Quantitative Framework for Reliable Safety Analysis

Haitao Huang, Claire S. Adjiman, and Nilay Shah
Centre for Process Systems Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BY, U.K.

The effectiveness of any methodology used to identify hazards in chemical processes affects both safety and economics. To achieve maximum safety at minimum cost, a conservative, but realistic, analysis must be carried out. An approach to hazard identification is proposed based on a detailed process model which includes nonlinear dynamics and uncertainty. A new modeling framework, the region-transition model (RTM), is developed, which enables the simulation of regions of the operating space through an extension of the hybrid state transition system formalism. The RTM is illustrated on a nonlinear batch reactor with parameter uncertainty. A safety-verification algorithm identifies regions of the input space (initial conditions and external inputs) which guarantee safe operation. The algorithm is successfully applied to three examples: a tank with overflow and underflow, a batch reactor with an exothermic reaction, and a CSTR with feed preheating.

Introduction

Although long overlooked by the academic community, safety has always been a critical issue in the design and operation of chemical plants. The occurrence of catastrophic accidents such as Seveso (Kleindorfer and Kunreuther, 1987), Bhopal (Kurzman, 1987), and Flixborough (Lewis, 1989) has resulted in less public acceptance of the chemical industry and led to the development of new safety standards and regulations, such as the European directive adopted after the Seveso incident (OJEC, 1997), or the OSHA standards for the management of highly hazardous chemicals (OSHA, 1992) in the U.S. It is now essential for chemical companies to carry out systematic analyses that convincingly demonstrate the safety of their processes to the increasingly wary regulatory agencies and general public.

An essential phase of the safety analysis of any process is concerned with the early phase of *hazard identification.* The hazards cataloged at this stage determine the focus of the remainder of the safety analysis. A number of approaches have been proposed for this important task. Most techniques advocated are based on qualitative information (Kelly, 1991; Kletz, 1991; Eley, 1992; Jones, 1992; Göring and Schecker, 1993; Catino and Ungar, 1995; Kumamoto and Henley, 1996), but a few quantitative methods have also been devised in recent years (Dimitriadis et al., 1997; Park and Barton, 1997; Moon et al., 1992). These existing methodologies are first briefly reviewed in order to identify needs for the future. We then present a new methodology which addresses some of the issues raised.

Correspondence concerning this article should be addressed to C. S. Adjiman.

Qualitative hazard identification

Qualitative techniques are widely used in industry. They capitalize on the experience acquired through years of practice and revolve around a team effort to analyze new processes. The most prominent approaches are checklists, preliminary hazard analysis (PHA), and hazard and operability studies (HAZOP). The concept of qualitative simulation has also emerged recently to help automate the task of hazard identification.

Checklists (Eley, 1992) rely on expert knowledge of the plant and safety regulations in order to identify disturbances in the operation of the process that can lead to unsafe situations (*initiating events*). A checklist is simply a list of all anticipated sources of hazards, such as pressure containers, pumps,

corrosion, and fire. Preliminary hazard analysis (PHA) (Kumamoto and Henley, 1996) goes beyond the checklist to identify hazards in terms of the sequence of events that lead from the initiating event to the final consequence. This additional information may be used to facilitate risk mitigation. Because checklists and PHA rely heavily on past experience, they are ill-suited to innovative projects. Additionally, little or no provision is made for interactions between different process units.

HAZOP (HAZard and OPerability) studies enable a more comprehensive analysis starting from a process and instrumentation diagram (P&ID) of the plant (Kelly, 1991; Kletz, 1991; Jones, 1992; Göring and Schecker, 1993). A team of safety experts analyzes each portion of the diagram trying to evaluate the potential hazards resulting from deviations from normal operation. A significant difference from checklists is that no assumptions are made at the outset on the types of dangers on which the team should focus. Creative thinking is encouraged through the use of keywords such as *more*, *less*, *faster*, *lower*, and so on. The progression of these virtual changes is then followed through the P&ID and hazardous consequences are thus identified. A drawback of HAZOP studies lies in their intensity: highly qualified personnel are required to spend considerable amounts of time poring over the P&IDs. For instance, ICI (Kumamoto and Henley, 1996) estimate that 100 man-hours are necessary for every million dollars in capital investment. The reliance on exclusively structural information to describe the process dynamics leads to a poor understanding of the interactions between different complex units and therefore threatens the reliability of the hazard identification. In addition, the systematic use of entirely qualitative keywords often leads to scenarios that are physically unrealizable, but that are nonetheless considered in the design of a safety system, resulting in the so-called "gold plating" of the plant. Despite their disadvantages, HAZOP studies constitute a central part of most industrial safety assessments because they produce conservative results and offer better guarantees than other techniques that significant hazards have indeed been eliminated. As a result, there have been efforts to systematize and facilitate their use throughautomation (Srinivasan and Venkatasubramanian, 1998; Viswanathan et al., 1998).

Qualitative simulation, which first arose in the Artificial Intelligence community (Kleer and Brown, 1984; Kuipers, 1986), is based on a model of the process in terms of trends, that is, positive or negative changes in process variables. The effect of initiating events corresponding to abnormal process states such as equipment failure is studied. This approach has been used to study the safety of chemical processes by a number of researchers (Catino and Ungar, 1995; Waters and Ponton, 1989). In this case, a qualitative model of the process and knowledge of possible initiating events are required and may be incorporated into a simulation tool. Once the behavior of the system has been generated for a given set of initiating events, the engineer can examine trends to determine whether a hazardous situation has occurred. Provided that the plant model is reliable and that the library of faults is complete, qualitative simulation has the potential to be a more powerful approach than HAZOP studies as it deals better with complex plant interactions. Like HAZOP studies, however, it can be very time-consuming because each initial event or combination of events must be examined in order to discover possible hazards.

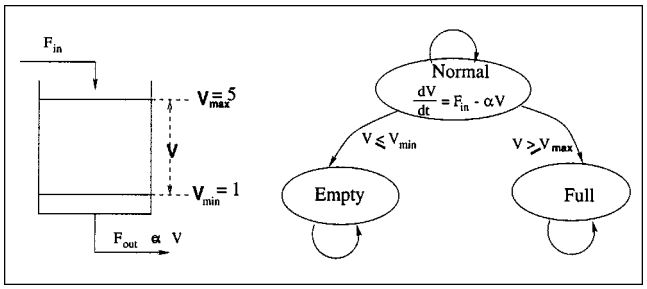


Figure 1. State-based representation of a tank.

α is a constant parameter.

Quantitative model-based approaches

One of the major problems with the hazard identification techniques presented so far is that they are based on mostly qualitative information which does not represent the complexity of chemical processes well. While they are often effective at identifying a list of potential hazards, it is usually difficult to determine whether these hazards can actually occur and how this might happen. A more realistic understanding of the hazards associated with any given initiating event could be obtained by using all known information on the process, including quantitative models. Such an approach would facilitate the consideration of the interactions of possible events. Several researchers have worked on approaches that meet this goal, by developing new modeling frameworks and tools to analyze the models. Quantitative model-based techniques all require knowledge of regions of the operating space that correspond to unsafe plant operation (such as a maximum operating pressure), and a list of potential disturbances to the process. Thus, they should really be viewed as a complement to qualitative techniques rather than a replacement.

The quantitative approaches are based on the fact that chemical processes are fundamentally hybrid systems, that is, they operate partly in a continuous way and partly in a discrete way (Barton and Pantelides, 1994). The simple system of Figure 1, for example, shows a tank with three states—normal operation, empty, and full—and the sets of continuous variables and equations that represent its dynamic behavior. In the context of hazard identification, the unsafe states (empty and full in the case of the tank) can generally be considered as "terminal" states: once one of these states is entered, the possibility of the hazard occurring has been ascertained and there is no need to model the return of the system to a safe state. In general, a hybrid system may evolve between a number of nonterminal safe states and terminal unsafe states.

The study of hybrid state-transition systems such as this one has been an active topic of research in the computer science and electrical engineering communities for a number of years (Wallace and Fujii, 1989; Grossman et al., 1993; Alur et al., 1996). Recently, researchers in the chemical engineering community have also turned their attention to this issue (Barton and Pantelides, 1994; Dimitriadis et al., 1996; Barton and Park, 1997; Dimitriadis et al., 1997).

A hybrid system can be represented by a directed graph as the one in Figure 1. The main elements of the graph are listed below.

A set $\mathcal{S}$ of vertices called *locations* or *states*.

A set of *continuous variables* and *equations* corresponding to each location. The equations take on the general form

$$\boldsymbol{f}^{(s)}\left(\dot{\boldsymbol{x}}^{(s)}(t), \boldsymbol{x}^{(s)}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{u}(t)\right) = 0, \forall s \in \mathcal{S}. \quad (1)$$

where $\boldsymbol{x}^{(s)}(t)$ is the vector of variables representing state s at time t, $\boldsymbol{y}^{(s)}(t)$ is a vector of algebraic variables for state s and $\boldsymbol{u}(t)$ is a vector of external influences (controls or disturbances) on the system.

A set of *edges* $\mathcal{E}$ that correspond to transitions between locations. A transition $e = (s, s', l_{ss'}, I_{ss'})$ consists of a source state or location s, a target state s', a transition relation $l_{ss'}(\boldsymbol{x}^{(s)}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{u}(t))$, which is a logical condition and a state initialization

$$I_{ss'}\left(\dot{\boldsymbol{x}}^{(s)}(t), \dot{\boldsymbol{x}}^{(s')}(t), \boldsymbol{x}^{(s)}(t), \boldsymbol{x}^{(s')}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{y}^{(s')}(t), \boldsymbol{u}(t)\right) = 0.$$

A number of assumptions are usually made regarding the types of equations which describe system behavior and the nature of possible transitions. A model must be able to describe a "run" of the hybrid system based on a set of initial conditions and external influences. A run is the trajectory of the system, and it can be defined in terms of the times of the state transitions and the values of the relevant variables as the system enters each new state. Two main approaches have emerged to model hybrid systems. One assigns a binary variable to each state, defining whether the system is in that state or not (Dimitriadis et al., 1997). One and only one such variable must be equal to one at any given time. Binary variables are also used to specify which transition relations are satisfied and which transition actually takes place at any given time. Variables describing equations and initializations for all states are incorporated within a single model and are activated or deactivated depending on the value of the binary variable representing the appropriate state. The alternative modeling strategy uses Boolean variables rather than binary variables to represent the state the system is in (Alur et al., 1995).

Once a representative model of the system has been built, two broad strategies have been advocated in order to probe the models and identify possible hazards:

(1) Reachability analysis (Alur et al., 1995; Asarin et al., 1995; Dill and Wong-Toi, 1995): Given a set of possible initial conditions, identify the (unsafe) states the system can reach over an infinite time horizon. Model checking using computational tree logic (CTL) (Moon et al., 1992; Park and Barton, 1997; Clarke et al., 1986), which applies to linear Boolean systems only, is essentially a restricted, but highly efficient, form of reachability analysis.

(2) Optimization-based worst-case analysis (Dimitriadis et al., 1996, 1997): Given a set of possible initial conditions and controls and a finite time horizon, minimize the time needed for the system to reach an unsafe state. This quantitative approach has been combined with automated HAZOP analysis (Srinivasan et al., 1997, 1998).

In order to guarantee the validity of the results obtained through these approaches, the analysis tools used must be based on a global analysis of the model. In the case of reachability analysis, rigorously valid over- and under-approximations of the model must usually be built, while for worst-case analysis, the global solution of the dynamic optimization problem is required. Given the theoretical difficulties this poses, these approaches have unfortunately been restricted to linear systems, and as such are applicable to a limited number of chemical processes. While it is common practice to linearize process models to increase tractability, such an approach is not suitable in the context of process safety as hazardous conditions are typically substantially different from the steady-state point used for linearization and are often caused by nonlinear behavior. In view of the capability of quantitative hazard identification to yield more realistic safety analysis than qualitative techniques, this work discusses the development of reliable models and techniques for nonlinear, uncertain systems. In particular, we aim to address the safety of nonlinear, uncertain systems through an analysis which will answer the following question:

Given a set of possible initial conditions and inputs and a finite time horizon, identify the set of initial conditions and inputs which lead to unsafe behavior.

A new modeling framework, the "region-transition model," is developed. A methodology is then presented to analyze the safety of the system using this model. Finally, the proposed approach is applied to three examples: a linear tank problem, a batch reactor with uncertain parameters, and a CSTR with feed preheating.

Region-Transition Model (RTM)

Representation of system state: regions

The objective of the present work is to develop a methodology which enables the analysis of regions of the operating space in terms of safety. Traditionally, an understanding of operating regions is achieved through the combination of optimization techniques with a modeling framework that allows the simulation of the system behavior given specific values for the initial conditions, inputs, and model parameters—a "point model." In contrast, we propose a new modeling tool which allows the simulation of the system based on *sets* of values for the initial conditions, inputs, and model parameters—a "region model." Thus, the system no longer evolves from point to point in the space of states, but rather *from region to region*. This concept has been briefly discussed in Adjiman (1999) and Huang et al. (2000). Two types of regions are introduced:

- A *simple region* r is defined by a unique state s and a hyper-rectangle $[\underline{\boldsymbol{x}}^{(s)}, \bar{\boldsymbol{x}}^{(s)}]\times[\underline{\boldsymbol{y}}^{(s)}, \bar{\boldsymbol{y}}^{(s)}]$, which provides bounds on the state variables that describe state s. The region r is then denoted by $(s, [\underline{\boldsymbol{x}}^{(s)}, \bar{\boldsymbol{x}}^{(s)}]\times[\underline{\boldsymbol{y}}^{(s)}, \bar{\boldsymbol{y}}^{(s)}])$.
- A *general region* $\mathcal{R}$ is defined as a union of simple regions. It may consist of H_s simple regions in a given state s which cover disjoint hyper-rectangles, such that $\mathcal{R} = \cup_{i=1,\ldots,H_s}(s,[\underline{\boldsymbol{x}}^{(i,s)}, \bar{\boldsymbol{x}}^{(i,s)}]\times[\underline{\boldsymbol{y}}^{(i,s)}, \bar{\boldsymbol{y}}^{(i,s)}])$. More broadly, the simple regions may be in different states in the set $\mathcal{S}_M$, such that $\mathcal{R} = \cup_{s\in\mathcal{S}_M}(s, \chi^{(s)})$ where $\chi^{(s)} = \cup_{i=1,\ldots,H_s}(s, [\underline{\boldsymbol{x}}^{(i,s)}, \bar{\boldsymbol{x}}^{(i,s)}]\times[\underline{\boldsymbol{y}}^{(i,s)}, \bar{\boldsymbol{y}}^{(i,s)}])$.

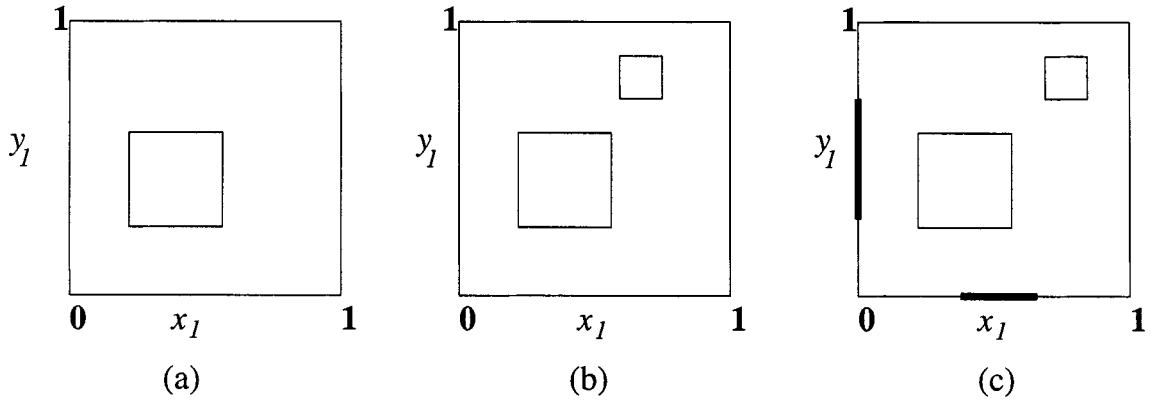


Figure 2. Mole fraction of component 1 in the liquid (x_1) and gas (y_1) phase for two-component mixture.

The system is in (a) a simple region in the state of gas/liquid coexistence; (b) a general region in the coexistence state; (c) a general region in all three possible states.

These concepts are illustrated in Figure 2 which considers a two-component fluid in a heated tank. The fluid can be in one of three states: supercooled liquid, coexisting liquid and gas, and superheated gas. The system is characterized by its temperature and the mole fractions of component 1. In the liquid state, only the liquid mole fraction x_1 is relevant. In the gas state, only its gas mole fraction y_1 is meaningful. Finally, in the coexistence state, both x_1 and y_1 must be known. The traditional "point" description of a system is a special case of the region description.

In order to manipulate regions which combine several states and hyper-rectangles, it is necessary to modify standard set operations. Given two regions $\mathfrak{R}_1$ and $\mathfrak{R}_2$, the following definitions will be used:

Intersection of Two Regions. $\mathfrak{R}_1 \cap \mathfrak{R}_2$ is obtained by first identifying the states common to both $\mathfrak{R}_1$ and $\mathfrak{R}_2$ and, then, for each of these states, identifying the intersection of hyper-rectangles.

Union of Two Regions. $\mathfrak{R}_1 \cup \mathfrak{R}_2$ is the union of all simple regions in $\mathfrak{R}_1$ and $\mathfrak{R}_2$.

Subset of a Region. $\mathfrak{R}_1 \subseteq \mathfrak{R}_2$, if and only if all states in $\mathfrak{R}_1$ are also in $\mathfrak{R}_2$ *and*, for each state $s \in \mathfrak{R}_1$, the set of hyper-rectangles $\chi_1^{(s)}$ is a subset of the set of hyper-rectangles $\chi_2^{(s)}$ corresponding to state s in $\mathfrak{R}_2$.

The systems considered in this work are represented within the hybrid state transition network formalism outlined in the subsection on quantitative model-based approaches. However, several modifications are introduced to ensure that realistic systems can be represented and to enable the modeling of regions. These are discussed below.

Model assumptions

Determinism. It is assumed that the differential and algebraic equations describing the behavior of the system in state s can be expressed as the differential-algebraic system

$$\dot{\boldsymbol{x}}^{(s)}(t) = \boldsymbol{f}^{(s)}\left(\boldsymbol{x}^{(s)}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{u}(t)\right),$$

$$\boldsymbol{g}\left(\boldsymbol{x}^{(s)}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{u}(t)\right) = 0 \qquad (2)$$

and that the model has a unique solution for any given set of consistent initial conditions, inputs and model parameters. No linearity assumptions are made.

Time-Invariance. The equations describing a state, the transition relations and the state initializations are assumed to be time-invariant. Thus, the memory of the process is limited to its latest state.

Instantaneous Transitions. Some researchers reduce the set of state initializations at time t, $I_{ss'}(\dot{\boldsymbol{x}}^{(s)}, \dot{\boldsymbol{x}}^{(s')}, \boldsymbol{x}^{(s)}, \boldsymbol{x}^{(s')}, \boldsymbol{y}^{(s)}, \boldsymbol{y}^{(s')}, \boldsymbol{u}) = 0$, to $\boldsymbol{x}^{(s)} = \boldsymbol{x}^{(s')}$ and $\boldsymbol{y}^{(s')} = \boldsymbol{y}(s)$, so that the same variable set describes every state in the system (Asarin et al., 1995). However, in general, the variable vector changes upon a transition: this is the case for a phase change where the disappearance of a phase removes the need for mole fractions describing that phase. No particular form is assumed for the state initializations in this work. However, it is assumed that transitions are instantaneous.

Model components

Uncertainty Representation. Using a nonlinear model which represents the system better than a linear model is not sufficient, if a reliable safety analysis is to be performed. The presence of uncertain information may be reflected in two ways: through probability distributions or through bounds on the uncertain data. In this work, we adopt the *bounds* description since hazard analysis is concerned with the possibility for hazards to occur rather than with their likelihood—an issue tackled in the subsequent phase of a risk management exercise. For instance, consider that a rate constant k for a first-order reaction $A \rightarrow B$ has been measured as $1.6\ \mathrm{s}^{-1}$ with a $\pm 5\%$ experimental error. Then, the rate constant is represented by a set of values $\{k\}$ bounded by two real numbers $\underline{k}$ and $\bar{k}$ equal to $1.52\ \mathrm{s}^{-1}$ and $1.68\ \mathrm{s}^{-1}$, respectively. The reaction rate can then be expressed as a set

$$\{r\} = \left\{r : r = kC_A,\ k \in \{k\} = \left[\underline{k}, \bar{k}\right]\right\} \qquad (3)$$

where C_A is the concentration of species A. The uncertainty in the kinetic parameter is thus propagated and a set of values is obtained for the reaction rate.

In the above example, it is assumed that although the value of the parameter is uncertain, its intrinsic value is constant. In some cases, some uncertainty may arise because the parameter value varies with time. That case is not considered in this article.

Model Equations. The introduction of sets of parameter values to represent uncertainty, and sets of inputs and initial conditions to represent operating regions changes the nature of the model equations. The righthand side of differential equations of the form $\dot{\boldsymbol{x}}(t) = f[\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)]$ becomes a set of functions so that the system

$$\dot{\boldsymbol{x}}(t) = \{\boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)]\}$$
$$\boldsymbol{g}[\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)] = 0 \quad (4)$$

now describes a set of trajectories $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}$.

Since the evaluation of every solution to this set of equations is clearly impossible, we seek to bound the set of solutions with trajectories $\underline{\boldsymbol{x}}(t)$, $\bar{\boldsymbol{x}}(t)$, $\underline{\boldsymbol{y}}(t)$ and $\bar{\boldsymbol{y}}(t)$ such that

$$(\boldsymbol{x}(t), \boldsymbol{y}(t)) \in \{\boldsymbol{x}(t), \boldsymbol{y}(t)\} \Rightarrow (\boldsymbol{x}(t), \boldsymbol{y}(t)) \in [\underline{\boldsymbol{x}}(t), \bar{\boldsymbol{x}}(t)] \times [\underline{\boldsymbol{y}}(t), \bar{\boldsymbol{y}}(t)]. \quad (5)$$

The trajectories $\underline{\boldsymbol{x}}(t)$, $\bar{\boldsymbol{x}}(t)$, $\underline{\boldsymbol{y}}(t)$ and $\bar{\boldsymbol{y}}(t)$ may coincide with the exact bounding trajectories of $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}$, or they may provide an overestimate of the set $\{\boldsymbol{x}(t), \boldsymbol{y}(t)\}$. This conservative modeling of the system behavior is desirable in the context of safety analysis to ensure that all possible unsafe situations are identified.

The derivation of bounding trajectories is a crucial aspect of the modeling framework. The strategy proposed is based on a discretization of the equations and the use of interval arithmetic (Moore, 1966). We consider the case of systems of ordinary differential equations where the system is represented by $\dot{\boldsymbol{x}} = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t)]$. First, an approximation to the point model is constructed through an explicit discretization scheme, and the differential equations are replaced by a set of algebraic equations. In the case of a forward discretization scheme, they take the form

$$\boldsymbol{x}_{i+1} = f(\boldsymbol{x}_i, \boldsymbol{u}_i), \quad i = 0, \ldots, N. \quad (6)$$

where N is the number of discrete time intervals. It is, of course, essential to ascertain the stability and accuracy of this discretization scheme when choosing the time step. Next, an inclusion for $f(\boldsymbol{x}_i, \boldsymbol{u}_i)$ is derived, that is, a function $F = F(\underline{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i, \underline{\boldsymbol{u}}_i, \bar{\boldsymbol{u}}_i)$ such that $f(\boldsymbol{x}_i, \boldsymbol{u}_i) \in F$, $\forall \boldsymbol{x}_i \in [\underline{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_i]$, $\forall \boldsymbol{u}_i \in [\underline{\boldsymbol{u}}_i, \bar{\boldsymbol{u}}_i]$. Such an inclusion could, for instance, be constructed as a natural extension, or could be based on a Taylor-series form (Moore, 1966). In the simple case of the tank in Figure 1 with initial condition V_0, the initial volume of liquid, and input $F_{\text{in}}(t)$, the inlet flow rate of liquid, the (linear) model equation can be discretized as

$$V_{i+1} = F_{\text{in},i}\,\delta t + (1 - \alpha\delta t)V_i, \quad i = 0, \ldots, N. \quad (7)$$

The inclusion for V_{i+1} based on natural extension is given by

$$V_{i+1} \in \left[\underline{F}_{\text{in},i}\,\delta t + (1 - \alpha\delta t)\underline{V}_i,\ \bar{F}_{\text{in},i}\,\delta t + (1 - \alpha\delta t)\bar{V}_i\right], \quad i = 0, \ldots, N, \quad (8)$$

provided that $\alpha\delta t \le 1$. Due to the linearity of the equation, this inclusion provides exact bounds on the set of discretized trajectories. If there is some uncertainty in the parameter α such that $\alpha \in [\underline{\alpha}, \bar{\alpha}]$, where $\underline{\alpha} \ge 0$ and $\bar{\alpha}\delta t \le 1$, then the exact inclusion becomes

$$V_{i+1} \in \left[\underline{F}_{\text{in},i}\,\delta t + (1 - \bar{\alpha}\delta t)\underline{V}_i,\ \bar{F}_{\text{in},i}\,\delta t + (1 - \underline{\alpha}\delta t)\bar{V}_i\right], \quad i = 0, \ldots, N. \quad (9)$$

When the system is represented by a set of differential-algebraic equations, the generation of bounding trajectories is more difficult. While we do not present a general derivation of bounding trajectories for such systems in this article, we show how this issue can be handled for certain types of algebraic equations in one of the case studies in the next section.

System Transitions. As the system evolves from region to region, parts of the region may undergo transitions to different states. In order to identify the transitions taking place at time t, we must evaluate the logical conditions for the entire region in which the system is found at time t. Let us consider that the system is in the simple region $r = (s, [\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t)] \times [\boldsymbol{y}^{(s)}(t)\ \bar{\boldsymbol{y}}^{(s)}(t)])$ and that the range of possible inputs is given by $[\underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)]$. Due to the possible presence of nonlinearities in the logical conditions defining a transition from s to s', $l_{ss'}(\boldsymbol{x}^{(s)}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{u}(t))$, it is not usually sufficient to evaluate $l_{ss'}$ at the vertex points of the simple region. In general, an inclusion of $L_{ss'} = [\underline{l}_{ss'}, \bar{l}_{ss'}]$ of $l_{ss'}$ is obtained using the rules of interval arithmetic so that

$$l_{ss'}(\boldsymbol{x}^{(s)}(t), \boldsymbol{y}^{(s)}(t), \boldsymbol{u}(t)) \in L_{ss'}, \forall \boldsymbol{x}^{(s)}(t) \in \left[\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t)\right], \forall \boldsymbol{y}^{(s)}(t) \in \left[\underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t)\right], \forall \boldsymbol{u}(t) \in \left[\underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)\right]. \quad (10)$$

This inclusion provides exact bounds on $l_{ss'}$ when it is linear, and an overestimate when it is nonlinear. In all cases, three outcomes may occur:

(1) $\underline{l}_{ss'} \ge 0$, meaning that $l_{ss'}$ is TRUE for the entire region and the transition occurs.

(2) $\bar{l}_{ss'} < 0$, meaning that $l_{ss'}$ is FALSE for the entire region and the transition does not occur.

(3) $\underline{l}_{ss'} < 0$ and $\bar{l}_{ss'} \ge 0$, meaning that $l_{ss'}$ may be TRUE for part or all of the region and FALSE for part or all of the region: the transition may or may not occur.

To represent this last case in which the transition relation is neither TRUE nor FALSE, a new value must be introduced for Boolean variables: UNDECIDED. Furthermore, propositional logic operators must be extended to handle UNDECIDED propositions as shown in Table 1.

The identification of an UNDECIDED transition condition leads to two possible actions:

- If the condition l'_{ss} depends on one of the state variables only, so that $l_{ss'}(\boldsymbol{x}^{(s)}(t), \boldsymbol{y}(t), \boldsymbol{u}(t)) = l_{ss'}(x_k^{(s)}(t))$, for some k, it may be possible to identify a critical value $x_{k,C}^{(s)}(t)$ of $x_k^{(s)}(t)$ at which a transition occurs. This is the most common case and occurs, for instance, for a transition from a safe tank level to a below-normal tank level: the transition relation takes the form $V(t) \le V_{\min}$ and the critical value of $V(t)$ is simply $V_{\min}$. Based on the critical value, it is possible to iden-

Table 1. Outcome of Modified Boolean Operators

Proposition	P_1	P_2	Outcome
$P_1 \wedge P_2$	True	Undecided	Undecided
	False	Undecided	False
	Undecided	Undecided	Undecided
$P_1 \vee P_2$	True	Undecided	True
	False	Undecided	Undecided
	Undecided	Undecided	Undecided
$P_1 \otimes P_2$	True	Undecided	Undecided
	False	Undecided	Undecided
	Undecided	Undecided	Undecided
$\neg P_1$	Undecided	—	Undecided
$\sim P_1$	True	—	True
	False	—	False
	Undecided	—	True
$\rightharpoonup P_1$	True	—	False
	False	—	True
	Undecided	—	True

$\wedge$ is AND, $\vee$ is inclusive OR, $\otimes$ is exclusive OR and $\neg$ is NOT. New symbols $\sim$ and $\rightharpoonup$ mean POSSIBLY and POSSIBLY NOT, respectively. All combinations not specfified here are the same as for classical propositional logic.

tify two sets of values for the state variable vector $\boldsymbol{x}^{(s)}(t)$. The first is $\chi_T^{(s)}(t)$, the set of values such that the transition relation is TRUE, and the second is $\chi_F^{(s)}$, the set of values such that the transition relation is FALSE. The outcome region is then split into two subregions: $r_1 = [s,\ \chi_T^{(s)}(t)]$ which undergoes a transition to state s', and $r_2 = (s,\ \chi_F^{(s)})$ which remains in state s. At time $t + \epsilon$, the system is then in a region composed of two simple regions in different states s and s'.

- If the condition $l_{ss'}$ is nonlinear, it is generally not possible to determine critical values of the states and inputs. Thus, two possibilities must be considered: the system may remain in state s or undergo a transition to s'. Two identical simple regions are considered at time t: $r_1 = r_2 = r$. Region r_1 is allowed to progress to time $t + \epsilon$, while remaining in state s. Region r_2 undergoes an instantaneous transition to state s'. Once again, the system is composed of two simple regions in two states.

Initialization Relations. When a region $r = (s,\ [\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t)] \times [\underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t)])$ undergoes a transition to state s' at time t, the vectors $\boldsymbol{x}^{(s')}(t)$ and $\boldsymbol{y}^{(s')}(t)$ must be initialized. More specifically, bounds on these vectors are required. They are obtained through an interval solution of the initialization relation

$$0 \in I_{ss'}\Big(\Big[\underline{\dot{\boldsymbol{x}}}^{(s)}, \bar{\dot{\boldsymbol{x}}}^{(s)}\Big], \Big[\underline{\dot{\boldsymbol{x}}}^{(s')}(t), \bar{\dot{\boldsymbol{x}}}^{(s')}(t)\Big], \Big[\underline{\boldsymbol{x}}^{(s)}, \bar{\boldsymbol{x}}^{(s)}\Big], \Big[\underline{\boldsymbol{x}}^{(s')}, \bar{\boldsymbol{x}}^{(s')}\Big], \Big[\underline{\boldsymbol{y}}^{(s)}, \bar{\boldsymbol{y}}^{(s)}\Big], \Big[\underline{\boldsymbol{y}}^{(s')}, \underline{\boldsymbol{y}}^{(s')}\Big], [\underline{\boldsymbol{u}}, \bar{\boldsymbol{u}}]\Big).$$

A Newton-based interval solution technique such as that described in Kearfott and Novoa III (1990), Neumaier (1990), and Schnepper and Stadtherr (1996) can be used. In many cases, initialization relations require a simple assignment or evaluation of equations rather than the solution of a nonlinear system of equations.

Summary of Model. Based on the features described so far, the "run" of a system from region to region can be obtained through a hybrid state and region transition model which combines the following components:

A set $\mathcal{S}$ of vertices called *locations* or *states*.

A set of *continuous variables* and *equations* corresponding to each location. The equations take on the form

$$\underline{\dot{\boldsymbol{x}}}^{(s)}(t) = \underline{\boldsymbol{f}}^{(s)}\Big(\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t), \underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t), \underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)\Big) = 0$$

$$\bar{\dot{\boldsymbol{x}}}^{(s)}(t) = \bar{\boldsymbol{f}}^{(s)}\Big(\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t), \underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t), \underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)\Big) = 0$$

$$\underline{\boldsymbol{g}}\Big(\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t), \underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t), \underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)\Big) \le 0$$

$$\bar{\boldsymbol{g}}\Big(\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t), \underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t), \underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)\Big) \ge 0$$

$$\underline{\boldsymbol{x}}^{(s)}(0) = \underline{\boldsymbol{x}}_0^{(s)}$$

$$\bar{\boldsymbol{x}}^{(s)}(0) = \bar{\boldsymbol{x}}_0^{(s)}$$

$$t \in [0, T] \qquad (11)$$

where $[\underline{\boldsymbol{x}}^{(s)}(t),\ \bar{\boldsymbol{x}}^{(s)}(t)]$ is the set of trajectories of the state variables for the system in state s, $[\underline{\boldsymbol{y}}^{(s)}(t),\ \bar{\boldsymbol{y}}^{(s)}(t)]$ is the set of trajectories of the algebraic variables for the system in state s, $[\underline{\boldsymbol{u}}^{(s)},\ \bar{\boldsymbol{u}}^{(s)}(t)]$ is the set of input ranges, and $[\underline{\boldsymbol{x}}_0^{(s)},\ \bar{\boldsymbol{x}}_0^{(s)}]$ is the set of initial conditions.

A set of *edges* $\mathcal{E}$ that correspond to transitions between locations. A transition $e = (s,\ s',\ L_{ss'},\ I_{ss'})$ consists of:

- a source state or location s;
- a target state s',
- a transition relation $L_{ss'}([\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t)], [\underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t)], [\underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)])$, which is a logical condition which can be TRUE, FALSE, or UNDECIDED;
- a state initialization relation

$$0 \in I_{ss'}\Big(\Big[\underline{\dot{\boldsymbol{x}}}^{(s)}(t), \bar{\dot{\boldsymbol{x}}}^{(s)}(t)\Big], \Big[\underline{\dot{\boldsymbol{x}}}^{(s')}(t), \bar{\dot{\boldsymbol{x}}}^{(s')}(t)\Big], \Big[\underline{\boldsymbol{x}}^{(s)}(t), \bar{\boldsymbol{x}}^{(s)}(t)\Big], \Big[\underline{\boldsymbol{x}}^{(s')}(t), \bar{\boldsymbol{x}}^{(s')}(t)\Big], \Big[\underline{\boldsymbol{y}}^{(s)}(t), \bar{\boldsymbol{y}}^{(s)}(t)\Big], \Big[\underline{\boldsymbol{y}}^{(s')}(t), \bar{\boldsymbol{y}}^{(s')}(t)\Big], [\underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)]\Big). \qquad (12)$$

Simulations with the region-transition model

Having described the formalism of the region-transition model (RTM), we now turn our attention to its use as a simulation tool. In this section, we first describe an algorithm that evaluates a run of the system. We then discuss the issue of overestimation of the outcome region and show how the model's properties can be used to overcome this problem. Finally, we study an application of the proposed framework on the simulation of a nonlinear batch reactor.

Algorithmic Statement. Let us consider at time t_1 a system in the region R_{t_1} defined by a union of simple regions, $\mathcal{R}_{t_1} = \cup_{s \in S_{t_1}} (s,\ \chi_{t_1}^{(s)} = [\underline{\boldsymbol{x}}^{(s)}(t_1),\ \bar{\boldsymbol{x}}^{(s)}(t_1)] \times [\underline{\boldsymbol{y}}^{(s)}(t_1),\ \bar{\boldsymbol{y}}^{(s)}(t_1)])$. We aim to determine the region $\mathcal{R}_{t_2}$ in which the system will be found at time $t_2 = t_1 + \epsilon$ given the input set $\mathcal{U}_{t_1} = [\underline{\boldsymbol{u}}(t_1), \bar{\boldsymbol{u}}(t_1)]$. We consider each simple region in $\mathcal{R}_{t_1}$ and compute its evolution based on the model equations and the occur-

rence of possible transitions. Mathematically, $\mathfrak{R}_{t_2}$ is expressed as

$$\mathfrak{R}_{t_2} = \bigcup_{s \in \mathfrak{s}_{t_1}} \bigcup_{(\boldsymbol{x}^{(s)}(t_1), \boldsymbol{y}^{(s)}(t_1)) \in \chi_{t_1}^{(s)}} \bigcup_{s' \in \mathfrak{J}_s} \times \begin{pmatrix} \sim L_{ss't_1}\left(\chi_{t_1}^{(s)}, \mathfrak{U}_{t_1}\right) \\ 0 \in \boldsymbol{I}_{ss'}\left(\chi_{t_1}^{(s')}, \chi_{t_1}^{(s)}, \mathfrak{U}_{t_1}\right) \\ \wedge \\ \underline{\boldsymbol{x}}_{t_2}^{(s')} = \underline{\boldsymbol{f}}^{(s')}\left(\chi_{t_1}^{(s')}, \mathfrak{U}_{t_1}\right) \\ \bar{\boldsymbol{x}}_{t_2}^{(s')} = \bar{\boldsymbol{f}}^{(s')}\left(\chi_{t_1}^{(s')}, \mathfrak{U}_{t_1}\right) \\ \underline{\boldsymbol{g}}^{(s')}\left(\chi_{t_1}^{(s')}, \mathfrak{U}_{t_1}\right) \le 0 \\ \wedge \\ \bar{\boldsymbol{g}}^{(s')}\left(\chi_{t_1}^{(s')}, \mathfrak{U}_{t_1}\right) \ge 0 \end{pmatrix} \quad (13)$$

In the above representation, the set $\mathfrak{J}_s$ denotes the set of possible transitions from state s. The truth or partial truth of the transition condition for each transition is asserted through $\sim L_{ss't_1}(\chi_{t_1}^{(s)}, \mathfrak{U}_{t_1})$. If this expression is TRUE, the corresponding initialization relations are applied $[0 \in \boldsymbol{I}_{ss'}(\chi_{t_1}^{(s')}, \chi_{t_1}^{(s)}, \mathfrak{U}_{t_1})]$. The relevant model equations are finally used to compute the outcome. The region-transition model therefore recursively predicts the run of the system under uncertainty, given an initial region and disturbances.

Overestimation of the Outcome Region. When the region transition model is linear and separable in the variables and the uncertain parameters, the *exact* progression from region to region can be calculated. However, when nonlinearities are introduced, the outcome region computed with the RTM may represent an overestimate of the actual outcome region. In general, the degree of overestimation depends on the nonlinearity of the model and on the size of the initial box.

While conservatism is a desirable feature in the context of safety analysis, our aim is to develop a methodology that leads to a realistic assessment. Two important properties of the RTM, inclusion monotonicity and convergence, are used to ensure that this goal is achieved.

Inclusion Monotonicity of the Region Transition Model. Given two initial regions $\mathfrak{R}_{a,0}$ and $\mathfrak{R}_{b,0}$ at time t_0, such that $\mathfrak{R}_{b,0} \subseteq \mathfrak{R}_{a,0}$, a time $t_1 > t_0$, and a set of inputs $[\underline{\boldsymbol{u}}(t), \bar{\boldsymbol{u}}(t)]$, $t \in [t_0, t_1]$, the actual outcome regions $\mathfrak{R}_{a,1}^*$ and $\mathfrak{R}_{b,1}^*$ at time t_1 are such that $\mathfrak{R}_{b,1}^* \subseteq \mathfrak{R}_{a,1}^*$ and the regions computed using the region-transition model $\mathfrak{R}_{a,1}$ and $\mathfrak{R}_{b,1}$ are such that $\mathfrak{R}_{b,1} \subseteq \mathfrak{R}_{a,1}$. This property derives from the monotonicity of interval inclusions (Moore, 1966).

Convergence of the Region Transition Model. The width of an interval vector $X = [\underline{x}, \bar{x}]$ of dimension n is given by $w(X) = \max_{i=1,\ldots,N} (\bar{x}_i - \underline{x}_i)$. The width of a general region $\mathfrak{R}$ containing hyper-rectangles in N states is given by $w(\mathfrak{R}) = \max_{s=1,\ldots,N} \max_{i=1,\ldots,n_s} (\bar{x}_i^{(s)} - \underline{x}_i^{(s)})$. Interval inclusions such as natural extensions possess a convergence property expressed as follows. Consider a real-valued function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$. Let $F(X)$ denote an inclusion of $f(\boldsymbol{x})$ over X. Let $f(X)$ denote the exact set of values of $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$. Then, $\lim_{w(X) \to 0} F(X) = f(X)$. Similarly, given a region-transition model *RTM* based on interval arithmetic, and the exact outcome $\mathfrak{R}_*$ of a region $\mathfrak{R}$, the following convergence property holds

$$\lim_{w(\mathfrak{R}) \to 0} RTM(\mathfrak{R}) = \mathfrak{R}_*. \quad (14)$$

Based on the monotonicity and convergence properties, partitioning of the input space (initial conditions and external inputs) can be used to reduce overestimation of the outcome region. Thus, if two regions $\mathfrak{R}_1$ and $\mathfrak{R}_2$ form a partition of a region $\mathfrak{R}$ with exact outcome $\mathfrak{R}_*$, the following relationship is satisfied

$$R_* \subseteq RTM(\mathfrak{R}_1) \cup RTM(\mathfrak{R}_2) \subseteq RTM(\mathfrak{R}), \quad (15)$$

As the partition of $\mathfrak{R}$ becomes more refined, convergence to $\mathfrak{R}_*$ is achieved

$$\mathfrak{R}_* = \lim_{n \to \infty} \bigcup_{i=1}^{n} RTM(\mathfrak{R}_i) \subseteq RTM(\mathfrak{R}) \quad (16)$$

The computational expense incurred during the evaluation of an outcome region transition model increases rapidly with the number of subregions. As a result, a compromise must be found between accuracy and effort. It is difficult to identify *a priori* the minimum number of subregions necessary to achieve a given level of accuracy. An optimal partitioning pattern will typically not be uniform over the input and parameter spaces as certain quantities contribute more to the system's nonlinearity than others, and certain regions of space will exhibit more nonlinear behavior than others. The issue of how to identify the inputs and parameters which lead to the largest overestimation will be addressed in the next section.

Batch reactor example

A nonlinear example is used to illustrate the concepts presented so far. We are concerned with a first-order exothermic reaction $A \to B$ taking place in a batch reactor fitted with a cooling jacket, subject to typical assumptions. The model for this system is given by the following mass and energy balances

$$\frac{dX}{dt} = k_0 \exp\left(-\frac{E_a}{RT}\right)(1 - X)$$

$$C_{A0} V C_p \frac{dT}{dt} = Q - \Delta H_R k_0 \exp\left(-\frac{E_a}{RT}\right)(1 - X) C_{A0} V \quad (17)$$

where X is the conversion, T is the reactor temperature in K, C_{A0} is the initial concentration of reactant A (10 mol/m^3), V is the reactor volume (0.1 m^3), C_p is the total heat capacity of the reactor contents, assumed constant (60 J/mol K), k_0 is the kinetic rate constant (0.022 s^{-1}), E_a is the activation energy (6,000 J/mol), R is the gas constant, and ΔH_R is the heat of reaction ($-140{,}000$ J/mol). Q is the rate of heat removal in W. For simplicity, it is assumed to be a constant at -385 W if the reactor temperature is greater than the coolant temperature $T_a = 290$ K and to be equal to zero if the reactor temperature is greater than or equal to T_a.

The behavior of the process is studied by obtaining bounding trajectories for the two state variables X and T. The maximum batch time considered is one hour. The initial batch temperature T_0 is taken to vary between 350 K and 360 K. The initial conversion is taken as zero.

Equations 17 are discretized as

$$X_{i+1} = k_0 \exp\left(-\frac{E_a}{RT_i}\right)(1-X_i)\,\delta t + X_i$$

$$T_{i+1} = \frac{Q\delta t}{C_{A0}VC_p} - \Delta H_R \frac{k_0}{C_p}\exp\left(-\frac{E_a}{RT_i}\right)(1-X_i)\,\delta t + T_i \tag{18}$$

Several bounding strategies are used in the RTM.

Natural Interval Extension. Bounds for the above equations can be derived by natural interval extensions. Exploiting the characteristics of the data to simplify the expressions, we obtain the following

$$\underline{X}_{i+1} = k_0 \underline{A}\,\delta t + \underline{X}_i$$

$$\overline{X}_{i+1} = k_0 \overline{A}\,\delta t + \overline{X}_i$$

$$\underline{T}_{i+1} = \frac{\underline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R k_0}{C_p}\underline{A}\,\delta t + \underline{T}_i$$

$$\overline{T}_{i+1} = \frac{\overline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R k_0}{C_p}\overline{A}\,\delta t + \overline{T}_i \tag{19}$$

where

$$\underline{A} = \min\left\{\exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\underline{X}_i), \exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\overline{X}_i),\right.$$
$$\left.\exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\underline{X}_i), \exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\overline{X}_i)\right\} \tag{20}$$

and

$$\overline{A} = \max\left\{\exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\underline{X}_i), \exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\overline{X}_i),\right.$$
$$\left.\exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\underline{X}_i), \exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\overline{X}_i)\right\} \tag{21}$$

Examining the form of the above equations, we note that overestimation is introduced as each occurrence of the state variables is treated independently. Thus, given $T_i \geq 0$ and $0 \leq [\underline{X}_i, \overline{X}_i] \leq 1$, $\underline{A}$ is equal to $\exp(-E_a/R\underline{T}_i)(1-\overline{X}_i)$ and therefore conversion appears both as $\underline{X}_i$ and $\overline{X}_i$ in the equation for $\underline{X}_{i+1}$. This results in a relaxation of the physical constraints on the variables and there is no guarantee that the bounds on conversion will be between the known bounds of 0 and 1, or that the bounds on the temperature will remain positive.

Use of Physical Bounds. An improvement on the bounds above can be obtained by ensuring that $[\underline{X}_i, \overline{X}_i] \in [0, 1]$ for all i and that the temperature of the reactor remains greater than the coolant temperature, T_a ($T_i \geq T_a$ for all i). This removes the possibility of sign changes in $(1-\underline{X}_i)$, $(1-\overline{X}_i)$, $-E_a/(R\underline{T}_i)$ and $-E_a/(R\overline{T}_i)$ and yields the following bounding system

$$\underline{X}_{i+1} = \max\left\{0, \min\left\{1, k_0 \exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\overline{X}_i)\,\delta t + \underline{X}_i\right\}\right\}$$

$$\overline{X}_{i+1} = \max\left\{0, \min\left\{1, k_0 \exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\underline{X}_i)\,\delta t + \overline{X}_i\right\},\right\}$$

$$\underline{T}_{i+1} = \max\left\{T_a, \frac{\underline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R k_0}{C_p}\exp\left(-\frac{E_a}{R\underline{T}_i}\right)\right.$$
$$\left.\times(1-\overline{X}_i)\,\delta t + \underline{T}_i\right\}$$

$$\overline{T}_{i+1} = \max\left\{T_a, \frac{\overline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R k_0}{C_p}\exp\left(-\frac{E_a}{R\overline{T}_i}\right)\right.$$
$$\left.\times(1-\underline{X}_i)\,\delta t + \overline{T}_i\right\} \tag{22}$$

This bounding system takes the general form

$$\underline{X}_{i+1} = \underline{X}_{i+1}(\underline{X}_i, \overline{X}_i, \underline{T}_i)$$

$$\overline{X}_{i+1} = \overline{X}_{i+1}(\underline{X}_i, \overline{X}_i, \overline{T}_i)$$

$$\underline{T}_{i+1} = \underline{T}_{i+1}(\overline{X}_i, \underline{T}_i, \underline{Q})$$

$$\overline{T}_{i+1} = \overline{T}_{i+1}(\underline{X}_i, \overline{T}_i, \overline{Q}). \tag{23}$$

Monotonicity Analysis. With further mathematical analysis of the righthand sides of the discretized system (Eq. 18), it is possible to identify monotonic behavior in the equations by examining the sign of several partial derivatives

$$\frac{\partial X_{i+1}}{\partial X_i} \geq 0 \text{ if } k_0\delta t \leq 1; \quad \frac{\partial X_{i+1}}{\partial T_i} \geq 0 \text{ if } X_i \leq 1; \quad \frac{\partial X_{i+1}}{\partial Q} = 0$$

$$\frac{\partial T_{i+1}}{\partial X_i} \leq 0 \qquad \frac{\partial T_{i+1}}{\partial T_i} \geq 0 \text{ if } X_i \leq 1; \quad \frac{\partial T_{i+1}}{\partial Q} \geq 0$$

This results in the following bounding system

$$\underline{X}_{i+1} = \max\left\{0, \min\left\{1, k_0 \exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\underline{X}_i)\,\delta t + \underline{X}_i\right\}\right\}$$

$$\overline{X}_{i+1} = \max\left\{0, \min\left\{1, k_0 \exp\left(1-\frac{E_a}{R\overline{T}_i}\right)(1-\overline{X}_i)\,\delta t + \overline{X}_i\right\}\right\}$$

$$\underline{T}_{i+1}$$

$$= \max\left\{T_a, \frac{\underline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R k_0}{C_p}\exp\left(-\frac{E_a}{R\underline{T}_i}\right)\left(1-\overline{X}_i\right)\delta t + \underline{T}_i\right\}$$

$$\overline{T}_{i+1}$$

$$= \max\left\{T_a, \frac{\overline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R k_0}{C_p}\exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\underline{X}_i)\delta t + \overline{T}_i\right\} \tag{24}$$

This system has the general form

$$\begin{aligned}
\underline{X}_{i+1} &= \underline{X}_{i+1}(\underline{X}_i, \underline{T}_i)\\
\overline{X}_{i+1} &= \overline{X}_{i+1}\left(\overline{X}_i, \overline{T}_i\right)\\
\underline{T}_{i+1} &= \underline{T}_{i+1}\left(\overline{X}_i, \underline{T}_i, \underline{Q}\right)\\
\overline{T}_{i+1} &= \overline{T}_{i+1}\left(\underline{X}_i, \overline{T}_i, \overline{Q}\right)
\end{aligned} \tag{25}$$

The state variables appearing in the discretized system are no longer treated as independent and the bounding system (Eq. 25) can be expected to produce less overestimation than the bounding system (Eq. 23). However, because the equations for lower and upper bounding trajectories are coupled through the dependence of $\underline{T}_{i+1}$ on $\overline{X}_i$ and that of $\overline{T}_{i+1}$ on $\underline{X}_i$, the bounding system will still lead to an overestimate of the exact boundary of the original set of equations.

Comparison of Bounding Strategies. In order to compare the accuracy of the different bounding strategies, a small input region $[\underline{T}_0, \overline{T}_0] = [350, 360]$ is considered. The resulting bounds on the conversion X and the temperature T are shown in Figure 3. Bounding strategy (Eq. 19) leads to a rapid divergence of the boundaries and to unphysical values for the conversion and temperature. The introduction of physical constraints in bounding strategy (Eq. 22) results in a marked improvement of the bounds on conversion and the lower bound on temperature. However, the upper bound on temperature is still unrealistic. When monotonicity analysis is used as in strategy (Eq. 24), the upper bound on the temperature exhibits more realistic behavior, and a further improvement on the conversion bounds is achieved. Figure 4 shows several point simulations for the batch reactor with initial temperature T_0 chosen between 350 K to 360 K at 2 K intervals. All trajectories are within the boundaries obtained with Eq. 24, and the figure illustrates the degree of overestimation that is inherent in the technique.

In order to reduce the degree of overestimation, the inclusion monotonicity property can be exploited. Thus, the input region is divided into two smaller regions $[\underline{T}_{0,1}, \overline{T}_{0,1}] = [350, 355]$ and $[\underline{T}_{0,2}, \overline{T}_{0,2}] = [355, 360]$. Simulations are run for these two regions using bounding system Eq. 24. As shown in Figure 5, the two output regions are within the boundaries obtained for the larger single region. The union of the two output regions is significantly smaller than the output region obtained previously.

Modeling Uncertainty. If uncertainty is introduced in the model parameters in the form of intervals, the bounding system needs to be modified accordingly. For instance, given $k_0 \in [\underline{k}_0, \overline{k}_0]$, the bounding system (Eq. 24) becomes

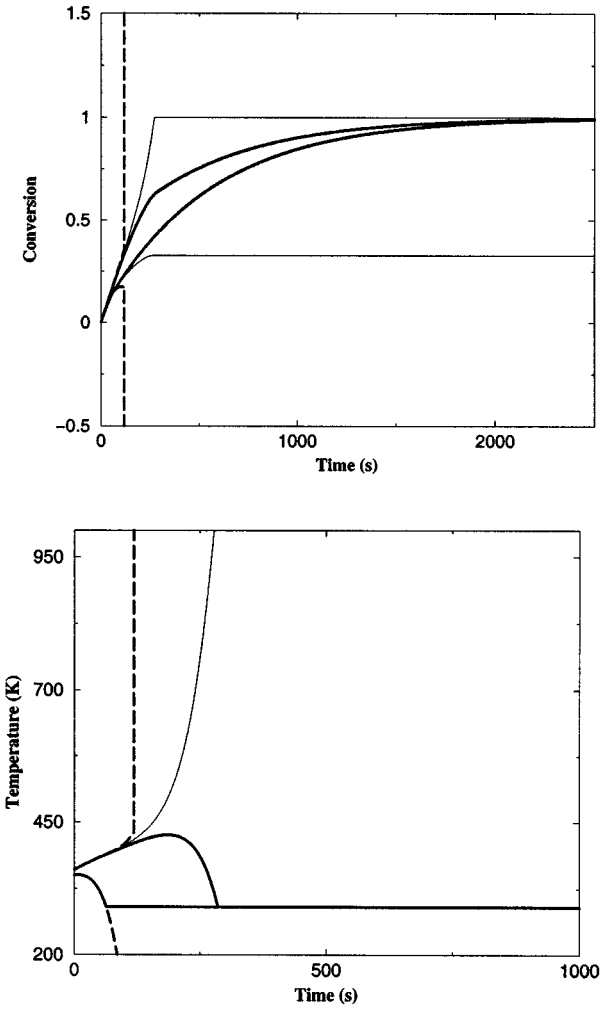


Figure 3. Boundaries for conversion and temperature using $T_0 \in [350, 360]$, as computed with bounding strategies (Eq. 19) (– –) (Eq. 22) (—) (Eq. 24) (—).

$$\underline{X}_{i+1} = \underline{k}_0 \exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\underline{X}_i)\delta t + \underline{X}_i$$

$$\overline{X}_{i+1} = \overline{k}_0 \exp\left(-\frac{E_a}{R\overline{T}_i}\right)\left(1-\overline{X}_i\right)\delta t + \overline{X}_i$$

$$\underline{T}_{i+1} = \frac{\underline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R \underline{k}_0}{C_p}\exp\left(-\frac{E_a}{R\underline{T}_i}\right)\left(1-\overline{X}_i\right)\delta t + \underline{T}_i$$

$$\overline{T}_{i+1} = \frac{\overline{Q}\delta t}{C_{A0}VC_p} - \frac{\Delta H_R \overline{k}_0}{C_p}\exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\underline{X}_i)\delta t + \overline{T}_i \tag{26}$$

If the uncertainty in k_0 is $\pm 5\%$, the corresponding interval is $k_0 \in [0.0209, 0.0231]$. Figure 6 shows the effect of this uncertainty on the computed outcome region for the input region

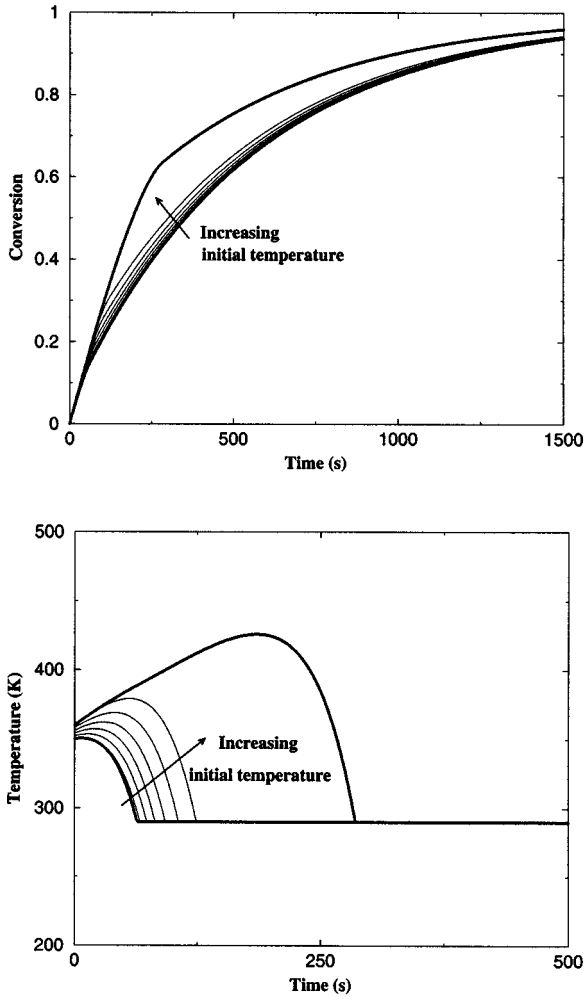


Figure 4. Boundaries (—) and point simulations (—) for conversion and temperature using $T_0 \in$ [350, 360].

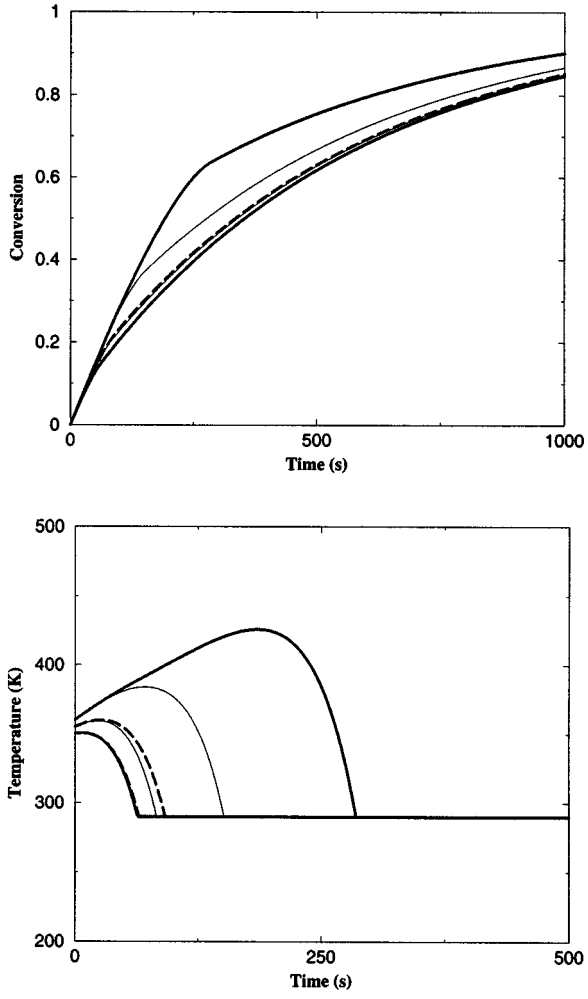


Figure 5. Boundaries with $T_0 \in$ [350, 360] (—), $T_0 \in$ [350, 355] (– –), and $T_0 \in$ [355, 360] (—).

used previously. The uncertainty in k_0 is seen to have a dramatic effect on the predicted outcome region, highlighting the sensitivity of the reactor to the reaction rate constant. The behavior shown is partly a result of overestimation which can be reduced by considering the union of several runs with smaller ranges of k_0.

Safety Analysis with the Region Transition Model

This section focuses on the full characterization of the operating space of a process defined through the proposed region-transition model. To achieve this goal, we perform a reachability analysis so that all outcome regions that can be reached given certain initial conditions and disturbances are identified. It has been shown, however, that even for hybrid systems that satisfy the strict assumption of linearity, reachability is not decidable (Alur et al., 1993; Kesten et al., 1993). In other words, no *general* methodology for infinite-time safety analysis can be devised, although in practice it is possible to identify infinite-time safety for certain nonlinear systems (Dimitriadis, 1997). This leads us to impose a time horizon on the problem. However, we will see that in practice, it is sometimes possible to guarantee infinite-time safety, provided that the system exhibits convergent behavior. In this case, the "invariant set" of the system has been found.

A number of algorithms for safety verification of state-transition systems have been proposed based on the notion of approximations of the reachable states (Alur et al., 1995, 1996b). However, the models were either linear or with bounded slopes. On the other hand, for the case of entirely continuous systems, the control community has dedicated much effort to devising methods that identify the invariant set of inputs for which the system always reaches the same pre-defined region (Chen, 1984; Jaulin and Walter, 1997). For instance, Jaulin and Walter (1997) proposed an algorithm based on interval arithmetic to identify the invariant sets for a nonlinear deterministic system.

In this section, we describe an algorithm which, given a region-transition model for a system, a time-horizon, and a set of initial conditions and disturbances, identifies the subsets of initial conditions and disturbances which ensure safe operation and those which lead to unsafe operation over the course of the time horizon.

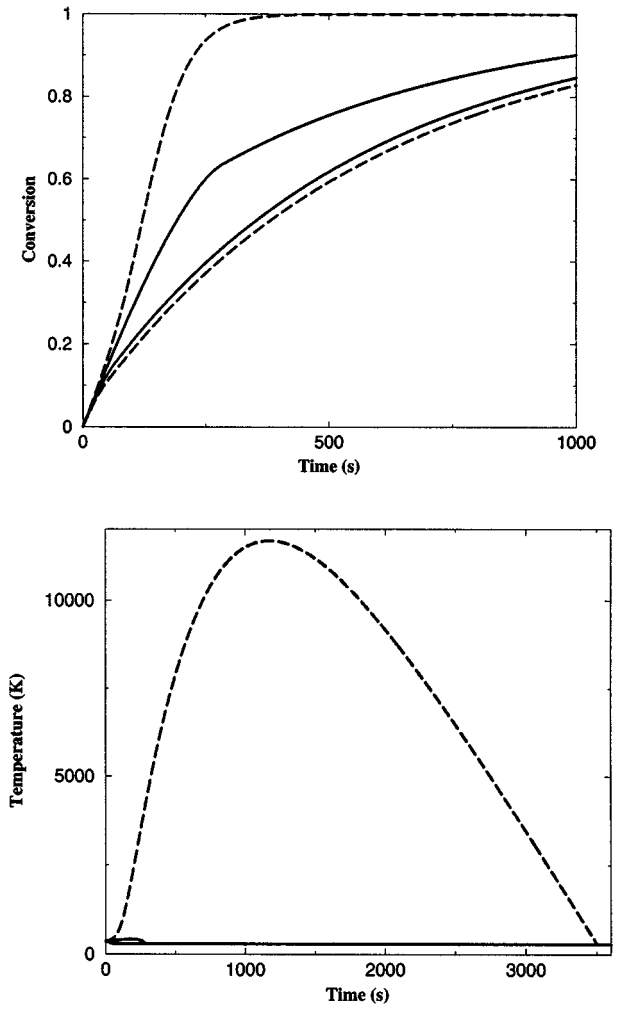


Figure 6. Boundaries with $k_0 = 0.022\text{s}^{-1}$ (—) and $k_0 \in [0.0209, 0.0231]$ (– –) for $T_0 \in [350, 360]$.

Initial conditions and disturbances

The initial conditions for the system are given in terms of a region $\mathfrak{R}_I = \bigcup_{s \in \mathfrak{S}_I} (s, \chi_I^{(s)})$, where $\mathfrak{S}_I$ is the set of initial states, and $\chi_I^{(s)}$ is the set of initial hyper-rectangles for the system in state s. In most cases, the initial conditions consist of a single hyper-rectangle in a safe state s. The disturbances $\boldsymbol{u}(t)$ are represented by a set of hyper-rectangles $\mathfrak{R}_D$. The overall region of interest is therefore given by $\mathfrak{R}_0 = \mathfrak{R}_I \times R_D$.

Algorithmic procedure

General Algorithmic Structure. In order to determine which subregions of the initial condition and disturbance region $\mathfrak{R}_0$ result in unsafe operation before the end of the time horizon, a branch and bound strategy combining the partitioning of $\mathfrak{R}_0$ and the simulation of subregions is used to build up three sets of regions:

- $\mathfrak{R}_U$, the set of regions of $\mathfrak{R}_0$ which have an unsafe outcome region by the end of the time horizon;
- $\mathfrak{R}_S$, the set of regions of $\mathfrak{R}_0$ which have a safe outcome region by the end of the time horizon; and
- $\mathfrak{R}_{UD}$, the set of regions of $\mathfrak{R}_0$ which have not been proven to belong to $\mathfrak{R}_U$ or $\mathfrak{R}_S$. These regions are identified as *undecidable*.

At the end of the safety analysis procedure, the three sets $\mathfrak{R}_U$, $\mathfrak{R}_S$ and $\mathfrak{R}_{UD}$ form a partition of $\mathfrak{R}_0$. The initial region that results in unsafe behavior is underestimated by $\mathfrak{R}_U$ and overestimated by $\mathfrak{R}_U \cup \mathfrak{R}_{UD}$. The size of $\mathfrak{R}_{UD}$ depends on the uncertainty present in the model, the quality of the bounding trajectories, and a user-specified tolerance.

At the beginning of the algorithm, $\mathfrak{R}_0$ is labeled as UNDECIDED and is identified as the single element of a list of UNDECIDED regions, referred to as the STACK. The three lists representing $\mathfrak{R}_U$, $\mathfrak{R}_S$ and $\mathfrak{R}_{UD}$ are empty. The algorithm is composed of an outer loop which identifies subregions of $\mathfrak{R}_0$ whose safety should be asserted, and an inner loop which determines the safety of the given input region based on a conservative estimate of reachable regions through an RTM simulation. In the outer loop, a region $R_{\text{in}} \subseteq \mathfrak{R}_0$ is taken from the STACK. It is passed to the inner loop which assesses the safety of its reachable (outcome) region $\mathfrak{R}_{\text{out}}$. Three types of outcome regions are possible: entirely SAFE, entirely UNSAFE, or PARTIALLY SAFE. This information is returned to the outer loop and R_{in} is assigned to one of four lists: if R_{out} is entirely SAFE, R_{in} is added to the $\mathfrak{R}_S$ list. If R_{out} is entirely UNSAFE, R_{in} is added to the $\mathfrak{R}_U$ list. If it is PARTIALLY SAFE, its size $w(R_{\text{in}})$ is computed. If $w(R_{\text{in}})$ is less than a user-specified tolerance, R_{in} is added to the $\mathfrak{R}_{UD}$ list as further computations to determine its safety are deemed unnecessary. Otherwise, R_{in} is split in a number of subregions, and these subregions are added to the STACK. This procedure is repeated until the STACK is empty.

The pseudo-code for *safety-analysis*(), the outer loop of the algorithm, is shown in Table 2. The pseudo-code for the inner loop, *reachable-region-safety*(REGION), is shown in Table 3. The procedure *region-safety*(REGION), used in Table 3, determines whether the specified region contains only safe states, only unsafe states, or a combination of the two and returns this information (SAFE, UNSAFE, PARTIALLY SAFE, respectively).

Table 2. Pseudo-Code for the Outer Loop of the Safety Verification Algorithm

```
PROCEDURE safety-analysis()
  Set tolerance ϵ;
  Set stack counter i = 1;
  Initialize regions and lists
    Initial region ℜ0 = ℜI × ℜD;
    Initialize lists L_ℜS = ∅, L_ℜU = ∅, L_ℜUD = ∅;
    Set STACK = {ℜ0};
  DO (STACK ≠ ∅)
    Unstack ℜi; Set R_in = ℜi and i = i − 1;
    Compute safety = reachable-region-safety (R_in);
    IF safety == SAFE, add R_in to L_ℜS;
    ELSE IF safety == UNSAFE, add R_in to L_ℜU;
    ELSE IF w(R_in) < ϵ, add R_in to L_ℜUD;
    ELSE split R_in into n regions and add new regions
      to STACK; Set i = i + n;
  OD;
  RETURN (L_ℜS, L_ℜU, L_ℜUD);
END safety-analysis;
```

Table 3. Pseudo-Code for the Inner Loop of the Safety Verification Algorithm

```
PROCEDURE reachable-region-safety (R_in)
  Set time horizon H, step size δt and number of time steps
    N = H/δt;
  Set counter i = 0;
  Initialize input region: R_0 = R_in;
  Set safety = region-safety (R_0);
  IF safety == UNSAFE, return (safety);
  DO (i ≤ N)
    Compute successor region R_{i+1} of R_i using RTM;
    safety = region-safety(R_{i+1});
    IF safety == UNSAFE, return (UNSAFE);
    Set i = i + 1;
  OD;
  RETURN (safety);
END reachable-region-safety;
```

Reducing Overestimation. The overestimation resulting from the *reachable-region-safety*(REGION) procedure will lead to the classification of entirely SAFE or UNSAFE regions as PARTIALLY SAFE. It is possible to refine the outcome of the *reachable-region-safety*(REGION) procedure by implementing an internal branching scheme, which would consider subregions by reducing the range of the initial conditions, disturbances, and uncertain model parameters. However, provided the PARTIALLY SAFE regions identified by *reachable-region-safety*(REGION) are sufficiently large, they will be split into smaller regions during the *safety-analysis*() procedure and retained for further analysis. Thus, the partitioning step in the outer loop of the algorithm, which is primarily designed to separate safe and unsafe subregions of the input region, also allows the reduction of the overestimation resulting from the bounding strategy. Uncertain parameters, on the other hand, are not branched on at the level of the outer loop. Therefore, a decrease of the overestimation arising from this uncertainty can only be achieved if a branching strategy is implemented in the inner loop. An example of such an approach will be presented in the next subsection.

Region Splitting. The nature of the algorithm is computationally intensive as the entire operating space must be resolved. Given n_I state variables and n_D disturbances, ranges for the initial conditions $[\underline{x}_i(0),\ \bar{x}_i(0)]$, $i = 1,\ \ldots,\ n_I$, ranges for the disturbances $[\underline{u}_i,\ \bar{u}_i]$, $i = 1,\ \ldots,\ n_D$, and a tolerance ϵ, the final partition of the operating space may, in the worst case, consist exclusively of regions of size ϵ^n, where $n = n_I + n_D$. The maximum number of such regions depends on the size of the original region and is given by

$$N_R = \epsilon^{-n} \prod_{i=1}^{n_I} \left(\bar{x}_i(0) - \underline{x}_i(0)\right) \prod_{j=1}^{n_D} \left(\bar{u}_j - \underline{u}_j\right).$$

Assuming that a region is split into p subregions at each iteration of the algorithm, the total number of iterations is given by

$$N_{\text{Iter}} = \sum_{i=1}^{\log_p(N_R)} p^k = \frac{pN_R - 1}{p-1}.$$

For a small example with one state variable, one disturbance, an initial region of size 1, $\epsilon = 0.0001$ and $p = 4$, the worst-case behavior gives $N_{\text{Iter}} = 1{,}333{,}333$.

The splitting, or branching, of regions to identify safe initial regions or reduce overestimation plays a crucial role in avoiding worst-case behavior. Among several possible alternative branching strategies which have been used in the context of global optimization (see, for example, Adjiman et al. (1998)), we will examine the effect of the following three:

n-Branching. Each region is split into n subregions where n is the total number of input quantities (initial conditions and disturbances).

Longest-Axis Branching. Each region is split into 2 subregions. The input variable which has least been branched on so far is selected.

Sensitivity-Driven Branching. Each region is split into 2 subregions. Certain input quantities have a larger effect on the overall safety of the process than others and their identification can help improve the performance of the algorithm. Since the transitions from safe to unsafe states occur upon a change in the value of some of the state variables, we can focus on the input quantities which affect these state variables most. To do so, we compute the sensitivities of the state variables to the inputs at a given time t

$$\frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{x}(0)} \text{ and } \frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{u}}. \tag{27}$$

These sensitivities are, of course, local values which depend on the choice of t, $\boldsymbol{x}(0)$ and $\boldsymbol{u}$. In practice, we choose t to be the smallest of the time of the first transition and the final time, and $\boldsymbol{x}(0)$ and $\boldsymbol{u}$ to be midpoints of the current region of interest

$$\boldsymbol{x}(0) = \frac{\bar{\boldsymbol{x}}(0) - \underline{\boldsymbol{x}}(0)}{2}, \qquad \boldsymbol{u} = \frac{\bar{\boldsymbol{u}} - \underline{\boldsymbol{u}}}{2}. \tag{28}$$

The sensitivities can be calculated through finite differencing, requiring $(n_I + n_D + 1)$ integrations of a system of n_I differential equations, or through the integration of a larger system of $n_I(n_I + n_D)$ differential equations. The additional cost of sensitivity computations is therefore high, and their effect on algorithmic performance will be investigated.

Case studies

Three case studies are presented to illustrate the use of the region transition model and the safety verification algorithm. All three are concerned with the analysis of hazard for open-loop behavior, as this constitutes the first step in establishing the inherent safety characteristics of a process. The first example is a simple linear problem describing tank overflow/underflow. It provides some insight into the methodology through comparisons with worst-case analysis. The second example revisits the exothermic reaction in a batch reactor discussed earlier and illustrates how nonlinearity and uncertainty are handled in the context of safety. Finally, the third example examines an exothermic reaction in a CSTR with feed preheating and considers the issue of integrated systems.

Tank Underflow and Overflow. The simple tank example discussed in the previous section is revisited. There are three modes of operation for this tank:

State 1. When the liquid volume in the tank V is between two bounds $V_{\min} = 1$ m^3 and $V_{\max} = 5$ m^3, the system is in a safe state and its behavior is governed by the single ODE

$$\dot{V} = F_{\text{in}}(t) - \alpha V(t),\ t \in [0,\ H]. \tag{29}$$

where F_{in} is the inflow at time t, α is a constant, and H is the time horizon.

State 2. When the liquid volume in the tank is less than $V_{\min}$, the system is in the unsafe state of underflow.

State 3. When the liquid volume in the tank is greater than $V_{\max}$, the system is in the unsafe state of overflow.

There is no need to define system equations for states 2 and 3, as our only concern is whether they are reached. Furthermore, these unsafe states are treated as terminal so that no transitions are possible from states 2 to 1 and 3 to 1.

The possible transitions at any given time step i are:

- $S_1 \rightarrow S_2$ when $V_t < V_{\min}$,
- $S_1 \rightarrow S_3$ when $V_t > V_{\max}$.

For the safety analysis of this linear system, we consider the input region defined by the initial tank level $V_0 \in [2,\ 3]$ (m^3) and the inlet flow rate $F_{\text{in}} \in [0,\ 1]$ (m^3/s), assumed to be constant with time. We also study the effect of uncertainty in the proportionality constant α which has a nominal value of 0.15 s^{-1}.

For this simple case, it is possible to derive conditions on V_0 and F_{in} which guarantee safe behavior for an infinite time horizon. When $V = V_{\max}$, the tank remains in safe state 1 provided that

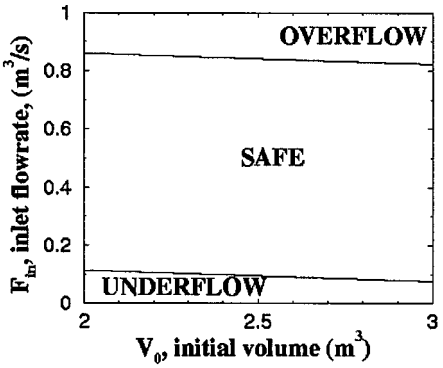
$$\dot{V} = F_{\text{in}} - \alpha V_{\max} \le 0 \Leftrightarrow F_{\text{in}} \le \alpha V_{\max} = 0.75 \text{ m}^3/\text{s}. \tag{30}$$

Similarly, when $V = V_{\min}$, the system remains safe if and only if

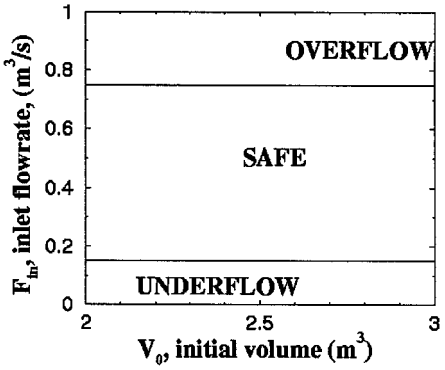
$$\dot{V} = F_{\text{in}} - \alpha V_{\min} \ge 0 \Leftrightarrow F_{\text{in}} \ge \alpha V_{\min} = 0.15 \text{ m}^3/\text{s} \tag{31}$$

As might be expected, the ultimate safety of the system is independent of the initial volume—provided it is within allowed bounds—and depends only on F_{in}.

Identification of Safe Input Regions. We initially consider a time horizon of 10 s and a tolerance $\epsilon = 10^{-4}$, and apply the safety analysis algorithm to the system. Bounding equations that provide an exact boundary for the system trajectories can be derived from interval arithmetic. As shown in Figure 7a, the input domain is split into three regions: safe (state 1), underflow (state 2), and overflow (state 3). The lines separating these regions correspond to undecidable regions of size ϵ^2. In this case, parts of the safe region for $H = 10$ s do not meet the necessary conditions on F_{in} for infinite time safety. With a longer time horizon of 100 s, the partitioning of the input region corresponds to the analytical solution, as seen in Figure 7b. It is in fact possible to identify regions of space exhibiting infinite time safety by specifying a large time horizon and testing for cyclic behavior, based on the fact that F_{in} is constant. Thus, if the tank volume at time step i is $V_i \in [\underline{V}_i,\ \overline{V}_i]$, with $\underline{V}_i \ge V_{\min}$ and $\overline{V}_i \le V_{\max}$, and if $V_{i+1} \in [\underline{V}_{i+1},\ \overline{V}_{i+1}] \subseteq [\underline{V}_i,\ \overline{V}_i]$, the system will remain safe for all subsequent time steps.



(a) $H = 10$ s



(b) $H = 100$ s

Figure 7. Safety analysis for tank example with different time horizons and $\epsilon = 10^{-4}$.

A comparison between this analysis and the worst-case approach of Dimitriadis et al. (1997) is instructive. With $\alpha = 0.15$ s^{-1}, the fastest occurring hazard is underflow. The most favorable conditions for this state to be reached are a low initial volume ($V_0 = 2$ m^3) and no inflow. This corresponds to the bottom lefthand corner of the underflow region in Figure 7. This point is identified as unsafe for a time horizon as short as 5 s.

Effect of Branching Strategy. The three branching strategies described earlier have been applied to this problem. The numbers of subregions in the sets of safe, unsafe and undecidable regions are shown in Table 4. Figure 8 shows the subregions examined in the safe and unsafe sets: in the case of sensitivity-driven branching, the inlet flow rate is systematically chosen for branching, in keeping with analytical results. This results in a dramatic reduction in the number of safe and unsafe regions explored. The number of undecidable regions is constant for all strategies as it only depends on ϵ and the bounding strategy. Finally, it is worthwhile noting that the worst case scenario for the algorithm requires the analysis of 1,333,333 subregions, and that much better performance is observed for all branching strategies, with only 32,794 regions explored in the best case.

Table 4. Numbers of Regions Analyzed in the Tank Example for Different Branching Strategies

Branching Strategy	Total No. of Regions	No. of Safe Regions	No. of Unsafe Regions	No. of Undecidable Regions
n-branching	98,306	13,122	52,416	32,768
Longest-axis	65,532	6,556	26,208	32,768
Sensitivity-driven	32,794	8	18	32,768

The Effect of Model Uncertainty. Assuming that the proportionality constant α is known within $\pm 5\%$, a new bounding system is derived for the tank

$$\underline{V}_{i+1} = \underline{F}_{\text{in}}\,\delta t + (1 - \overline{\alpha}\delta t)\underline{V}_i, \quad i = 1, \ldots, N$$

$$\overline{V}_{i+1} = \overline{F}_{\text{in}}\,\delta t + (1 - \underline{\alpha}\delta t)\overline{V}_i, \quad i = 1, \ldots, N \qquad (32)$$

where $[\underline{\alpha}, \overline{\alpha}] = [0.142, 0.158]$. Since the two equations are decoupled, they provide exact boundaries for the tank behavior. As shown in Figure 9, the size of the undecidable regions has increased. Since there is no overestimation in this system, this situation reflects the cost of uncertainty. The undecidable region between the safe and overflow states is larger than that between the safe and underflow states. The analysis of the equations highlights this disparity. From Eqs. 32, when $V_i = V_{\max}$ the system is guaranteed to remain safe if and only if

$$F_{\text{in}} \le \underline{\alpha} V_{\max} = 0.710 \text{ m}^3/\text{s}. \qquad (33)$$

In the case of $V_i = V_{\min}$, safe operation is ensured by

$$F_{\text{in}} \ge \overline{\alpha} V_{\max} = 0.158 \text{ m}^3/\text{s}. \qquad (34)$$

These critical values of F_{in} correspond to the worst-case values of α in the given uncertainty range, and result in a smaller safe region than previously. When best-case values of α are used, we find $F_{\text{in}} \le 0.79$ m^3/s prevents overflow, and $F_{\text{in}} \ge 0.142$ m^3/s prevents underflow. These values correspond to the highest and lowest boundaries of the undecidable regions respectively.

Batch Reactor. We revisit the batch reaction example discussed previously. The model is modified to account for the heat-transfer term more accurately by introducing $Q = UA(T_a - T)$, where U is the overall heat-transfer coefficient and A is the heat-transfer area. A nominal value of $UA = 3$ W/K is used. The system has four possible states:

State 1. Normal operating conditions, for all times during the 25 min batch time. The system equations are

$$\frac{dX}{dt} = k_0 \exp\left(-\frac{E_a}{RT}\right)(1 - X),$$

$$C_{A0} V C_p \frac{dT}{dt} = UA(T_a - T) - \Delta H_R k_0 \exp\left(-\frac{E_a}{RT}\right)(1 - X) C_{A0} V. \qquad (35)$$

State 2. Overheating, which occurs when the reactor temperature exceeds 540 K.

State 3. Successful run completion at time $t = 1{,}500$ s, with a conversion X greater than or equal to 97.5% and a temperature less than or equal to 540 K.

State 4. Off-spec run, at time $t = 1{,}500$ s, with a conversion less than 97.5%.

States 2 to 4 are terminal states so that the only possible transitions (shown in Figure 10) are:

- $S_1 \to S_2$ for $T(t) > T_{\max} = 540$ K,
- $S_1 \to S_3$ for $t = 1{,}500$ s and $X \ge 97.5\%$,
- $S_1 \to S_4$ for $t = 1{,}500$ s and $X < 97.5\%$,

The parameters used in the model are the same as discussed earlier. The input region considered is in State 1, with $X(0) = 0$, $T_0 \in [310, 540]$, and $T_a \in [290, 310]$. The bounding equations for Eq. 35 are obtained through monotonicity analysis

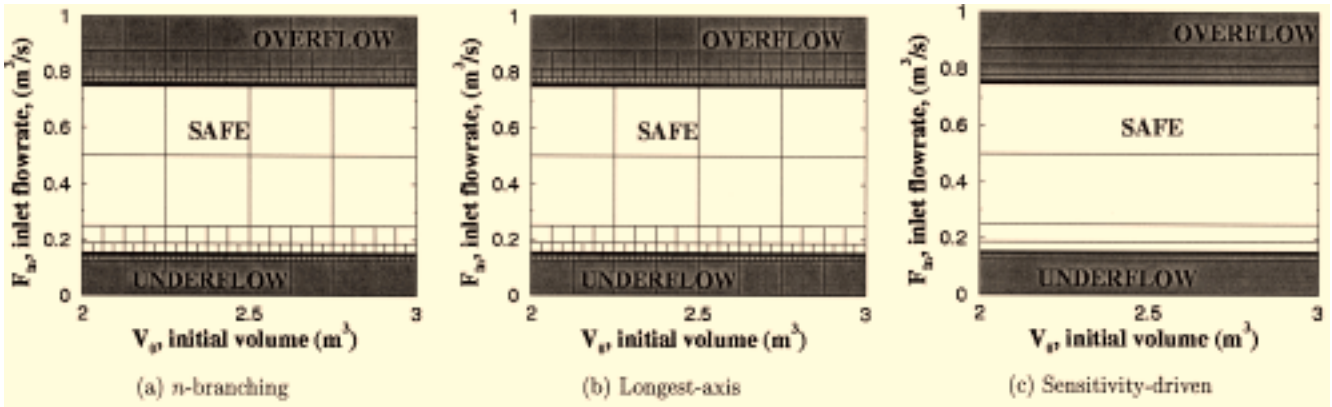


Figure 8. Region partitions for tank example using different branching strategies and $\epsilon = 10^{-4}$.

The shaded areas are unsafe.

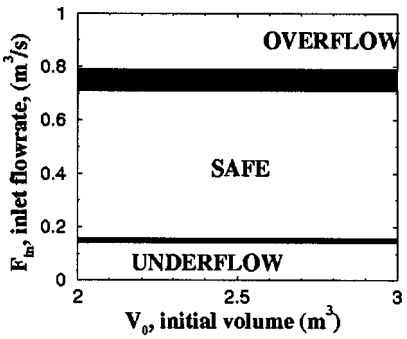


Figure 9. Undecidable regions (in black) for tank example with an uncertainty in α of $\pm 5\%$.

$$\underline{X}_{i+1} = k_0 \exp\left(-\frac{E_a}{R\underline{T}_i}\right)(1-\underline{X}_i)\,\delta t + \overline{X}_i,$$

$$\overline{X}_{i+1} = k_0 \exp\left(-\frac{E_a}{R\overline{T}_i}\right)(1-\overline{X}_i)\,\delta t + \underline{X}_i,$$

$$\underline{T}_{i+1} = \frac{UA\,\delta t}{C_{A0}VC_p}(\underline{T}_a - \underline{T}_i) - \Delta H_R k_0 \exp\left(-\frac{E_a}{R\underline{T}_i}\right) \times (1-\overline{X}_i)\,\delta t + \underline{T}_i,$$

$$\overline{T}_{i+1} = \frac{UA\,\delta t}{C_{A0}VC_p}(\overline{T}_a - \overline{T}_i) - \Delta H_R k_0 \exp\left(-\frac{E_a}{R\overline{T}_i}\right) \times (1-\underline{X}_i)\,\delta t + \overline{T}_i. \quad (36)$$

The results of the safety analysis algorithm are shown in Figure 11. The nonrectangular shape of the safe region is due to the nonlinearity of the model. The percentage of the total input area occupied by undecidable regions is 0.6% for $\epsilon = 0.1$ and 0.08% for $\epsilon = 0.01$. This seven-fold decrease in uncertainty comes at a cost of an eight-fold increase in computational requirements.

Effect of Branching Strategy. As shown in the tank example, the strategy used for the construction of subregions affects the efficiency of the algorithm. Nonlinearity in the model

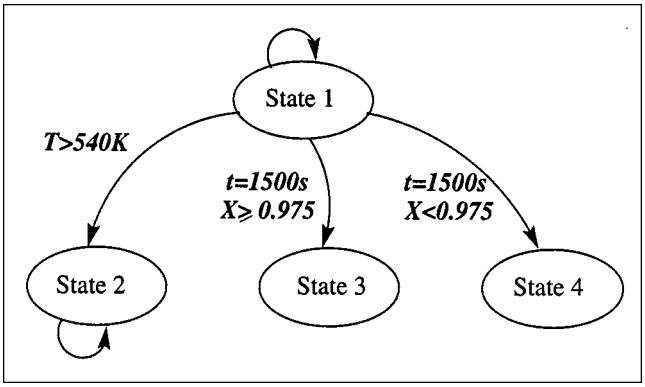
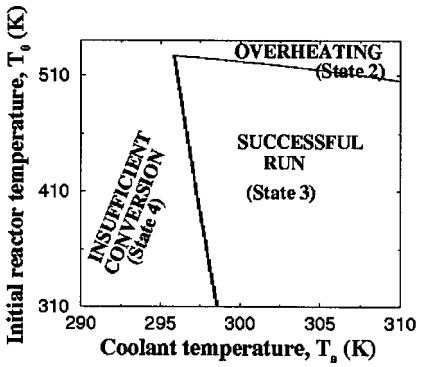
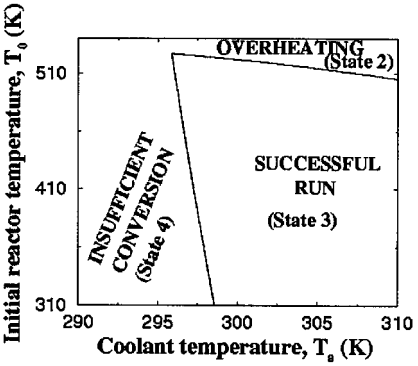


Figure 10. State-transition network for batch reactor example.



(a) $\epsilon = 0.1$



(b) $\epsilon = 0.01$

Figure 11. Results of safety analysis for batch reactor example for two values of ϵ.

The black areas represent undecidable regions.

is expected to reinforce this feature. For sensitivity-driven branching, the derivatives of the state variables with respect to T_0 and T_a are computed at the midpoint of the current input region using finite differences. This involves three integrations of Eqs. 35, during which the system will evolve to states 2, 3, or 4. The following three cases are therefore distinguished:

State 2. When the system enters state 2, the integration is stopped. The last values of the state variables are used to compute the sensitivity of T to T_0 (T_{T_0}) and T_a (T_{T_a}). If $|T_{T_0}| \geq |T_{T_a}|$, T_0 is branched on; otherwise, T_a is chosen.

State 3. If the run is completed successfully, the sensitivities at time $t = 1{,}500$ s are used. If $|T_{T_0}| + |X_{T_0}| \geq |T_{T_a}| + |X_{T_a}|$, T_0 is branched on; otherwise, T_a is chosen.

State 4. If the run is completed with insufficient conversion, the sensitivities of X to T_0 and T_a at time $t = 1{,}500$ s are used. If $|X_{T_0}| \geq |X_{Ta}|$, T_0 is branched on; otherwise, T_a is chosen.

The results for the three branching strategies with $\epsilon = 0.1$ are summarized in Table 5.

Sensitivity-driven branching gives the best performance in terms of number of regions explored. The nonlinearity of the problem is reflected in the fact that the number of undecidable regions depends on the branching strategy. Computa-

Table 5. Numbers of Regions Analyzed in the Batch Reactor Example for Different Branching Strategies

Branching Strategy	Total No. of Regions	No. of Safe Regions	No. of Overheat Regions	No. of Low Conversion Regions	No. of Undecidable Regions
n-branching	38,341	12,757	3,627	9,169	12,788
Longest-axis	13,248	3,382	393	3,008	6,465
Sensitivity-driven	11,105	2,355	985	1,300	6,465

tional time is strongly dependent on the branching strategy. A 46% decrease in computational time is observed from n-branching to longest-axis branching. A further 7% decrease is achieved from longest-axis branching to sensitivity-driven branching, in spite of the overhead incurred by sensitivity calculations.

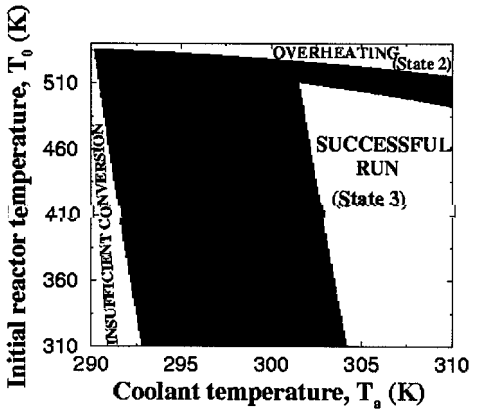
Model Uncertainty. If it is assumed that the rate constant k_0 is known to within $\pm 2\%$, the safety analysis yields the results shown in Figure 12a. A large fraction (57.6%) of the total input region is classified as undecided, and the safe region is smaller than when no uncertainty is present. In order to reduce overestimation, the uncertainty interval for k_0 is split into a different number N_I of subintervals which are analyzed separately in the inner loop of the algorithm. As shown in Figures 12b and c, this results in a smaller uncertain region. For only two subintervals, the undecided area covers 31% of the total region. For five subintervals, it represents 14.9%, and for ten subintervals, 9.5%. Branching on the uncertain parameters in the inner loop introduces additional computational expense, but this is compensated by a reduction in the overall number of regions which must be simulated. Thus, the use of two subintervals is 43% less expensive than the use of one subinterval, and the use of five subintervals is 61% less expensive than that of one. However, computational requirements would increase if a much finer partitioning of the uncertain space parameter was used. Overall, branching on uncertain parameters within the inner loop is a beneficial strategy to reduce both uncertainty and computational time.

Alternative Input Region. The successful completion of a run given a range of values of the heat-transfer coefficient is also of interest. In this case, we consider the input region given by $T_0 \in [310, 540]$ and $UA \in [0, 6]$ for a value of T_a of 298 K. The results shown in Figure 13 for $\epsilon = 0.1$ highlight the critical role played by U, as only a narrow nonlinear region ensures successful completion of a batch.

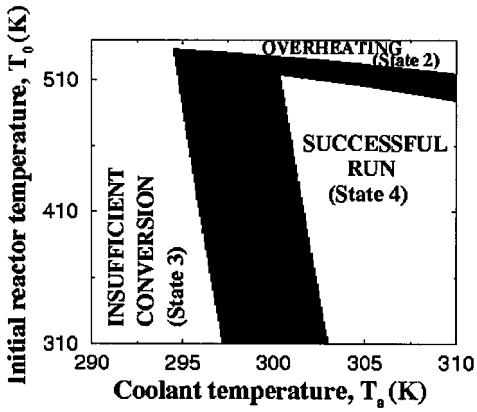
CSTR with Feed Preheating. This case study focuses on an integrated system in order to show how feedback loops affect the calculation of bounding trajectories and how this issue can be tackled during safety analysis. The system consists of a CSTR with a first-order exothermic reaction and a heat exchanger. As shown in Figure 14, the feed to the reactor is preheated in the exchanger using the outlet from the CSTR.

The CSTR is used to produce propylene glycol through the hydrolysis of propylene oxide. Data for this reaction are obtained from Fogler (1999). Due to the consequence of evaporation of propylene oxide, the reactor temperature cannot exceed $T_{\max} = 324$ K. There are two possible states for this system.

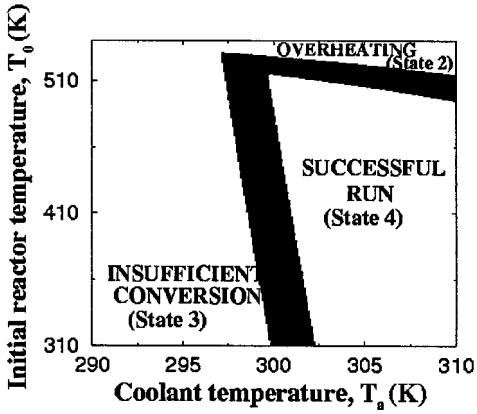
State 1. Normal operation. Assuming perfect mixing and constant total flow rate heat capacity, the model equations



(a) $N_I = 1$



(b) $N_I = 2$



(c) $N_I = 5$

Figure 12. Safety analysis with $k_0 \in [0.02156, 0.02244]$ and $\epsilon = 0.1$.

The k_0 uncertainty interval has been split into N_I subintervals in each plot.

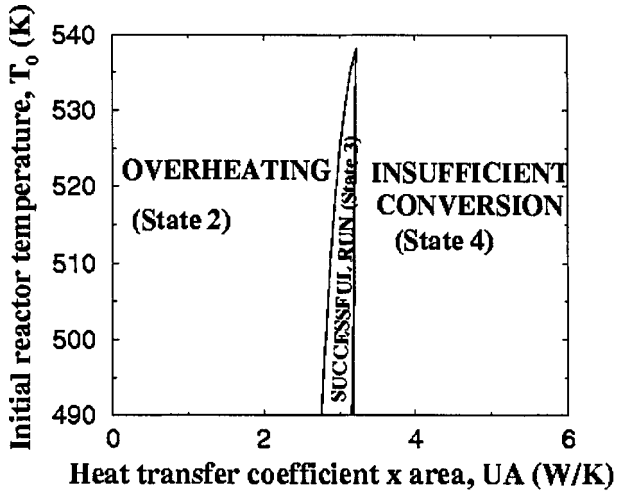


Figure 13. Safety analysis for batch reactor example for $\epsilon = 0.1$.

The black areas represent undecidable regions.

are

$$\frac{dX}{dt} = -\frac{X}{\tau} + k_0 \exp\left(-\frac{E_a}{RT}\right)(1-X)$$

$$\frac{dT}{dt} = -\frac{1}{\tau}(T - T_{\text{in}}) + \frac{(-\Delta H_R)k_0 \exp\left(-\frac{E_a}{RT}\right)(1-X)F_{A0}}{FC_p}$$

$$T_{\text{in}} = \frac{UAT + FC_pT_0}{UA + FC_p} \tag{37}$$

where X is the outlet conversion of A, $\tau = 442.44$ s is the residence time in the reactor, $k_0 = 4.71 \times 10^9 \text{s}^{-1}$ is the kinetic rate constant, $E_a = 75{,}295$ J/mol is the activation energy, $R = 8.314$ J/mol·K is the gas constant, $\Delta H_R = -84{,}666$ kJ/kmol is the heat of reaction, $FC_p = 5{,}084.4$ W/K is the constant flow rate heat capacity of the inlet and outlet from the reactor, $F_{A0} = 3.01$ mol/s is the inlet of flow rate of propylene oxide, U is the heat-transfer coefficient in W/m^2·K, and A is the heat-transfer area in m^2. The temperatures T_0, T_{in}, and T are in K, and as defined in Figure 14.

State 2. Unsafe state. Evaporation of propylene oxide, for $T > T_{\max}$. State 2 is a terminal state so that the only transition is $S_1 \rightarrow S_2$ when $T > T_{\max}$. T, the reactor outlet temperature, is the highest temperature in the system, so the condition $T \le T_{\max}$ suffices to ensure that the reactant does not evaporate.

We consider the startup of the reactor from a uniform temperature T_0 until steady state is reached. The input region of interest consists of T_0 and UA.

Bounding System. The system (Eq. 37) is then discretized and its monotonicity is analyzed to yield a bounding system of the following form

$$\begin{aligned}
\underline{X}_{i+1} &= \underline{X}_{i+1}(\underline{X}_i, \underline{T}_i) \\
\overline{X}_{i+1} &= \overline{X}_{i+1}(\overline{X}_i, \overline{T}_i) \\
\underline{T}_{i+1} &= \underline{T}_{i+1}(\overline{X}_i, \underline{T}_i, \underline{T}_{\text{in},i}) \\
\overline{T}_{i+1} &= \overline{T}_{i+1}(\underline{X}_i, \overline{T}_i, \overline{T}_{\text{in},i}) \\
\underline{T}_{\text{in},i} &= \underline{T}_{\text{in},i}\ (\underline{T}_i, \underline{UA}, \underline{T}_0) \\
\overline{T}_{\text{in},i} &= \overline{T}_{\text{in},i}(\overline{T}_i, \overline{UA}, \overline{T}_0)
\end{aligned} \tag{38}$$

Due to the linear dependence of T_{in} on T and T_0, the above bounding system is readily solved at each time step.

Safety Analysis of the Integrated System. The region $T_0 \times UA = [290, 300] \times [50, 60]$ is analyzed. The presence of feedback in the system, coupled with the exothermic nature of the reaction, results in a larger degree of overestimation than previously observed. This in turn requires a tighter tolerance ϵ to achieve a given level of accuracy. Figure 15 shows the results of the safety verification algorithm for $\epsilon = 0.01$. The inlet temperature has a much more important effect on safety than the performance of the heat exchanger as measured by UA. The width of the undecidable region is larger than in previous examples, suggesting that the safe region may be wider than detected by the algorithm. A thinner boundary between the safe and unsafe regions would be obtained with a smaller ϵ value.

A decomposition approach based on the uncoupling of the safety of the CSTR and the heat exchanger can be used to gain a better understanding of the issue of feedback. For this purpose, the safety of the CSTR alone is first studied. The

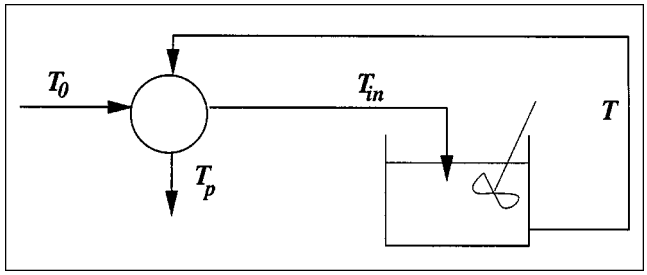


Figure 14. CSTR with feed preheating example.

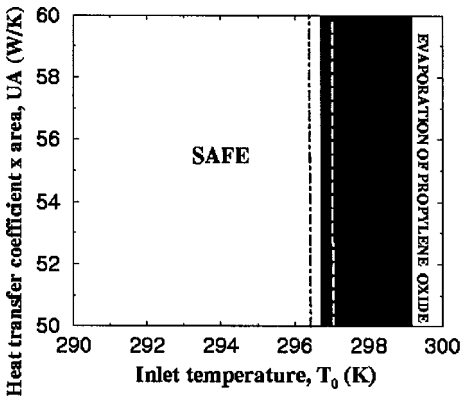


Figure 15. Safety analysis for CSTR example for $\epsilon = 0.01$.

Shown are the undecidable region (in black), the "exact" boundary between safe and unsafe regions, obtained through 25,000 simulations (– –, in white), the boundary between safe and unsafe regions obtained through decomposition (———).

input region consists only of the inlet temperature to the CSTR, T_{in}. With $\epsilon = 0.01$, the safety verification algorithm finds that the CSTR is safe (that is, its outlet temperature does not exceed 324 K) for 290 K $\leq T_{in} \leq$ 296.71 K. For 299.11 K $\leq T_{in}$, the CSTR is unsafe. Thus, the uncertainty region for the CSTR alone ([296.71, 299.11]) is almost as large as that for the integrated system ([296.69, 299.17]). This observation points to the nonlinear nature of the exothermic reaction as a cause for the size of the undecidable region. The critical value of the inlet temperature to the CSTR is $T_{in}^C = 296.71$ K. Based on this value, the safety analysis of the heat exchanger can be carried out. As shown in Eqs. 37, T_{in} depends on the input variables T_0 and UA as well as on T. The T dependence needs to be removed to separate the heat exchanger from the CSTR. A worst-case approach is taken, whereby T is set to 324 K, the maximum temperature. A boundary between the safe and unsafe regions is thus given by the line

$$UA\left(T_{max} - T_{in}^C\right) - FC_p\left(T_{in}^C - T_0\right) = 0. \qquad (39)$$

Figure 15 shows a comparison of the decomposition approach with the application of the safety analysis algorithm to the entire system. The boundary (Eq. 39) obtained through decomposition leads to a smaller safe region than the analysis for the combined system. This is a result of the approximation used during the analysis of the heat exchanger ($T = T_{max}$). Thus, despite the large uncertain region obtained, the safety analysis of the entire system yields more realistic results than the decomposition approach. The actual safe region can be expected to be yet larger. In order to measure the cost of uncertainty, 25,000 simulations of the combined system were performed at points within the uncertain region. This led to the identification of the likely exact boundary between the safe and unsafe regions. Interestingly, 86% of the uncertain region actually falls in the unsafe region, so that the rigorous combined analysis underestimates the size of the safe region by only 0.35 K.

This example illustrates the fact that feedback within the process can be handled successfully by the approach, and that the nonlinear dynamics of the system play a major role in determining the degree of uncertainty in the results. Since this behavior is most severe in the unsafe region, the safe region identified is close to the exact safe region. Most importantly, the region identified as safe through both the decomposition and integrated approaches is indeed unconditionally safe for the time horizon studied.

Conclusions

The assessment of the safety of a given process requires the reliable identification of safe operating conditions. However, the completion of this task to yield a realistic representation of possible hazards is hindered by the presence of nonlinearities and uncertainties as is typical in chemical plants. In this context, the framework proposed in this work for the modeling and reliable safety analysis of general processes can serve as a useful complement to the qualitative techniques which are generally used in hazard assessment, such as HAZOP. The formalism of the "region-transition model" relies on the concepts of hybrid systems and interval arithmetic and takes into account the full dynamics of the system. Nonlinearity and uncertainty are naturally handled within the approach. A significant advantage of the proposed methodology is that it operates on regions of the operating space, rather than on points within the space. As a result, nonlinear relationships between disturbances and initial conditions that guarantee safe operation are identified. A recursive safety analysis algorithm which follows a branch-and-bound approach has been implemented and successfully applied to three case studies: a linear tank example with uncertainty, an exothermic reaction in a batch reactor with uncertainty in the kinetic parameter, and an exothermic reaction in a CSTR with feed preheating.

The examples studied in this article all have low-dimensional input spaces. Furthermore, only constant deviations from nominal operation have been considered (constant disturbances). Current work therefore focuses on extending the approach to larger problems and more realistic disturbances. Two key issues are of importance: the derivation of tighter bounding trajectories, where differential inequalities (Walter, 1970) may be particularly helpful, and the representation of high-dimensional results in a compact and useful form.

Acknowledgments

The authors gratefully acknowledge grant number GR/N02269 from the U.K. government Engineering and Physical Sciences Research Council for financial support.

Literature Cited

Adjiman, C. S., "Safety Verification in Chemical Plants: A New Quantitative Approach," *Comput. and Chem. Eng. Suppl.*, S581 (1999).

Adjiman, C. S., I. P. Androulakis, and C. A. Floudas, "A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLPs: II. Implementation and Computational Results," *Comput. and Chem. Eng.*, **22**, 1159 (1998).

Alur, R., C. Courcoubetis, N. Halbwachs, T. Henziger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The Algorithmic Analysis of Hybrid Systems," *Theor. Comput. Sci.*, **138**, 3 (1995).

Alur, R., C. Courcoubetis, T. Henziger, and P.-H. Ho, "Hybrid Automata: an Algorithm Approach to the Specification and Verification of Hybrid Systems," *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds., *Lectures Notes in Computer Science*, Vol. 736, Springer-Verlag, Berlin, p. 209 (1993).

Alur, R., T. Henziger, and P.-H. Ho, "Automatic Symbolic Verification of Embedded Systems," *IEEE Trans. on Software Eng.*, **22**, 181 (1996a).

Alur, R., T. Henziger, and E. Sontag, *Hybrid Systems: III. Verification and Control, Lecture Notes in Computer Science*, Vol. 1066, Springer-Verlag, Berlin (1996b).

Asarin, E., O. Maler, and A. Pnueli, "Reachability Analysis of Dynamical Systems having Piecewise-Constant Derivatives," *Theor. Comput. Sci.*, **138**, 35 (1995).

Barton, P., and C. Pantelides, "Modeling of Combined Discrete-Continuous Processes," *AIChE J.*, **40**, 966 (1994).

Barton, P., and T. Park, "Analysis and Control of Combined Discrete/Continuous Systems: Progress and Challenges in the Chemical Processing Industries," *AIChE Symp. Ser.*, 5-316 (1997).

Catino, C., and L. Ungar, "Model-Based Approach to Automated Hazard Identification of Chemical Plants," *AIChE J.*, **41**, 97 (1995).

Chen, C.-T., *Linear System Theory and Design*, Holt, Rinehart and Winston, New York (1984).

Clarke, E., E. Emerson, and A. Sistla, "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications," *ACM Trans. on Programming Languages and Syst.*, **8**, 244 (1986).

Dill, D., and H. Wong-Toi, "Verification of Real-Time Systems by Successive Over and Under Approximation," *Computer Aided Verification*, **939**, 409 (1995).
Dimitriadis, V., N. S. J. Hackenberg, and C. Pantelides, "A Case Study in Hybrid Process Safety Verification," *Comput. and Chem. Eng.*, **20**, S503 (1996).
Dimitriadis, V., N. Shah, and C. Pantelides, "Modeling and Safety Verification of Discrete/Continuous Processing Systems," *AIChE J.*, **43**, 1041 (1997).
Dimitriadis, V. D., "Modelling, Safety Verification and Design of Discrete/Continuous Processing Systems," PhD Thesis, Imperial College (1997).
Eley, C., "Compliance Audit Checklists for Hazardous Chemicals," *Hybroc. Process.*, **71**, 97 (1992).
Fogler, H. S., *Elements of Chemical Reaction Engineering*, Prentice Hall, Upper Saddle River, NJ, 3rd ed. (1999).
Göring, M., and H. Schecker, "HAZEXPERT—An Integrated Expert System to Support Hazard Analysis in Process Plant Design," *Comput. and Chem. Eng.*, **17**, S429 (1993).
Grossman, R., A. Nerode, A. Ravn, and H. Rischel, *Hybrid Systems, Lectures Notes in Computer Science*, Vol. 736, Springer-Verlag, Berlin (1993).
Huang, H., C. S. Adjiman, and N. Shah, "Model-Based Safety Verification Under Uncertainty," *Euro. Symp. on Computer Aided Process Eng.-10*, S. Pierucci, ed., Elsevier Science B. V. (2000).
Jaulin, L., and E. Walter, "Global Numerical Approach to Nonlinear Discrete-Time Control," *IEEE Trans. on Automatic Control*, **42**, 872 (1997).
Jones, D., "Lessons from HAZOP Experiences," *Hybroc. Process.*, **71**, 77 (1992).
Kearfott, R. B., and M. Novoa III, "INTBIS, a Portable Interval Newton/Bisection Package," *ACM Trans. Math. Soft.*, **16**, 152 (1990).
Kelly, W., "Oversights and Mythology in a HAZOP Program," *Hydroc. Process.*, **70**, 114 (1991).
Kesten, Y., A. Pnueli, J. Sifakis, and S. Yovine, "Integration Graphs: a Class of Decidable Hybrid Systems," *Hybrid Systems*, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds., *Lectures Notes in Computer Sci.*, Vol. 736, Springer-Verlag, Berlin, p. 179 (1993).
Kleer, J. D., and J. Brown, "A Qualitative Physics Based on Confluence," *Artificial Intelligence*, **24**, 7 (1984).
Kleindorfer, P., and H. Kunreuther, *Insuring and Managing Hazardous Risks — From Seveso to Bhopal and Beyond*, Springer-Verlag, Berlin (1987).
Kletz, T., "Incidents that Could Have Been Prevented by HAZOP," *J. of Loss Prevention in the Process Ind.*, **4**, 128 (1991).
Kuipers, B., "Qualitative Simulation," *Artificial Intelligence*, **29**, 289 (1986).
Kumamoto, H., and E. Henley, *Probabilistic Risk Assessment and Management for Engineers and Scientists*, 2nd ed., IEEE Press, NJ, (1996).
Kurzman, D., *A Killing Wind: Inside Union Carbide and the Bhopal Catastrophe*, McGraw-Hill, New York (1987).
Lewis, D., "Flixborough," *Chem. Eng.*, **466**, 6 (1989).
Moon, I., G. Powers, J. Burch, and E. Clarke, "Automatic Verification of Sequential Control Systems using Temporal Logic," *AIChE J.*, **38**, 67 (1992).
Moore, R., *Interval Analysis*, Prentice Hall, Englewood Cliffs, NJ (1966).
Neumaier, A., *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge (1990).
OJEC, "Seveso II Directive 96/82/EC," *Official J. of the Euro. Communities* (1997).
OSHA, "Process Safety Management of Highly Hazardous Chemicals," U.S. Dept. of Labor, OSHA (1992).
Park, T., and P. Barton, "Implicit Model Checking of Logic-Based Control Systems," *AIChE J.*, **43**, 2246 (1997).
Schnepper, C. A., and M. A. Stadtherr, "Robust Process Simulation using Interval Methods," *Comput. Chem. Eng.*, **20**, 187 (1996).
Srinivasan, R., V. Dimitriadis, N. Shah, and V. Vekatasubramanian, "Integrating Knowledge-Based and Mathematical Programming Approaches for Process Safety Verification," *Comput. and Chem. Eng.*, **21**, S905 (1997).
Srinivasan, R., V. Dimitriadis, N. Shah, and V. Venkatasubramanian, "Safety Verification using a Hybrid Knowledge-Based Mathematical Programming Framework," *AIChE J.*, **44**, 361 (1998).
Srinivasan, R., and V. Venkatasubramanian, "Automating HAZOP Analysis of Batch Chemical Plants: I. The Knowledge Representation Framework," *Comput. and Chem. Eng.*, **22**, 1345 (1998).
Viswanathan, S., C. Johnsson, R. Srinivasan, V. Venkatasubramanian, and K. E. Arzen, "Automating HAZOP Analysis of Batch Chemical Plants: II. Implementation and Application," *Comput. and Chem. Eng.*, **22**, 1687 (1998).
Wallace, D., and R. Fujii, "Software Verification and Validation: an Overview," *IEEE Software*, **6**, 10 (1989).
Walter, W., *Differential and Integral Inequalities*, Springer-Verlag, Berlin (1970).
Waters, A., and J. Ponton, "Qualitative Simulation and Fault Propagation in Process Plants," *Chem. Eng. Res. and Des.*, **67**, 407 (1989).

Manuscript received Aug. 15, 2000, and revision received June 7, 2001.