

Fault Detection in Nonlinear Continuous-Time Systems with Uncertain Parameters

Youdong Lin* and Mark A. Stadtherr[†]

Department of Chemical and Biomolecular Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

(August 24, 2007)
(revised, February 20, 2008)

*Current address: LINDO Systems, Inc., 1415 North Dayton Street, Chicago, IL 60622, USA

[†]Author to whom all correspondence should be addressed. E-mail: markst@nd.edu

Abstract

In model-based fault diagnosis for dynamic systems with uncertain parameters, an envelope of all fault-free behaviors can be determined from the model and used as a reference for detecting faults. We demonstrate here a method for generating an envelope that is rigorously guaranteed to be complete, but without significant overestimation. The method is based on an interval approach, but uses Taylor models to reduce the overestimation often associated with interval methods. To speed fault detection, a method that uses bounded-error measurement data and a constraint propagation procedure is proposed for shrinking the envelope. Several fault detection scenarios involving nonlinear, continuous-time systems are used to evaluate this approach.

Introduction

Fault diagnosis is an important topic in modern control theory and practice. A fault may be defined as “a non permitted deviation of a characteristic property which leads to the inability to fulfill the intended purpose.”¹ Such a fault may disturb normal operation, thus reducing the performance of a system or even leading to breakdowns or catastrophic failure. There is an abundance of literature on process fault diagnosis, including detection and isolation, as reviewed recently by Venkatasubramanian et al.²⁻⁴ Methods can be classified into three general categories, statistical-based, knowledge-based, and analytical.⁵ In the last category, a common model-based method for fault detection is the analytical-redundancy (or model-reference) approach. In this case, a fault is detected by comparing the actual measured behavior of a system to a model-generated estimation of its normal (fault-free) behavior. If there are any discrepancies, expressed in terms of residuals, that exceed some threshold, then this indicates a fault. The model may be obtained from first principles or through use of identification techniques.⁶ Linearized models can be used, but may result in poor performance when the system is sufficiently nonlinear.⁷ Likewise, discrete-time difference equation models can be used, but for the demands of many real-world applications, continuous-time models are more realistic.⁸ Thus, our focus here is on using nonlinear, continuous-time models. While model development may require some effort, models used for fault detection potentially may be used in connection with the related problems of fault isolation and diagnosis, or for other problems involving process dynamics, and also provide general insight into process behavior.⁹

Since a model is rarely exact, normal behavior is generally represented by some range, or envelope, of values. A system is then determined to be faulty if any measurement goes outside of its envelope of normal values. The properties of the envelope are important in determining the performance of the fault detection process. An envelope is *complete* if it contains all possible normal behaviors. An envelope is *sound* if it contains only normal behaviors. An ideal (exact) envelope

would be both complete and sound. However, in practice, envelopes will be either overestimated (complete but not sound) or underestimated (sound but not complete). Clearly, the completeness and soundness of the envelope play a critical role in the robustness and sensitivity of the fault detection. If the envelope is overbounded, it could fail to detect some faults in a timely manner. On the other hand, if the envelope is underbounded, it may result in false alarms. It is generally considered preferable to have a complete, slightly overestimated envelope.¹⁰

Armengol et al.¹⁰ have reviewed several methods for envelope generation. One common type of approach is to use a model to obtain nominal values for normal behavior, and to then superimpose uncertainties in the form of a tolerance (or threshold), thus creating an envelope. The threshold may be either fixed or adaptive, but in either case it is difficult to guarantee completeness, leading to the use of overly large thresholds and a highly overbounded envelope. Another type of approach is to include uncertainties directly in the model, by regarding one or more model parameters as uncertain. A widely used method of this type is Monte Carlo simulation. The model is solved using a large number of different values for the uncertain parameters, and these are combined to create the envelope of normal behaviors. The Monte Carlo envelope is always underestimated (sound but not complete), but as the number of trials increases will approach the exact envelope. To obtain a complete envelope that is guaranteed, a number of methods based on interval analysis¹¹ have been explored. In this case, intervals are used to represent the uncertain model parameters. However, interval methods have a reputation of yielding highly overbounded envelopes. This is due¹⁰ primarily to the dependency (multi-incidence) problem, which is inherent in interval arithmetic, and the wrapping problem, which arises when interval calculations are done in state space. Both of these difficulties are discussed in more detail below. The work described here is motivated by the desire to obtain and use an envelope that is guaranteed to be complete, but without the large overestimations associated with interval methods. For obtaining this complete, slightly overestimated

envelope, we propose a completely new approach that combines traditional interval methods with the use of Taylor models.¹²

In this paper, we consider a model-reference approach for fault detection in systems described by nonlinear, ordinary differential equation (ODE) models. To account for uncertainties, the model parameters and initial states are taken to be interval valued. Intervals contain information about upper and lower bounds only; thus, in using intervals, no assumptions are made about the probability distribution of the uncertainties or about the independence or correlation of parameters. For obtaining a reference envelope of normal behavior, we use a new method¹³ for solving interval-valued initial-value problems (IVPs). This method is based on a traditional interval approach,¹¹ but uses Taylor models¹² to address overestimation issues. The envelope computed using this method is *rigorously guaranteed* to be complete, yet generally without significant overestimation. For using this reference envelope, two fault detection methods are described here. The first method makes direct use of the envelope, while in the second method, a constraint propagation procedure based on Taylor models is used, together with error-bounded measurements, to enable fault detection earlier than with the first method.

The remainder of this paper is organized as follows. In the next section, we present a mathematical statement of the problem to be solved. This is followed by a section that provides background on interval analysis and Taylor models, a section in which we summarize the method used for envelope generation, and a section in which we describe the fault detection methods that use this envelope-generation technique. Finally, we present example problems in which we demonstrate the effectiveness of these methods in fault detection, and compare their performance to a standard Monte Carlo simulation approach.

Problem Statement

Consider a nonlinear, continuous-time system represented by the following model:

$$\begin{aligned} \mathbf{x}' &= \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \boldsymbol{\theta}), \end{aligned} \tag{1}$$

where \mathbf{x} is the m -dimensional state vector, $\boldsymbol{\theta}$ is a p -dimensional time-invariant parameter vector, and \mathbf{y} is the r -dimensional output vector. Output measurements $\hat{\mathbf{y}}_j$ at $t = t_j$ are available with error $\mathbf{v}_j = \hat{\mathbf{y}}_j - \mathbf{y}_j$, where $\mathbf{y}_j = \mathbf{h}(\mathbf{x}_j, \boldsymbol{\theta})$ and $\mathbf{x}_j = \mathbf{x}(t_j)$. The initial states \mathbf{x}_0 are assumed to lie in a known interval \mathbf{X}_0 . The parameter vector $\boldsymbol{\theta}$ is assumed to be constant and to belong to a known interval $\boldsymbol{\Theta}$, which represents the “normal set” of parameter values for a fault-free system. The measurement error \mathbf{v}_j is bounded and assumed to belong to a known interval \mathbf{V}_j at each t_j . Therefore, the output vector \mathbf{y}_j belongs to a known box $\mathbf{Y}_j = \hat{\mathbf{y}}_j - \mathbf{V}_j$. The structure of the model, that is, the function $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$, is assumed to be known (if the model structure is not known with certainty, or if the model structure is poorly chosen, wider parameter intervals may be needed to fully capture normal behaviors). We assume that \mathbf{f} is $(k-1)$ -times continuously differentiable with respect to the state variables \mathbf{x} . Here k is the order of the truncation error in the interval Taylor series (ITS) method to be used in the integration procedure. We also assume that \mathbf{f} and \mathbf{h} are $(q+1)$ -times continuously differentiable with respect to the uncertain quantities (initial states \mathbf{x}_0 and parameters $\boldsymbol{\theta}$), where q is the order of the Taylor model to be used to represent dependence on these quantities.

A fault is defined to occur when one or more of the system parameters is no longer in its normal set. The envelope of normal system output, generated using the parametric model with the normal set of parameter values, describes the fault-free system behavior. Once faults occur, the output \mathbf{Y}_j will lie outside the boundary of the envelope, and then a fault is reported. The goals of fault detection are to report faults as soon as possible if they occur, and to avoid false alarms.

Background

The approach described here for fault detection is based on interval analysis and employs Taylor models. Thus, as background, we provide a brief summary of interval analysis and of Taylor models. Much more detail on both topics is available elsewhere.

Interval analysis

A real interval $X = [\underline{X}, \overline{X}]$ is defined by $X = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}$. We use an underline to indicate the lower bound of an interval and an overline to indicate the upper bound. A real interval vector $\mathbf{X} = (X_1, X_2, \dots, X_n)^T \subset \mathbb{R}^n$ has n real interval components and can be interpreted geometrically as an n -dimensional rectangle or box. Unless noted otherwise, we use uppercase to indicate intervals and lowercase (or uppercase with underline or overline) to indicate real numbers. Arithmetic operations with intervals are defined by $X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}$, where $\text{op} \in \{+, -, \times, \div\}$. Interval versions of the elementary functions can be similarly defined. For dealing with exceptions, such as division by an interval containing zero, extended models for interval arithmetic are available, often based on the extended real system $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$. The concept of containment sets (csets) provides a valuable framework for constructing models for interval arithmetic with consistent handling of exceptions.^{14,15} Implementations of interval arithmetic and elementary functions are readily available, and recent compilers from Sun Microsystems directly support interval arithmetic and an interval data type.

For an arbitrary function $f(\mathbf{x})$, the interval extension $F(\mathbf{X})$ encloses the range of $f(\mathbf{x})$ over \mathbf{X} . It is often computed by substituting \mathbf{X} into $f(\mathbf{x})$ and then evaluating the function using interval arithmetic. This “natural” interval extension may be wider than the actual range of function values, though it always includes the actual range. This overestimation of the function range is due to the dependency (or multi-incidence) problem, which may arise when a variable occurs more than once in

a function expression. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not recognized when interval arithmetic is used. In effect, when interval arithmetic is used, the range computed for the function is the range that would occur if each instance of a particular variable were allowed to take on a *different* value in its interval range. For the case in which $f(\mathbf{x})$ is a single-use expression, that is, an expression in which each variable occurs only once, interval arithmetic will always yield the true function range. For more general situations, there are a variety of other approaches that can be used to try to tighten interval extensions, including the use of Taylor models, as described in the next subsection. Several good introductions to interval analysis, including interval arithmetic and other aspects of computing with intervals, are available.^{14,16–18}

Taylor models

Makino and Berz¹² have described a remainder differential algebra (RDA) approach for bounding function ranges and controlling the dependency problem of interval arithmetic.¹⁹ This method is based on representing a function with a model consisting of a Taylor polynomial and an interval remainder bound. Based on a Taylor expansion about the point $\mathbf{x}_0 \in \mathbf{X} \subset \mathbb{R}^n$, the q -th order Taylor model T_f of $f(\mathbf{x})$ consists of a q -th order polynomial function in $(\mathbf{x} - \mathbf{x}_0)$, p_f , and an interval remainder bound R_f , such that $f \in T_f = p_f + R_f$ for all $\mathbf{x} \in \mathbf{X}$. The function f is then bounded by seeking bounds on the Taylor model T_f , which is also denoted $T_f = (p_f, R_f)$.

Arithmetic operations with Taylor models can be done using the remainder differential algebra described by Makino and Berz,^{12,19,20} which includes addition, multiplication, reciprocal, and intrinsic functions. Using these, it is possible to start with simple functions such as the constant function $f(x) = k$, for which $T_f = (k, [0, 0])$, and the identity function $f(x_i) = x_i$, for which $T_f = (x_{i0} + (x_i - x_{i0}), [0, 0])$, and then to compute Taylor models for very complicated functions.

Therefore, by using simple operator overloading with RDA operations, it is possible to compute a Taylor model for any function representable in a computer environment.

An interval bound on a Taylor model $T = (p, R)$ over \mathbf{X} is denoted by $B(T)$ and is given by $B(T) = B(p) + R$, where $B(p)$ is an interval bound on the polynomial part p . The range bounding of the polynomial $B(p) = P(\mathbf{X} - \mathbf{x}_0)$ is an important issue, which directly affects the performance of Taylor model methods. However, exact range bounding of an interval polynomial is NP hard, and direct evaluation using interval arithmetic is very inefficient, often yielding only loose bounds. Various bounding schemes^{21–23} have been used, mostly focused on exact bounding of the first- and second-order polynomial terms. However, exact bounding of a general interval quadratic is also computationally expensive (in the worst case, exponential in the number of variables). Thus, in our implementation of Taylor models,^{13,24,25} we have used a compromise approach, in which only the first-order and the diagonal second-order terms are considered for exact bounding, and other terms are evaluated directly. That is,

$$B(p) = \sum_{i=1}^n \left[a_i (X_i - x_{i0})^2 + b_i (X_i - x_{i0}) \right] + Q, \quad (2)$$

where we seek to bound the summation exactly, with Q , the interval bound of all other terms, obtained by direct evaluation with interval arithmetic. In the summation, since X_i occurs twice, there is a dependency problem that appears to prevent the use of interval arithmetic to obtain exact bounds. However, for $|a_i| \geq \omega$, where ω is a small positive number, we can rearrange Eq. (2) so that each X_i occurs only once in the summation; that is,

$$B(p) = \sum_{i=1}^n \left[a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} \right] + Q. \quad (3)$$

Using Eq. (3), the exact bounds on the summation can be obtained using interval arithmetic. If $|a_i| < \omega$, evaluation with Eq. (2) is used instead. It has been shown that, compared to other rigorous bounding methods, the Taylor model often yields sharper bounds for modest to complicated

functional dependencies.^{12,19,23}

Method for Envelope Generation

As noted previously, in model-based fault detection an envelope of normal states may be used, and there are various methods for envelope generation.¹⁰ It is desired to obtain an envelope that is complete, so that there is no possibility of false alarms, but with little overestimation, so that faults are detected quickly. To generate such an envelope for nonlinear continuous-time systems, we need an IVP solver for nonlinear ODEs that can compute rigorous bounds on the state variables \boldsymbol{x} for the case in which the initial values and parameters are given by intervals. Interval methods (also called validated methods or verified methods) for ODEs provide a natural approach for computing the desired enclosure of the state variables.

Traditional interval methods¹¹ usually consist of two processes applied at each integration step. In the first process, existence and uniqueness of the solution are proved using the Picard-Lindelöf operator and the Banach fixed point theorem, and a rough enclosure of the solution is computed. In the second process, a tighter enclosure of the solution is computed. In general, both processes are realized by applying interval Taylor series (ITS) expansions with respect to time, and using automatic differentiation to obtain the Taylor coefficients. An excellent review of the traditional interval methods has been given by Nedialkov et al.,²⁶ and more recent work has been reviewed by Neher et al.²⁷ For addressing this problem, there are various packages available, including AWA,²⁸ VNODE^{29,30} and COSY VI,³¹ all of which consider uncertainties (interval valued) in initial values only. In this study, we will use a new validated solver¹³ for parametric ODEs, which is used to produce guaranteed bounds on the solutions of dynamic systems with interval-valued initial states and parameters. The method makes use, in a novel way, of the Taylor model approach^{12,19,20} to deal with the dependency problem on the uncertain variables (parameters and initial values). We

will summarize here the basic ideas of the method used.

The ODE problem of interest is

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbf{X}_0, \quad \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad (4)$$

where $t \in [t_0, t_N]$ for some $t_N > t_0$, and \mathbf{X}_0 and $\boldsymbol{\Theta}$ represent enclosures of initial values and parameters, respectively, that correspond to normal, fault-free behavior. It is desired to determine a verified enclosure of all possible solutions to this IVP. We denote by $\mathbf{x}(t; t_j, \mathbf{X}_j, \boldsymbol{\Theta})$ the set of solutions $\{\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \mid \mathbf{x}_j \in \mathbf{X}_j, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta})$ denotes a solution of $\mathbf{x}' = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ for the initial condition $\mathbf{x} = \mathbf{x}_j$ at t_j . We will summarize a method for determining enclosures \mathbf{X}_j of the state variables at each time step $j = 1, \dots, N$, such that $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \boldsymbol{\Theta}) \subseteq \mathbf{X}_j$.

Assume that at t_j we have an enclosure \mathbf{X}_j of $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \boldsymbol{\Theta})$, and that we want to carry out an integration step to compute the next enclosure \mathbf{X}_{j+1} . Then, in the first phase of the method, the goal is to find a step size $h_j = t_{j+1} - t_j > 0$ and a rough enclosure $\widetilde{\mathbf{X}}_j$ of the solution such that a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \in \widetilde{\mathbf{X}}_j$ is guaranteed to exist for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. We apply a traditional interval method, with high order enclosure, to the parametric ODEs by using an interval Taylor series (ITS) with respect to time. That is, we determine h_j and $\widetilde{\mathbf{X}}_j$ such that for $\mathbf{X}_j \subseteq \widetilde{\mathbf{X}}_j^0$,

$$\widetilde{\mathbf{X}}_j = \sum_{i=0}^{k-1} [0, h_j]^i \mathbf{F}^{[i]}(\mathbf{X}_j, \boldsymbol{\Theta}) + [0, h_j]^k \mathbf{F}^{[k]}(\widetilde{\mathbf{X}}_j^0, \boldsymbol{\Theta}) \subseteq \widetilde{\mathbf{X}}_j^0. \quad (5)$$

Here k denotes the order of the Taylor series, and the coefficients $\mathbf{F}^{[i]}$ are interval extensions of the Taylor coefficients $\mathbf{f}^{[i]}$ of $\mathbf{x}(t)$ with respect to time, which can be obtained recursively in terms of $\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$. When Eq. (5) is satisfied, it demonstrates³² that there exists a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \in \widetilde{\mathbf{X}}_j$ for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

In the second phase of the method, we compute a tighter enclosure $\mathbf{X}_{j+1} \subseteq \widetilde{\mathbf{X}}_j$, such that $\mathbf{x}(t_{j+1}; t_0, \mathbf{X}_0, \boldsymbol{\Theta}) \subseteq \mathbf{X}_{j+1}$. This will be done by using an ITS approach to compute $\mathbf{T}_{\mathbf{x}_{j+1}}(\mathbf{x}_0, \boldsymbol{\theta})$,

a Taylor model of \mathbf{x}_{j+1} in terms of the initial values \mathbf{x}_0 and parameters $\boldsymbol{\theta}$, and then obtaining the enclosure $\mathbf{X}_{j+1} = B(\mathbf{T}_{\mathbf{x}_{j+1}})$. For the Taylor model computations, we begin by representing the interval initial states and parameters by the Taylor models (identity functions) $\mathbf{T}_{\mathbf{x}_0}$ and $\mathbf{T}_{\boldsymbol{\theta}}$, respectively. Then, we can determine Taylor models $\mathbf{T}_{\mathbf{f}^{[i]}}$ of the interval Taylor series coefficients $\mathbf{f}^{[i]}(\mathbf{x}_j, \boldsymbol{\theta})$ by using RDA operations to compute $\mathbf{T}_{\mathbf{f}^{[i]}} = \mathbf{f}^{[i]}(\mathbf{T}_{\mathbf{x}_j}, \mathbf{T}_{\boldsymbol{\theta}})$. Using an interval Taylor series for \mathbf{x}_{j+1} with coefficients given by $\mathbf{T}_{\mathbf{f}^{[i]}}$, and incorporating a novel approach for using the mean value theorem on Taylor models, one can obtain $\mathbf{T}_{\mathbf{x}_{j+1}}(\mathbf{x}_0, \boldsymbol{\theta})$, the desired Taylor model of \mathbf{x}_{j+1} in terms of the parameters $\boldsymbol{\theta}$ and initial states \mathbf{x}_0 . It is also necessary to address the wrapping effect.¹¹ This occurs because the set $\mathbf{x}(t_{j+1}; t_0, \mathbf{X}_0, \boldsymbol{\Theta})$ of solutions that we seek to enclose is rarely an interval. When this set is wrapped in an interval, overestimation occurs. If interval enclosures are used to propagate solution ranges from one time step to the next, this overestimation is also propagated and can grow rapidly. Thus, to control the wrapping effect, the state enclosures are propagated using a new type of Taylor model consisting of a polynomial and a parallelepiped remainder bound. Complete details of the computation of $\mathbf{T}_{\mathbf{x}_{j+1}}$ are given by Lin and Stadtherr.¹³

An implementation of this approach, called VSPODE (Validating Solver for Parametric ODEs), has been developed and tested by Lin and Stadtherr,¹³ who compared its performance with results obtained using the popular VNODE package.^{29,30} For the test problems used, VSPODE provided tighter enclosures on the state variables than VNODE, and required significantly less computation time. The examples presented below provide further evidence that this approach can provide a rigorously complete envelope with relatively little overestimation. Information about the availability of VSPODE can be obtained by contacting the authors.

Fault Detection Methods

Two methods for model-based fault detection are considered. The first is a simple method that requires envelope generation only. The second is more complex, and involves the use of bounded-error output measurements in a constraint propagation scheme to shrink the envelope of normal outputs and accelerate fault detection.

Method 1

This is a basic method in which direct comparison is made to the computed envelope of normal outputs. Using VSPODE for envelope generation, as described in the previous section, with the normal sets of parameter values and initial states, Taylor models $\mathbf{T}_{\mathbf{x}_j}(\mathbf{x}_0, \boldsymbol{\theta})$ of the state variables at each time step j can be determined. Using Taylor model operations, we can then compute Taylor models of the outputs $\mathbf{T}_{\mathbf{y}_j} = \mathbf{h}(\mathbf{T}_{\mathbf{x}_j}, \mathbf{T}_{\boldsymbol{\theta}})$ at each time step j . The envelope of normal outputs is then determined from $B(\mathbf{T}_{\mathbf{y}_j})$, $j = 1, \dots, N$; that is, by bounding $\mathbf{T}_{\mathbf{y}_j}$ over the normal sets $\boldsymbol{\theta} \in \Theta$ and $\mathbf{x}_0 \in \mathbf{X}_0$. This envelope can then serve as a reference of fault-free system behaviors. If at some time step j , the bounded-error measurement Y_{ij} of any output component i lies entirely outside of the envelope ($B(T_{y_{ij}}) \cap Y_{ij} = \emptyset$), then a fault has been detected, and an alarm can be triggered. Since the enclosure is rigorously guaranteed to include all possible solutions of the nonlinear ODE system, the envelope is complete. Therefore, false alarms are effectively eliminated. In this method, however, the measurement information has not been thoroughly utilized.

Method 2

In this method, the output measurements are used to accelerate the fault detection process. The measurements provide information that can be used to reduce, at each time step, the uncertainties in the model parameters and initial states, and to identify correlations between parameters if they

exist.³³ That is, information provided by the measurements is used to shrink Θ and \mathbf{X}_0 . With the reduction of these uncertainties, a better (tighter) envelope can be obtained, thus resulting in earlier detection of faults. If no faults have occurred, then at each measurement time t_j , the predicted fault-free output should be consistent with the error-bounded measurement. That is, $\mathbf{y}_j \in \mathbf{Y}_j$. This condition can be satisfied (conservatively) by imposing the constraint $B(\mathbf{T}_{\mathbf{y}_j}) \subseteq \mathbf{Y}_j$. Using this in a constraint propagation procedure, as described next, parts of Θ and \mathbf{X}_0 that are incompatible with the measurements can be eliminated. If any component of either Θ or \mathbf{X}_0 is ever completely eliminated, then it means that the predicted fault-free outputs and the measurements have become completely incompatible, indicating detection of a fault.

We will now describe a constraint propagation procedure (CPP) for using the constraint $B(\mathbf{T}_{\mathbf{y}_j}) \subseteq \mathbf{Y}_j$ to reduce the uncertain quantities Θ and \mathbf{X}_0 . In this discussion it is convenient to denote $\mathbf{Z} = (\mathbf{X}_0, \Theta)^T \subset \mathbb{R}^n$, with $\mathbf{z} = (\mathbf{x}_0, \boldsymbol{\theta})^T \in \mathbf{Z}$ and $n = m + p$. Constraint propagation is widely used in various forms (e.g., hull consistency) in connection with interval methods.^{14,16} Since it is implemented using interval arithmetic, application of the CPP does not affect the completeness of the envelope. Thus this method will not cause false alarms. We have found similar constraint propagation strategies to be useful in other contexts.^{24,25} The procedure is described here in the context of the fault detection problem.

Consider output component i at time step j , and say that the Taylor model $T_{y_{ij}}(\mathbf{z})$ has been determined using VSPODE for $\mathbf{z} \in \mathbf{Z}$, and that the bounded-error output measurement Y_{ij} is also available. We want to use the constraint $B(T_{y_{ij}}) \subseteq Y_{ij}$ to determine if any part of \mathbf{Z} is incompatible with the measurement and thus can be eliminated to shrink \mathbf{Z} . We first compute $B(T_{y_{ij}})$ and check to see if either $\overline{B(T_{y_{ij}})} < \underline{Y_{ij}}$ or $\underline{B(T_{y_{ij}})} > \overline{Y_{ij}}$. If so, then clearly $B(T_{y_{ij}}) \not\subseteq Y_{ij}$ and no part of the current \mathbf{Z} is consistent with the measurement. Thus the predicted fault-free output is incompatible with the measurement and a fault has been detected. Doing these checks is equivalent to Method

1.

In Method 2, we continue the constraint propagation by first expressing $B(T_{y_{ij}})$ in the form of Eq. (3). For some component k of \mathbf{Z} , and assuming $|a_k| \geq \omega$, this leads to

$$B(T_{y_{ij}}) = a_k \left(Z_k - z_{k0} + \frac{b_k}{2a_k} \right)^2 - \frac{b_k^2}{4a_k} + S_k, \quad (6)$$

where

$$S_k = \sum_{\substack{l=1 \\ l \neq k}}^n \left[a_l \left(Z_l - z_{l0} + \frac{b_l}{2a_l} \right)^2 - \frac{b_l^2}{4a_l} \right] + Q. \quad (7)$$

Here $z_{k0} \in Z_k$ and is usually the midpoint $z_{k0} = m(Z_k)$; the value of z_{k0} will not change during the CPP. We can reduce the computational effort to obtain S_k by recognizing that this quantity is just $B(T_{y_{ij}})$ less the k -th term in the summation, and $B(T_{y_{ij}})$ was already computed earlier in the CPP. Thus, for each k , S_k can be determined by dependent subtraction using

$$S_k = B(T_{y_{ij}}) \ominus \left[a_k \left(Z_k - z_{k0} + \frac{b_k}{2a_k} \right)^2 - \frac{b_k^2}{4a_k} \right]. \quad (8)$$

The dependent subtraction operation for intervals is defined by $C \ominus A = [\underline{C} - \underline{A}, \overline{C} - \overline{A}]$ and is valid if C depends additively on A . Now define the intervals $U_k = Z_k - z_{k0} + \frac{b_k}{2a_k}$ and $V_k = \frac{b_k^2}{4a_k} - S_k$, so that $B(T_{y_{ij}}) = a_k U_k^2 - V_k$. The goal is to identify and retain only the part of Z_k that contains values of z_k for which it is *possible* to satisfy the constraint $B(T_{y_{ij}}) = a_k U_k^2 - V_k \subseteq Y_{ij}$, or $U_k^2 \subseteq W_k$, with $W_k = (V_k + Y_{ij})/a_k$. This corresponds to the requirement that

$$\overline{U_k^2} \leq \overline{W_k} \text{ and } \underline{U_k^2} \geq \underline{W_k}. \quad (9)$$

The set U_k that satisfies Eq. (9) can be determined to be

$$U_k = \begin{cases} \emptyset & \text{if } \overline{W_k} < 0 \\ [-\sqrt{\overline{W_k}}, \sqrt{\overline{W_k}}] & \text{if } \underline{W_k} \leq 0 \leq \overline{W_k} \\ -\sqrt{\overline{W_k}} \cup \sqrt{\overline{W_k}} & \text{if } \underline{W_k} > 0. \end{cases} \quad (10)$$

The part of Z_k to be retained is then $Z_k = Z_k \cap \left(U_k + z_{k0} - \frac{b_k}{2a_k} \right)$. Only a part of Z_k in which the constraint is guaranteed not to be satisfied has been eliminated.

If $|a_k| < \omega$ and $|b_k| \geq \omega$, then Eq. (3) should not be used, but Eq. (2) can be used instead. Following a procedure similar to that used above, we now have $B(T_{y_{ij}}) = b_k U_k - V_k$ with $U_k = Z_k - z_{k0}$ and $V_k = -(B(T_{y_{ij}}) \ominus b_k(Z_k - z_{k0}))$. Note that all quadratic terms are now included in V_k . In identifying bounds on the part of Z_k in which it is possible to satisfy the constraint, the set U_k can be determined to be $U_k = (V_k + Y_{ij})/b_k$. The part of Z_k to be retained is then $Z_k = Z_k \cap (U_k + z_{k0})$. If both $|a_k|$ and $|b_k|$ are less than ω , then no CPP will be applied on Z_k .

The overall CPP is implemented by beginning with $k = 1$ and proceeding component by component. If, for any k , the result $Z_k = \emptyset$ is obtained, then there is an inconsistency between the predicted fault-free output and the measurement. This means that a fault has been detected and that the CPP can be stopped. Otherwise the CPP proceeds until all components of \mathbf{Z} have been updated. Note that, in principle, each time an improved (smaller) Z_k is found, it could be used in computing S_k for subsequent components of \mathbf{Z} . However, this requires recomputing the bound $B(T_{y_{ij}})$, which is expensive. Thus, the CPP for each component is done using the bounds $B(T_{y_{ij}})$ computed from the original \mathbf{Z} . If, after each component is processed, \mathbf{Z} has been sufficiently reduced (by more than 10% by volume), then a new bound $B(T_{y_{ij}})$ is obtained, now over the smaller \mathbf{Z} , and a new CPP is started. Otherwise, the CPP terminates.

Method 2 can now be summarized as follows: Beginning at $t = 0$, with given initial state interval \mathbf{X}_0 and parameter interval Θ , and time step counter set at $j = 1$,

1. Integrate the system using VSPODE and obtain the Taylor model $T_{\mathbf{x}_j}(\mathbf{x}_0, \theta)$ of the states at $t = t_j$.
2. Calculate the Taylor models $T_{\mathbf{y}_j}(\mathbf{x}_0, \theta)$ of the outputs.

3. Perform the CPP using the constraint $B(\mathbf{T}_{\mathbf{y}_j}) \subseteq \mathbf{Y}_j$ to try to shrink \mathbf{X}_0 and Θ .
 - (a) If any component of either \mathbf{X}_0 and Θ is eliminated in the CPP, declare a FAULT at t_j .
 - (b) Otherwise, continue.
4. Set $j = j + 1$ and go to Step 1 to do the next time step in the integration (using the updated \mathbf{X}_0 and Θ).

Examples

We now consider fault detection in several scenarios involving three different nonlinear systems: a system of three coupled tanks, an electromechanical positioning system, and an exothermic batch reactor with cooling jacket. This is a diverse set of examples involving different types of nonlinearities. In addition to using the new fault detection methods described here, we also use a standard Monte Carlo simulation (MCS) approach to provide a basis for comparison. In the MCS method, an envelope of normal outputs is generated by solving Eq. (1) many times using initial states \mathbf{x}_0 randomly selected from \mathbf{X}_0 and parameters θ randomly selected from Θ . If at some time step j , the bounded-error measurement Y_{ij} of any output component i lies entirely outside of the MCS envelope, a fault is declared. The MCS envelope is sound but not complete, thus this method is subject to false alarms, though if a sufficiently large number of simulations is done, the risk of false alarms can be made quite small. In the examples here, the MCS envelope is determined using 100 simulations. For Methods 1 and 2, when VSPODE is used for envelope generation, this is done using ITS order $k = 17$ and Taylor model order $q = 3$.

Three coupled tanks

As the first example problem with which to demonstrate the fault detection methods suggested here, we use a system of three coupled tanks.³⁴ As shown in Fig. 1, this is a nonlinear system consisting of three cylindrical tanks which are coupled through connecting pipes of circular cross-section. Tank 1 has an incoming flow Q_1 which is controlled by a pump, and the outflow is from Tank 3. All three liquid levels h_1 , h_2 and h_3 are measured. Sainz et al.³⁴ also use this example to demonstrate an interval-based fault detection method. However, their method requires that the model be treated as a set of discrete-time difference equations. Using the new approach described here, we can directly use the underlying continuous-time ODE model.

The dynamical model of the system is derived using material balances on each tank. Assuming constant fluid density and using Torricelli's Law, this gives:

$$\begin{aligned} A_1 h_1' &= Q_1 - c_1 \sqrt{h_1 - h_2} - l_1 \sqrt{h_1} \\ A_2 h_2' &= c_1 \sqrt{h_1 - h_2} - c_2 \sqrt{h_2 - h_3} - l_2 \sqrt{h_2} \\ A_3 h_3' &= c_2 \sqrt{h_2 - h_3} - c_3 \sqrt{h_3} - l_3 \sqrt{h_3} \end{aligned} \quad (11)$$

where the A_i , $i = 1, 2, 3$, are the cross-sectional areas of the tanks, and the c_i and l_i , $i = 1, 2, 3$, denote the valve constants and leakage constants, respectively. Here, it is assumed that $h_1 \geq h_2 \geq h_3$. The output variables y_i , $i = 1, 2, 3$, are the state variables (liquid levels).

The value of the incoming flow is available as an interval corresponding to an imprecise measurement; that is, $Q_1 \in [0.00495, 0.00505]$ m³/s. Output measurements (fluid levels) \hat{y}_i , $i = 1, 2, 3$, are available, but with noise bounded by $V_i = [-0.01, 0.01]$ m, $i = 1, 2, 3$. Thus, the true fluid levels y_i are enclosed by $y_i \in Y_i = \hat{y}_i - V_i$, $i = 1, 2, 3$. All areas are assumed to be known exactly, and to have the values $A_1 = A_2 = A_3 = 1$ m². In a fault-free system, the valve constants are only known to belong to the interval $c_i \in [0.0099, 0.0101]$ m^{5/2}/s, $i = 1, 2, 3$. It is also assumed that the system is normal if the leakage constants are less than 0.0001; that is, the normal set for these parameters

is $l_i \in [0, 0.0001] \text{ m}^{5/2}/\text{s}$, $i = 1, 2, 3$. The initial liquid levels are also not known precisely, and are given by $h_{10} \in [0.59, 0.61] \text{ m}$, $h_{20} \in [0.49, 0.51] \text{ m}$ and $h_{30} \in [0.39, 0.41] \text{ m}$.

VSPODE was used, with the interval-valued parameters and initial states given above, to generate the envelope of normal outputs. This envelope is shown by the solid curves in Fig. 2. It is rigorously guaranteed to be complete, but is not sound. To get some indication of how much this envelope overestimates the true range of normal outputs, we also determined an envelope using MCS. This provides a sound, but not complete, envelope, which is shown by the shaded area in Fig. 2, and which is an underestimation of the true range of normal outputs. Table 1 provides a more quantitative comparison of the two envelopes at selected values of time. Comparison of the VSPODE envelope and the Monte Carlo envelope suggests that by using VSPODE we were able to obtain a rigorously guaranteed complete envelope with very little overestimation.

We consider here four fault scenarios involving clogging (decrease of a valve constant) and/or leaking (increase of a leakage constant). In all scenarios, the fault occurs at $t = 0$. A set of measurement data for each scenario was generated by simulation using the nominal (midpoint) values of the parameters and initial states, and then adding measurement noise, which was a random number between -0.01 and $+0.01 \text{ m}$.

In the first scenario, some clogging occurs between tanks 2 and 3, which decreases the valve constant. Specifically, the valve constant is reduced to $c_2 = 0.008 \text{ m}^{5/2}/\text{s}$. Fig. 3 shows the fault detection results for the clogging scenario. At $t = t_j$, each bounded-error measurement Y_{ij} is shown by a vertical bar. Two envelopes are shown for each output. The larger envelope is the same as shown in Fig. 2 and was generated by VSPODE without use of the CPP. This envelope is used in fault detection Method 1. It can be seen that the level measurement in tank 3 falls completely outside of this envelope at $t = 26 \text{ s}$, thus triggering an alarm. The alarm would be triggered at $t = 21 \text{ s}$ (not shown in the Figure) if the incomplete MCS envelope is used as the reference for

normal outputs. The smaller (tighter) envelope is the envelope generated using VSPODE with the CPP, and is the envelope used for fault detection Method 2. Now, at $t = 12$ s, we see that the envelope for the level in tank 2 has become empty. This occurs because in the CPP it has been determined that there is an inconsistency between the measurements and the normal set of parameter values, thus triggering an alarm. By using the CPP and taking advantage of real-time measurement data, the speed of the fault detection has been significantly improved. The average CPU time for fault detection at each time step (1 s) in Method 2 was 0.08 s (Intel P4 3.2 GHz machine). For this scenario, and all the scenarios considered for this and the other example systems, the computation time required for fault detection with Method 2 was much less than the sampling period, thus allowing real-time implementation.

In the second scenario, excessive leakage occurs from tank 2, which increases the leakage constant. Specifically, the leakage constant is increased to $l_2 = 0.001 \text{ m}^{5/2}/\text{s}$. Fig. 4 shows the fault detection results for the excessive leakage scenario. Again, the larger envelope is the same as shown in Fig. 2 and was generated by VSPODE without use of the CPP (Method 1). For this case, the fault was detected at $t = 45$ s, when the level measurement for tank 2 falls entirely below the envelope. The fault is detected at $t = 20$ s (not shown) if the incomplete Monte Carlo envelope is used. The tighter envelope comes from VSPODE with the CPP (Method 2). This leads to quicker fault detection, at $t = 17$ s.

For the third scenario, we allowed both of these clogging and leaking faults to occur at the same time ($t = 0$). For this scenario, a fault was reported at $t = 20$ s using either Method 1 or the MCS method, and at $t = 17$ s using Method 2. Detailed results are not shown for this case. Because the measurement data used have random noise, and thus are not the same as used in connection with either of the first two scenarios, these performance results are not directly comparable to the earlier results.

In the final fault scenario, there is an incipient leak from tank 2 that begins at $t = 0$ and then slowly increases with time. Specifically, the leakage constant is assumed to be $l_2 = 0.00005 + 0.00002t$. Fig. 5 shows the fault detection results for this scenario. Again the larger envelope corresponds to Method 1, and the smaller envelope to Method 2. A fault was reported at $t = 56$ s using Method 1 and at $t = 51$ s using MCS (not shown). For Method 2, the fault was detected much sooner, at $t = 30$ s. Results on this and the other scenarios indicate that Method 2 provides significantly faster fault detection than Method 1, which is not surprising since Method 2 uses real-time measurements and Method 1 does not. Both Methods 1 and 2 are based on rigorously guaranteed complete envelopes of normal output, and thus are not susceptible to false alarms. The MCS method can provide earlier alarms than Method 1, but it is based on an incomplete (underestimated) envelope, with an accompanying risk of false alarms, and this must be considered to make a fair comparison.

Electromechanical positioning system

We consider an electromechanical positioning system³⁵ in which a bar with length l and mass m is positioned (rotated) in a vertical plane using a DC motor with mechanical damping. A spring is attached to the bar to provide additional stiffness. This system is governed by the following second order system:

$$J\theta'' + f\theta' + \zeta\theta + mgl \sin(\theta) = k_a u. \quad (12)$$

Here θ is the angular displacement of the motor shaft, $u = A \sin(\omega t)$ is a sinusoidal voltage applied to the DC motor armature through an electronic power amplifier with gain k_a , J is the joint inertia of the bar and the motor armature, f is the viscous damping coefficient of the mechanical damper, ζ is the spring stiffness constant and g is the acceleration due to gravity. The system can be rewritten

as a first order autonomous ODE system

$$\begin{aligned}
\xi_1' &= \xi_2 \\
\xi_2' &= -b\xi_2 - c\xi_1 - k_1 \sin(\xi_1) + A_k \sin(\omega\xi_3) + w(\xi_3) \\
\xi_3' &= 1
\end{aligned} \tag{13}$$

with states $\xi_1 = \theta$, $\xi_2 = \theta'$, $\xi_3 = t$, and parameters $b = f/J \in [0.95, 1.05] \text{ s}^{-1}$, $c = \zeta/J \in [0.39, 0.41] \text{ s}^{-2}$, $k_1 = mgl/J = 0.1 \text{ s}^{-2}$, $A_k = Ak_a/J = 0.5 \text{ s}^{-2}$ and $\omega = 0.25\pi \text{ s}^{-1}$. The additional term $w(\xi_3)$ accounts for the faults that will be considered for this system. For fault-free behavior, $w = 0$. The initial conditions are $(\xi_1, \xi_2, \xi_3)_0^T = (0, 0, 0)^T$. The output is $y = \xi_1$ and is measured every 0.2 s, but subject to noise bounded by $V = [-0.05, 0.05]$.

The envelope of normal outputs generated using VSPODE for Method 1 is shown by the solid curves in Fig. 6. An envelope was also generated using MCS, and this is shown as the shaded area in Fig. 6. Table 2 gives a comparison of the two envelopes at selected values of time. Considering that the MCS envelope is an underestimate of the true envelope of normal outputs, it appears that VSPODE was able to determine a rigorously complete envelope with very little overestimation.

Two fault scenarios are considered: 1. There is a malfunctioning of the power electronics resulting in an extraneous constant voltage being applied to the motor. In this case w will be a nonzero constant, which we take to be $w = 0.02 \text{ s}^{-2}$. 2. There is an electronics malfunction resulting in an extraneous time-varying voltage being applied to the motor. In this case, we take $w = 0.001t = 0.001\xi_3 \text{ s}^{-2}$. In both scenarios, the fault occurs at $t = 0$. A set of measurement data for each scenario was generated by simulation using nominal (midpoint) values of parameters, and then adding measurement noise, which was a random number between -0.05 and 0.05 .

Fig. 7 shows the fault detection results for the constant voltage scenario. In this case, Method 1, Method 2, and MCS all report the fault at $t = 4.6 \text{ s}$. The envelope generated by Method 2 is very thin and not easily seen in Fig. 7. This shows that Method 2 will not always detect a fault

faster than Method 1, but since the envelope used by Method 2 is always within the envelope used by Method 1, the performance of Method 2 will never be worse than Method 1. Fig. 8 shows the fault detection results for the varying voltage scenario. Method 1 reports the fault at $t = 14.4$ s, with the MCS method at $t = 10$ s. Method 2 detects the fault at $t = 9.4$ s. Here Method 2 provides rapid fault detection that, unlike MCS, is not subject to false alarms.

Batch reactor system

We consider a first-order exothermic reaction $A \rightarrow B$ in a batch reactor fitted with a segmented, variable area cooling jacket. The model³⁶ is given by the following material and energy balances

$$\begin{aligned}\frac{dX}{dt} &= k_0 \exp\left(-\frac{E_a}{RT}\right) (1 - X) \\ \frac{dT}{dt} &= \frac{UA}{C_{A0}VC_p}(T_a - T) - \frac{\Delta H_R k_0}{C_p} \exp\left(-\frac{E_a}{RT}\right) (1 - X),\end{aligned}\quad (14)$$

where X is the conversion and T is the reactor temperature. Other quantities and their values are shown in Table 3. The values shown for the heat transfer constant UA and the coolant temperature T_a are nominal values, since these parameters are not known with certainty. These two uncertain parameters are assumed (arbitrarily) to be $UA \in [2.7, 2.9]$ W/K, and $T_a \in [298, 302]$ K. The output is the temperature, $y = T$, which is measured every 10 s. These temperature measurements are subject to noise bounded by $V = [-2, 2]$ K. The initial conditions are $X_0 = 0$ and $T_0 \in [398, 402]$ K. The time required in the batch reactor is 300 s.

Fig. 9 shows the envelope of normal outputs determined using VSPODE (solid curves) and using MCS (shaded area). These two envelopes are also compared in Table 4 for selected values of time. The MCS envelope is sound but not complete, underestimating the true envelope. The VSPODE envelope is rigorously complete but not sound, overestimating the true envelope. However, it is clear from the results in Fig. 9 and Table 4 that the amount of overestimation is extremely small.

Two fault scenarios are considered: 1. A clogging in a segment of the cooling jacket at $t = 0$,

causing the heat transfer constant to decrease from its nominal value 2.8 W/K to 2.6 W/K. 2. A gradual increase in coolant temperature from its nominal value 300 K to 310 K at a rate of 0.2 K/s, beginning at $t = 50$ s. A set of measurement data for each scenario was generated by simulation using nominal values of parameters, and then adding measurement noise, which was a random number between -2 and 2 K.

Fig. 10 shows the fault detection results for the first scenario (drop of heat transfer constant). For this case, Method 1, Method 2, and MCS all report the fault at $t = 30$ s. Fig. 11 shows the fault detection results for the second scenario (gradual increase of coolant temperature). Both Method 1 and MCS report the fault at $t = 170$ s, and Method 2 reports the fault at $t = 130$ s. As in the previous examples, Method 2 reduces the fault detection time, and does so on the basis of a rigorously complete envelope, not subject to false alarms.

Concluding Remarks

We have demonstrated here an approach for generating an envelope of normal, fault-free outputs for nonlinear systems described by continuous-time models with uncertain parameters and/or initial states. This approach is rigorously guaranteed to be complete, but does not cause significant overestimation. Two fault detection methods were described based on this approach. The first method made direct use of the envelope, while in the second method, a constraint propagation procedure based on Taylor models was used, together with error-bounded output measurements, thus enabling fault detection earlier than when using the first method. Several fault detection scenarios involving nonlinear, continuous-time systems were used to successfully demonstrate these methods.

There are many other analytical approaches to model-based fault detection. However, as emphasized by Venkatasubramanian et al.,² these are mostly limited to the use of linear models, or

very specific nonlinear models. The strategy presented here provides a rigorous approach for dealing with general nonlinear models, without having to resort to potentially poor linearizations. The residuals used for fault detection in the methods described here are based simply on the differences in the error-bounded output measurements compared to the envelopes of expected, fault-free outputs obtained from the model. While such residuals may be used to detect a fault, they may not be directly useful in isolating the fault. Thus, in many other approaches for fault detection there is an emphasis on determining diagnostic residuals that are sensitive to only a subset of the possible faults. Standard techniques² for generating such residuals include use of diagnostic observers, parity relationships and Kalman filters. Enhanced techniques for generating more selective residuals include the directional-residual^{37,38} and structural-residual³⁹ methods. In the approach presented here, it is possible to use any residual that is a function of the error-bounded measurements and the state bounds determined from the model. However, we have not considered how to best choose the residuals for purposes of fault isolation and diagnosis.

Another type of method for model-based fault detection is parameter estimation.¹ Online measurements can be used to estimate parameters in a model, based on techniques such as least squares, and if a parameter deviates sufficiently from its normal value, a fault is detected. The second method described here incorporates a technique of this type. If the parameter interval that is consistent with the error-bounded measurements becomes inconsistent with the normal range of the parameter, then this method detects a fault. If the parameter that has deviated from its normal range has real physical meaning, as in the case of a first-principles model, then this approach may be directly useful for fault isolation and diagnosis.

Acknowledgments

This work was supported in part by the State of Indiana 21st Century Research and Technology Fund under Grant #909010455, and by the Department of Energy under Grant DE-FG02-05CH11294.

References

1. Isermann R. Process fault detection based on modeling and estimation methods: A survey. *Automatica*. 1984;20:387–404.
2. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuir SN. A review of process fault detection and diagnosis part I: Quantitative model-based methods. *Comput Chem Eng*. 2003;27:293–311.
3. Venkatasubramanian V, Rengaswamy R, Kavuir SN. A review of process fault detection and diagnosis part II: Qualitative models and search strategies. *Comput Chem Eng*. 2003;27:313–326.
4. Venkatasubramanian V, Rengaswamy R, Kavuir SN, Yin K. A review of process fault detection and diagnosis part III: Process history based methods. *Comput Chem Eng*. 2003;27:327–246.
5. Chiang LH, Russell EL, Braatz RD. *Fault Detection and Diagnosis in Industrial Systems*. Springer, 2001.
6. Walter É, Pronzato L, Norton J. *Identification of Parametric Models from Experimental Data*. New York, NY: Springer-Verlag, 1997.
7. Rajaraman S, Hahn J, Mannan MS. A methodology for fault detection, isolation, and identification for nonlinear processes with parametric uncertainties. *Ind Eng Chem Res*. 2004; 43:6774–6786.
8. Barton PI, Lee CK, Yunt M. Optimization of hybrid systems. *Comput Chem Eng*. 2006; 30:1576–1589.
9. Isermann R. Model-based fault-detection and diagnosis—Status and applications. *Annu Rev Control*. 2005;29:71–85.

10. Armengol J, Travé-Massuyè L, Vehí J, de la Rosa JL. A survey on interval model simulators and their properties related to fault detection. *Annu Rev Control.* 2000;24:31–39.
11. Moore RE. *Interval Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1966.
12. Makino K, Berz M. Remainder differential algebras and their applications. In: Berz M, Bishof C, Corliss G, Griewank A, eds., *Computational Differentiation: Techniques, Applications, and Tools.* Philadelphia: SIAM, 1996; 63–74.
13. Lin Y, Stadtherr MA. Validated solutions of initial value problems for parametric ODEs. *Appl Num Math.* 2007;57:1145–1162.
14. Hansen E, Walster GW. *Global Optimization Using Interval Analysis.* New York: Marcel Dekker, 2004.
15. Pryce JD, Corliss GF. Interval arithmetic with containment sets. *Computing.* 2006;78:251–276.
16. Jaulin L, Kieffer M, Didrit O, Walter É. *Applied Interval Analysis.* London: Springer-Verlag, 2001.
17. Neumaier A. *Interval Methods for Systems of Equations.* Cambridge, UK: Cambridge University Press, 1990.
18. Kearfott RB. *Rigorous Global Search: Continuous Problems.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 1996.
19. Makino K, Berz M. Efficient control of the dependency problem based on Taylor model methods. *Reliab Comput.* 1999;5:3–12.
20. Makino K, Berz M. Taylor models and other validated functional inclusion methods. *Int J Pure Appl Math.* 2003;4:379–456.

21. Makino K, Berz M. Taylor model range bounding schemes. In: *Third International Workshop on Taylor Methods*. Miami Beach, FL, 2004; .
22. Makino K, Berz M. Verified global optimization with taylor model-based range bounders. *Transactions on Computers*. 2005;11:1611–1618.
23. Neumaier A. Taylor forms - Use and limits. *Reliab Comput*. 2003;9:43–79.
24. Lin Y, Stadtherr MA. Deterministic global optimization for parameter estimation of dynamic systems. *Ind Eng Chem Res*. 2006;45:8438–9448.
25. Lin Y, Stadtherr MA. Deterministic global optimization of nonlinear dynamic systems. *AIChE J*. 2007;53:866–875.
26. Nedialkov NS, Jackson KR, Corliss GF. Validated solutions of initial value problems for ordinary differential equations. *Appl Math Comput*. 1999;105:21–68.
27. Neher M, Jackson KR, Nedialkov NS. On Taylor model based integration of ODEs. *SIAM J Numer Anal*. 2007;45:236–262.
28. Lohner RJ. Computations of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In: Cash J, Gladwell I, eds., *Computational Ordinary Differential Equations*. Oxford, UK: Clarendon Press, 1992; 425–435.
29. Nedialkov NS. *Computing Rigorous Bounds on the Solution of An Initial Value Problems for An Ordinary Differential Equation*. Ph.D. thesis, University of Toronto, Toronto, Canada, 1999.
30. Nedialkov NS, Jackson KR, Pryce JD. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliab Comput*. 2001;7:449–465.

31. Berz M, Makino K. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliab Comput.* 1998;4:361–369.
32. Corliss GF, Rihm R. Validating an prior enclosure using high-order taylor series. In: Alefeld G, Frommer A, eds., *Scientific Computing: Computer Arithmetic, and Validated Numerics*. Akademie Verlag, Berlin, 1996; 228–238.
33. Lin Y, Stadtherr MA. Guaranteed nonlinear state and parameter estimation for continuous-time systems. *Ind Eng Chem Res.* 2007;46:7198–7207.
34. Sainz MA, Armengol J, Vehí J. Fault detection and isolation of the three-tank system using the modal interval analysis. *J Process Contr.* 2002;12:325–338.
35. Martínez-Guerra R, Garrido R, Osorio-Miron A. The fault detection problem in nonlinear systems using residual generators. *IMA J Math Control I.* 2005;22:119–136.
36. Huang H, Adjiman CS, Shah N. Quantitative framework for reliable safety analysis. *AIChE J.* 2002;48:78–96.
37. Gertler J, Monajemy R. Generating directional residuals with dynamic parity relations. *Automatica.* 1995;31:627–635.
38. Yin K. Minimax methods for fault isolation in the directional residual approach. *Chem Eng Sci.* 1998;53:2921–2931.
39. Gertler J, Singer D. A new structural framework for parity equation-based failure detection and isolation. *Automatica.* 1990;26:381–388.

Table 1: Envelope of normal outputs for the three tank example determined by VSPODE and by Monte Carlo simulation (MCS). For each value of time, the outputs are given in the order h_1 , h_2 , h_3 .

t (s)	Envelope (m)	
	VSPODE	MCS
10	[0.6054, 0.6279]	[0.6071, 0.6254]
	[0.4878, 0.5099]	[0.4896, 0.5063]
	[0.3609, 0.3819]	[0.3623, 0.3801]
20	[0.6184, 0.6432]	[0.6214, 0.6393]
	[0.4849, 0.5084]	[0.4878, 0.5034]
	[0.3374, 0.3593]	[0.3399, 0.3559]
30	[0.6293, 0.6563]	[0.6332, 0.6516]
	[0.4817, 0.5071]	[0.4855, 0.5010]
	[0.3182, 0.3407]	[0.3212, 0.3361]
40	[0.6386, 0.6675]	[0.6432, 0.6622]
	[0.4786, 0.5053]	[0.4831, 0.4987]
	[0.3024, 0.3255]	[0.3058, 0.3198]
50	[0.6464, 0.6773]	[0.6517, 0.6714]
	[0.4756, 0.5036]	[0.4808, 0.4965]
	[0.2892, 0.3129]	[0.2932, 0.3069]

Table 2: Envelope of normal outputs for the positioning system example, as determined by VSPODE and by Monte Carlo simulation (MCS).

t (s)	Envelope	
	VSPODE	MCS
5	[0.4843, 0.5324]	[0.4883, 0.5292]
10	[-0.1197, -0.0823]	[-0.1171, -0.0866]
15	[-0.3936, -0.3494]	[-0.3909, -0.3544]
20	[0.5887, 0.6568]	[0.5948, 0.6526]

Table 3: Batch reactor parameters.

Parameter	Description	Value
k_0	Kinetic rate constant	0.022 s^{-1}
C_{A0}	Initial concentration of A	10 mol/m^3
V	Volume of the reactor	0.1 m^3
C_p	Total heat capacity	60 J/mol K
E_a	Activation energy	6000 J/mol
R	Gas constant	8.314 J/mol/K
ΔH_R	Heat of reaction	$-140,000 \text{ J/mol}$
UA	Heat transfer constant	2.8 W/K
T_a	Coolant temperature	300 K

Table 4: Envelope of normal outputs for the batch reactor example, as determined by VSPODE and by Monte Carlo simulation (MCS).

t (s)	Envelope (K)	
	VSPODE	MCS
50	[486.71, 512.93]	[488.23, 510.79]
100	[455.55, 482.79]	[455.86, 481.73]
150	[406.73, 425.93]	[407.19, 425.14]
200	[374.24, 386.38]	[374.67, 385.95]
250	[355.26, 363.53]	[355.68, 363.32]
300	[343.53, 349.94]	[343.88, 349.81]

List of Figure Captions

Figure 1. Coupled three-tank system.

Figure 2. Envelope of normal tank levels computed using VSPODE (solid curves) compared to envelope obtained by Monte Carlo simulation (shaded area).

Figure 3. Fault detection for scenario of clogging between tanks 2 and 3. See text for discussion.

Figure 4. Fault detection for scenario of excess leakage from tank 2. See text for discussion.

Figure 5. Fault detection for scenario of slowly increasing leakage from tank 2. See text for discussion.

Figure 6. Envelope of normal positions computed using VSPODE (solid curves) compared to envelope obtained by Monte Carlo simulation (shaded area).

Figure 7. Fault detection for scenario of constant voltage. See text for discussion.

Figure 8. Fault detection for scenario of varying voltage. See text for discussion.

Figure 9. Envelope of normal batch reactor temperature computed using VSPODE (solid curves) compared to envelope obtained by Monte Carlo simulation (shaded area).

Figure 10. Fault detection for scenario of drop of heat transfer constant at $t = 0$. See text for discussion.

Figure 11. Fault detection for scenario of rising coolant temperature (beginning at $t = 50$). See text for discussion.

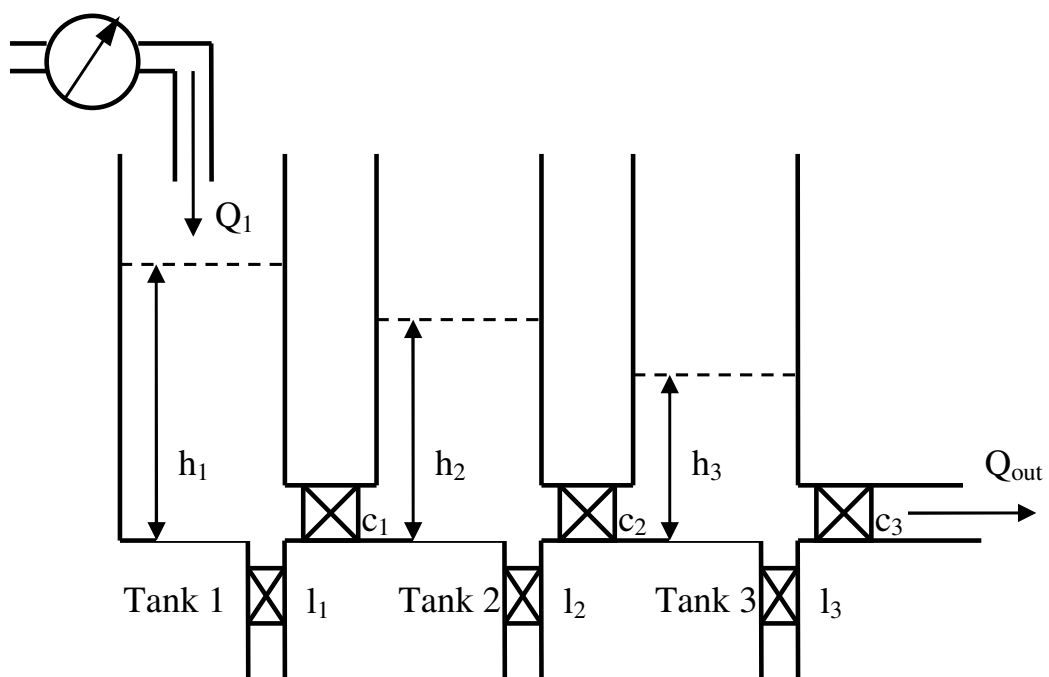


Figure 1: Coupled three-tank system.

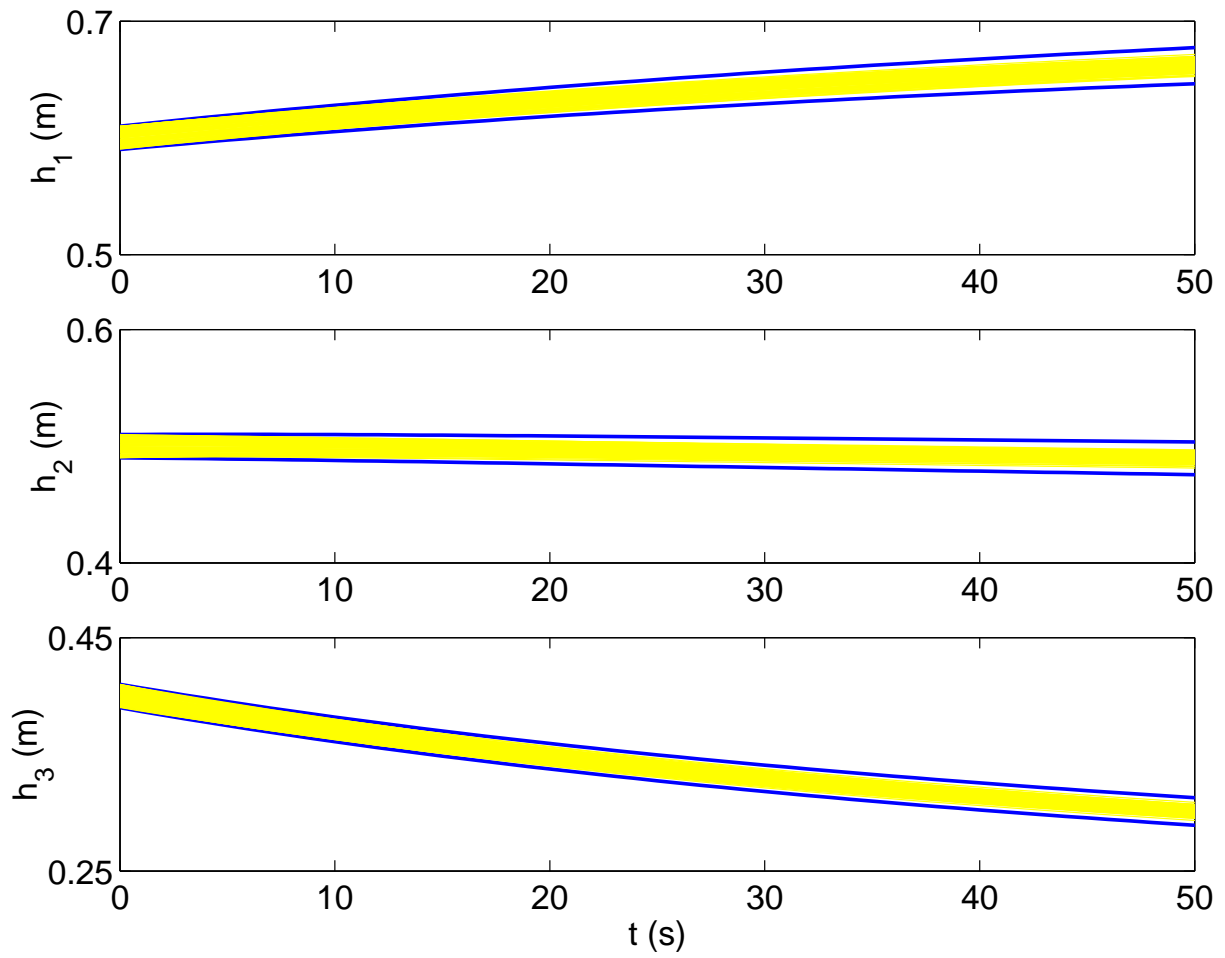


Figure 2: Envelope of normal tank levels computed using VSPODE (solid curves) compared to envelope obtained by Monte Carlo simulation (shaded area).

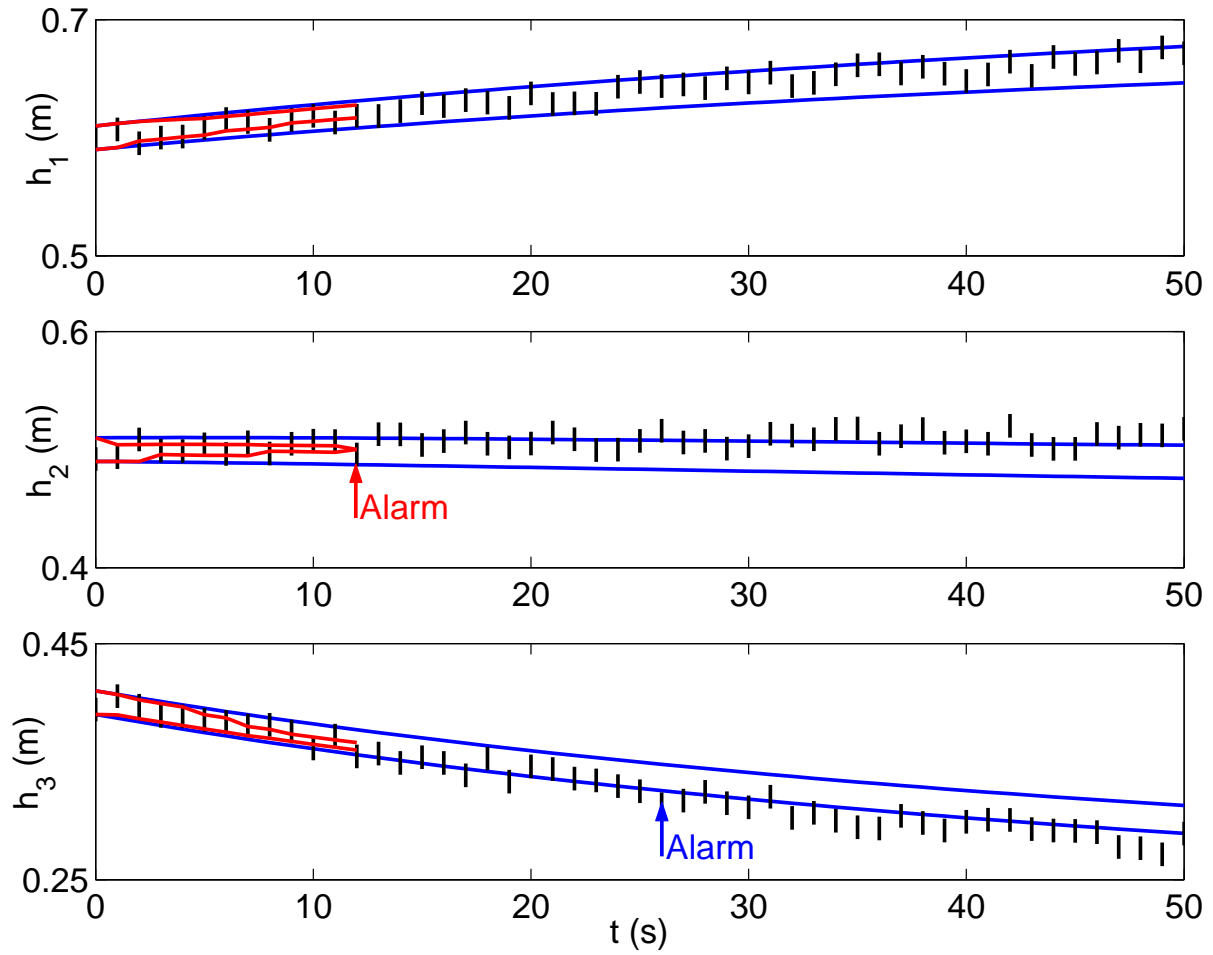


Figure 3: Fault detection for scenario of clogging between tanks 2 and 3. See text for discussion.

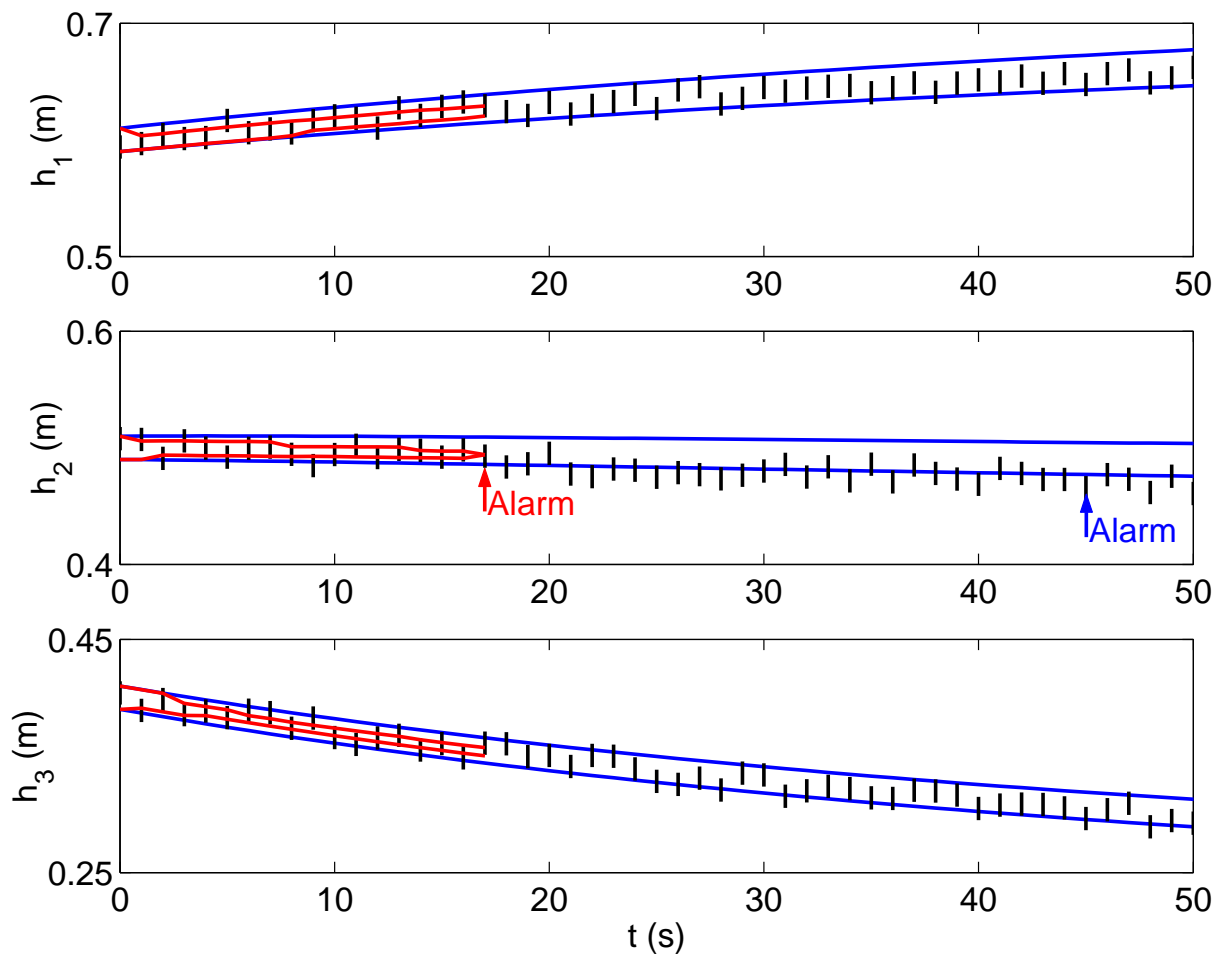


Figure 4: Fault detection for scenario of excess leakage from tank 2. See text for discussion.

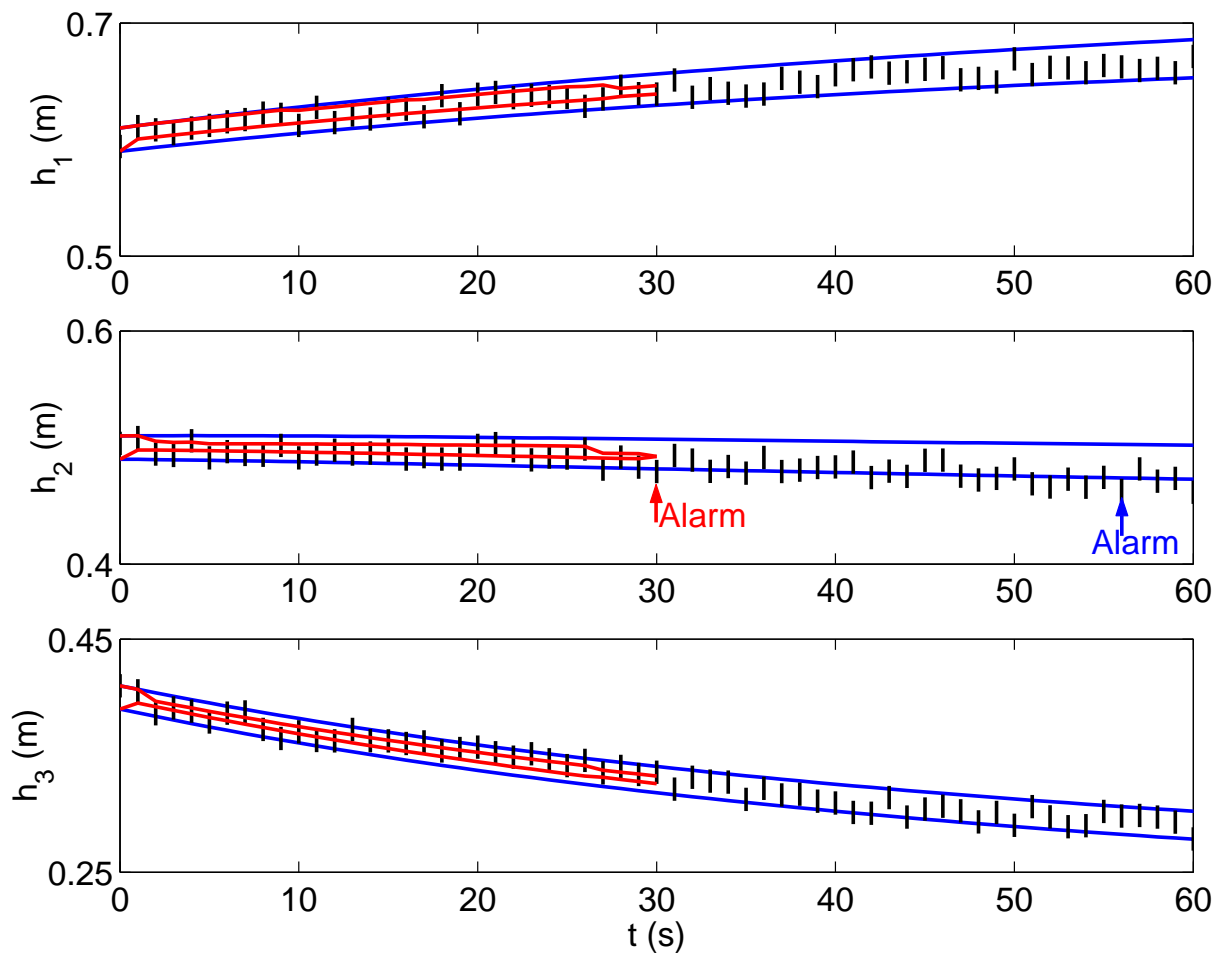


Figure 5: Fault detection for scenario of slowly increasing leakage from tank 2. See text for discussion.

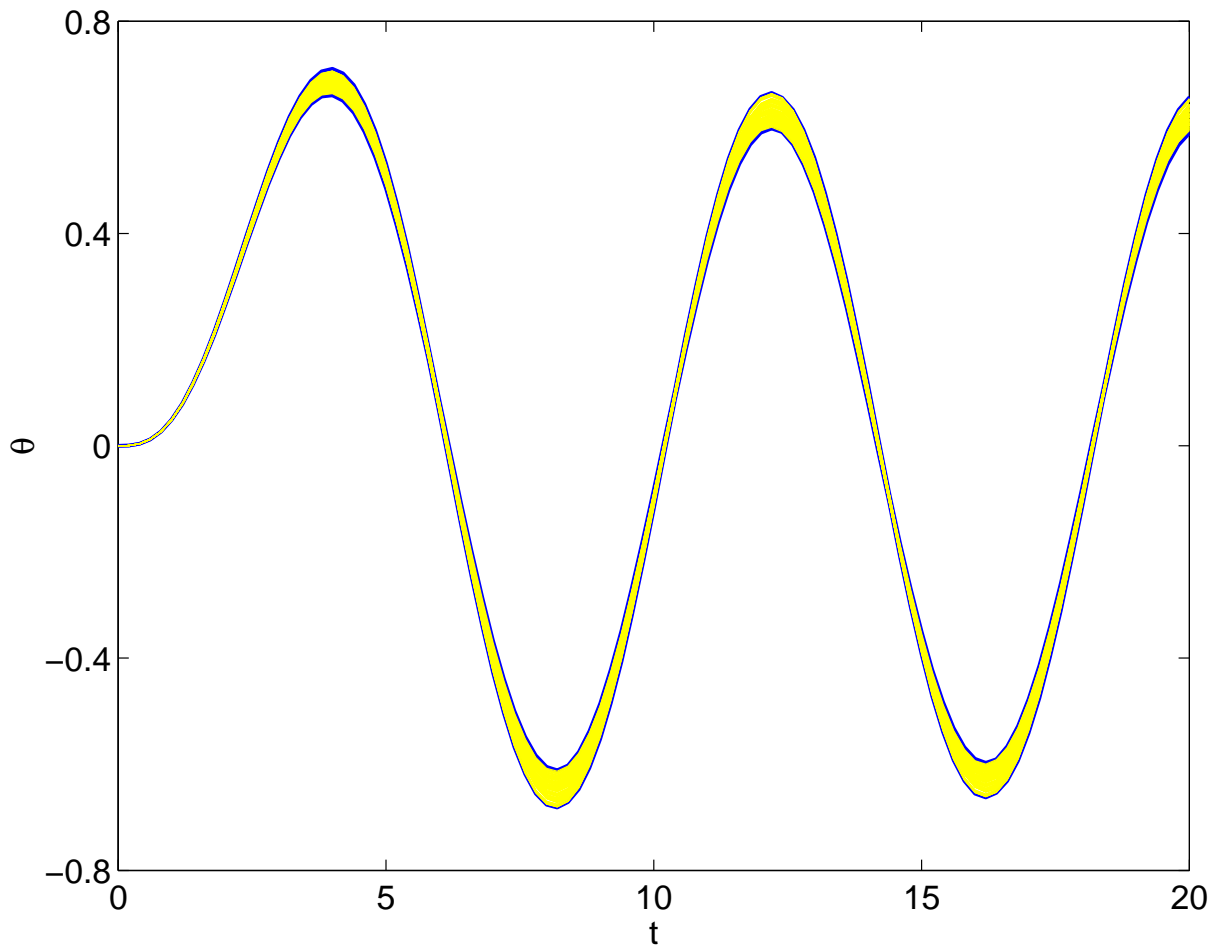


Figure 6: Envelope of normal positions computed using VSPODE (solid curves) compared to envelope obtained by Monte Carlo simulation (shaded area).

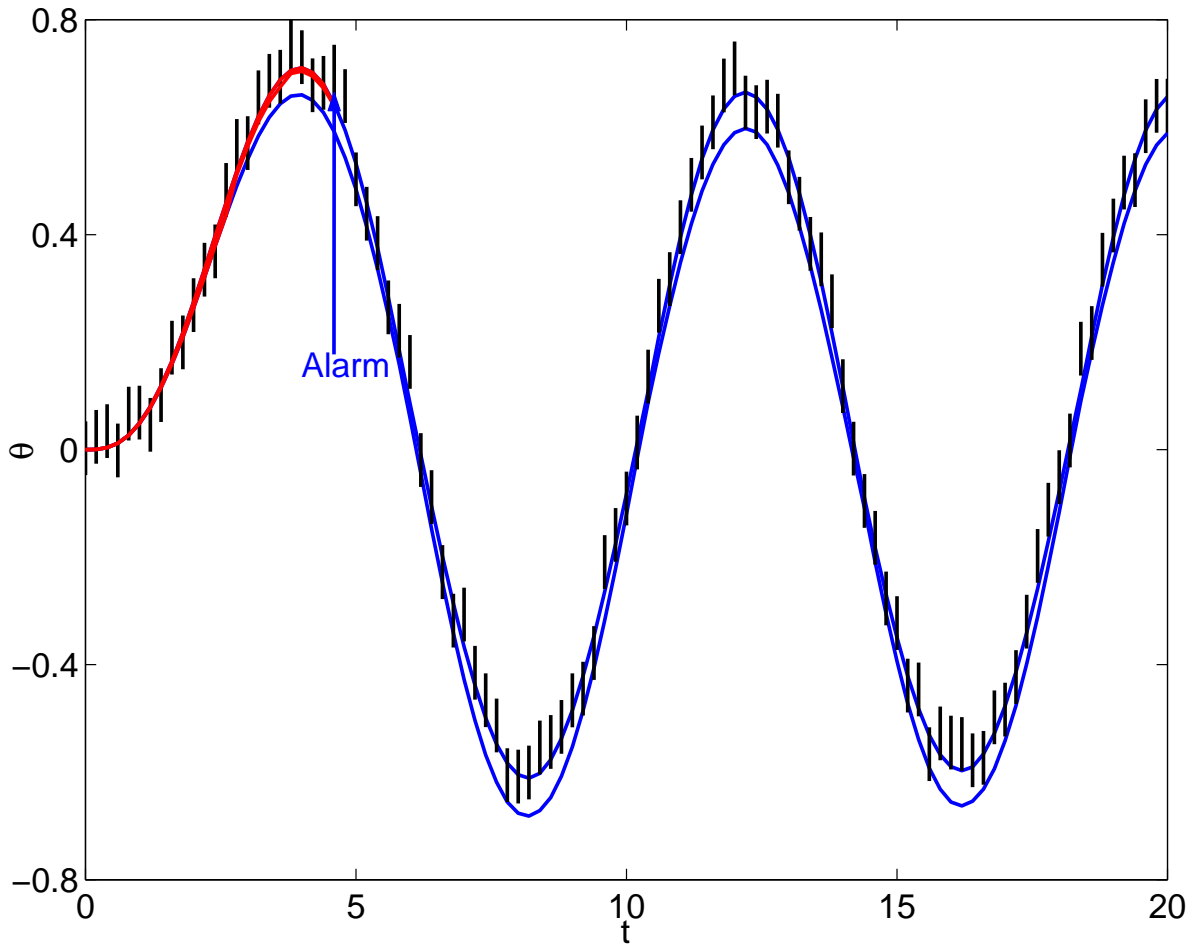


Figure 7: Fault detection for scenario of constant voltage. See text for discussion.

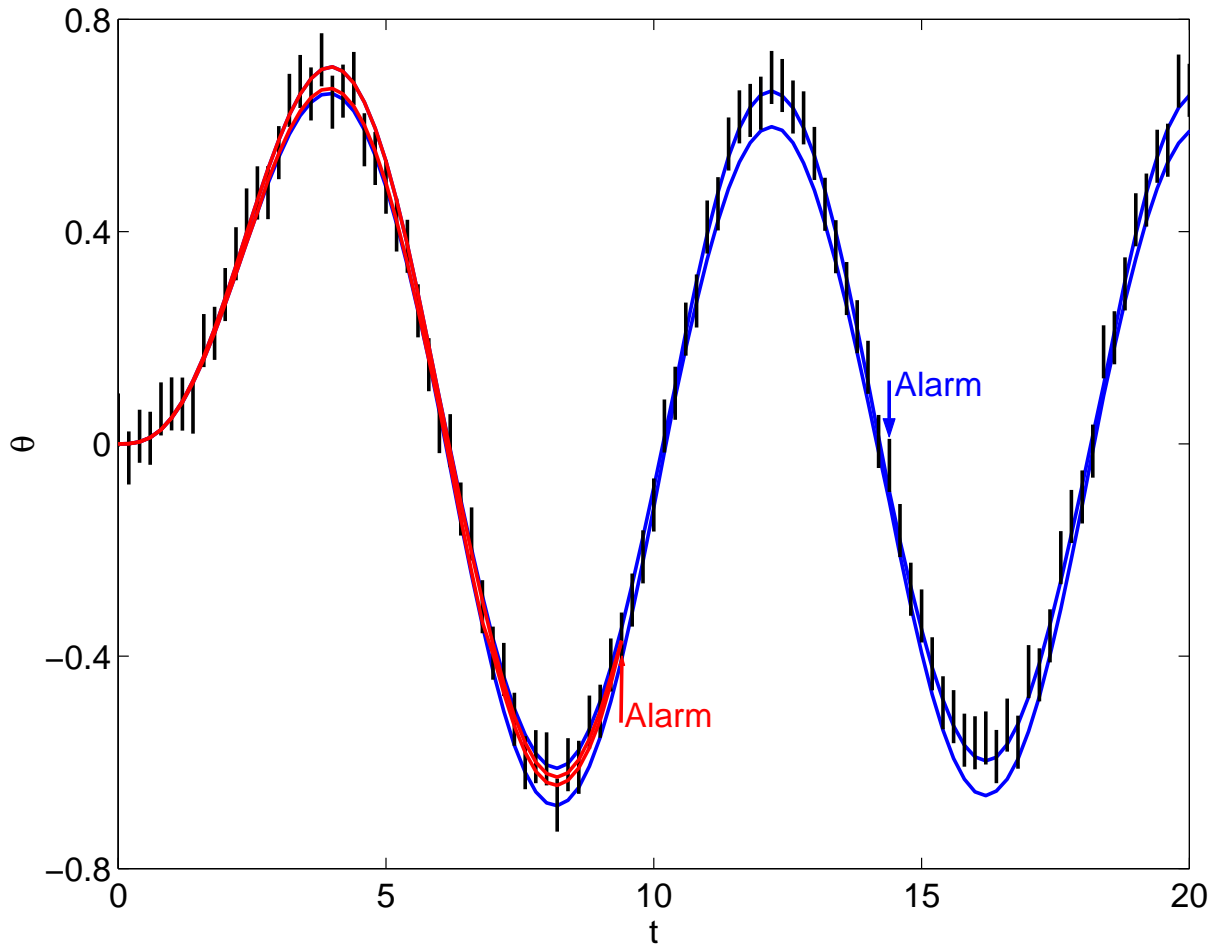


Figure 8: Fault detection for scenario of varying voltage. See text for discussion.

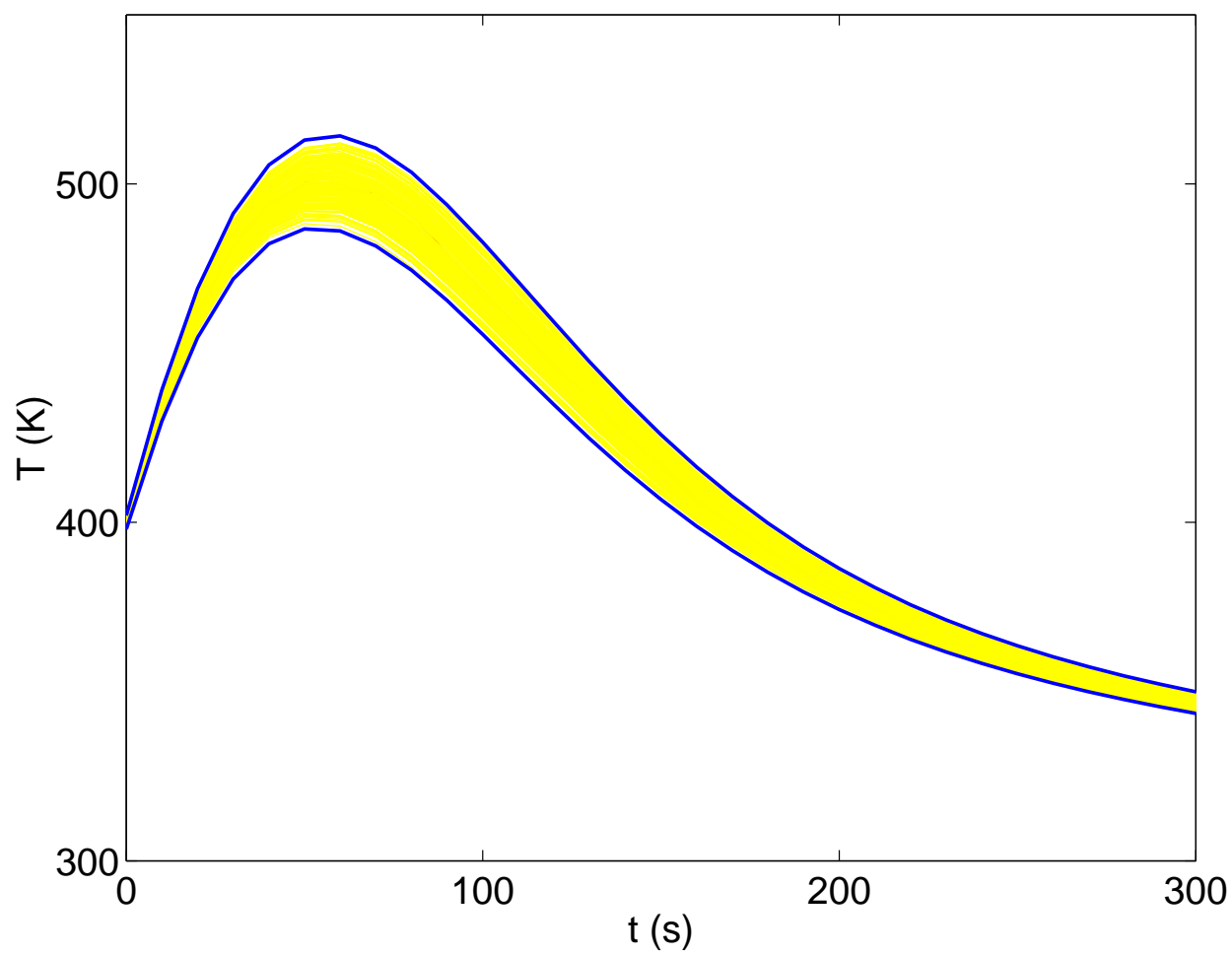


Figure 9: Envelope of normal batch reactor temperature computed using VSPODE (solid curves) compared to envelope obtained by Monte Carlo simulation (shaded area).

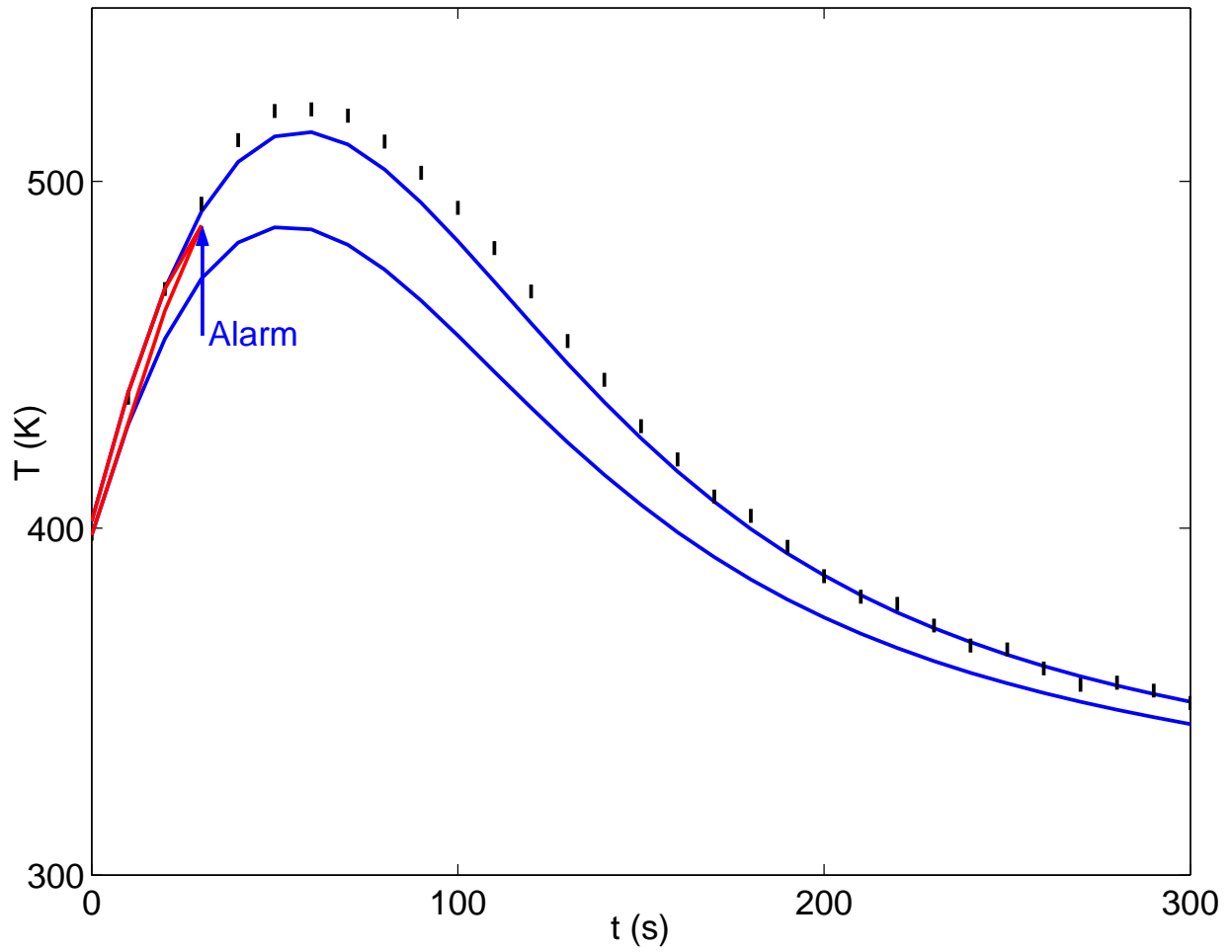


Figure 10: Fault detection for scenario of drop of heat transfer constant at $t = 0$. See text for discussion.

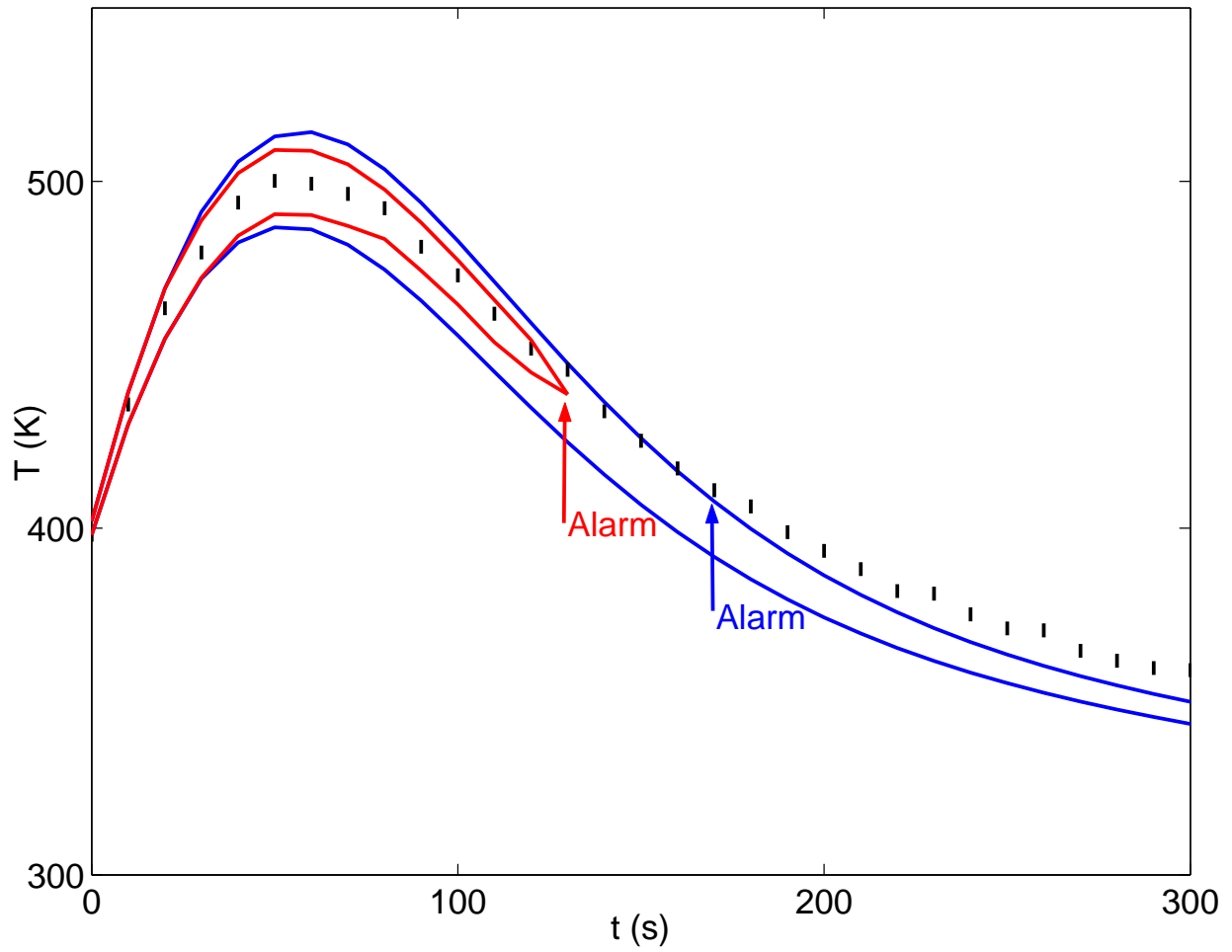


Figure 11: Fault detection for scenario of rising coolant temperature (beginning at $t = 50$). See text for discussion.