in $m$, and $c$ is the coefficient of $m$. For the manipulations regarding the variables in the monomials in Definition 4 we consider only those that are uncertain.

**Definition 4 (Inverse uncertainty measure):** The measure $\omega \in \mathbb{R}_+$ of a monomial $m$ is defined as

$$\omega(m) = (c \cdot (\tilde{\xi}(k) - \xi(k))^\nu)^{-1},$$

where $\xi(k)$ and $\tilde{\xi}(k)$ correspond to elementwise smallest and largest admissible values of $\xi(k)$. ■
Although two monomials might have the same average value, it would be beneficial to lift the one with the smaller range. This is true because the maximal distance between the true value of the bilinear product and its linear relaxation grows proportionally to the uncertainty size of the involved variables.

To introduce the next measure we define first the notion of monomial division. We say that monomial $m_1 = \psi^{N_1}$ divides monomial $m_2 = \psi^{N_2}$, or

$$m_1|m_2, \quad \text{if} \quad N_1 \leq N_2 \quad \text{for each component.}$$

In other words, every variable of $m_1$ is present in $m_2$ with greater or equal power.

**Definition 5 (Repetition measure):** Measure $\rho \in \mathbb{N}_+$ of a monomial $m$ is equal to the amount of monomials in $FP$ that are divided by $m$.

Lifting a monomial, repeated multiple times within the formulation, will introduce a more significant reduction of the constraint complexity compared to a similar monomial, that only appears in the formulation once.

Before using the measures, we define the set $D$ of all of the monomials $m$ that divide at least one monomial in $FP$. To each $m$ we assign a tuple of values, corresponding to each of the discussed measures. The tuple consists of orderings for each measure.

$$ord(m) = (ord_1(m), \ldots, ord_\delta(m)),$$

where $ord_i$ is the value of the corresponding measure of the monomial (we use $i = 3$ for the measures presented).

To be able to directly compare two monomials using the tuples $ord(\cdot)$ we introduce a scalarization function $scf(\cdot)$. It generates a total order within the set of monomials, i.e. for any two monomials $m_i$, $m_j$ at least one of the following two relations should hold:

$$scf(ord(m_i)) \geq scf(ord(m_j)),$$

$$scf(ord(m_i)) \leq scf(ord(m_j)).$$

We consider an additive scalarization function of the form

$$scf(ord(m)) = \sum_{i=1}^{\delta} \frac{ord_i(m)}{\beta_i},$$

with $\beta_i$ denoting scaling coefficients that normalize the range of values of the corresponding ordering.

**Remark 2:** Each choice for $scf(\cdot)$ produces a different output and thus could lead to a different decision of which variables to be lifted. We refer to [2] for other examples of scalarization functions.

Once the scalarization function orders the monomials, we determine a threshold value $\gamma \in \mathbb{R}$ and lift all the monomial that are with greater scalarization value. Since we assume no dynamics of the lifted variables, we additionally decrease the problem formulation by aggregating the monomials of the resulted formulation that only consist of the lifted variables.

Algorithm 1 formalizes the proposed reduction technique.

### Algorithm 1 Lifting through multi-objective ordering

**Require:** feasibility problem $FP_\xi$; set $D$ of monomials $m$ dividing each monomial in $FP_\xi$; ordering tuples $ord(m)$ for each $m \in D$; scalarization function $scf(\cdot)$; threshold value $\gamma$

- set $Q_\xi = Q_\emptyset$
- order $D$ using $scf : scf(ord(m_i)) \geq scf(ord(m_{i+1}))$
- set $i = 1$

while $scf(ord(m_i)) \geq \gamma$ do

- set $l_i(\xi) = m_i$
- replace $m_i$ with $\mu_i$ in $Q_\xi$
- if $m_i|m_j$ for $j > i$ then

- replace the $m_i$ part in $m_j$ with $\mu_i$

end if

- $i +=$

end while

**Discussion:** The proposed reduction technique includes multiple parameters, that can affect its performance. The threshold value $\gamma$ determines the amount of lifted components of the formulation $FP$, and hence the amount of severe interconnections and considered dynamics constrains. Decreasing this value will reduce the computational complexity of $FP_{proj}$, however, can lead to more conservative approximations.

One of the possible ways to overcome this reduced quality of the diagnosis is by increasing the length of the diagnosis time window $K$. Considering a longer measurement window with a simpler model can still result in a less computationally demanding problem compared to the initial formulation.

Another degree of freedom, as stated in Remark 2, is the choice of a scalarization function. Considering, for example, a multiplicative scalarization function can significantly change the order of the monomials and thus lead to a different reduced model. This choice can be made based on the model structure as well as the chosen objective criteria.

In the next section we demonstrate the presented complexity reduction method to diagnose a fault in a heating, ventilation and air conditioning system of a building.

### IV. Example

We illustrate the presented reduction method considering a heating, ventilation and air conditioning (HVAC) system, depicted in Fig. 1. The presented example is based on [1].

The system consists of seven rooms with the same area, with the heater placed in room 2. The inner walls, separating rooms $i$ and $j$, are transferring the heat with the speed $q(T_j - T_i)$, where $q$ is the heat transfer coefficient, and $T_i$, $T_j$ denote temperatures of the corresponding rooms. The outer walls, separating the rooms from the outside, are similarly transferring the heat from the rooms with the speed...
Fig. 1: Multi-room setup. Room 2 contains the heater, and the fault (damaged wall insulation) occurs in room 5.

$q_a(T_i - T_a)$, where $T_a$ is the ambient outdoors temperature, and $q_a$ is the heat transfer coefficient of the outer wall. The heater in room 2 analogously warms it with the speed $q_H(T_H - T_a)$.

**Fault scenario:** We consider a possible fault scenario, when an insulation of the outer wall in room 5 degrades, increasing the heat transfer coefficient between the room and the outside by $q_F$. The discrete-time model of the system is

\[
T_i(k+1) = T_i(k) + \Delta t \cdot \Gamma_i(k), \quad i \in \{1, 3, 4, 6, 7\}
\]

\[
T_2(k+1) = T_2(k) + \Delta t (\Gamma_2(k) + q_H(T_H - T_2(k)))
\]

\[
T_3(k+1) = T_3(k) + \Delta t (\Gamma_3(k) - s \ w_i q_F (T_5(k) - T_a))
\]

where $s$ denotes whether the fault has occurred, $N_i$ is a set of neighboring rooms, and $w_i$ is the amount of outer walls of the room $i$. The corresponding values are listed in Table I.

**TABLE I: Room layout parameters**

<table>
<thead>
<tr>
<th>Room number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbors ($N_i$)</td>
<td>{2, 5}</td>
<td>{1, 3, 6}</td>
<td>{2, 7}</td>
<td>{5}</td>
<td>{1, 4, 6}</td>
<td>{2, 5, 7}</td>
<td>{3, 6}</td>
</tr>
<tr>
<td>Outer walls ($w_i$)</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Problem reduction:** For each variable of model (1) we determine the causal order, uncertainty bounds and the amount of times it is repeated in the formulation. Parameter bounds are chosen with $\pm 2.5$ percent relative uncertainty around the nominal values, as listed in Table II. The measurement data is simulated for the fault scenario with nominal values and added absolute uncertainty of $\pm 0.2$ degrees (see Figure 2).

**TABLE II: Nominal parameter values**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$q_h$</th>
<th>$q_a$</th>
<th>$q_H$</th>
<th>$q_F$</th>
<th>$T_a$</th>
<th>$T_H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal value</td>
<td>0.091</td>
<td>0.042</td>
<td>0.225</td>
<td>0.063</td>
<td>15.15</td>
<td>30</td>
</tr>
</tbody>
</table>

These values are employed to determine the uncertainty measure of each monomial, and the proposed additive scalarization function is applied. The threshold $\gamma$ is chosen so that the 12 monomials with the lowest objective value are lifted:

$q_a T_a$, $q_F$, $q_i T_i$ ($i \in \{2, 3, 7\}$), $q_a T_i$ ($i \in \{1, \ldots, 7\}$).

Following Algorithm 1 we introduce new lifting variables that replace these monomials within model (1). This makes the states $T_5(k+1)$ and $T_7(k+1)$ decoupled from the rest of the system, and hence they can be placed in the lifting subproblem $FP_{proj}$ can then be presented as follows:

$$T_2(k+1) = T_2(k) + \Delta t (q_i(T_i(k) - 2T_1(k)) + \mu_1(k))$$

$$T_3(k+1) = T_3(k) + \Delta t (q_i(T_i(k) + T_6(k)) + q_H(T_H - T_2(k)) + \mu_2(k))$$

$$T_5(k+1) = T_5(k) + \Delta t (q_i(T_i(k) + T_3(k) + T_6(k) - 3T_5(k)) - s q_F (T_5(k) - T_a)) + \mu_4(k))$$

$$T_6(k+1) = T_6(k) + \Delta t (q_F(T_5(k) - 3T_6(k)) + \mu_5(k))$$

with $\mu_i$ replacing the sums of corresponding lifting monomials, and $q_F$ is kept for ease of notation, since it does not appear in a lifting monomial on its own.

**Simulation results:** Using simulated measurement data, starting with initial conditions $T_i(0) = 19 \pm 0.2$ degrees, we determine the minimal length of the diagnosis time window that allows to diagnose the presence of the fault. We employ the sampling time of $\Delta t = 0.25$ h, and determine the bounds of the lifted variables $\mu_i$ for each time step using interval arithmetic, as discussed in Section III. We compare the size and computation time of both the full and the structurally relaxed models and present the summarized results in Table III.

**Discussion:** As one can observe in Figure 2, the parametric uncertainty is large enough to prevent diagnosis after the new steady state is reached. This emphasises the need for fast yet precise diagnosis in the transient phase.

For given simulated data both full and reduced models were able to determine the presence of the fault after 13 time steps (3.25 hours). The reduced formulation was completed 55 percent faster than the original problem despite having substantially more variables and inequality constraints.

This illustrates the main advantage of the approach, as the obtained relaxed model discards the interconnections of system states that would not play a major role in our ability to diagnose the fault. If the complexity of the initial
model were higher (e.g. with polynomial constraints of higher order), the reduction could even decrease the size of the model (cf. e.g. [13]), since the transformation into a quadratic problem would introduce a possibly large number of additional variables.

V. CONCLUSIONS

In this contribution, we propose a multi-objective complexity reduction approach for polynomial systems subject to abrupt faults. It builds upon the causal order reduction technique presented in [14]. We employ a set-based framework to detect the presence of a fault while allowing bounded uncertainties on process parameters and noise. The proposed approach combines multiple criteria to order the monomials according to their importance for the diagnosis of a fault. The unimportant constraint components are then substituted, preserving the information on their bounds.

Although, as seen from the presented example, the size of the problem is not guaranteed to decrease, the solution time is reduced significantly. In spite of the heuristic nature of the criteria for monomial ordering, the method succeeds in providing the diagnosis result with a minimal impact on the length of the required time horizon. Due to the conservatism of the relaxation procedure the diagnosis result retains guarantees, meaning the method does not discard the actual fault scenario.

The proposed approach has several opportunities for extensions. One promising direction is incorporating more measures such that different aspects of complexity are addressed. A different direction is developing a procedure to evaluate the parameter values used in the algorithm. Another, more challenging extension, could be in considering a more elaborate function bases instead of the monomial decomposition presented in this work.

REFERENCES