

Comparison of Advanced Set-Based Fault Detection Methods with Classical Data-Driven and Observer-Based Methods for Uncertain Nonlinear Processes

Bowen Mu^a, Xuejiao Yang^a, Joseph K. Scott^{a,*}

^a*School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, 311 Ferst Dr., Atlanta, 30318, GA, USA*

Abstract

Automated fault detection (FD) methods are essential for safe and profitable operation of complex engineered systems. Both data-driven and model-based methods have been extensively studied, and some are widely used in practice. However, distinguishing faults from acceptable process variations remains a critical challenge, making both false alarms and missed faults commonplace. In principle, set-based FD methods can rigorously address this challenge. However, existing methods are often much too conservative, particularly for nonlinear systems. Moreover, few if any published comparisons clearly demonstrate the supposed advantages of set-based methods relative to conventional methods. This paper first presents a new set-based FD method based on discrete-time differential inequalities and demonstrates increased fault sensitivity through several case studies. Next, a detailed comparison of set-based methods with representative data-driven and model-based approaches is presented. The results verify some key advantages of the set-based approaches, but also highlight key challenges for future work.

Keywords: Fault detection, Set-based methods, Uncertain systems, Nonlinear systems

1. Introduction

Due to the level of complexity, integration, and automation in modern engineered systems, equipment malfunctions and other abnormal events are frequent and unavoidable. These events, termed *faults*, often have serious economic, safety, and environmental consequences if not detected quickly (Venkatasubramanian et al., 2003b). At the same time, false alarms (i.e., fault declarations during normal operation) caused by benign disturbances can lead to shut-downs or other operational changes that also do significant economic harm. Thus, automated methods for detecting faults quickly and accurately are essential. This paper introduces a new set-based fault detection algorithm based on discrete-time differential inequalities and presents a detailed comparison of set-based fault detection methods with conventional data-driven and model-based approaches.

To date, the theory and practice of fault detection (FD) has been dominated by data-driven approaches (Chiang et al., 2000; Venkatasubramanian et al., 2003b). In these methods, historical data is analyzed to identify important statistics, and the variability of these statistics under normal operating conditions is quantified in terms of thresholds. Online, new measurements are compared with the historical data and a fault is declared if the current statistics violate the computed thresholds. In ideal cases (e.g., stationary Gaussian data), well-established statistical methods such as principal component analysis (PCA) can be used to identify appropriate statistics and set thresholds to achieve any desired rate of false alarms (Chiang et al., 2000).

These methods are simple, scalable, and widely used in industry (Joe Qin, 2003). However, for general non-Gaussian data, these simple methods can result in inappropriate statistics or thresholds, often leading to high false alarm rates. Several advanced data-driven methods based on independent component analysis (ICA), dynamic PCA, kernel PCA, and other machine learning techniques have been developed to address this, but these have considerably higher costs and are not well established in practice (Lee et al., 2006; Ku et al., 1995; Choi et al., 2005; Lee et al., 2007). Another significant disadvantage of all data-driven methods is that they require historical data that is appropriate for the current operating point. If the current operating point is different from that of the historical data, either intentionally or due to a persistent disturbance or transient, the process statistics can deviate significantly from historical values, leading to persistent false alarms that render the method unusable (see experiments in §5).

To address these limitations, another class of FD methods makes use of process models in place of historical data (Venkatasubramanian et al., 2003b; Isermann, 2005; Patton and Chen, 1997). The most standard approach is to use an observer (i.e., state estimator) to predict the most likely output values at the next sampling time. A fault is then declared if the measured outputs deviate from the predictions by more than a prescribed threshold (Patton and Chen, 1997). This eliminates the need for historical data at the current operating point and naturally handles non-steady operations. In principle, the model also provides a means to completely characterize the output statistics under fault-free conditions, which is crucial for setting thresholds that accurately distinguish faults from disturbances. However, this requires both an accurate model and accurate knowl-

*Corresponding author

Email address: joseph.scott@chbe.gatech.edu (Joseph K. Scott)

edge of the disturbance probability distributions, which is often impractical. Moreover, even when these distributions are available, propagating them through a nonlinear model to obtain output distributions is extremely difficult. Thus, significant approximations are needed, such as the use of linearization and Gaussian distributions in methods based on extended Kalman filters (Patton and Chen, 1997). In general, this can lead to inaccurate output statistics and, as a result, fault insensitivity or excessive false alarms (see experiments in §5).

A third class of FD methods called set-based methods attempts to address the shortcomings of observer-based methods by modeling all disturbances and measurement noises in terms of deterministic bounds rather than probability distributions. Specifically, these inputs are assumed to be bounded within known compact sets, but nothing is assumed about their distributions. Set-based computations are then used to rigorously test if a new measured output is consistent with the process model given these bounds, and a fault is declared if not. This approach is attractive because obtaining bounds on disturbances and measurement noises is often easier than obtaining accurate probability distributions. Moreover, model uncertainty can be easily incorporated using bounded time-invariant parameters, which lessens the need for a highly accurate model. Finally, these methods completely eliminate false alarms provided that the input bounds are valid. However, accurate set-based computations are required to achieve high fault sensitivity, which remains a major challenge.

Many set-based FD methods are available for linear systems using computations with intervals (Seron and De Doná, 2009; Efimov et al., 2013), polytopes (Blesa et al., 2012), ellipsoids (Reppa and Tzes, 2011), zonotopes (Scott et al., 2013; Ingimundarson et al., 2009; Tabatabaeipour et al., 2012), and constrained zonotopes (Scott et al., 2016). However, testing the consistency of a measured output with a nonlinear model is significantly more difficult. One approach is to use set-based parameter estimation, wherein measurements are used to compute an enclosure of the set of consistent model parameters. A fault is then declared when this enclosure has no overlap with a known set of possible parameter values. In (Jaubertie et al., 2013), this is done using a differential algebraic approach and interval-based set inversion techniques. However, the computational cost scales exponentially with the number of uncertain parameters. This method is extended to systems with probabilistic noises using a Bayesian framework in (Fernández-Canti et al., 2013), but this does not provide rigorous bounds.

Another approach to set-based FD is to apply set-based state estimation. At each sampling time, a set-based state estimator provides a guaranteed enclosure of the set of states consistent with the model, the bounded uncertainties, and all past measurements. This is then used to compute an enclosure of the possible model outputs, and a fault is declared if the measured output is outside of this set. The key challenge is to compute sufficiently accurate enclosures fast enough for online fault detection. A method for continuous-time systems based on upper and lower Luenberger observers and cooperativity theory is proposed in (Raïssi et al., 2010). In (Combastel, 2016), an Extended Zonotopic and Gaussian Kalman Filter is proposed for

discrete-time systems with uncertainties composed of bounded and unbounded parts. However, both methods rely on conservative linearizations of the dynamics over the entire state domain, which can lead to weak enclosures compared to adaptive linearizations such as those in (Alamo et al., 2005; Combastel, 2005). The FD method in (Rostampour et al., 2017) also uses a set-based state estimator, but forgoes rigorous enclosures in favor of smaller sets based on a prescribed false alarm rate. Thus, this method is not guaranteed to avoid false alarms. Moreover, it requires the solution of nonlinear chance constrained optimization problems at each sampling time, which is prohibitive. To reduce conservatism and increase efficiency, some approaches use approximate models with simpler structure. In (Wang and Puig, 2016), nonlinear models are linearized before constructing the observer, as in the extended Kalman filter. Similarly, (Chai et al., 2013) approximates nonlinear input-output models using a Takagi-Sugeno fuzzy neural network that is linear in the uncertain parameters. Ellipsoidal (Chai et al., 2013) and zonotopic (Wang and Puig, 2016) enclosures are then computed for the approximate models and used for fault detection. However, these enclosures are not necessarily valid for the original system. Finally, (Tulsyan and Barton, 2016) proposes a set-based FD method for continuous-time systems using advanced reachable set bounding techniques based on differential inequalities. However, measurements are not used to refine the predicted enclosures as in a true set-based state estimator, which is a serious limitation.

Despite this prior work, performing set-based computations with sufficient accuracy for effective fault detection remains a major challenge for nonlinear systems. Moreover, although the potential advantages of set-based FD methods have been articulated in many prior studies, to the best of our knowledge, no detailed studies comparing set-based FD methods to more conventional data-driven and observer-based methods are available in the literature. In this context, this article makes two main contributions. First, we present a new set-based FD algorithm based on the set-based state estimator recently developed in (Yang and Scott, 2018a). This estimator computes interval enclosures using the theory of discrete-time differential inequalities (DTDI) and has been shown to produce significantly more accurate enclosures than other state-of-the-art set-based state estimators for nonlinear test cases in (Yang and Scott, 2018a). However, this method has not previously been applied for FD. To this end, we develop a new FD algorithm using DTDI and demonstrate through case studies that it offers significantly improved fault sensitivity.

Second, we present a detailed comparison of set-based FD methods with more conventional data-driven and observer-based methods using three case studies. Specifically, we compare against the standard PCA method described in (Joe Qin, 2003) and the extended Kalman filter (EKF) method from (Fathi et al., 1993). These methods were chosen because they are representative of the state-of-practice in data-driven and observer-based FD, respectively. We acknowledge that many alternative methods exist and may perform better in specific scenarios. Yet, none are as well established or widely used, and so it seems most informative to first understand how set-based meth-

ods compare to these classical benchmarks.

Comparing data-driven, observer-based, and set-based FD methods is challenging and somewhat ill-posed because they are based on fundamentally different assumptions about the process noises and require different information about the process that may not be accurately known in practice. Thus, any comparison necessarily involves applying the methods in cases that violate some of their assumptions. Yet, this is exactly the case when applying these methods to real systems, so it is important to study how they perform in such circumstances. To this end, we develop a framework for comparing data-driven, observer-based, and set-based FD methods using simulated systems with various noise distributions, various types of model uncertainty, various fault-free scenarios representing different operating conditions, and various types of faults. In every case, we make the most sensible approximation of how each method would be applied in practice without knowledge of the actual process and noise distributions used in the simulation.

The remainder of this paper is organized as follows. Section 2 gives a formal problem statement. Section 3 describes the representative data-driven and observer-based FD methods selected for our comparisons. Section 4 introduces the set-based FD paradigm, describes existing methods we compare against, and presents our new set-based FD algorithm. In Sections 5, 6, and 7 detailed comparisons of all FD methods are provided using three case studies. Finally, Section 8 provides further discussion and concluding remarks.

2. Problem Statement

This paper considers FD algorithms for uncertain nonlinear discrete-time systems of the form

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k, \mathbf{w}_k), \quad (1)$$

$$\mathbf{y}_k = \mathbf{g}(k, \mathbf{x}_k, \mathbf{v}_k). \quad (2)$$

Above, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the output, $\mathbf{w}_k \in \mathbb{R}^{n_w}$ is the disturbance, $\mathbf{v}_k \in \mathbb{R}^{n_v}$ is the measurement noise, and $k \in \mathbb{K} \equiv \{0, \dots, K\}$. The functions $\mathbf{f}$ and $\mathbf{g}$ have the form $\mathbf{f} : \mathbb{K} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_x}$ and $\mathbf{g} : \mathbb{K} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_y}$.

Different FD methods make different assumptions about the nature of the noises $\mathbf{w}$ and $\mathbf{v}$ and the required properties of the functions $\mathbf{f}$ and $\mathbf{g}$. The assumptions for the conventional FD methods are discussed in §3 while those for the set-based methods are given in §4. In our numerical comparisons, different kinds of noises are used in different test cases. The system is said to be fault-free if equations (1) and (2) hold and the noises $\mathbf{w}_k$ and $\mathbf{v}_k$ obey their specified bounds or probability distributions over the entire time horizon. Faults are modelled by discrete changes in either the governing equations or the nature of the noises at some time k . The objective of fault detection methods is to detect this situation as quickly as possible. If a fault is not detected by a particular method, we call it a *missed fault*. In contrast, if the simulated system is fault-free but a fault is declared by a method, we call it a *false alarm*. These two indices plus the detection speed are used for comparing methods in the numerical examples.

3. Conventional Fault Detection Methods

This section describes the basic principles of the conventional FD methods compared in §5–7.

3.1. Principal Component Analysis (PCA) Method

PCA is widely used for fault detection in industrial processes. The standard PCA-based FD method is described in (Chiang et al., 2000) and is chosen here as a representative of conventional data-driven methods. This approach does not assume a system model or any information about the noises $\mathbf{w}_k$ and $\mathbf{v}_k$. Instead, it assumes that we have a historical dataset $\mathbf{Y} \in \mathbb{R}^{m \times n_y}$ composed of m measured output vectors $\mathbf{y}_k \in \mathbb{R}^{n_y}$ collected during some period when the process was operating at steady-state without faults. Moreover, these outputs are assumed to obey a multivariate Gaussian distribution. This will hold when (1)–(2) are linear and $\mathbf{w}_k$ and $\mathbf{v}_k$ are serially uncorrelated Gaussian noises, but not necessarily otherwise.

The offline portion of the method involves first normalizing $\mathbf{Y}$ so that each column has zero mean and unit standard deviation, and then computing the eigendecomposition of the covariance matrix $\mathbf{S} \equiv \frac{1}{m-1} \mathbf{Y}^T \mathbf{Y}$. The eigenvectors are sorted by their eigenvalues, which indicate how much of the variance in the data is explained by each eigenvector. The eigenvectors corresponding to large eigenvalues (enough to describe a prescribed percentage of the total variance) are called loading vectors and define the *principal component subspace*, while the rest constitute the *residual subspace*.

When detecting faults online, each new measurement $\mathbf{y}_k$ is first normalized in the same way as the columns of $\mathbf{Y}$, and then two statistics are computed that are used as residuals for fault detection. For the first, the component of $\mathbf{y}_k$ along each loading vector is computed. The resulting vector is called the *score vector* and, once each component is normalized by the corresponding eigenvalue, should obey a multivariate Gaussian distribution with zero mean and identity covariance matrix in the fault-free case. The residual for fault detection is defined as the squared two-norm of this vector, which represents the deviation from the mean. A threshold containing this statistic with a specified probability is calculated using the fact that the residual follows the T^2 distribution. Online, a fault is declared whenever this residual crosses the threshold, and the specified probability determines the probability of a false alarm (Joe Qin, 2003). The second residual used in the PCA method is the squared prediction error (SPE), which is the square of the Euclidean distance from $\mathbf{y}_k$ to the principal component subspace. Since the SPE obeys the Q statistic, a threshold can also be generated by specifying a desired probability of false alarms. Compared with T^2 , SPE is more robust to the inaccuracy of small eigenvalues.

Since the comparisons in this paper are done by simulation, historical datasets are synthesized using 60,000 output vectors from the simulated fault-free system starting at the nominal steady state. The principal components are chosen by retaining the k eigenvectors with largest eigenvalues such that the summation of their eigenvalues exceeds 90% of the total eigenvalue summation. For all of the test cases in §5–7 this heuristic retained all eigenvectors (i.e., no dimension reduction was neces-

sary), so fault detection was done with the T^2 statistic only. A 95% confidence level was selected for the T^2 threshold. Thus, we expect a 5% false alarm rate for systems with normally distributed outputs. However, the systems we consider are nonlinear and sometimes have non-Gaussian distributions, so this confidence level is only approximate.

3.2. Extended Kalman Filter (EKF) Method

Extended Kalman Filters are used extensively in industry for state estimation of nonlinear systems. Accordingly, we chose the EKF-based FD method in (Fathi et al., 1993) as a representative of conventional observer-based FD methods. In this method, a standard EKF is applied to (1)–(2) assuming that $\mathbf{f}$ and $\mathbf{g}$ are continuously differentiable and $\mathbf{w}_k$ and $\mathbf{v}_k$ are serially uncorrelated Gaussian noises. At each sampling time, the EKF furnishes an innovation vector ϵ defined as the difference between the predicted outputs and the measured values. A fault is declared if any component of ϵ has a significant bias away from 0. For each dimension i , this is determined using a sequential probability ratio test (SPRT) that compares the likelihoods of the following hypotheses: (i) ϵ_i is Gaussian with mean 0 (fault-free), and (ii) ϵ_i is Gaussian with mean a for some specified $a \neq 0$ (faulty). Note that the first hypothesis is true of the fault-free system if $\mathbf{w}_k$ and $\mathbf{v}_k$ are serially uncorrelated zero-mean Gaussian noises and (1)–(2) are linear, but not necessarily otherwise. Moreover, the assumption that a fault merely causes a shift in the mean may not be accurate in practice. Thus, the use of the SPRT for fault detection should be considered heuristic.

The test statistic of the SPRT is the logarithm of the joint likelihood ratio (LLR) function, $l_{i,k}$, which can be generated by a simple recursion under the assumption of Gaussian innovations. In hypothesis testing, $l_{i,k}$ typically has both negative and positive thresholds. Hypothesis (i) is deemed correct if $l_{i,k}$ falls below the negative threshold, while Hypothesis (ii) is deemed correct if $l_{i,k}$ exceeds the positive threshold. However, for fault detection it is only of interest to confirm Hypothesis (ii). Therefore, the following modified recursion is used to minimize detection time (Chen and Adams, 1976):

$$l_{i,k} = \max\{l_{i,k-1} + a[\tilde{\epsilon}_{i,k} - 0.5a], 0\}. \quad (3)$$

The innovation $\tilde{\epsilon}_{i,k}$ above is normalized by the standard deviation of ϵ_i (as estimated by the EKF) so that a can be chosen as ± 1 to detect positive and negative biases (Fathi et al., 1993). Given a specified probability of false alarms α and missed faults β , the threshold λ for $l_{i,k}$ is computed by solving

$$\begin{aligned} e^\lambda - \lambda - 1 &= -\left(B + A \frac{e^B - 1}{1 - e^A}\right), \\ A &= \ln(\beta/(1 - \alpha)), \\ B &= \ln((1 - \beta)/\alpha). \end{aligned} \quad (4)$$

In our implementation, fault detection is done using two residuals for each i , one with $a = 1$ and another with $a = -1$. However, in our numerical results, we only plot a single residual for each scenario. In fault-free scenarios, methods are compared according to their false alarm rates, so we select the residual with most false alarms. In faulty scenarios, we select the

residual that indicates the fault earliest. To compare with PCA, the false alarm and missed fault probabilities are selected to be 95%. However, since the innovations are not guaranteed to be Gaussian in our examples, this threshold is only approximate.

4. Set-based Fault Detection

In this section, we present a generic algorithm for set-based FD that makes use of a set-based state estimator. We then describe five distinct methods that arise from applying this algorithm with different state estimators.

In set-based methods, the initial conditions, disturbances, and measurement noises are assumed to be bounded by known compact sets:

$$(\mathbf{x}_0, \mathbf{w}_k, \mathbf{v}_k) \in C_0 \times W \times V, \quad \forall k \in \mathbb{K}. \quad (5)$$

Let $\mathbf{y}_{0:K} = (\mathbf{y}_0, \dots, \mathbf{y}_K)$ denote an observed output sequence. The goal of set-based state estimation is to characterize the sets $X_{k|k}(\mathbf{y}_{0:k})$ and $X_{k+1|k}(\mathbf{y}_{0:k})$ defined as the sets of all possible states at step k and $k+1$, respectively, that are consistent with the model (1)–(2), the bounds (5), and the measurements up to step k . Unfortunately, these sets can be arbitrarily complex and cannot be computed directly. Instead, set-based methods aim to compute guaranteed enclosures of these sets, denoted by $\hat{X}_{k|k}(\mathbf{y}_{0:k})$ and $\hat{X}_{k+1|k}(\mathbf{y}_{0:k})$. Toward this end, a key observation is that $X_{k|k}(\mathbf{y}_{0:k})$ and $X_{k+1|k}(\mathbf{y}_{0:k})$ satisfy a recursive relationship. Specifically, for any $\mathbf{y} \in \mathbb{R}^{n_y}$, define the measurement set

$$X_k^m(\mathbf{y}) \equiv \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{y} = \mathbf{g}(k, \mathbf{x}, \mathbf{v}), \mathbf{v} \in V\}. \quad (6)$$

Then, the following recursion holds (Le et al., 2013):

$$X_{0|0} = C_0, \quad (7)$$

$$X_{k|k}(\mathbf{y}_{0:k}) = X_{k|k-1}(\mathbf{y}_{0:k}) \cap X_k^m(\mathbf{y}_k), \quad (8)$$

$$X_{k+1|k}(\mathbf{y}_{0:k}) = \{\mathbf{f}(k, \mathbf{x}, \mathbf{w}) : (\mathbf{x}, \mathbf{w}) \in X_{k|k}(\mathbf{y}_{0:k}) \times W\}. \quad (9)$$

Although (7)–(9) cannot be implemented algorithmically due to the complexity of the sets $X_{k|k}(\mathbf{y}_{0:k})$ and $X_{k+1|k}(\mathbf{y}_{0:k})$, these relations suggest a recursive procedure for computing the enclosures $\hat{X}_{k|k}(\mathbf{y}_{0:k})$ and $\hat{X}_{k+1|k}(\mathbf{y}_{0:k})$. To write this procedure in a sufficiently general form for our purposes, suppose that an *a priori* enclosure satisfying the following assumption is known.

Assumption 1. A set $G \subset \mathbb{R}^{n_x}$ is known such that $\mathbf{x}_k \in G$ for all $k \in \mathbb{K}$ provided that (5) holds.

This assumption is not restrictive since we may always choose $G = \mathbb{R}^{n_x}$. However, the solutions of many systems are known to be positive, satisfy conservation laws, or lie in more general invariant sets (Scott and Barton, 2013; Shen and Scott, 2017). This information can be used to define nontrivial choices of G , which can then be exploited by some state estimation methods to obtain tighter enclosures. Under Assumption 1, enclosures $\hat{X}_{k|k}(\mathbf{y}_{0:k})$ and $\hat{X}_{k+1|k}(\mathbf{y}_{0:k})$ can be computed by any recursive procedure that satisfies the following inclusions:

$$\hat{X}_{0|0} \supset C_0, \quad (10)$$

$$\hat{X}_{k|k}(\mathbf{y}_{0:k}) \supset \hat{X}_{k|k-1}(\mathbf{y}_{0:k-1}) \cap X_k^m(\mathbf{y}_k) \cap G, \quad (11)$$

$$\hat{X}_{k+1|k}(\mathbf{y}_{0:k}) \supset \{\mathbf{f}(k, \mathbf{x}, \mathbf{w}) : (\mathbf{x}, \mathbf{w}) \in (\hat{X}_{k|k}(\mathbf{y}_{0:k}) \cap G) \times W\}. \quad (12)$$

The inclusions (11) and (12) are called the *correction* and *prediction* steps, respectively. Specific set-based state estimators differ in the details of how the sets $\hat{X}_{k|k}(\mathbf{y}_{0:k})$ and $\hat{X}_{k+1|k}(\mathbf{y}_{0:k})$ are represented computationally and how each set is computed from the last so as to satisfy (10)–(12). This is discussed further below and illustrated by the example in §4.1.

The set-based FD methods we consider attempt to detect faults by checking if the real output is consistent with the fault-free model in each time step. Define the set of possible output vectors at time k given $\mathbf{y}_{0:k-1}$ by

$$Y_{k|k-1}(\mathbf{y}_{0:k-1}) \equiv \{\mathbf{g}(k, \mathbf{x}, \mathbf{v}) : (\mathbf{x}, \mathbf{v}) \in X_{k|k-1}(\mathbf{y}_{0:k-1}) \times V\}. \quad (13)$$

A measured output sequence $\mathbf{y}_{0:k}$ is inconsistent with the fault-free model if and only if $\mathbf{y}_k \notin Y_{k|k-1}(\mathbf{y}_{0:k-1})$. If this happens, then a fault must have occurred. Since $Y_{k|k-1}(\mathbf{y}_{0:k-1})$ is also generally impossible to compute, set-based FD methods instead compute an enclosure $\hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1})$ satisfying

$$\hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1}) \supset \{\mathbf{g}(k, \mathbf{x}, \mathbf{v}) : (\mathbf{x}, \mathbf{v}) \in (\hat{X}_{k|k-1}(\mathbf{y}_{0:k-1}) \cap G) \times V\}. \quad (14)$$

If $\mathbf{y}_k \notin \hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1})$, then $\mathbf{y}_{0:k}$ is inconsistent with the fault-free model and a fault must have occurred. However, this is now a one-sided guarantee and nothing can be concluded if $\mathbf{y}_k \in \hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1})$. A general FD algorithm based on this concept is given in Algorithm 1. This concept is also illustrated in Figure 1, which shows an example of a complex set $Y_{k|k-1}(\mathbf{y}_{0:k-1})$ (blue) along with two simple enclosures $\hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1})$ representing the results of two different set-based FD methods (orange interval and green polytope). If the observed measurement $\mathbf{y}_k$ is one of the triangular markers, then it is consistent with the fault-free model. Since it also lies inside of the enclosures, neither method will declare a fault. Thus, set-based FD methods never give false alarms. If $\mathbf{y}_k$ is one of the square markers, then a fault has occurred but neither method can detect it. Finally, if $\mathbf{y}_k$ is one of the circles, then a fault has occurred, the interval method can detect it, but the polytope method cannot. Clearly, fault sensitivity depends critically on how tight the enclosure $\hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1})$ is to the true set $Y_{k|k-1}(\mathbf{y}_{0:k-1})$.

Algorithm 1 Set-based FD using a set-based state estimator

```

1: function SETBASEDFD( $C_0, W, V, G$ )
2:    $\hat{X}_{0|-1} \supset C_0$ 
3:   for  $k = 0$  to  $K$  do
4:      $\hat{Y}_{k|k-1} \supset \{\mathbf{g}(k, \mathbf{x}, \mathbf{v}) : (\mathbf{x}, \mathbf{v}) \in (\hat{X}_{k|k-1} \cap G) \times V\}$ 
5:     Measure  $\mathbf{y}_k$ 
6:     if  $\mathbf{y}_k \notin \hat{Y}_{k|k-1}$  then
7:       break (Fault detected)
8:     end if
9:      $\hat{X}_{k|k} \supset \hat{X}_{k|k-1} \cap X_k^m(\mathbf{y}_k) \cap G$ 
10:     $\hat{X}_{k+1|k} \supset \{\mathbf{f}(k, \mathbf{x}, \mathbf{w}) : (\mathbf{x}, \mathbf{w}) \in (\hat{X}_{k|k} \cap G) \times W\}$ 
11:  end for
12: end function
```

We now describe the specific set-based FD methods compared in this paper, all of which use Algorithm 1 with different set-based state estimators satisfying (10)–(12) and (14). The sets C_0 , W , and V are assumed to be intervals throughout.

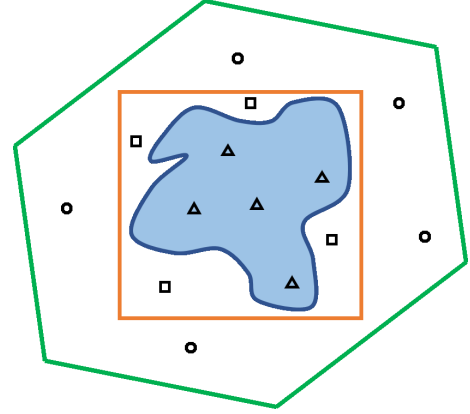


Figure 1: Illustration of the set of all possible outputs $Y_{k|k-1}(\mathbf{y}_{0:k-1})$ (blue) along with two enclosures $\hat{Y}_{k|k-1}(\mathbf{y}_{0:k-1})$ representing the results of two different set-based FD methods (orange interval and green polytope). The markers represent the measured output $\mathbf{y}_k$ in different fault-free (Δ) and faulty ($\square, \circ$) cases.

The first three methods are based on standard set-based state estimators from the literature (Kieffer et al., 1998; Alamo et al., 2005; Combastel, 2005). None of them make use of *a priori* information; i.e., $G = \mathbb{R}^{n_x}$. In the first method, called the *standard interval method*, the sets $\hat{X}_{k|k}$, $\hat{X}_{k|k-1}$, and $\hat{Y}_{k|k-1}$ are intervals, the intersection in (11) is computed exactly, and the images in (12) and (14) are computed by simply evaluating $\mathbf{f}$ and $\mathbf{g}$ in interval arithmetic. This method is simple, but is known to lead to very conservative bounds. Much sharper bounds can often be achieved by representing $\hat{X}_{k|k}$, $\hat{X}_{k|k-1}$, and $\hat{Y}_{k|k-1}$ as zonotopes. Thus, we also compare against two state-of-the-art zonotopic methods (Alamo et al., 2005; Combastel, 2005). Although these methods are well-known, to our knowledge they have not previously been used for set-based fault detection. The functions $\mathbf{f}$ and $\mathbf{g}$ are assumed to be continuously differentiable in (Alamo et al., 2005) and twice continuously differentiable in (Combastel, 2005). In the prediction step, both methods compute $\hat{X}_{k+1|k}$ by linearizing $\mathbf{f}$, using standard zonotope computations for linear mappings, and then adding a rigorous bound on the linearization error. This is done using the mean-value theorem in (Alamo et al., 2005) and via Taylor expansion in (Combastel, 2005). The computation of $\hat{Y}_{k|k-1}$ is done analogously by conservative linearization of $\mathbf{g}$. The condition $\mathbf{y}_k \notin \hat{Y}_{k|k-1}$ can be checked exactly by solving a linear program, or conservatively but more cheaply by checking if $\mathbf{y}_k$ lies outside of the interval hull of $\hat{Y}_{k|k-1}$. We do the latter in all numerical experiments here since the former did not result in improved fault detection. For the correction step (11), the procedure in (Bravo et al., 2006) is used for both methods because it was found to give consistently tighter enclosures. This procedure first encloses $X_k^m(\mathbf{y}_k)$ by an intersection of n_y strips and then conservatively bounds the intersection of $\hat{X}_{k|k-1}(\mathbf{y}_{0:k-1})$ with each strip successively. Even when $X_k^m(\mathbf{y}_k)$ is an interval, its intersection with $\hat{X}_{k|k-1}(\mathbf{y}_{0:k-1})$ is not necessarily a zonotope, so this step can lead to significant overestimation. Applying these prediction and correction procedures recursively produces zonotopes of increasing complexity. Therefore, conservative *order reduction* techniques are

needed to maintain computational efficiency (Yang and Scott, 2018b). In our numerical experiments, we reduced the order of $\hat{X}_{k+1|k}(\mathbf{y}_{0:k})$ to 10 after the prediction step for every k using the reduction method in (Alamo et al., 2005).

Next, we describe two new FD methods based on recent advances in set-based state estimation in (Yang and Scott, 2018a, 2020). Since this requires several definitions and concepts from set-based computing that may be unfamiliar to some readers, we also demonstrate the computations involved for a tutorial example in §4.1. Let $Z = [\mathbf{z}^L, \mathbf{z}^U]$ denote the n -dimensional interval $\{\mathbf{z} \in \mathbb{R}^n : \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U\}$. Denote the space of all such intervals by $\mathbb{I}^n$. For any $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, an interval function $[\mathbf{h}] : \mathbb{I}^n \rightarrow \mathbb{I}^m$ is an *inclusion function* for $\mathbf{h}$ if

$$\mathbf{h}(Z) \equiv \{\mathbf{h}(\mathbf{z}) : \mathbf{z} \in Z\} \subset [\mathbf{h}](Z), \quad \forall Z \in \mathbb{I}^n. \quad (15)$$

We assume that functions $[\mathbf{f}] : \mathbb{K} \times \mathbb{I}^{n_x} \times \mathbb{I}^{n_w} \rightarrow \mathbb{I}^{n_x}$ and $[\mathbf{g}] : \mathbb{K} \times \mathbb{I}^{n_x} \times \mathbb{I}^{n_y} \rightarrow \mathbb{I}^{n_y}$ are available such that $[\mathbf{f}](k, \cdot, \cdot)$ and $[\mathbf{g}](k, \cdot, \cdot)$ are inclusion functions for $\mathbf{f}(k, \cdot, \cdot)$ and $\mathbf{g}(k, \cdot, \cdot)$, respectively, for all $k \in \mathbb{K}$. These can be readily obtained from interval arithmetic (Moore et al., 2009). Next, given any set $A \subset \mathbb{R}^{n_x}$, let the *interval refinement operator* $\mathcal{I}[\cdot, A] : \mathbb{I}^{n_x} \rightarrow \mathbb{I}^{n_x}$ satisfy

$$(Z \cap A) \subset \mathcal{I}[Z, A], \quad \forall Z \in \mathbb{I}^{n_x}. \quad (16)$$

In words, $\mathcal{I}[Z, A]$ attempts to shrink the interval Z by eliminating regions that are outside of A , resulting in a new interval. We assume such an operator is available for the set $A = G \cap X_k^m(\mathbf{y}_k)$, $\forall k \in \mathbb{K}$. Methods for refining a given interval enclosure based on a set of constraints have been widely studied. Methods appropriate for this application are discussed in detail in §IV in (Yang and Scott, 2020). For the examples considered in this paper, both G and $X_k^m(\mathbf{y}_k)$ are expressible in terms of linear constraints, so we define $\mathcal{I}[\cdot, G \cap X_k^m(\mathbf{y}_k)]$ exactly as in (Yang and Scott, 2018a) using Algorithm 1 from (Yang and Scott, 2018c). Finally, for every $i \in \{1, \dots, n_x\}$, define the *face selection operators* $\beta_i^L, \beta_i^U : \mathbb{I}^{n_x} \rightarrow \mathbb{I}^{n_x}$ by

$$\beta_i^L([\mathbf{z}^L, \mathbf{z}^U]) \equiv \{\mathbf{z} \in [\mathbf{z}^L, \mathbf{z}^U] : z_i = z_i^L\}, \quad (17)$$

$$\beta_i^U([\mathbf{z}^L, \mathbf{z}^U]) \equiv \{\mathbf{z} \in [\mathbf{z}^L, \mathbf{z}^U] : z_i = z_i^U\}. \quad (18)$$

The state estimator developed in (Yang and Scott, 2018a) describes all sets as intervals and is defined as follows. Given $\hat{X}_{k|k-1}$ and a new measurement $\mathbf{y}_k$ at time k , the correction step is given by

$$\hat{X}_{k|k} = \mathcal{I}[\hat{X}_{k|k-1}, G \cap X_k^m(\mathbf{y}_k)]. \quad (19)$$

The prediction for $k+1$ is then computed componentwise as follows, where $[f_i^L, f_i^U]$ denotes the i^{th} component of $[\mathbf{f}]$:

$$\hat{x}_{i,k+1|k}^L = f_i^L(k, \mathcal{I}[\beta_i^L(\hat{X}_{k|k}), G \cap X_k^m(\mathbf{y}_k)], W), \quad (20)$$

$$\hat{x}_{i,k+1|k}^U = f_i^U(k, \mathcal{I}[\beta_i^U(\hat{X}_{k|k}), G \cap X_k^m(\mathbf{y}_k)], W).$$

This is the same as the standard interval method except that $f_i^{L/U}$ is evaluated at $\mathcal{I}[\beta_i^{L/U}(\hat{X}_{k|k}), G \cap X_k^m(\mathbf{y}_k)]$ rather than $\hat{X}_{k|k}$. The resulting bounds are therefore significantly tighter, but remain valid by Theorem 3 in (Yang and Scott, 2020) provided

that $\mathcal{I}$ satisfies a Lipschitz condition and $\mathbf{f}$ satisfies a certain monotonicity condition. The Lipschitz requirement is established for our choice of $\mathcal{I}$ in (Yang and Scott, 2018a) and for several other choices in (Yang and Scott, 2020). Moreover, by Corollary 4 in (Yang and Scott, 2020), the monotonicity requirement holds whenever $\mathbf{f}$ takes the Euler-discretized form $\mathbf{f}(k, \mathbf{x}_k, \mathbf{w}_k) = \mathbf{x}_k + h\tilde{\mathbf{f}}(k, \mathbf{x}_k, \mathbf{w}_k)$ with any locally Lipschitz continuous $\tilde{\mathbf{f}}$ and any step size h below a simple upper bound given in (Yang and Scott, 2020). These conditions have been verified in (Yang and Scott, 2020) for all of the examples considered here. The prediction step (20) is motivated by similar methods for continuous-time reachability analysis based on the theory of differential inequalities (DI). Consequently, this method is referred to as discrete-time differential inequalities (DTDI). In the comparisons herein, we further distinguish two specific DTDI methods. The first, termed standard DTDI (sDTDI), uses the trivial *a priori* enclosure $G = \mathbb{R}^{n_x}$, and hence replaces $\mathcal{I}[\cdot, G \cap X_k^m(\mathbf{y}_k)]$ with $\mathcal{I}[\cdot, X_k^m(\mathbf{y}_k)]$ everywhere. This is often much more effective than the standard interval method due to the use of $\beta_i^{L/U}$, but still results in weak bounds in many cases (Yang and Scott, 2020). The second, termed redundancy-based DTDI (rDTDI), results from choosing a nontrivial *a priori* enclosure. Such an enclosure is sometimes known from physical insights. More generally, one can be manufactured by lifting the system into a higher-dimensional state space. This procedure is described in (Shen and Scott, 2017) and is demonstrated by example in §5.7 here. Examples in (Shen and Scott, 2017; Yang and Scott, 2020) show that this often results in much tighter bounds than alternative methods.

Although DTDI shows promising state estimation results, it has not been previously applied for fault detection. To adapt the sDTDI and rDTDI methods to FD, we simply use the corresponding state estimator in Algorithm 1. To perform the test $\mathbf{y}_k \notin \hat{Y}_{k|k-1}$, the predicted output interval is computed as

$$\hat{Y}_{k|k-1} = [\mathbf{g}](k, \mathcal{I}[\hat{X}_{k|k-1}, G], V). \quad (21)$$

Remark 1. Since the use of G significantly improves the performance of DTDI, it is reasonable to consider whether it would similarly enhance the standard interval and zonotope methods. As shown in (Yang and Scott, 2020), the answer is negative for the standard interval method. Specifically, the use of the face selection operators $\beta_i^{L/U}$ makes refinement based on G much more potent in DTDI. However, methods for refining zonotopic enclosures using G have very recently been developed in (Rego et al., 2021) and show promising results. Applying these methods to fault detection is a worthwhile project, but is beyond the scope of this article.

4.1. Interval-based FD Methods Tutorial

This section demonstrates the computations required for the interval-based FD methods described above using a simple example. We cover the standard interval, sDTDI, and rDTDI methods. The zonotope methods are not covered as they are not the main contribution of this paper and require significant additional background. The reader is referred to (Kühn, 1998;

Alamo et al., 2005; Bravo et al., 2006) for further details on computations with zonotopes.

Consider the following two-dimensional system:

$$\begin{aligned} x_{1,k+1} &= 0.9x_{1,k} + 0.1x_{1,k}x_{2,k}, \\ x_{2,k+1} &= 0.9x_{2,k} - 0.1x_{1,k}x_{2,k}, \\ y_k &= x_{1,k} + v_k. \end{aligned} \quad (22)$$

Assume that $v_k \in V \equiv [-0.1, 0.1]$, $\forall k \in \mathbb{K}$. To illustrate a single step of the interval FD methods, assume that at time k we have the initial enclosure $\hat{X}_{k|k-1} = ([0, 1], [0, 1])$. This notation is intended to indicate that $\hat{X}_{k|k-1}$ is a two-dimensional interval vector with both components equal to $[0, 1]$.

Let f_1 , f_2 , and g denote the right-hand sides of the dynamics and output equation in (22). All three interval FD methods require inclusion functions for these functions, which can be constructed using interval arithmetic. We require the standard interval rules for addition, subtraction, and multiplication, which are as follows:

$$\begin{aligned} [a, b] + [c, d] &= [a + c, b + d], \\ [a, b] - [c, d] &= [a - d, b - c], \\ [a, b] \times [c, d] &= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]. \end{aligned} \quad (23)$$

Applying these rules, the inclusion functions $[f_i] = [f_i^L, f_i^U]$ and $[g] = [g^L, g^U]$ are defined for an arbitrary argument $X = ([x_1^L, x_1^U], [x_2^L, x_2^U])$ by,

$$\begin{aligned} f_1^L(X) &= 0.9x_1^L + 0.1 \min(x_1^L x_2^L, x_1^L x_2^U, x_1^U x_2^L, x_1^U x_2^U), \\ f_1^U(X) &= 0.9x_1^U + 0.1 \max(x_1^L x_2^L, x_1^L x_2^U, x_1^U x_2^L, x_1^U x_2^U), \\ f_2^L(X) &= 0.9x_2^L - 0.1 \max(x_1^L x_2^L, x_1^L x_2^U, x_1^U x_2^L, x_1^U x_2^U), \\ f_2^U(X) &= 0.9x_2^U - 0.1 \min(x_1^L x_2^L, x_1^L x_2^U, x_1^U x_2^L, x_1^U x_2^U), \\ g^L(X, V) &= x_1^L + v^L, \\ g^U(X, V) &= x_1^U + v^U. \end{aligned} \quad (24)$$

For the standard interval method, a single iteration of Algorithm 1 is computed as follows. Starting with $\hat{X}_{k|k-1} = ([0, 1], [0, 1])$, the interval $\hat{Y}_{k|k-1}$ in line 4 is computed as

$$\hat{Y}_{k|k-1} = [g](\hat{X}_{k|k-1}, V) = [0, 1] + [-0.1, 0.1] = [-0.1, 1.1]. \quad (25)$$

Next, a measurement y_k is taken and compared to $\hat{Y}_{k|k-1}$. If $y_k \notin \hat{Y}_{k|k-1}$, the algorithm terminates. Assume that $y_k = 0.5$ so that $y_k \in \hat{Y}_{k|k-1}$. Then, the algorithm proceeds to line 9, where, by (6),

$$X_k^m(y_k) = \{(x_1, x_2) : 0.4 \leq x_1 \leq 0.6\}. \quad (26)$$

The correction set $\hat{X}_{k|k}$ is then computed using (19). This step requires the refinement operator $\mathcal{I}$, which we have defined as Algorithm 1 in (Yang and Scott, 2018c) throughout this paper. Since $G = \mathbb{R}^{n_x}$ for the standard interval method and $X_k^m(y_k)$ is a simple interval in this case, it can be shown that the refinement step here simply reduces to the intersection $\hat{X}_{k|k-1} \cap X_k^m(y_k)$. Thus, we obtain $\hat{X}_{k|k} = ([0.4, 0.6], [0, 1])$. To compute the new prediction set in line 10, we compute

$\hat{X}_{k+1|k} = [\mathbf{f}](\hat{X}_{k|k})$ using (24) with $X = \hat{X}_{k|k}$. The result is $\hat{X}_{k+1|k} = ([0.36, 0.6], [-0.06, 0.9])$.

The computations for the sDTDI method are exactly the same as for the standard interval method until line 10, resulting again in $\hat{X}_{k|k} = ([0.4, 0.6], [0, 1])$. However, line 10 is computed via (20). For example, to compute the second lower bound $\hat{x}_{2,k+1|k}^L$, we first compute $\beta_2^L(\hat{X}_{k|k}) = ([0.4, 0.6], [0, 0])$. Next, this interval is refined to compute $\mathcal{I}[\beta_2^L(\hat{X}_{k|k}), G \cap X_k^m(y_k)]$. Since we choose $G = \mathbb{R}^{n_x}$ in sDTDI, this refinement again reduces to a simple intersection with $X_k^m(y_k)$, which gives

$$\mathcal{I}[\beta_2^L(\hat{X}_{k|k}), G \cap X_k^m(y_k)] = ([0.4, 0.6], [0, 0]). \quad (27)$$

In this case, this intersection accomplishes nothing that wasn't already accomplished by the intersection with $X_k^m(y_k)$ in line 9. This is a consequence of the fact that $X_k^m(y_k)$ is an interval in this example and is not the case in general. Note that the intervals $\beta_2^L(\hat{X}_{k|k})$ and $\mathcal{I}[\beta_2^L(\hat{X}_{k|k}), G \cap X_k^m(y_k)]$ computed here are not intended to be valid enclosures of the true state $\mathbf{x}_k$. Their only purpose is to be used in (20) to compute $\hat{X}_{k+1|k}$, which is a valid enclosure of $\mathbf{x}_{k+1}$. The fact that they can be used for this purpose despite not enclosing $\mathbf{x}_k$ is the central result of the DTDI theory in (Yang and Scott, 2020). Finally, the interval (27) is given as input to $f_2^L(X)$ in (24). This gives $\hat{x}_{2,k+1|k}^L = 0$, which is better than the value -0.06 obtained by the standard interval method due to the use of the face selection operator β_2^L . The remaining three bounds are computed analogously and happen to all yield the same results as the standard interval method in this case. The final result is $\hat{X}_{k+1|k} = ([0.36, 0.6], [0, 0.9])$.

To apply the rDTDI method, we first need to define an *a priori* enclosure G . Since this system does not satisfy any known invariants, we will manufacture one by lifting the system into a higher dimensional state space as proposed in (Shen and Scott, 2017). The first step is to introduce new redundant state variables defined as functions of the original states. For reasons discussed below, in this case we choose to define a single new variable x_3 by

$$x_{3,k} = x_{1,k} + x_{2,k}. \quad (28)$$

Combining equations in (22), it can be shown that this variable satisfies the difference equation

$$x_{3,k+1} = 0.9x_{3,k}. \quad (29)$$

Augmenting (22) with this new equation, we obtain a three-dimensional system whose solutions lie in the following set by design:

$$G = \{(x_{1,k}, x_{2,k}, x_{3,k}) : x_{1,k} + x_{2,k} - x_{3,k} = 0\}. \quad (30)$$

We will apply Algorithm 1 to this lifted system.

In the procedure above, we could have defined x_3 as any function of x_1 and x_2 , with different choices yielding different lifted systems and different sets G . The rationale for defining x_3 as we did is that it results in the relatively simple difference equation (29). In particular, the nonlinear term $x_{1,k}x_{2,k}$, which is likely to cause much of the overestimation in sDTDI, is completely cancelled out. Such simplifications often enable

accurate bounds on the right-hand sides of the new states to be computed even using simple interval arithmetic. If so, then DTDI will produce accurate bounds on the new states ($x_{3,k}$ in this case), which can then be used to refine the bounds on the original states ($x_{1,k}$ and $x_{2,k}$) using the manufactured invariants (28). When this is done in each time step, as in rDTDID, it can strongly mitigate the accumulation of overestimation error over time. The reader is referred to (Shen and Scott, 2017; Yang and Scott, 2020) for more details on the mechanisms by which manufactured invariants can lead to improved bounds, strategies for choosing the new states effectively, and numerous examples showing the generality of this approach.

We now proceed with the steps of Algorithm 1 for a generic step k of the rDTDID method. For the previous methods, we assumed the initial enclosure $\hat{X}_{k|k-1} = ([0, 1], [0, 1])$. Since Algorithm 1 is applied to the lifted three-dimensional system for rDTDID, we instead assume that $\hat{X}_{k|k-1} = ([0, 1], [0, 1], [0, 2])$. The bound $x_{3,k} \in [0, 2]$ is directly inferred from (28) via $\hat{X}_{3,k|k-1} = \hat{X}_{1,k|k-1} + \hat{X}_{2,k|k-1} = [0, 1] + [0, 1]$. This is always how $\hat{X}_{k|k-1}$ is computed at $k = 0$ when manufactured invariants are used. However, at later time steps $\hat{X}_{k|k-1}$ comes directly from the steps of Algorithm 1 and may not satisfy $\hat{X}_{3,k|k-1} = \hat{X}_{1,k|k-1} + \hat{X}_{2,k|k-1}$ (see the results at $k + 1$ below). Although we consider a generic step k here, defining $\hat{X}_{k|k-1}$ as $([0, 1], [0, 1], [0, 2])$ makes for a fair comparison because it ensures that, at the beginning of the step, rDTDID does not have any information that was not available to the other methods.

Given $\hat{X}_{k|k-1}$, the first step is to compute $\hat{Y}_{k|k-1}$ in line 4. This is done using (21), which requires that we first compute $\mathcal{I}[\hat{X}_{k|k-1}, G]$. In this case, the operator $\mathcal{I}$ defined in (Yang and Scott, 2018a) essentially attempts to infer tighter bounds on each individual state $x_{i,k}$ using the relation (28) and the known bounds on the other two states in $\hat{X}_{k|k-1}$. Because of the way the third component of $\hat{X}_{k|k-1}$ was defined above, the refinement has no effect here. As a result, (21) yields $\hat{Y}_{k|k-1} = [-0.1, 1.1]$, exactly as in the previous two methods. Next, the correction set $\hat{X}_{k|k}$ is computed using (19), which requires the set $\mathcal{I}[\hat{X}_{k|k-1}, G \cap X_k^m(y_k)]$, where

$$X_k^m(y_k) = \{(x_1, x_2, x_3) : 0.4 \leq x_1 \leq 0.6\}. \quad (31)$$

The behavior of the refinement operator in this case is to first intersect $\hat{X}_{k|k-1}$ with $X_k^m(y_k)$ to obtain $([0.4, 0.6], [0, 1], [0, 2])$ and then refine this set based on (28). In the latter step, the combination of (28) with the improved bound $x_{1,k} \in [0.4, 0.6]$ obtained from intersection with $X_k^m(y_k)$ allows us to infer that $x_{3,k} \in [0.4, 1.6]$, which is better than the original bound $[0, 2]$. However, the bound on $x_{2,k}$ cannot be improved. The final result of the correction step is $\hat{X}_{k|k} = ([0.4, 0.6], [0, 1], [0.4, 1.6])$.

Finally, the new prediction set $\hat{X}_{k+1|k}$ in line 10 is computed via (20). For example, to compute the second lower bound $\hat{x}_{2,k+1|k}^L$, we first compute $\beta_2^L(\hat{X}_{k|k}) = ([0.4, 0.6], [0, 0], [0.4, 1.6])$. Next, this interval is refined to compute $\mathcal{I}[\beta_2^L(\hat{X}_{k|k}), G \cap X_k^m(y_k)]$. The behavior of the refinement operator here is again to first intersect $\beta_2^L(\hat{X}_{k|k})$ with $X_k^m(y_k)$, which has no effect, and then refine this set based on (28). In the latter step, the combination of (28) with the bound

$x_{2,k} \in [0, 0]$ obtained through the use of the face selection operator allows us to infer that $x_{3,k} \in [0.4, 0.6]$. Therefore,

$$\mathcal{I}[\beta_2^L(\hat{X}_{k|k}), G \cap X_k^m(y_k)] = ([0.4, 0.6], [0, 0], [0.4, 0.6]). \quad (32)$$

As with sDTDID, $\beta_2^L(\hat{X}_{k|k})$ and $\mathcal{I}[\beta_2^L(\hat{X}_{k|k}), G \cap X_k^m(y_k)]$ are not intended to be valid enclosures of the true state $\mathbf{x}_k$. Their only purpose is to be used in (20) to compute $\hat{X}_{k+1|k}$, which is a valid enclosure of $\mathbf{x}_{k+1}$ as shown in (Yang and Scott, 2020).

In the last step, (32) is given as input to $f_2^L(X)$ in (24). This gives $\hat{x}_{2,k+1|k}^L = 0$. Following the same procedure for the remaining five bounds results in $\hat{X}_{k+1|k} = ([0.36, 0.6], [0, 0.9], [0.4, 1.44])$. The results for x_1 and x_2 are no better than sDTDID for this single step. This is in fact guaranteed by the way we initialized the third component of $\hat{X}_{k|k-1}$. However, the computed bounds for x_3 at $k + 1$ now contain new information that $\hat{X}_{3,k+1|k} = [0.4, 1.44]$ is tighter than what can be inferred from (24) via $\hat{X}_{1,k+1|k} + \hat{X}_{2,k+1|k} = [0.36, 1.5]$. These bounds can now be used to improve the bounds on x_1 and x_2 in the next step through the use of the refinement operator, ultimately leading to a better result than sDTDID. Indeed, the bounds $\hat{X}_{k+2|k+1}$ obtained in the next step (assuming $y_{k+1} = 0.5$) for the standard interval, sDTDID, and rDTDID methods, respectively, are $([0.3564, 0.594], [-0.108, 0.8136])$, $([0.36, 0.594], [0, 0.774])$, and $([0.36, 0.5904], [0, 0.774], [0.36, 1.296])$. Thus, rDTDID provides a slightly tighter upper bound for x_1 than sDTDID. In general, such small improvements tend to compound, leading to significant differences in accuracy over longer time horizons.

5. Fault Detection in a CSTR

Consider the following model of a continuous stirred tank reactor (CSTR) from (Shen and Scott, 2017):

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + h[-u_{3,k}x_{1,k}x_{2,k} - k_2x_{1,k}x_{3,k} + \tau^{-1}(u_{1,k} - 2x_{1,k})], \\ x_{2,k+1} &= x_{2,k} + h[-u_{3,k}x_{1,k}x_{2,k} + \tau^{-1}(u_{2,k} - 2x_{2,k})], \\ x_{3,k+1} &= x_{3,k} + h[u_{3,k}x_{1,k}x_{2,k} - k_2x_{1,k}x_{3,k} - 2\tau^{-1}x_{3,k}], \\ x_{4,k+1} &= x_{4,k} + h[k_2x_{1,k}x_{3,k} - 2\tau^{-1}x_{4,k}], \\ y_{1,k} &= x_{2,k} + v_{1,k}, \\ y_{2,k} &= x_{3,k} + v_{2,k}, \\ y_{3,k} &= x_{4,k} + v_{3,k}. \end{aligned} \quad (33)$$

Above, x_i is the concentration (M) of species i , y_i is the measurement of x_i , u_i and v_i are disturbances and measurement noises (specified further in each scenario considered below), $h = 0.015$ min, $\tau = 20$ min, and $k_2 = 0.4$ M⁻¹min⁻¹.

The states of this system do not satisfy any known *a priori* enclosure G . Therefore, to apply rDTDID, we manufacture invariants by lifting the system into a higher dimensional state space similarly as in §4.1. Here, we choose to introduce four

Table 1: Plot markers for set-based FD methods

Index	Method Description	Marker
(i)	Standard interval method	□
(ii)	Zonotope method (Combastel, 2005)	◇
(iii)	Zonotope method (Alamo et al., 2005)	○
(iv)	sDTDI	△
(v)	rDTDI	★

new states defined specifically by

$$\begin{aligned}
z_{1,k} &= -\frac{1}{3}x_{1,k} - \frac{1}{3}x_{2,k} + \frac{1}{3}x_{3,k}, \\
z_{2,k} &= -\frac{1}{3}x_{1,k} - \frac{1}{3}x_{3,k} + \frac{1}{3}x_{4,k}, \\
z_{3,k} &= -x_{1,k} + 2x_{2,k} + x_{3,k}, \\
z_{4,k} &= x_{1,k} - x_{2,k} + x_{4,k}.
\end{aligned} \tag{34}$$

Next, we augment the original system with difference equations for these new states derived by combining (33) and (34):

$$\begin{aligned}
z_{1,k+1} &= z_{1,k} + h[u_{3,k}x_{1,k}x_{2,k} - \frac{1}{3}\tau^{-1}(u_{1,k} + u_{2,k}) - 2\tau^{-1}z_{1,k}], \\
z_{2,k+1} &= z_{2,k} + h[k_2x_{1,k}x_{3,k} - \frac{1}{3}\tau^{-1}u_{1,k} - 2\tau^{-1}z_{2,k}], \\
z_{3,k+1} &= z_{3,k} + h\tau^{-1}[2(u_{2,k} - z_{3,k}) - u_{1,k}], \\
z_{4,k+1} &= z_{4,k} + h\tau^{-1}[u_{1,k} - u_{2,k} - 2z_{4,k}].
\end{aligned} \tag{35}$$

By design, the solutions of this augmented system satisfy (34), regardless of the values of the disturbances. Therefore, the rDTDI method described in §4 can be applied with

$$G = \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^8 : (34) \text{ holds}\}. \tag{36}$$

The specific definitions (34) were chosen such that some uncertain and nonlinear terms in (33) cancel out when forming (35).

In the following subsections, the conventional and set-based FD methods described in §3.4 are compared in several fault-free and faulty scenarios. Each set-based method is assigned a consistent marker in all plots. These are collected in Table 1 for easy reference. The fault-free and faulty scenarios for this example are listed in Table 2. In addition to these scenarios, we also consider different possibilities for how the disturbances $\mathbf{u}_k$ and measurement noises $\mathbf{v}_k$ are distributed in the simulation of the real system (both prior to a fault and after a fault unless otherwise specified in the fault description in Table 2). These cases are organized into subsections 5.1–5.5. To mimic the situation that occurs when applying FD methods to real systems, these distributions are not precisely known to the FD methods and may violate some of their underlying assumptions. The implementation of the EKF and set-based methods in such cases are discussed below. For PCA, historical data is always generated from fault-free steady-state simulations using the specified distributions for $\mathbf{u}_k$ and $\mathbf{v}_k$, so this data is different in each subsection.

5.1. Truncated Gaussian Disturbances

We first consider the case where the system disturbances u_1 (M), u_2 (M), and u_3 ($\text{M}^{-1}\text{min}^{-1}$), and the measurement noises

Table 2: Fault-free and faulty scenarios for the CSTR example. Unless otherwise specified, all scenarios begin from the nominal steady-state $\mathbf{x}_0 = (0.036, 0.038, 0.36, 0.052)$ and the uncertainties $\mathbf{u}_k$ and $\mathbf{v}_k$ obey the probability distributions specified in Sections 5.1–5.5

Index	Scenario Description
(a)	(Fault Free) Normal operation
(b)	(Fault Free) $\mathbf{x}_0$ is perturbed away from the nominal steady state by 10% to (0.032, 0.034, 0.32, 0.047)
(c)	(Fault Free) Starting from $t = 6$ min, all disturbances take constant values far from their means but within their bounds: $u_1 = 1.05$, $u_2 = 0.95$ and $u_3 = 45$
(d)	(Faulty) Starting from $t = 6$ min, the inlet concentration takes a constant value outside its bounds: $u_1 = 0.5$
(e)	(Faulty) Starting from $t = 6$ min, the residence time decreases by 40% to $\tau = 12$ min

v_1 (M), v_2 (M), and v_3 (M) are all serially uncorrelated truncated Gaussian random variables. Specifically, Gaussian distributions with means μ_i and standard deviations σ_i are truncated to $[\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$. This case is considered first because it most closely satisfies the assumptions of all methods at once. Specifically, $\mathbf{u}_k$ and $\mathbf{v}_k$ are bounded, as required by set-based methods, and (very nearly) Gaussian, as assumed by EKF and PCA. The EKF method is given the means and standard deviations of $\mathbf{u}_k$ and $\mathbf{v}_k$: $\mu_{\mathbf{u}} = (1.0, 0.9, 30)$, $\mu_{\mathbf{v}} = (0, 0, 0)$, $\sigma_{\mathbf{u}} = (\frac{0.1}{3}, \frac{0.1}{3}, \frac{20}{3})$, and $\sigma_{\mathbf{v}} = (\frac{10^{-2}}{3}, \frac{10^{-3}}{3}, \frac{10^{-3}}{3})$. The set-based methods are given the truncation bounds: $u_{1,k} \in [0.9, 1.1]$, $u_{2,k} \in [0.8, 1.0]$, $u_{3,k} \in [10, 50]$, $v_{1,k} \in [-10^{-2}, 10^{-2}]$, and $v_{2,k}, v_{3,k} \in [-10^{-3}, 10^{-3}]$. Both methods are given the exact initial condition in Table 2.

Figure 2 shows the performance of all methods in the fault-free Scenario (a). The top subfigure shows the T^2 values generated by PCA (black). The middle subfigure shows the residuals generated by EKF from (3) (blue). For both methods, a fault is detected if the residual exceeds the threshold (red dashed). Although EKF generates a distinct residual for each measured variable, we consistently plot only the one with the most false alarms (fault-free scenarios) or the one that detects the fault fastest (faulty scenarios), in this case, l_4 . In the bottom subfigure, each pair of lines represents the upper and lower bounds of the predicted output set $\hat{Y}_{k|k-1}$ from (14) computed by one of the set-based methods (see Table 1). For brevity, bounds are only shown for a single measured variable, in this case, y_1 . The measured value of y_1 is the solid yellow curve in the middle. Under fault-free operation, this curve should stay within the bounds generated by each method. If it goes outside the bounds, then a fault is declared by that method.

Figure 2 shows that the set-based methods never generate false alarms, while both PCA and EKF generate false alarms fairly often. This is a major advantage of the set-based approaches and is guaranteed because $\mathbf{u}_k$ and $\mathbf{v}_k$ are bounded.

Among the set-based methods, rDTDI provides the tightest output bounds, followed closely by sDTDI. The zonotope methods outperform the standard interval method, but are still very conservative. This is largely caused by the correction

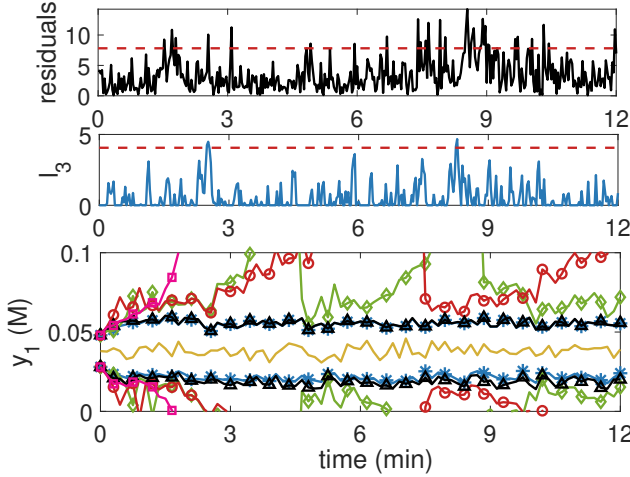


Figure 2: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table I) for the CSTR example in Scenario (a) with truncated Gaussian noises.

step, which is more challenging for zonotopes. The algorithm used for this step involves discrete choices (Bravo et al. 2006), which also accounts for the erratic oscillatory behavior of the zonotope bounds. The sDTDI bounds are a drastic improvement over the standard interval method, showing the value of the DI approach. The use of manufactured invariants in rDTDI leads to even tighter prediction sets. This difference appears small in Figure 2 because the widths of the sDTDI and rDTDI bounds are dominated by the measurement error. Moreover, since the sDTDI and rDTDI bounds are made similar again after the correction step, the difference in prediction sets does not accumulate over time. Nonetheless, the improved accuracy of rDTDI proves to be very significant for FD (see Scenarios (d) and (e) below).

In Scenario (b), no fault occurs, but the system starts away from steady-state and operates transiently for a short time. Both EKF and the set-based methods are given the exact initial condition. Figure 3 shows that PCA gives a clear false alarm in this case. In contrast, EKF and the set-based methods operate normally, with the set-based methods giving no false alarms and EKF giving false alarms about as often as in Scenario (a). This is because both EKF and the set-based methods are based on dynamic models that can explain the data from this scenario without asserting a fault, whereas PCA is trained using steady-state data, which makes the transient data appear anomalous. It is arguable that this scenario should be considered a fault because it violates the assumptions of PCA. However, in practice, faults are defined by the application rather than by the FD method (e.g., as those events needing intervention), and it is clearly desirable to have methods that are flexible with regard to the range of fault-free behaviors they can model. Thus, we consider this to be a negative result for PCA, indicating that, at least in the standard form implemented here, it is not appropriate for systems with transients.

Scenario (c) considers a large, persistent disturbance. Specifically, u_1 , u_2 , and u_3 obey their usual distributions until $t = 6$,

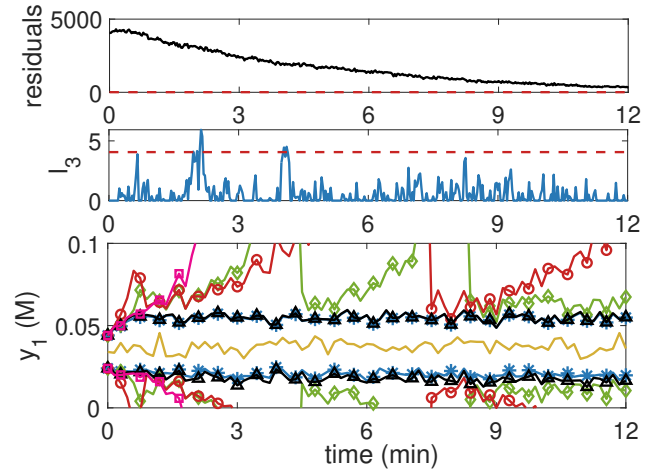


Figure 3: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table I) for the CSTR example in Scenario (b) with truncated Gaussian noises.

but afterwards take constant values that deviate significantly from their means but remain within their interval bounds. Because these values remain within their bounds, the set-based methods do not consider this a fault and give no alarms, as shown in Figure 4. In contrast, both PCA and EKF regard this as a fault. As in Scenario (b), it is debatable which conclusion is correct. The value of u_k after $t = 6$ still satisfies the basic assumptions of the set-based methods, while it violates those of EKF and shifts the system to a new operating point that is inconsistent with the historical data used by PCA. Thus, all methods are correct in the sense that they properly identify whether their notion of fault-free behavior has been violated. Yet, again, faults should be defined by the application rather than the method. In practice, moderate shifts in the operating point of a plant, either due to disturbances or intentional changes, are common, and FD algorithms should be able to distinguish these from more significant shifts that warrant alarms. To investigate this, we deliberately designed Scenarios (c) and (d) as, respectively, fault-free and faulty persistent disturbances. We argue that the ability of set-based methods to encode such distinctions precisely by setting the allowable range of u_k is a significant advantage. In contrast, the inconsistency of new operating conditions with historical training data is a well-known limitation of purely data-driven FD methods (Venkatasubramanian et al., 2003a).

Scenario (d) considers another large, persistent disturbance. Specifically, u_1 , u_2 , and u_3 obey their usual distributions until $t = 6$, but afterwards u_1 takes a constant value that deviates significantly from its mean and falls outside of its interval bound. In contrast to Scenario (c), this scenario should be regarded as a fault in all methods. Figure 5 shows that PCA detects the fault earliest at $t = 6.45$, followed by EKF at $t = 6.57$. Among the set-based methods, rDTDI is the only one to detect the fault and does so at $t = 7.47$. This is observed in the bottom subfigure as the yellow curve leaving the blue bounds.

Scenario (e) considers another fault arising from a change in

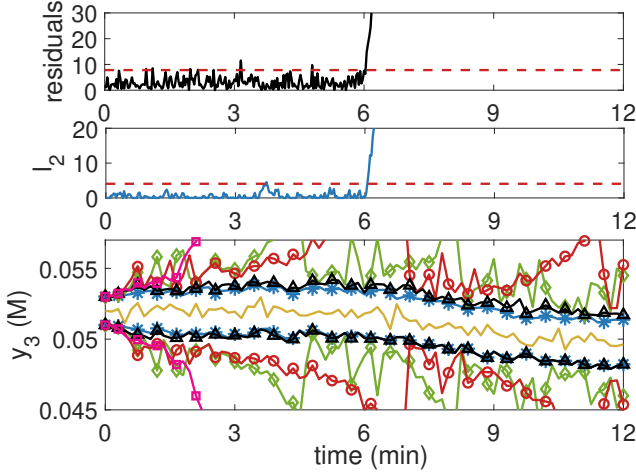


Figure 4: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table I) for the CSTR example in Scenario (c) with truncated Gaussian noises.

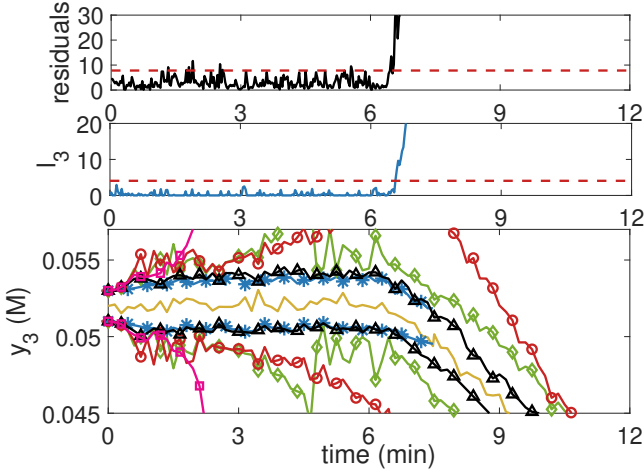


Figure 5: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table I) for the CSTR example in Scenario (d) with truncated Gaussian noises.

the model rather than the disturbances. Starting from $t = 6$, the residence time decreases by 40%, which could result from channeling in the reactor. Figure 6 shows that PCA and EKF detect the fault at $t = 6.06$ and $t = 6.15$, respectively. Again, rDTDI is the only set-based method able to detect the fault and does so at $t = 6.72$.

Although we only show the results for one run in Scenarios (a)–(e), we actually ran all of our numerical tests over 100 times with different randomly generated disturbance and measurement noise sequences to ensure stability of the results. In particular, in Scenarios (d) and (e), rDTDI is the only set-based method to detect the fault in all runs. This method is more sensitive to faults because it generally provides tighter enclosures $\hat{Y}_{k|k-1}$ of the true set of outputs consistent with the fault-free model, $Y_{k|k-1}$.

We conclude that rDTDI is the most effective set-based FD

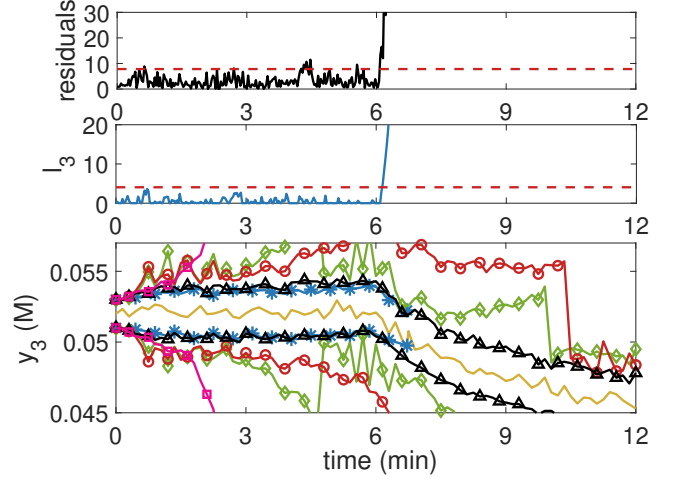


Figure 6: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table I) for the CSTR example in Scenario (e) with truncated Gaussian noises.

method tested. It is also very efficient. The average computational time of rDTDI per step is 0.0013 s, which is much faster than the sampling time for this problem, less than the 0.0033 s and 0.0023 s required by the zonotope methods (ii) and (iii), respectively, and only about 10 times slower than the standard interval method. Although all set-based methods eliminate the false alarms suffered by PCA and EKF, most of them failed to detect any faults. In contrast, the fault sensitivity of rDTDI is competitive with PCA and EKF, although rDTDI is slower.

5.2. Unbounded Disturbances

We now consider the case where $\mathbf{w}_k$ and $\mathbf{v}_k$ are serially uncorrelated Gaussian random variables. We use the same means and variances as in §5.1 but the distributions are no longer truncated, so $\mathbf{w}_k$ and $\mathbf{v}_k$ are unbounded. This matches the assumptions of PCA and EKF exactly, but violates the basic assumption of the set-based methods. To apply these methods, W and V are specified using 99.7% confidence intervals of the form $[\mu - 3\sigma, \mu + 3\sigma]$ for each variable. Thus, all methods are implemented exactly as in §5.1 but the simulated system is different.

In Scenario (a), PCA and EKF perform exactly as before. However, rDTDI generates false alarms in 12% of the nominal trajectories simulated. In theory, the other set-based methods can also generate false alarms in this case, but their bounds are too conservative. Specifically, when the bounds V and W are not truly valid enclosures, a false alarm will occur if a measurement is observed that lies in $Y_{k|k-1}$ but not $\hat{Y}_{k|k-1}$. Clearly, this is much more likely if $\hat{Y}_{k|k-1}$ is smaller, which is the case for rDTDI here. The results for Scenarios (b)–(e) are also very similar to those in §5.1 except for the generation of false alarms by rDTDI (figures omitted for brevity).

While it is reasonable to assume that $\mathbf{w}_k$ and $\mathbf{v}_k$ are bounded in any real system, it is unreasonable to assume that rigorous bounds W and V will always be known or used, especially since choosing W and V very conservatively degrades fault sensitivity. Therefore, it is important to consider how set-based meth-

ods will behave when the inclusions $\mathbf{w}_k \in W$ and $\mathbf{v}_k \in V$ only hold with high probability. This reality is seldom if ever discussed in the set-based FD literature, yet our experiments show that there are some critical issues to resolve. First, even if it is very unlikely that $\mathbf{w}_k \notin W$ or $\mathbf{v}_k \notin V$ in any given time step (e.g., 0.3% in our experiments), the probability that at least one such violation will occur over a horizon $k = 0, \dots, K$ compounds with increasing K . Therefore, if the set-based state estimator is accurate, false alarms can be quite common. This alone is not necessarily a serious problem since the false alarm rate might still be much lower than in conventional methods. However, restarting a set-based method after a false alarm is nontrivial. In particular, resetting the set-based state estimator requires an enclosure of the possible states at the current time, which is no longer available. For observable linear systems, such an enclosure can be inferred from future measurements (Meslem and Ramdani, 2020). However, aside from the trivial case of full state measurement, no such method has yet been proposed for nonlinear systems. Thus, at present, there is no way to continue a set-based FD method once a false alarm has occurred. Until this deficiency is addressed, it must be concluded that set-based FD methods are not usable when the inclusions $\mathbf{w}_k \in W$ and $\mathbf{v}_k \in V$ are not certain, which puts their practical utility in serious doubt.

To gain further insights into the potential advantages of the set-based approach relative to conventional approaches, all following experiments consider cases where $\mathbf{w}_k$ and $\mathbf{v}_k$ are bounded, and hence the set-based methods function as intended.

5.3. Uniformly Distributed Disturbances

To investigate how PCA and EKF perform with non-Gaussian distributions, we now consider the case where $\mathbf{w}_k$ is uniformly distributed within the same interval W used in §5.1 and $\mathbf{v}_k$ follows the truncated Gaussian distribution from §5.1. PCA is implemented with historical data generated with these distributions, EKF is given the correct means and standard deviations of $\mathbf{w}_k$ and $\mathbf{v}_k$, and the set-based methods are given W and V .

In the fault-free Scenarios (a)–(c), the results for all methods are very similar to those in §5.1 except that PCA exhibits fewer false alarms. This is shown for Scenario (a) in Figure 7. In the faulty Scenarios (d)–(e), PCA and EKF both detect the faults at 6 min with almost no delay, even more rapidly than in §5.1. The set-based methods perform similarly to §5.1, with rDTDI detecting both faults at around 7 min (competitive with, but slower than, PCA and EKF), and all other methods failing to detect the faults at all. Figures for Scenarios (b)–(e) are omitted for brevity.

5.4. Other Non-Gaussian Disturbances

To further investigate the impact of non-Gaussianity, we consider several other distributions for the elements of $\mathbf{w}_k$, including asymmetric and bimodal distributions. In all cases, the distributions are truncated to the same bounds as in §5.1 and the distribution of $\mathbf{v}_k$ remain as in §5.1. Again, PCA is implemented with historical data generated with these distributions,

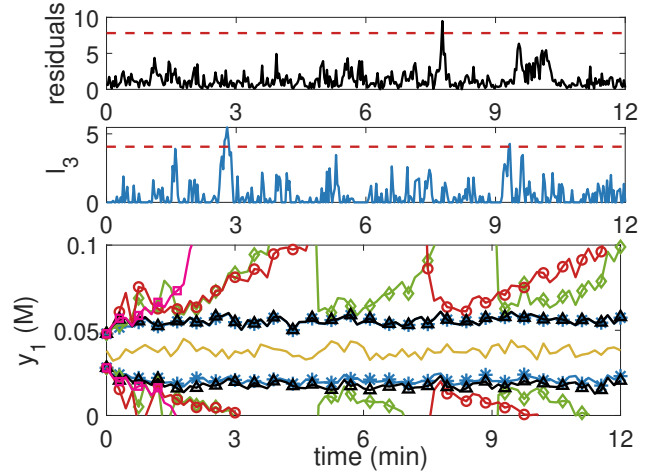


Figure 7: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table 1) for the CSTR example in Scenario (a) with uniformly distributed disturbances.

EKF is given the correct means and standard deviations, and the set-based methods are given W and V . The results are very similar to those in §5.1 so the details are omitted for brevity. This indicates that all methods are insensitive to moderate deviations from Gaussianity. This was expected for set-based methods, but somewhat surprising for PCA and EKF. Using non-Gaussian distributions consistently cause PCA and EKF to give slightly fewer false alarms in Scenario (a) relative to §5.1, suggesting a potential reduction in fault sensitivity. However, the detection times in faulty Scenarios (d) and (e) are essentially unchanged because the residuals rise so sharply in those cases.

5.5. Time-Invariant Uncertainties

So far, we have assumed that all uncertainties are time-varying, serially uncorrelated random variables. However, in practice, many uncertainties are time-invariant, or vary only over time-scales much longer than the sampling time. To investigate FD performance in such cases, we now assume that the reaction rate constant u_3 is a time-invariant uncertain parameter with the same upper and lower bounds as in §5.1. The remaining disturbances and measurement noises obey the same truncated Gaussian distributions as in §5.1.

This situation is easily handled by set-based methods since they only depend on the bounds of the uncertain variables. In contrast, to apply EKF, u_3 must be approximated as a time-varying Gaussian disturbance. The most sensible approach is to choose a Gaussian distribution centered on the midpoint of the u_3 bounds and with most of the probability density contained in those bounds. Therefore, EKF is implemented with the same mean and standard deviation as in §5.1. Implementing PCA correctly in this situation requires a bit more clarity about the nature of u_3 . If u_3 is truly time-invariant, then the value of u_3 during the time when the historical data was collected would be the same as the value describing the system in the present, when the fault detection algorithms are applied. This is good for PCA because the method will not be affected at all by the

uncertainty in u_3 . In contrast, if u_3 varies slowly, e.g., as would occur from catalyst deactivation, then the value of u_3 during the time that the historical data was collected may be different than the present value. Indeed, historical data may have been collected with multiple values of u_3 . To delineate these cases, we consider the case of truly time-invariant u_3 in §5.5.1 and the case of long-term time invariant u_3 in §5.5.2. In the former, the historical data from PCA was generated with same value of u_3 used to simulate the system during the fault detection tests. In the latter, the historical data was generated with random but fixed values of u_3 in different trajectories.

5.5.1. Truly time-invariant uncertainties

For this test, u_3 is set to 40 for all simulations. This value falls within the bounds [10, 50], but is significantly different than the midpoint 30 assumed to be the mean by EKF. The results for Scenario (a) are shown in Figure 2. PCA works normally with a false alarm rate similar to that in §5.1. Since u_3 has the same value in both the historical data and the current trajectory, the fact that this value is unknown has no effect on PCA. In contrast, the EKF residual rapidly diverges, giving a clear false alarm. Since EKF expects a mean value of $u_3 = 30$, the fact that $u_3 = 40$ causes a significant error between the EKF predicted output and real output. In the time-varying case, this will not be problematic because this error will not persist in future time steps, and will in fact be negated with high probability. However, when u_3 is time invariant, this error occurs with the same sign in every time step, causing a steady accumulation in the EKF residual. We conclude that the EKF method cannot be used for systems with uncertain time-invariant parameters, which is a significant drawback. In contrast, the set-based methods perform normally as expected, since they are not sensitive to the distribution of u_3 .

In Scenarios (b)–(e), EKF fails in the same way as in Scenario (a), while the results for PCA and the set-based methods are similar to §5.1. The details are omitted for brevity.

5.5.2. Long-term time-invariant uncertainties

For this test, u_3 is assumed to remain constant on the time-scale of fault detection but vary over longer times. Accordingly, the historical data for PCA was generated by simulating 1000 different trajectories starting from the nominal steady-state, each with a single u_3 value randomly sampled from the truncated Gaussian distribution in §5.1 and then held constant throughout the simulation.

In this case, EKF fails by the same mechanism described in §5.5.1 while the set-based methods still perform normally, as expected. Figure 9 shows the PCA results in Scenario (a). Two cases are considered. In the top subfigure, the constant value of u_3 during the FD test is the midpoint of its interval bounds, 30. The residuals generated by PCA are always significantly below the threshold, indicating a slight loss of sensitivity relative to §5.1. This is because the threshold is set using historical data that has more variance than the current process data since multiple u_3 values are used in the historical data. At the same time, the current value of u_3 is the mean value, which generates relatively small residuals. In contrast, in the bottom subfigure,

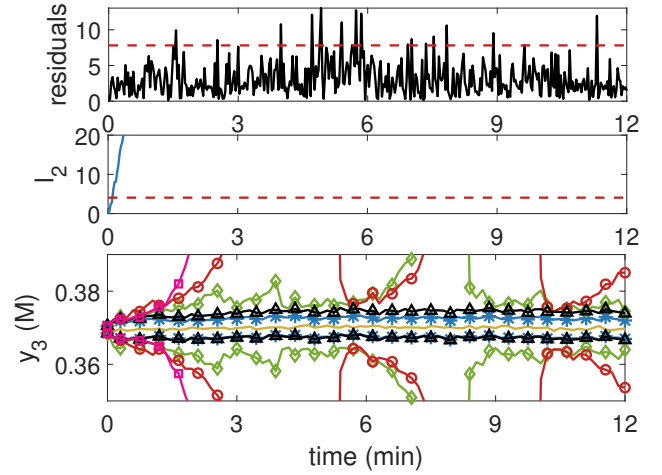


Figure 8: Results of the PCA (top), EKF (middle), and set-based (bottom) methods (markers defined in Table 1) for the CSTR example in Scenario (a) when u_3 is a truly time-invariant parameter taking value 40 and u_1 and u_2 are truncated Gaussian noises.

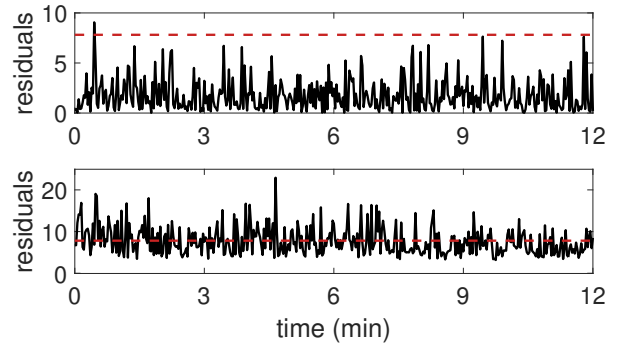


Figure 9: Results of the PCA method for the CSTR example in Scenario (a) when u_3 is a long-term time-invariant parameter taking the value 30 (top) and 50 (bottom), while u_1 and u_2 are truncated Gaussian noises.

the constant value of u_3 during the FD test is its upper bound, 50, which is far from the mean value. In this case, PCA generates nearly constant false alarms, despite the fact that u_3 is in its normal range. In the faulty scenarios (d) and (e), the PCA residuals rise sharply for both $u_3 = 30$ and $u_3 = 50$ cases, as in §5.1 (not shown). Thus, the faults are detected properly when $u_3 = 30$, but not when $u_3 = 50$ since the system is already in an alarm state. These results indicate that PCA is not effective for systems with slowly varying uncertainties without some means to match the training data to the current uncertainty value.

6. Fault Detection in a Batch Reactor

The following dynamics describe a six-species enzymatic reaction in a batch reactor, where x_i represents the concentration

(M) of species i (Scott and Barton, 2013):

$$\begin{aligned}
x_{1,k+1} &= x_{1,k} + h[-k_{1,k}x_{1,k}x_{2,k} + k_{2,k}x_{3,k} + k_{6,k}x_{6,k}], \\
x_{2,k+1} &= x_{2,k} + h[-k_{1,k}x_{1,k}x_{2,k} + k_{2,k}x_{3,k} + k_{3,k}x_{3,k}], \\
x_{3,k+1} &= x_{3,k} + h[k_{1,k}x_{1,k}x_{2,k} - k_{2,k}x_{3,k} - k_{3,k}x_{3,k}], \\
x_{4,k+1} &= x_{4,k} + h[k_{3,k}x_{3,k} - k_{4,k}x_{4,k}x_{5,k} + k_{5,k}x_{6,k}], \\
x_{5,k+1} &= x_{5,k} + h[-k_{4,k}x_{4,k}x_{5,k} + k_{5,k}x_{6,k} + k_{6,k}x_{6,k}], \\
x_{6,k+1} &= x_{6,k} + h[k_{4,k}x_{4,k}x_{5,k} - k_{5,k}x_{6,k} - k_{6,k}x_{6,k}].
\end{aligned} \tag{37}$$

The rate constants $\mathbf{k} = (k_1, \dots, k_6)$ are taken to be uncertain and modelled as disturbances $\mathbf{w}_k$. The initial condition is $\mathbf{x}_0 = (34, 20, 0, 0, 16, 0)$ and the step size is $h = 9 \times 10^{-5}$ h. Every state is measured with error, so $\mathbf{y}_k = \mathbf{x}_k + \mathbf{v}_k$.

As a batch reactor, this system does not operate at steady-state, which poses new challenges for fault detection. It has already been shown in the previous example (§5.1, Scenario (b)) that PCA is not usable for non-steady operation. Therefore, we only compare EKF and set-based methods in this section. Moreover, we only consider the case where $\mathbf{w}_k$ and $\mathbf{v}_k$ are serially uncorrelated truncated Gaussian uncertainties. Experiments with other distributions lead to the same conclusions already discussed in §5 and do not provide new insights. The bounds for the measurement noises are chosen as $[-0.2, 0.2]$ and for the rate constants as $[\hat{\mathbf{k}}, 10\hat{\mathbf{k}}]$, where $\hat{\mathbf{k}} = (0.1, 0.033, 16, 5, 0.5, 0.3)$. The distribution for each of these variables is formed by truncating the Gaussian distribution with mean μ equal to the midpoint of the bounds and standard deviation σ such that the bounds equal $[\mu - 3\sigma, \mu + 3\sigma]$.

To implement rDTDI, we use a nontrivial enclosure G derived from species conservation principles. Specifically, it is known that this reaction network satisfies the following reaction invariants (Scott and Barton, 2013):

$$\begin{aligned}
x_2 + x_3 &= 20, \\
x_5 + x_6 &= 16, \\
x_1 - x_2 + x_4 - x_5 &= -2.
\end{aligned} \tag{38}$$

In addition, the states are known to satisfy the natural bounds $\mathbf{x} = (x_1, \dots, x_6) \in X_{\text{nat}} = [0, 34] \times [0, 20] \times [0, 20] \times [0, 34] \times [0, 16] \times [0, 16]$. Thus, we apply rDTDI with

$$G = \{\mathbf{x} \in X_{\text{nat}} : (38) \text{ holds}\}. \tag{39}$$

Table 3: Fault-free and faulty scenarios for the batch reactor example

Index	Scenario Description
(a)	(Fault Free) Normal operation
(b)	(Faulty) Starting from $t = 0.009$ h, a sensor fault increases the variance of v_3 . The new distribution is a Gaussian with $\mu = 0$ and $\sigma = 0.1$ truncated to the 99.7% confidence interval $[-0.3, 0.3]$.
(c)	(Faulty) Starting from $t = 0.009$ h, k_3 takes a constant value outside its bounds: $k_3 = 8$.
(d)	(Faulty) Starting from $t = 0.009$ h, k_1 takes a constant value outside its bounds: $k_1 = 0.08$.

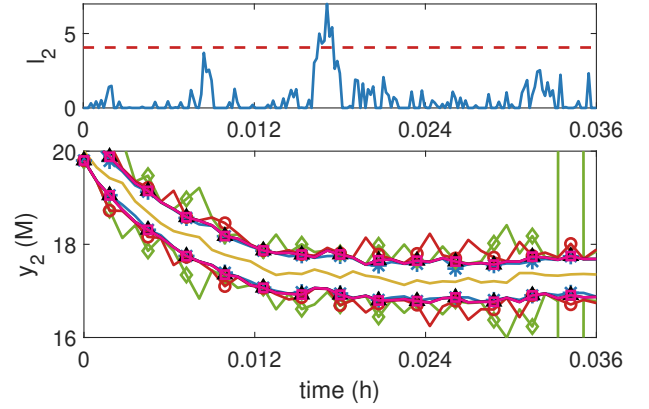


Figure 10: Results of the EKF (top) and set-based (bottom) methods (markers defined in Table 1) for the batch reactor example in Scenario (a) with truncated Gaussian noises.

The fault-free and faulty scenarios considered for this example are listed in Table 3.

Under normal operation (Scenario (a), Figure 10), EKF raises a small number of false alarms, while the set-based methods give no false alarms as expected. The rDTDI method provides the tightest output bounds, followed closely by sDTDI and the standard interval method. The standard interval method performs much better than in §5. This is attributed to the fact that all states are measured in this example, so the correction step prevents conservatism in the prediction step from accumulating over time. The two zonotope methods are the weakest due to the conservatism introduced in the correction step.

In Scenario (b), a sensor fault occurs at $t = 0.009$ that significantly increases the variance of v_3 . As shown in Figure 11, this immediately increases the variance of the EKF residual l_3 . However, since v_3 is serially uncorrelated and still has zero mean, the increased measurement errors do not cause the residual to accumulate over time, and hence l_3 does not shift definitively above its threshold. The end result looks more like an increased false alarm rate rather than a clear fault signature. In contrast, all set-based methods are very sensitive to this fault, detecting it quickly and almost simultaneously at $t = 0.00918$.

In Scenario (c), a fault occurs at $t = 0.009$ that causes k_3 to take the constant value $8 \notin [16, 160]$. Figure 12 shows that EKF detects this fault immediately, while all set-based methods fail to detect it within the simulated time horizon. EKF is very sensitive to this fault because k_3 deviates sharply from its pre-fault mean. This drives the states away from their expected mean values, causing accumulating residuals. However, the deviation of k_3 from the interval $[16, 160]$ known to the set-based methods is small relative to the width of this interval ($\sim 5\%$). Thus, this fault does not produce states that are significantly different from those achievable with $k_3 \in [16, 160]$, making it difficult for set-based methods to detect.

In Scenario (d), a fault occurs at $t = 0.009$ that causes k_1 to take the constant value $0.08 \notin [0.1, 1]$. Figure 13 shows that EKF is very sensitive to this fault as well, detecting it at $t = 0.0954$. However, in this case the set-based methods are

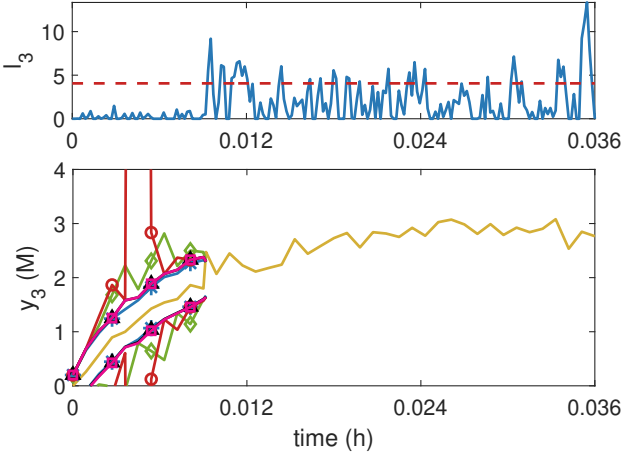


Figure 11: Results of the EKF (top) and set-based (bottom) methods (markers defined in Table I) for the batch reactor example in Scenario (b) with truncated Gaussian noises.

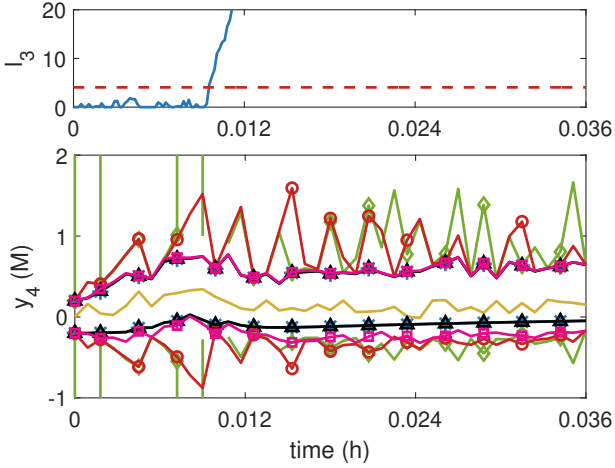


Figure 12: Results of the EKF (top) and set-based (bottom) methods (markers defined in Table I) for the batch reactor example in Scenario (c) with truncated Gaussian noises.

sensitive as well, albeit slower. The rDTDI method detects the fault earliest at $t = 0.01962$, followed by sDTDI at $t = 0.02565$ and the standard interval method at $t = 0.027$.

7. Fault Detection in a Sewer System

The following dynamics describe a sewer system with three tanks (Tornil-Sin et al., 2012):

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + h[u_{1,k} + u_{2,k} - \kappa_1 x_{1,k}], \\ x_{2,k+1} &= x_{2,k} + h[\kappa_1 x_{1,k} - \kappa_2 \sqrt{x_{2,k}}], \\ x_{3,k+1} &= x_{3,k} + h[\kappa_2 \sqrt{x_{2,k}} + u_{3,k} - \kappa_3 x_{3,k}], \\ y_{1,k} &= x_{2,k} + v_{1,k}, \\ y_{2,k} &= x_{3,k} + v_{2,k}. \end{aligned} \quad (40)$$

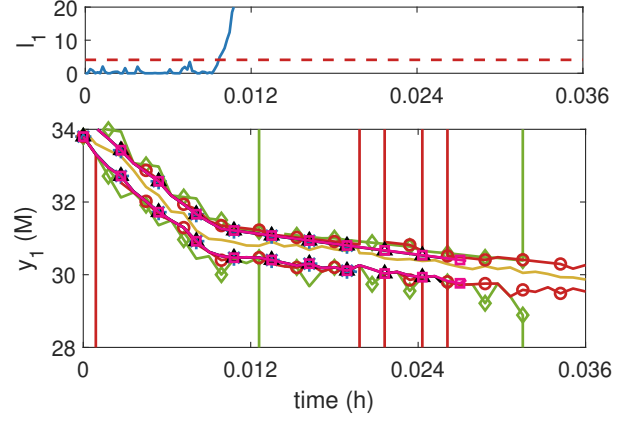


Figure 13: Results of the EKF (top) and set-based (bottom) methods (markers defined in Table I) for the batch reactor example in Scenario (d) with truncated Gaussian noises.

Above, x_i is the water volume in tank i (m^3), y_i is the measurement of x_i , v_i is the measurement noise, u_i is the inlet flowrate of rain (m^3s^{-1}), κ_i is the outlet valve constant, and $h = 30$ s. We assume that $u_i = d_i + w_i$, where $\mathbf{d} = (1, 2, 1)$ specifies the nominal inflows and w_i is a disturbance. The disturbances and measurement noises are taken to be serially uncorrelated truncated Gaussian random variables with bounds $w_i \in [-0.1, 0.1]$, $v_1 \in [-50, 50]$, and $v_2 \in [-100, 100]$. The corresponding distributions are generated by truncating Gaussian distributions with mean zero and standard deviations chosen so that the bounds above are equal to $[\mu - 3\sigma, \mu + 3\sigma]$. The valve constants κ_i are assumed to be long-term time-invariant uncertain parameters as explained in §5.5. The interval bounds are $\kappa_1 \in [4.8, 6.8] \times 10^{-4}$, $\kappa_2 \in [1.99, 2.01] \times 10^{-2}$, and $\kappa_3 \in [9.9, 10.1] \times 10^{-4}$. Each κ_i obeys a truncated Gaussian distribution constructed over these bounds in the usual way. For each trajectory simulated (either for generating PCA data or for fault detection tests), a single value of each κ_i is drawn from its distribution and then held constant throughout the trajectory. The PCA data consists of 1000 such trajectories. The initial condition is the nominal steady-state $\mathbf{x}_0 = (5180, 22500, 4000)$.

As discussed in §5, EKF is generally unusable for systems with time-invariant uncertain parameters. Thus, we only compare PCA and the set-based methods in this section. Moreover, we only consider the case where $\mathbf{w}_k$ and $\mathbf{v}_k$ obey truncated Gaussian distributions. Experiments with other distributions lead to the same conclusions already discussed in §5 and do not provide new insights.

The states of this system do not satisfy any known *a priori* enclosure G . Thus, to implement rDTDI, we manufacture invariants in the same way as in §5. Here, we define a single new state variable, $z_k = x_{1,k} + x_{2,k} + x_{3,k}$, and augment the original system with the redundant difference equation

$$z_{k+1} = z_k + h[u_{1,k} + u_{2,k} + u_{3,k} - \kappa_3 x_{3,k}].$$

Then, rDTDI is applied with

$$G = \{(\mathbf{x}, z) \in \mathbb{R}^4 : z = x_1 + x_2 + x_3\}. \quad (41)$$

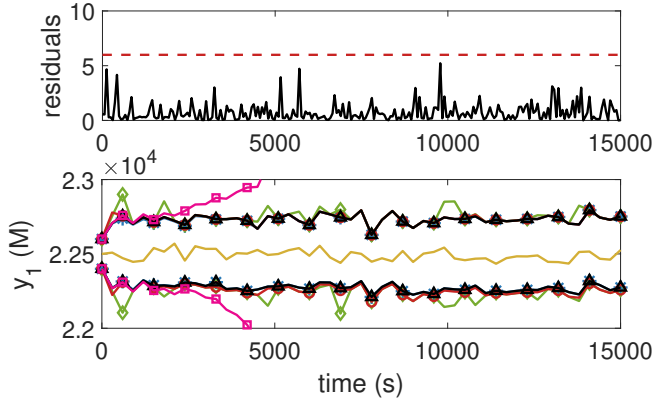


Figure 14: Results of the PCA (top) and set-based (bottom) methods (markers defined in Table I) for the sewer example in Scenario (a).

The fault-free and faulty scenarios studied for this example are listed in Table 4.

Table 4: Fault-free and faulty scenarios for the sewer system

Index	Scenario Description
(a)	(Fault Free) Normal operation.
(b)	(Fault Free) The initial condition is perturbed away from the steady state to $\mathbf{x}_0 = (5000, 20000, 3500)$.
(c)	(Fault Free) Starting from $t = 3000$, w_1 takes a constant value far from its mean but within its bounds: $w_1 = 0.08$
(d)	(Faulty) Starting from $t = 3000$, w_3 takes a constant value outside its bounds: $w_3 = 0.2$.
(e)	(Faulty) Starting from $t = 3000$, a leak happens in the second tank which leads to a change of dynamics: $x_{2,k+1} = x_{2,k} + h[\kappa_1 x_{1,k} - (\kappa_2 + 0.005) \sqrt{x_{2,k}}]$

Figure 14 shows the results under normal conditions (Scenario (a)). PCA does not give any false alarms in this case. This is attributed to the presence of long-term time-invariant parameters. Since these parameters take a range of different values in the historical data, but just a single value during the FD test, PCA expects a larger variance under normal operations than is actually observed (see further discussion in §5.5.2). The set-based methods also give no false alarms as usual. The rDTDI method again provides the tightest output bounds, followed closely by sDTDI and the zonotope methods. The standard interval method yields much weaker bounds. As in §5, the superior bound accuracy of rDTDI relative to sDTDI proves to be significant for fault detection in later scenarios, although it is hardly perceptible on the scale of Figure 14.

In Scenario (b), no fault occurs, but the system starts away from steady-state and operates transiently for a short time. Set-based methods are given the exact initial condition. Consistent with our findings in §5, PCA exhibits severe false alarms in this scenario, while the set-based methods handle the transient without issue, giving no false alarms. Figures are omitted for brevity.

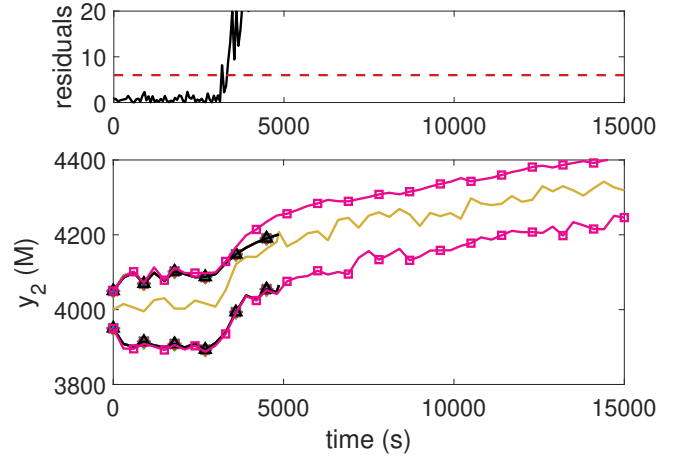


Figure 15: Results of the PCA (top) and set-based (bottom) methods (markers defined in Table I) for the sewer example in Scenario (d).

Scenario (c) considers a large, persistent disturbance. Specifically, all uncertainties obey their usual distributions until $t = 3000$, but afterwards w_1 takes a constant value that deviates significantly from its means but remains within its interval bounds. As discussed in detail in §5 (Scenario (c)), this is not considered a fault. Even so, the PCA residual increases sharply at $t = 3000$ and exceeds the threshold at $t = 3240$. In contrast, none of the set-based methods declare a fault, as expected. Figures are omitted for brevity.

Scenario (d) considers another large, persistent disturbance. Specifically, all uncertainties obey their usual distributions until $t = 3000$, but afterwards w_1 takes a constant value that falls outside of its interval bound. In contrast to Scenario (c), this scenario is considered a fault. Figure 15 shows that PCA detects the fault quickly at $t = 3300$, while rDTDI, sDTDI, and the two zonotope methods all detect it later at $t = 4800$. The standard interval method fails to detect the fault.

Scenario (e) considers a leak in the second tank starting at $t = 3000$. As detailed in Table 4, this fault changes the model rather than the disturbances. As shown in Figure 16, PCA detects the fault very quickly at $t = 3210$. The rDTDI method is the only set-based method able to detect the fault and does so at $t = 5850$.

8. Conclusions

In this paper, we introduced two new set-based fault detection (FD) algorithms based on the sDTDI and rDTDI set-based state estimators recently developed in (Yang and Scott, 2018a). We then studied the performance of these methods compared to both existing set-based FD methods and classical data-driven and observer-based FD methods using three detailed examples.

Among the set-based methods, rDTDI consistently provided the most accurate output bounds, followed by sDTDI. As a result, rDTDI also had the highest fault sensitivity, detecting most faults quickly and often being the only set-based method to detect the fault at all. The superiority of rDTDI was most pro-

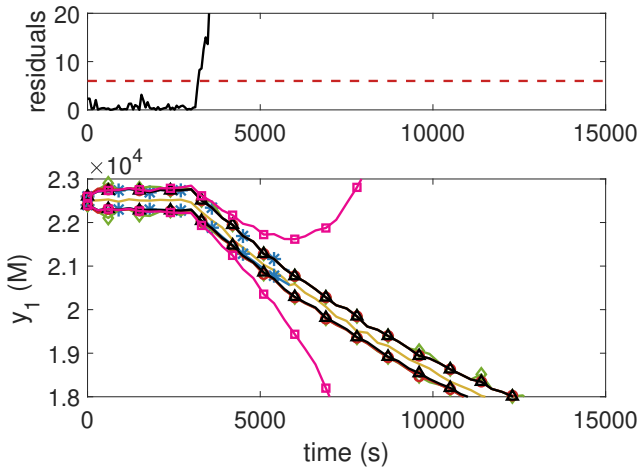


Figure 16: Results of the PCA (top) and set-based (bottom) methods (markers defined in Table I) for the sewer example in Scenario (e).

nounced when not all states were measured, as in the CSTR and sewer examples. Due to the high incidence of missed faults, we conclude that the other set-based methods were not accurate enough for effective fault detection in any of our examples. In contrast, the combination of DI with the use of model redundancy in rDTDI showed potential for significantly more effective set-based FD.

A critical weakness of the conventional PCA and EKF methods was the occurrence of frequent false alarms. Although the false alarm rates were reasonable in ideal circumstances, in several cases they were severe enough to render the methods unusable. For example, brief transients were found to cause severe false alarms in PCA, while the presence of time-invariant uncertain parameters caused them in EKF. Similarly, both PCA and EKF gave severe false alarms in the presence of persistent disturbances within the normal range. Since all of these situations occur commonly in practice, the inability to distinguish them from true faults that require intervention is a significant limitation. In contrast, all of these situations were handled easily by the set-based methods with no false alarms, provided that the bounds W and V were valid. Thus, the set-based approach showed significant potential to enable more robust fault detection for uncertain nonlinear systems.

On the other hand, in cases where false alarms were not prohibitive, PCA always detected faults the fastest, followed closely by EKF, and only then by rDTDI. Although rDTDI detected nearly all faults detected by PCA or EKF, the detection time was often significantly longer. It is presently unclear how much of this delay is due to inaccuracy in the rDTDI bounds and how much is inherent in the set-based approach (i.e., would persist even with exact bounds). Thus, while the fault sensitivity of rDTDI is competitive with classical methods, there is still substantial room for improvement.

Besides, we found that set-based approaches, including sDTDI and rDTDI, are not usable in their current form when the bounds W and V are not rigorous (i.e., they only contain w_k and v_k with high probability). This includes all cases where w_k

and v_k are in fact unbounded. In such cases, set-based methods are likely to give false alarms, which calls into question one of their main advantages. Moreover, general methods do not yet exist for restarting a set-based method once a false alarm occurs, since this requires a guaranteed enclosure of the current states. Since obtaining rigorous bounds W and V is difficult in practice, these issues must be addressed before the set-based approach can be effectively applied in many real applications.

Finally, we note that the case studies used here are relatively small. While data-driven methods have been successfully demonstrated on much larger systems (Chiang et al., 2000), observer-based and set-based methods have not, largely due to the difficulty in developing high-quality models. Moreover, it generally becomes more difficult to obtain accurate enclosures as system size increases, which is likely to lead to lower fault sensitivity for larger test cases. Thus, future research towards tighter enclosures is also needed to improve the applicability of set-based methods to larger systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRedit authorship contribution statement

Bowen Mu: Conceptualization, methodology, simulation, and original draft writing. **Xuejiao Yang:** Conceptualization, methodology, and simulation. **Joseph Scott:** Conceptualization, draft reviewing and editing, and supervision.

Acknowledgement

This material is based upon work supported by the National Science Foundation under grant number 1949748.

References

- Alamo, T., Bravo, J., Camacho, E., 2005. Guaranteed state estimation by zonotopes. *Automatica* 41, 1035–1043.
- Blesa, J., Puig, V., Saludes, J., 2012. Robust fault detection using polytope-based set-membership consistency test. *IET Control Theory Appl.* 6, 1767–1777.
- Bravo, J.M., Alamo, T., Camacho, E.F., 2006. Bounded error identification of systems with time-varying parameters. *IEEE Trans. Automat. Contr.* 51, 1144–1150.
- Chai, W., Qiao, J., Wang, H., 2013. Robust fault detection using set membership estimation and TS fuzzy neural network, in: *Proc. Am. Control Conf.*, IEEE. pp. 893–898.
- Chen, T.T., Adams, M., 1976. A sequential failure detection technique and its application. *IEEE Trans. Automat. Contr.* 21, 750–757.
- Chiang, L.H., Russell, E.L., Braatz, R.D., 2000. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media.
- Choi, S.W., Lee, C., Lee, J.M., Park, J.H., Lee, I.B., 2005. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemom. Intell. Lab. Syst.* 75, 55–67.
- Combastel, C., 2005. A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes, in: *Proc. IEEE Conf. Decis. Control*, pp. 7228–7234.

- Combastel, C., 2016. An extended zonotopic and Gaussian Kalman filter (EZGKF) merging set-membership and stochastic paradigms: Toward nonlinear filtering and fault detection. *Annu. Rev. Control* 42, 232–243.
- Efimov, D., Raïssi, T., Chebotarev, S., Zolghadri, A., 2013. Interval state observer for nonlinear time varying systems. *Automatica* 49, 200–205.
- Fathi, Z., Ramirez, W.F., Korbicz, J., 1993. Analytical and knowledge-based redundancy for fault diagnosis in process plants. *AIChE J.* 39, 42–56.
- Fernández-Cantí, R.M., Tornil-Sin, S., Blesa, J., Puig, V., 2013. Nonlinear set-membership identification and fault detection using a Bayesian framework: Application to the wind turbine benchmark, in: *Proc. IEEE Conf. Decis. Control*, pp. 496–501.
- Ingimundarson, A., Bravo, J.M., Puig, V., Alamo, T., Guerra, P., 2009. Robust fault detection using zonotope-based set-membership consistency test. *Int. J. Adapt. Control Signal Process.* 23, 311–330.
- Isermann, R., 2005. Model-based fault-detection and diagnosis—status and applications. *Annu. Rev. Control* 29, 71–85.
- Jauberthie, C., Verdière, N., Travé-Massuyès, L., 2013. Fault detection and identification relying on set-membership identifiability. *Annu. Rev. Control* 37, 129–136.
- Joe Qin, S., 2003. Statistical process monitoring: Basics and beyond. *J. Chemom.* 17, 480–502.
- Kieffer, M., Jaulin, L., Walter, E., 1998. Guaranteed recursive nonlinear state estimation using interval analysis, in: *Proc. IEEE Conf. Decis. Control*, pp. 3966–3971.
- Ku, W., Storer, R.H., Georgakis, C., 1995. Disturbance detection and isolation by dynamic principal component analysis. *Chemometr. Intell. Lab. Syst.* 30, 179–196.
- Kühn, W., 1998. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing* 61, 47–67.
- Le, V.T.H., Stoica, C., Alamo, T., Camacho, E.F., Dumur, D., 2013. Zonotopic guaranteed state estimation for uncertain systems. *Automatica* 49, 3418–3424.
- Lee, J.M., Qin, S.J., Lee, I.B., 2006. Fault detection and diagnosis based on modified independent component analysis. *AIChE J.* 52, 3501–3514.
- Lee, J.M., Qin, S.J., Lee, I.B., 2007. Fault detection of non-linear processes using kernel independent component analysis. *Can. J. Chem. Eng.* 85, 526–536.
- Meslem, N., Ramdani, N., 2020. A new approach to design set-membership state estimators for discrete-time linear systems based on the observability matrix. *Int. J. Control* 93, 2541–2550.
- Moore, R.E., Kearfott, R.B., Cloud, M.J., 2009. *Introduction to interval analysis*. SIAM.
- Patton, R.J., Chen, J., 1997. Observer-based fault detection and isolation: Robustness and applications. *Control Eng. Pract.* 5, 671–682.
- Raïssi, T., Videau, G., Zolghadri, A., 2010. Interval observer design for consistency checks of nonlinear continuous-time systems. *Automatica* 46, 518–527.
- Rego, B.S., Scott, J.K., Raimondo, D.M., Raffo, G.V., 2021. Set-valued state estimation of nonlinear discrete-time systems with nonlinear invariants based on constrained zonotopes. *Automatica* 129, 109638.
- Reppa, V., Tzes, A., 2011. Fault detection and diagnosis based on parameter set estimation. *IET Control Theory Appl.* 5, 69–83.
- Rostampour, V., Ferrari, R., Keviczky, T., 2017. A set based probabilistic approach to threshold design for optimal fault detection, in: *Proc. Am. Control Conf.*, IEEE. pp. 5422–5429.
- Scott, J.K., Barton, P.I., 2013. Bounds on the reachable sets of nonlinear control systems. *Automatica* 49, 93–100.
- Scott, J.K., Findeisen, R., Braatz, R.D., Raimondo, D.M., 2013. Design of active inputs for set-based fault diagnosis, in: *Proc. Am. Control Conf.*, IEEE. pp. 3561–3566.
- Scott, J.K., Raimondo, D.M., Marseglia, G.R., Braatz, R.D., 2016. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica* 69, 126–136.
- Seron, M.M., De Doná, J.A., 2009. Fault tolerant control using virtual actuators and invariant-set based fault detection and identification, in: *Proc. IEEE Conf. Decis. Control*, pp. 7801–7806.
- Shen, K., Scott, J.K., 2017. Rapid and accurate reachability analysis for nonlinear dynamic systems by exploiting model redundancy. *Comput. Chem. Eng.* 106, 596–608.
- Tabatabaeipour, S.M., Odgaard, P.F., Bak, T., Stoustrup, J., 2012. Fault detection of wind turbines with uncertain parameters: A set-membership approach. *Energies* 5, 2424–2448.
- Tornil-Sin, S., Ocampo-Martinez, C., Puig, V., Escobet, T., 2012. Robust fault detection of non-linear systems using set-membership state estimation based on constraint satisfaction. *Eng. Appl. Artif. Intell.* 25, 1–10.
- Tulsyan, A., Barton, P.I., 2016. Reachability-based fault detection method for uncertain chemical flow reactors. *IFAC-PapersOnLine* 49, 1–6.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003a. A review of process fault detection and diagnosis: Part III: Process history based methods. *Comput. Chem. Eng.* 27, 327–346.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003b. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Comput. Chem. Eng.* 27, 293–311.
- Wang, Y., Puig, V., 2016. Zonotopic extended Kalman filter and fault detection of discrete-time nonlinear systems applied to a quadrotor helicopter, in: *Conf. Control Fault-Toler. Syst. SysTol*, IEEE. pp. 367–372.
- Yang, X., Scott, J.K., 2018a. Accurate set-based state estimation for nonlinear discrete-time systems using differential inequalities with model redundancy, in: *Proc. IEEE Conf. Decis. Control*, pp. 680–685.
- Yang, X., Scott, J.K., 2018b. A comparison of zonotope order reduction techniques. *Automatica* 95, 378–384.
- Yang, X., Scott, J.K., 2018c. Efficient reachability bounds for discrete-time nonlinear systems by extending the continuous-time theory of differential inequalities, in: *Proc. Am. Control Conf.*, IEEE. pp. 6242–6247.
- Yang, X., Scott, J.K., 2020. Accurate uncertainty propagation for discrete-time nonlinear systems using differential inequalities with model redundancy. *IEEE Trans. Automat. Contr.* 65, 5043–5057.