SQsim: A Simulator for Imprecise ODE Models

Herbert Kay*

20 March 1996

Abstract

This article describes a method for representing and simulating Ordinary Differential Equation (ODE) systems which are imprecise – that is, where the ODE model contains both parametric and functional uncertainty. Such models, while useful in engineering contexts, are not used in practice because they require either special structures which limit the describable uncertainty or produce predictions which are extremely weak. This article describes SQsim (for SemiQuantitative SIMulator), a system which provides a general language for representing and reasoning about many common types of engineering uncertainty. By defining the model both qualitatively and quantitatively and by using a simulation method that combines inferences across the qualitative-to-quantitative spectrum, SQsim produces predictions that maintain a precision consistent with the model imprecision.

1 Introduction

Much work in engineering and science involves the construction of models. For continuous physical systems, Ordinary Differential Equations (ODE) models are a common representation. Such models consist of a system of differential equations that describe the trajectory of state variables over time. Making predictions from such models is quite straightforward and efficient given the availability of numerical ODE solvers such as LSODE and Runge-Kutta. However, this approach to modeling and simulation requires that the modeler determine the precise ODE for the physical system of interest. For many problems, a precise ODE may be difficult to find.

Consider the task of designing a physical device (e.g., a chemical plant or electronic circuit). The designer will typically construct a model of the device with precise values for its components (determined by detailed analysis), but real components have manufacturing tolerances, so it is unlikely that the real device will behave exactly the same as the model.

*This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-9216584 and IRI-9504138, by NASA grants NCC 2-760 and NAG 2-994, and by the Texas Advanced Research Program under grant no. 003658-242.

Consider the task of modeling an existing physical device, perhaps for the purpose of monitoring or diagnosis. While the modeler may have a good understanding of the physical principles involved in the device, he will not have precise knowledge of the values of every parameter and function that describes the system. If he creates a precise model, its predictions may not agree with the behavior of the real system.

In each of these cases, there is an inherent imprecision in the modeling task. This imprecision defines a *model space* that covers many different precise models. By using a single precise model, we ignore this imprecision, possibly making erroneous predictions. Ideally, we would like to use models that explicitly represent model imprecision and can produce a *useful* behavioral prediction – i.e., a prediction that is precise enough to distinguish desired from undesired behavior. Not surprisingly, there has been much work in this area. Some approaches to imprecise modeling are:

- Simply ignore the problem. For some classes of systems (for example, systems with tight feedback), small amounts of parametric uncertainty do not affect the results. For systems with larger amounts of uncertainty or nonlinearities, however, the model–device mismatch may be large.

- Use Monte-Carlo analysis [Kahaner *et al.*, 1989]. By running repeated simulations of the ODE system using different combinations of parameter values, it is possible to get some idea of the behavior of the models in the model space. Unfortunately, in addition to being slow, this approach may miss certain key combinations of parameters that produce a behavior not captured by the simulations that were run. Furthermore, it is difficult to represent functional uncertainty with this method.

- Represent uncertainty using probability distributions for imprecisely known parameters and use variance propagation methods to make predictions [Gelb, 1974, Reckhow, 1987]. This approach fails when the uncertainty cannot be modeled as probabilities (for example, if we have functional uncertainty). Furthermore, linearization is required to solve the variance equations which leads to a mismatch between the behavior of the true model and its linearization.

- Use a specialized model structure that separates the uncertainty from the ODE system [Lunze, 1989]. The ODE model and the uncertainty model can be solved separately and then combined to bound the model space. This method yields very precise predictions, but requires very specific model structures and is thus not generally applicable.

- Use intervals to bound parameters and use an interval ODE solver to compute predictions [Corliss, 1995a]. This method is fast and general, but produces very weak bounds.

We seek to develop a modeling language and simulation method that can describe imprecise ODE systems in a way that captures the types of uncertainty used by engineers, yet can provide useful predictions in that the predictions are as tight as possible. In this paper, we describe SQSIM (for SemiQuantitative SIMulator), a system for modeling and simulation which meets these requirements. Section 2 describes our method for modeling uncertainty, which is based on a combined qualitative-quantitative representation. Section 3 describes the simulation method, which makes use of several different inference methods that operate at different points along the qualitative-to-quantitative spectrum. Since the nature of the inferences that can be made are different at each point, SQsim combines them to produce a more useful behavioral description than any single method can. Section 4 contains an extended example, and Section 5 discusses other approaches to simulating uncertain models. Finally, Section 6 describes possible application areas as well as future work in improving the quality of SQsim predictions.

2 Modeling Uncertainty

When simulating an uncertain system, it is helpful to separate the information that *is* precisely known from that which *is not*, so that unambiguous inferences can be made from the precise information. Often, there will be precise information about the structural and qualitative properties of the model, while the numerical information will be imprecise. SQsim makes use of this distinction by using a multi-level representation based on the QSIM [Kuipers, 1986] representation for ODE systems. The levels are:

- The structural level (SDE).
  At this level, we describe the form of the ODE system in terms of the state variables and the constraints that link them. Constraints are described as arithmetic operators and functional relationships. The structural level provides the backbone of the modeling process. We call this representation the SDE (for Structural Differential Equation).

- The qualitative level (QDE).
  The qualitative level adds information about the domains of each model variable as well as the nature of the functional constraints. Each model variable is partitioned into a set of "landmark" values which represent important (to the modeler) values in the domain of the variable. For example, if A models the amount of fluid in a tank, it may be modeled by the *quantity space* $(0, FULL)$ which states that empty $(A = 0)$ and full $(A = FULL)$ are two interesting values in the space of the variable. The landmarks in a quantity space define an ordinal relationship on the values in the domain of the variable. Furthermore, sets of corresponding landmarks restrict the possible effects of constraints in the model. We call this representation the QDE (for Qualitative Differential Equation).

  The QDE also records information about the shape of functional forms in the model. Functional shapes can be defined as being monotonic over a region, S-shaped, U-shaped, etc. As we shall see, by making this information explicit, simulation methods that work at the qualitative level can infer facts about the behavior of a model that are unavailable using purely numerical methods.

- The quantitative level (SQDE).
  At this level, we record the uncertainty in the model. To be useful in engineering applications, a modeling language must capture the types of uncertainty that are commonly encountered by engineers. We define two classes of uncertainty in models:

  - Parametric imprecision models the uncertainty in the value of a scalar parameter of the model. This is the most common type of uncertainty and can be used to represent the the tolerance on a component or a bound on a reaction rate.
  - Functional imprecision models the uncertainty in the functional relationship between two or more variables. This type of uncertainty can be used to represent a family of functions which may not have a common parametric form yet share some qualitative property (such as monotonicity). Functional imprecision is a very useful feature for a model representation since it reduces the number of models needed to reason about a particular process.

  We represent parametric imprecision with numerical intervals that bound the uncertainty. We represent functional imprecision by defining *static envelopes* within which the functional constraint must lie. This type of imprecise knowledge has several advantages:

  - It is a form of knowledge readily available in engineering domains.
    Often, parametric and functional tolerances are described using intervals and bounding curves in the engineering literature.
  - There is an existing mathematics for computing with intervals and static envelopes.
  - We can represent precise models by allowing all interval widths and static envelope widths to go to zero. Thus, the more we know about a model, the more precise we can make it.

  We call this representation the SQDE (for Semiquantitative Differential Equation) because it contains both the QDE and imprecise quantitative information.

Figure 1 shows the representation of a simple one-tank system with a constant inflow. At each level, we further restrict the model space so that it eventually contains only one ODE. By using simulation techniques targeted for particular levels, we can thus utilize this information in a variety of ways.

3 Making Useful Predictions from Imprecise Models

By using a multi-level model, the modeler can represent both precise qualitative knowledge explicitly at the QDE level and imprecise numerical information at the SQDE level. We can then use inference methods on the QDE that are independent of the imprecise quantitative information. For example,

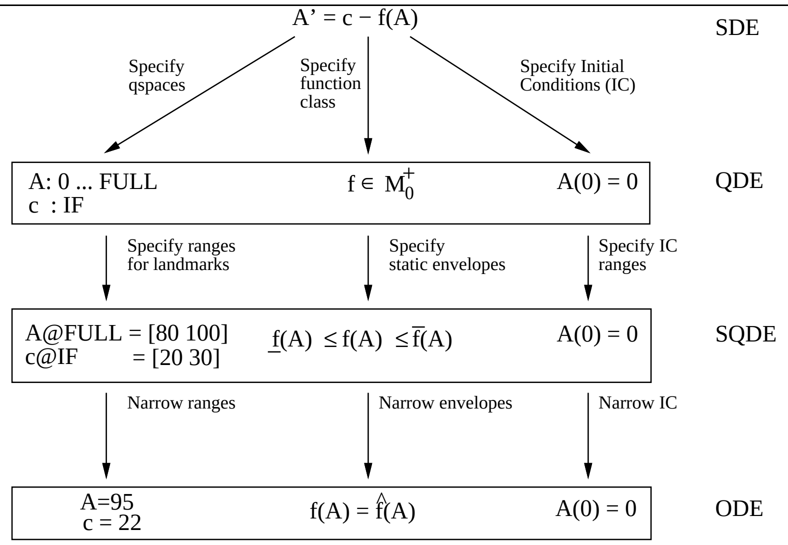


Figure 1: Multi-level model of an imprecisely-known system. At each level, the representation entails a space of ODEs. At the top is the purely structural SDE. The QDE adds qualitative information to the SDE, and the SQDE adds imprecise numerical information to the QDE. As we move downward we specify more information, thus reducing the size of the model space.

if we are describing a single tank system with a model that states that outflow is a monotonic function of amount then regardless of the precise functional relationship, we can absolutely conclude that an increase in amount will produce an increase in outflow.

As we shall see, inference at the qualitative level can improve the quality of predictions at the numerical level and vice versa. Our goal is thus to use a set of inference techniques at various points along the qualitative-to-quantitative spectrum and combine their results to yield tighter behavioral predictions. The next sections describe these methods.

3.1 Qualitative Inference

Qualitative inference produces predictions based on the QDE portion of the model. Recall that the qualitative description of a model is composed of the set of model variables whose domains are defined in terms of quantity spaces, the structural constraints that describe the relationship between variables in terms of arithmetic operators and functions, and shape constraints on the functional relations. Using Kuipers' QSIM algorithm [Kuipers, 1984, Kuipers, 1986], we can simulate the behavior of a QDE. QSIM simulates a QDE by tracking the magnitude and direction of change of each variable (which are called the *qmag* and *qdir* of the variable), where the magnitude is defined as being at or between two landmarks in the variable's quantity space and the direction of change is either increasing, steady, or decreasing starting from a user-specified (possibly incomplete) initial state.

By applying the rules of qualitative arithmetic [Kuipers, 1994], QSIM generates a set of variable descriptions consistent with the model constraints at the initial state. For instance, if x and y are related by a monotonic increasing function, $qdir(x) = \uparrow$ implies $qdir(y) = \uparrow$.

QSIM models time as a sequence of alternating points and intervals in time. The special variable $TIME$ represents time by a quantity space where each time-point state is at a landmark and each time-interval state is between two landmarks. The time landmark for the initial state is $T0$. Once a consistent assignment of qmags and qdirs is defined for the initial state, qualitative transition rules determine what will happen in the following time-interval. These transition rules enforce continuity constraints on the behavior trajectory. There are two different sets of transitions – those that hold when the simulation is at a time-point and those that occur when the simulation is in a time-interval. For example, if at time-point $T0$, $qmag(A) = 0$ and $qdir(A) = \uparrow$ then over the time-interval $(T0\ T1)$, $qmag(A) = (0\ FULL)$. Then at time-point $T1$, $qmag(A) = (0\ FULL)$[1] or $FULL$ since A may remain in the interval or increase to the next landmark. By repeating this process, QSIM produces a set of alternating time-point and time-interval qualitative descriptions of the system. Such a set of states is called a *qualitative behavior*. Normally, there will be multiple qualitative behaviors for a given model. This is because:

1. The qualitative model represents a family of ODE systems. Thus, if members of the family have different qualitative behaviors, each possible behavior will be represented.
2. Since qualitative mathematics is inherently ambiguous[2], some behaviors generated by this method will be *spurious*.

Figure 2 shows the result of simulating the QDE in Figure 1. It produces three behaviors corresponding to the cases where A overflows (first behavior), reaches a steady-state below the $FULL$ landmark (second behavior), or reaches a steady-state at $FULL$ (third behavior). These behaviors can be organized into a *behavior tree* which highlights where qualitative distinctions between behaviors occur.

One nice property of QSIM is that all real behaviors of the QDE are predicted. This *guaranteed coverage property* is of great importance to design (where we must be sure that all consequences of the design are explored) as well as diagnosis (where we want to be sure that all possible outcomes of our diagnostic hypotheses are considered) and will be preserved in all stages of the SQsim algorithm.

One weakness of QSIM is that some spurious behaviors may also be predicted. By utilizing a variety of additional qualitative information (for instance, information about the higher-order

[1] If the qmag of a variable is an interval between two landmarks at a time-point state and the qdir is steady, QSIM will create a new landmark in the qspace of the variable between the existing landmarks. Such landmarks are created because points where the derivative reaches zero are often interesting points in the trajectory of a variable and thus should be considered landmarks.

[2] For example, if we know that the directions of change of x and y are positive and negative, respectively, the direction of change for $x + y$ can be positive, negative, or zero.

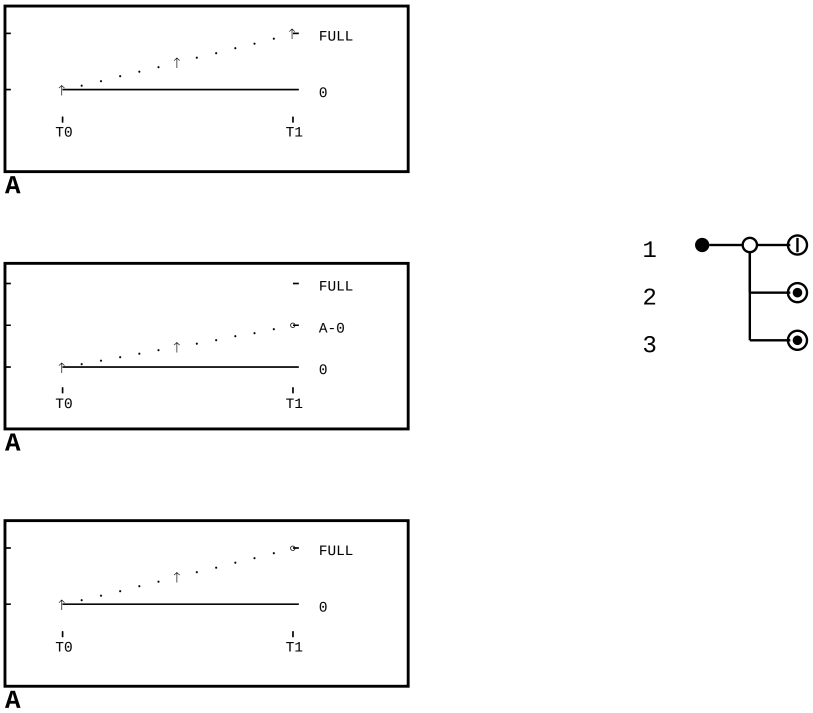


Figure 2: Simulation of the single-tank QDE. Three behaviors are produced from one initial state. Only the final state differs in each behavior (see graph at right). Each plot shows one possible behavior of the system. The qmag is determined by the height on the vertical axis and the qdir is determined by the plotted symbols ($\uparrow$ for increasing, $\downarrow$ for decreasing, and $\circ$ for steady). Each symbol is plotted either at a landmark, or midway between two landmarks. The connecting dots are purely visual aids and do not represent a specific trajectory taken by the prediction. In the first behavior, the tank overflows (A reaches $FULL$ and is increasing). In the second behavior, the amount of water reaches some equilibrium value between 0 and $FULL$ to which QSIM automatically assigns the landmark $A0$. In the third behavior, the amount of water reaches equilibrium just as the tank becomes full. All three behaviors are possible given the information in the QDE.

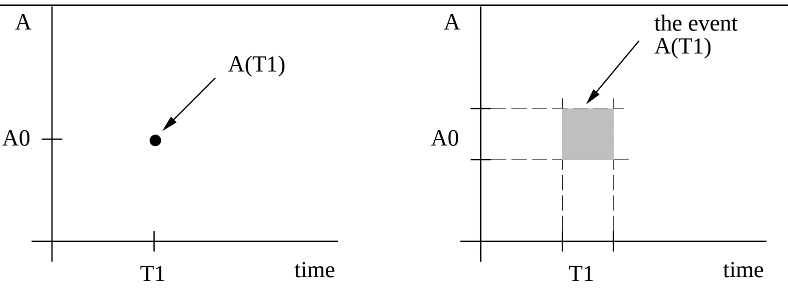


Figure 3: An event corresponds to the value of a variable at a time-point. As such, it represents a point in the state space (left graph). Due to uncertainty about the exact coordinates of the point, however, the event appears as a rectangle when viewed quantitatively.

derivatives of the system) QSIM can often refute branches of the behavior tree, thus reducing the number of spurious behaviors. As we shall see, numerical information can also be used to prune spurious qualitative behaviors thereby reducing prediction ambiguity.

3.2 Q2 – Inference at the Semiquantitative Level

Since QSIM ignores information at the quantitative level, it generates the set of all possible qualitative behaviors of ODEs entailed by the QDE, but it may also include behaviors of systems that are not part of the family described by the SQDE. Semiquantitative inference, however, does use the quantitative knowledge in the SQDE to attach range information to the qualitative behavior tree. In the process, some branches of the tree may be refuted because their qualitative descriptions are inconsistent with the quantitative information, while on other branches quantitative information refines the prediction.

The Q2 algorithm [Kuipers and Berleant, 1988] produces semiquantitative inferences from the SQDE. At each qualitative time-point, QSIM defines *events* for each variable. An event is the qmag of a variable together with the qmag of the $TIME$ variable at the time-point. For example if at some time-point state $TIME = T1$ and $A = A0$ then $A(T1)$ is an event for the variable A (see Figure 3). Since an event represents the instantaneous snapshot of a particular variable at a particular time, the SDE portion of the SQDE can be viewed as a set of equations relating the model variables, and the events must satisfy these equations. If we annotate the events with the range information described in the SQDE, we end up with a set of interval equations that must hold at each time-point. Given an initial assignment of ranges to certain landmarks, we can solve the system of equations using propagation of intervals through model constraints using interval mathematics [Moore, 1979][3]. Whenever we find an equation that computes the value of a model

[3] We propagate intervals through the unknown functional relationships by using the static envelopes defined in the SQDE.

variable, we intersect its existing range with the computed one. If their intersection reduces the range, we assert the updated range and follow the consequences of this new, smaller range. If the intersection is null, we refute the behavior since the interval equations are inconsistent with the qualitative behavior. We continue this process until either the state is refuted or we reach a fixed-point with respect to the ranges of the variables[4].

For example, in the final state of the first behavior in Figure 2, we have the event $A(T1) = FULL$. The SDE gives us the equation $A' = c - f(A)$ which must hold for this event. Annotating $FULL$ and c with the ranges given in the SQDE and using $\underline{f}(A) = 8\sqrt{A}$ and $\overline{f}(A) = 10\sqrt{A}$ leads to the following inferences:

$$
\begin{aligned}
A'(T1) &= [\underline{c}, \overline{c}] - [\underline{f}(\underline{FULL}), \overline{f}(\overline{FULL})] \\
&= [20, 30] - [\underline{f}(80), \overline{f}(100)] \\
&= [20, 30] - [71.5, 100] \\
&= [-80, -41.5]
\end{aligned}
$$

Since the qualitative prediction determines that the qdir of A is $\uparrow$, we know that A' must be greater than zero at $T1$ or $A'(T1) = [0, \infty]$. Intersecting this with the value computed above leads to a null interval, and so we may refute this behavior on the basis of the quantitative information.

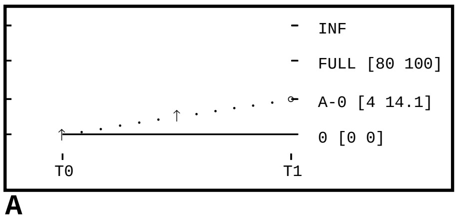
The propagation algorithm can be extended across time-intervals by using the Mean Value Theorem to relate the derivative of each state variable to the bounds of the state variable and the bounds on the difference in time between the two time-points. This type of inference reduces the width of the interval associated with the qmag of the time value at each time-point. Q2 thus refines the behavior description produced by QSIM by generating numeric bounds for both events and time-intervals such that:

- Each event determines a rectangle in the trajectory space that all behaviors must pass through.
- Between events, each time-interval description encloses all trajectories of the variable in a *dynamic envelope*, i.e., a pair of functions of time $\underline{A}(t)$ and $\overline{A}(t)$ such that $\underline{A}(t) \leq A(t) \leq \overline{A}(t)$ over the interval.

Figure 4 shows the results of running Q2 on the QDE version of the single-tank system. Given the numerical information, Q2 is able to eliminate both behaviors where the amount of water reaches $FULL$.

While Q2 is a powerful inference method for reducing the ambiguity in the qualitative behavior description, it may still permit spurious behaviors and overly wide dynamic trajectory envelopes. There are two reasons for this:

[4] We ensure that a fixed-point will be found in finite time by defining a minimum change ϵ below which a change in a range is considered insignificant.



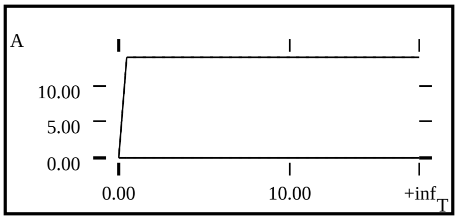


Figure 4: Q2 simulation of the single-tank SQDE with static envelope $\underline{f}(A) = 8\sqrt{A}$, $\overline{f}(A) = 10\sqrt{A}$. On the left, we display the same result as in Figure 2 but annotated with ranges. On the right, we display the same behavior as a dynamic envelope on a quantitative plot of A versus time.

- Interval arithmetic is inherently ambiguous because correlations between variables are ignored. Consider solving the equation $z = x - x$ where x is in the interval $[1, 2]$. Applying the rules of interval mathematics $z = [1, 2] - [1, 2] = [-1, 1]$ whereas the true solution is clearly $z = 0$. The problem is that the interval operator assumes that x can be 2 for one operand and 1 for the other, which is not possible since they must be the same.
- Q2 uses the Mean Value Theorem to describe changes between time-points in a behavior. Since the distance between time-points can be arbitrarily large, this means that Q2's prediction over intervals can be very coarse for the same reason that a numerical simulator with too large a time-step produces poor results.

3.3 Nsim – Inference at the Quantitative Level

Q2 bridges the gap between the qualitative and quantitative representations in the SQDE by using the qualitative model as a framework for generating quantitative interval equations at each qualitative time-point. While interval mathematics can infer numerical bounds on the behavior, there are other purely numerical inference methods that can tighten these bounds even further. We have developed such a method that uses a step-size that fits the numerical properties of the SQDE rather than relying on the distance between qualitative time-points. Our method, called Nsim [Kay and Kuipers, 1993], does this by constructing an ODE system that is guaranteed to bound the behaviors of the SQDE. This system is then simulated to generate a dynamic envelope for all behaviors of the SQDE.

Nsim constructs an *extremal* ODE system from a QDE by minimizing and maximizing each derivative equation in the SQDE. To compute an extremal equation, Nsim rewrites the SQDE using the translations in Table 1 to substitute terms. For example, the ODE system

$$\begin{aligned} A' &= c - f(A) \\ B' &= f(A) - f(B) \end{aligned}$$

e	$L(e)$	$U(e)$
c	$\underline{c}$	$\overline{c}$
x_j	$\underline{x}_j$	$\overline{x}_j$
x_i	$\beta(x_i)$	$\beta(x_i)$
$A+B$	$L(A)+L(B)$	$U(A)+U(B)$
$A-B$	$L(A)-U(B)$	$U(A)-L(B)$
$A\times B$	$min(C_{Mult}(A,B))$	$max(C_{Mult}(A,B))$
$A\div B$	$min(C_{Div}(A,B))$	$max(C_{Div}(A,B))$
$-A$	$-U(A)$	$-L(A)$
$M^+(A)$	$\underline{M}^+(L(A))$	$\overline{M}^+(U(A))$
$M^-(A)$	$\underline{M}^-(U(A))$	$\overline{M}^-(L(A))$

$$\begin{array}{rrcl} \text{where} & C_{Mult}(A,B) & = & \{L(A)\cdot L(B), U(A)\cdot L(B), L(A)\cdot U(B), U(A)\cdot U(B)\} \\ \text{and} & C_{Div}(A,B) & = & \{L(A)/L(B), U(A)/L(B), L(A)/U(B), U(A)/U(B)\}. \end{array}$$

Table 1: Translation table for extremal expressions of the equation $x_i' = f_i(\mathbf{x}_i)$. Let $\beta(f_i)$ be the desired bound on x_i' ($\beta = L$ or $\beta = U$). The table is applied recursively to the subexpressions of f_i. The symbol x_j is any state variable other that x_i, c is a constant, M^+ and M^- are monotonic functions, $\underline{c}$ and $\overline{c}$ return the lower or upper range values of c, $\underline{M}^*$ and $\overline{M}^*$ return the lower or upper functional envelope of the monotonic function. For the state variable x_i, the bound is the same as that for f_i.

becomes

$$\begin{aligned} \underline{A}' &= \underline{c} - \overline{f}(\underline{A}) \\ \underline{B}' &= \underline{f}(\underline{A}) - \overline{f}(\underline{B}) \\ \overline{A}' &= \overline{c} - \underline{f}(\overline{A}) \\ \overline{B}' &= \overline{f}(\overline{A}) - \underline{f}(\overline{B}) \end{aligned}$$

By applying both a lower and an upper translation for each equation, NSIM generates a new ODE system whose state variables are the upper and lower bounds for the state variables of the SQDE. Thus, we replace the original imprecisely-defined system with a precise ODE system of twice the order that is guaranteed to bound the original SQDE[5]. Since this new system is an ODE system, we can simulate it using conventional simulation methods to obtain a bound on the system[6]. Figure 5 shows the result of applying NSIM to the SQDE of the single-tank system. Since

[5]See Appendix A for a proof.

[6]In our work we use the LSODA subroutine from ODEPACK.

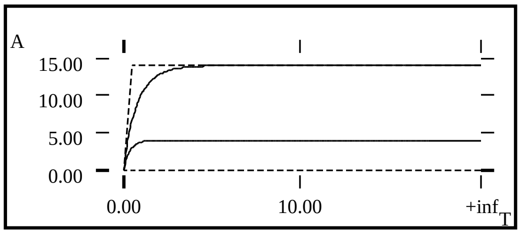


Figure 5: Simulation of the single-tank SQDE with NSIM. The solid curve is the NSIM dynamic envelope. The dotted box is the Q2 dynamic envelope defined by the initial state and equilibrium state events. The NSIM envelope is much tighter than in Figure 4 because the ODE solver determines the simulation step-size based on the numerical properties of the extremal system.

the simulation time-step is controlled by the numerical properties of the extremal system, NSIM's bounds are much tighter than those of Q2.

While NSIM produces tighter bounds than Q2 in many cases, it can also generate spurious behaviors in the form of overly-wide dynamic envelopes. These spurious behaviors stem from:

- Representing uncertainty with intervals.
  One can precisely describe the state-space uncertainty of a system at a given time by a set of points. Unfortunately, this type of description is intractable with respect to representation and inference since the set of points is normally infinite. A useful description must represent such sets finitely. NSIM uses hypercubes as an uncertainty representation. Such a simplification makes description and inference tractable, however it also admits spurious behaviors. Given that the hypercube $\mathbf{X}(t)$ describes the state of uncertainty at t, the set $\mathbf{S}$ which contains all states reachable from $\mathbf{X}(t)$ at time $t+h$ is not in general a hypercube. Thus, we must enclose the reachable states at $t+h$ in a hypercube so that they are representable. We thus include extra states in $\mathbf{X}(t+h)$ that are not reachable from $\mathbf{X}(t)$. As simulation proceeds, this set may include more and more states that are not reachable from the initial state, thus leading to bounds that diverge with time. This is known as the "wrapping problem" in the interval simulation literature [Moore, 1979] (see Figure 6). Note that wrapping is the property of any finite representation of a set of states. Depending on the system, is sometimes possible to reduce the effect of wrapping by using other representations such as ellipsoids [Moore, 1979] or by using coordinate transformations that orient the hypercube with respect to the trajectory of the system (see Section 5).

- Ignoring correlations in the SQDE.
  A model may contain multiple references to a function, however, the extremization process

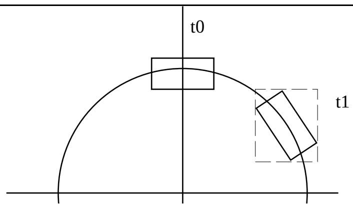


Figure 6: The wrapping effect. As we travel through the phase space from t_0 to t_1, the state hypercube orients itself with respect to the trajectory. Since intervals are described with respect to the coordinate axes, we must "wrap" the new bound with a new hypercube oriented with respect to the original coordinate system.

may make these references refer to different functions. For example, if we have a model

$$x' = c - f(x) \qquad y' = f(x) - g(y)$$

our intention is that the two occurrences of $f(x)$ refer to the same function. The extremal system for the lower bounds of x and y, however, will look like

$$\underline{x}' = \underline{c} - \overline{f}(\underline{x}) \qquad \underline{y}' = \underline{f}(\underline{x}) - \overline{g}(\underline{y}).$$

Unlike the original system, the second term in the first equation is no longer identical to the first term in the second one. NSIM is thus making predictions about models that are not part of the original model space and so these predictions admit spurious states. In some cases, model correlation can be eliminated by rewriting the SQDE so that only one reference to a function is made (see [Hüllermeier, 1995]). In the above example, we can introduce a new variable $z = x + y$ thus transforming the system into

$$x' = c - f(x) \quad z' = x' + y' = c - g(z - x)$$

which makes only one reference to f. The lower bound is now

$$\underline{x}' = \underline{c} - \overline{f}(\underline{x}) \qquad \underline{z}' = \underline{c} - \overline{g}(\underline{z} - \overline{x})$$

- Ignoring correlations between variable bounds.
  The extremization process, coupled with the use of the interval representation for uncertainty, causes NSIM to produce an ODE system that can, in some cases, ignore the correlation between upper and lower bounds of a variable in the original system. While the SQDE exists in an order-n state-space, the extremal system generated by NSIM is of order-$2n$. If the trajectories generated by each ODE and initial condition entailed by the SQDE and state hypercube at t_0 are mapped into $2n$-space, they will occupy an order-n subspace since, they are subject to

the added constraint $\forall_{i,t}\ \underline{x}_i(t) = \overline{x}_i(t)$. The extremal system, however, is not restricted to considering points in this subspace. Thus the extremal system may compute solutions based on points in $2n$-space where $\underline{x}_i(t) \neq \overline{x}_i(t)$ which cannot happen in any of the ODEs. For example, consider the harmonic oscillator

$$x' = v \qquad v' = -x$$

which has an extremal system defined as

$$\underline{x}' = \underline{v} \qquad \underline{v}' = -\overline{x} \qquad \overline{x}' = \overline{v} \qquad \overline{v}' = -\underline{x}. \tag{1}$$

For this system, the derivative of the extremal system state vector $[\underline{x}, \overline{x}, \underline{v}, \overline{v}]^T$ depends simultaneously on both upper and lower bounds of x and v. Thus, the correlations $\underline{x} = \overline{x}$ and $\underline{v} = \overline{v}$ are explicitly ignored (assuming that the bounds on x and v are of non-zero width). In contrast, consider the system

$$x' = c - x \qquad y' = x - y$$

which has the extremal system

$$\underline{x}' = \underline{c} - \underline{x} \qquad \underline{y}' = \underline{x} - \underline{y} \qquad \overline{x}' = \overline{c} - \overline{x} \qquad \overline{y}' = \overline{c} - \overline{y}$$

In this case, the extremal system separates into two second-order ODEs. The derivative of $[\underline{x}, \underline{y}]$ depends only on the lower bounds of x and y and the derivative of $[\overline{x}, \overline{y}]$ depends only on their upper bounds. Because neither system requires the upper and lower bounds of a system simultaneously, the correlations $\underline{x} = \overline{x}$ and $\underline{y} = \overline{y}$ are not ignored. The practical result is that the extremal system for this system produces tighter bounds than extremal system (1).

SQDEs that lead to separable extremal systems have the property that their bounds are determined by simulating "corners" of the hypercube. This class is identical to the class of quasi-monotonic systems [Walter, 1970] for which it can be shown that simulating the corners of the hypercube leads to an optimal bound [Corliss, 1995a].

These sources of spurious behavior suggest that we can improve the NSIM prediction by using different uncertainty representations or simulation methods. In Section 5 we will examine some of these possibilities. While such methods can reduce the width of dynamic envelopes in some cases, they require specialized simulation engines. It is worth noting that by casting the bounding problem as the solution to an ODE system, NSIM can take advantage of the speed and efficiency of existing numerical ODE simulators as well as exploit future advances in ODE simulation technology.

3.4 SQSIM – Combining Inference at the Different Levels

The predictions produced by QSIM, Q2, and NSIM use information in the SQDE in different ways. QSIM provides the qualitative shape of the trajectories. Q2 augments the qualitative events

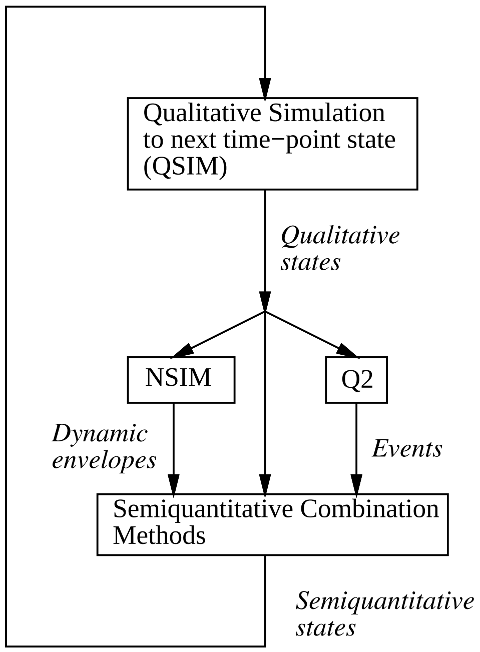


Figure 7: Information flow through SQSIM. QSIM generates a time-point state which Q2 and NSIM annotate with events and dynamic envelopes. This information is then passed through the semiquantitative combination methods. If the state is consistent, the cycle repeats with QSIM generating all successors to the state.

with interval ranges. NSIM produces dynamic envelopes directly from the imprecise numerical information. While each method guarantees that all real behaviors entailed by the original SQDE are predicted, they each admit spurious behaviors as well. QSIM generates them due to uncertainty in qualitative arithmetic. Q2 fails to refute them due to uncertainty in interval mathematics and the uncertainty in the distance between qualitative time-points. NSIM generates them because the process which generates the extremal system can ignore correlations between variables. By combining these inference methods, SQSIM is able to reduce the imprecision in predictions by ensuring consistency across the predictions.

Figure 7 shows the flow of information through SQSIM. SQSIM uses QSIM as the basis for simulation. At every qualitative time-point Q2 and NSIM are run to produce events and dynamic envelopes that describe the qualitative behavior up to the time-point. Then, each of the combination methods described below are run until a fixed-point with respect to Q2 events is found. If no contradiction is detected, SQSIM generates the successor states of the current state and repeats the process.

SQSIM uses four methods for combining QSIM, Q2, and NSIM inferences:

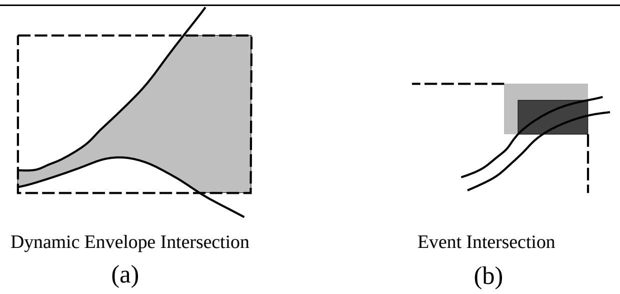


Figure 8: Q2 and NSIM intersection methods. We can intersect both Q2 events and dynamic envelopes with NSIM dynamic envelopes. For dynamic envelope intersection, the Q2 dynamic envelope (dashed box) and NSIM envelope (solid line) produce the shaded envelope. For event intersection a Q2 event (light rectangle) and the NSIM envelope (solid line) produce a reduced event description (darker rectangle).

Dynamic envelope intersection We can intersect the Q2 dynamic envelopes between two events with the NSIM prediction to yield a tighter bound. Figure 8(a) demonstrates this process. In this figure, the NSIM prediction starts out tight and diverges as time increases. In contrast, the Q2 dynamic envelope is a constant band over the interval. By intersecting both predictions, we obtain a behavior that is better than Q2 alone at the beginning of the interval and better than NSIM for later values of time. If there is a time at which the NSIM and Q2 dynamic envelopes do not intersect, the entire behavior can be refuted since their predictions must be consistent over the entire behavior.

Event intersection NSIM's dynamic envelopes can also be intersected with Q2 event descriptions. In this case, a behavior can be refuted if the NSIM envelope does not pass through any point in the event rectangle. If the event and dynamic envelope do overlap, it is possible to refine the event based on the intersection. In Figure 8(b), for example, the NSIM envelope restricts the starting time for the event as well as the maximum value for the event variable. By running Q2 on this reduced event description, SQSIM may further refine other events in the behavior.

Bump detection For some models, the NSIM dynamic envelope will exhibit clear qualitative properties. For example, the envelope may contain a "bump" where all trajectories reach a local minimum or maximum (see Figure 9). In such cases, the QSIM description should also contain a bump in the corresponding region of the behavior. The bump detector locates bumps within the dynamic envelope of a variable and ensures that there is a corresponding critical point in the QSIM description. Note that this test can only be used when the NSIM envelopes contain a recognizable bump. This is because a dynamic envelope of non-zero width does not normally provide information about the shape of the underlying trajectories. Therefore, every behavior that passes through the

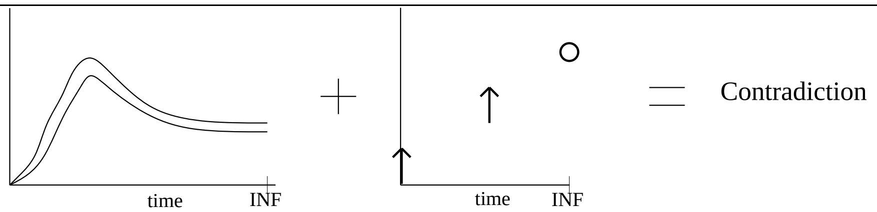


Figure 9: Bump detection ensures that an NSIM-detected extremum has a corresponding qualitative extremum. The dynamic envelope in the left graph shows that all behaviors must pass through a maximum before reaching quiescence. The qualitative plot on the right has no such maximum and therefore there is a contradiction.

envelope *may* pass through a local extremum. It is only when there is a bump in the envelope that we are guaranteed that every behavior *must* pass through an extremum.

Order reduction The Q2 dynamic envelope is a coarse trajectory envelope of the behavior. NSIM can produce a more precise bound, however its envelope can also diverge with time. By using the Q2 dynamic envelope, we can control NSIM so that it does not simulate an upper (or lower) bound when that bound is already at the maximum (or minimum) determined by Q2. Instead, we can set this bound's derivative to zero, thus reducing the order of the extremal system. Such an operation can greatly reduce the width of the dynamic envelope. This method can also be applied to the sign of the derivative – i.e., if the derivative of an upper (or lower) bound takes on a sign inconsistent with the Q2 prediction, the corresponding bound in the extremal system can be set to zero. Figures 18 and 19 in Section 4 demonstrate the use of order reduction to improve a dynamic envelope where bounds on the derivative of a state variable are known.

Resimulation SQSIM operates on a state-by-state basis, using all events from the current and previous states to refine the behavior. Sometimes, however, the bounds on events may be shrunk as a result of events generated at later time-points. This is especially true for equilibrium states (which occur at the end of behaviors) since in these states Q2 can produce tighter event bounds because the derivatives of all the state variables are known to be precisely 0. Furthermore, it is sometimes possible to bound the time over which derivatives of state variables have constant qdir and use this information to guide order-reduction. Resimulation is the process of analyzing a completed behavior for these types of non-local properties and then running all combination methods described above. Figures 18 and 19 in Section 4 illustrate the use of resimulation.

4 An Example

In this section, we present an example of using SQsim to determine the behavior of a chemical system. The intent is to demonstrate the behavior, as well as the strengths and limitations of SQsim.

The system that we simulate is an adiabatic Continuously-Stirred Tank Reactor (CSTR) with an irreversible exothermic first-order reaction $A \rightarrow B$. The equations of the system are :

$$\begin{aligned} \frac{dC_A}{dt} &= \frac{C_{Ai} - C_A}{\tau} - k_0 e^{-E/T} C_A \\ \frac{dT}{dt} &= \frac{T_i - T}{\tau} - h_r k_0 e^{-E/T} C_A \end{aligned} \tag{2}$$

where C_A is the concentration of reactant A in the tank, C_{Ai} is the inlet concentration of A, T is the tank temperature, T_i is the temperature of the inlet stream, τ is the residence time, h_r is the heat of reaction, k_0 is the rate constant, and E is the activation energy. This system is highly nonlinear due to the exponential containing T.

In the process of design or monitoring such a system, we may be interested in answering the following questions:

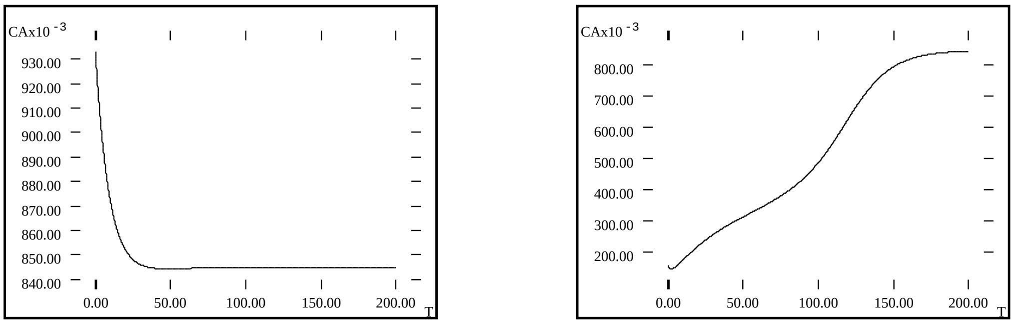
- What are the locations of equilibrium points in the state space?
- What are the qualitative trajectories of the system?
- What are the final values of the state variables and how do they compare to the initial values?
- Do the trajectories remain inside an acceptable region of the state space?
- Where and when do extrema occur in the trajectories of the state variables?
- How sensitive is the behavior to uncertainty in model parameters?

Our example follows the work of Dalle Molle [Dalle Molle, 1989] who made a detailed study of the adiabatic CSTR for the case where the system is in equilibrium and then C_{Ai} is negatively perturbed. He analyzed the numerical and qualitative properties of this system and determined that there are three equilibrium states for an ODE instance with the values described in Table 2. He then examined the effect of different perturbations from these equilibrium states. In all cases T decreases over time, however there are four possible qualitative behaviors of C_A. For our analysis, we focus on a decrease on C_{Ai} from 1.0 to 0.9. In this case, there are two different behaviors depending on which steady-state we start from (see Figure 10).

In the next section, we analyze the behavior of the precise model using SQsim. The only uncertainty that we allow is in the initial state description. Then, in the following section, we will analyze the behaviors produced when we allow parametric uncertainty in the model.

State number	C_A	T	Comments
1	0.9332	353.4	Low conversion
2	0.6090	418.2	Unstable
3	0.1566	508.7	High conversion

Table 2: The three possible equilibrium states of the CSTR given the parameters $C_{Ai} = 1.0$, $T_i = 340$, $\tau = 10$, $k_0 = 10000$, $E = 5000$, $h_r = -200$.



(a) (b)

Figure 10: The two behaviors of C_A when simulated numerically from (a) steady-state 1 and (b) steady-state 3. Note the local minimum in (a) at $t \approx 50$ and in (b) at $t \approx 6$.

4.1 The Precise CSTR under SQSIM

In this section, we consider the precise model of the CSTR simulated from an imprecise initial state.

4.1.1 Qualitative Inference

Figure 11 gives an overview of the results in this section and is intended as a road map for following the analysis of the example.

We begin with the SDE described by Equation 2 which defines the model variables and the relations between them. From this, we form a QDE by defining the quantity spaces for the model variables and then use QSIM to generate the qualitative equilibrium states. The quantity space for C_A contains the landmarks 0, CAI (which represents the input stream concentration), and INF (which represents the landmark at ∞, a value larger than any finite real number) while T contains the landmarks 0, TI (which represents the input stream temperature) and INF. These quantity spaces restrict C_A and T to positive values. By asserting that the qdirs of C_A and T are steady and that all constants are at fixed landmark values, QSIM is able to determine the equilibrium values of the qmags and qdirs for all model variables. QSIM determines that there is a single qualitative state consistent with the three equilibrium states in Table 2 (see Figure 12). This is because all three quantitative equilibria are located such that C_A is between 0 and $C_{Ai}(= 1.0)$ and T is between $T_i(= 340)$ and ∞.

Next, we perturb C_{Ai} by asserting that its qmag is less than its qmag in the equilibrium state. QSIM generates 8 behaviors from three possible initial states (which are due to uncertainty in the the value of $C_{Ai} - C_A$ in the first equation which may be positive, zero, or negative). In all behaviors T decreases monotonically over time and C_A has one of four qualitative shapes (see Figure 13) which include the two numerically produced behaviors in Figure 10. The other two behaviors occur when C_{Ai} is perturbed to values other than 0.9.

4.1.2 Semiquantitative Inference

Next, we add quantitative information to produce the SQDE. Since we are modeling the precise ODE, the semiquantitative information consists of zero-width intervals for each model parameter. Our first goal is to identify the equilibria of the system. We define the bounds on C_A and T to be $[0, +\infty]$ which encodes the knowledge that the equilibria lie in the positive portion of the state space of the system. SQSIM is then able to isolate the equilibria of the system by using a technique called Localization on the initial state. Localization addresses the "gap" problem with interval representations. For example, the narrowest interval representing $\sqrt{4}$ is $[-2, 2]$ even though it is really only the endpoints of the interval that are true solutions. There is thus a gap in the interval. Localization attempts to find gaps in the domain of a variable. Given a variable to localize and an initial range, the Localizer selects a point within the range of the variable and using Q2 to derive a contradiction, indicating that the initial range contains a gap. When a gap is found, it splits

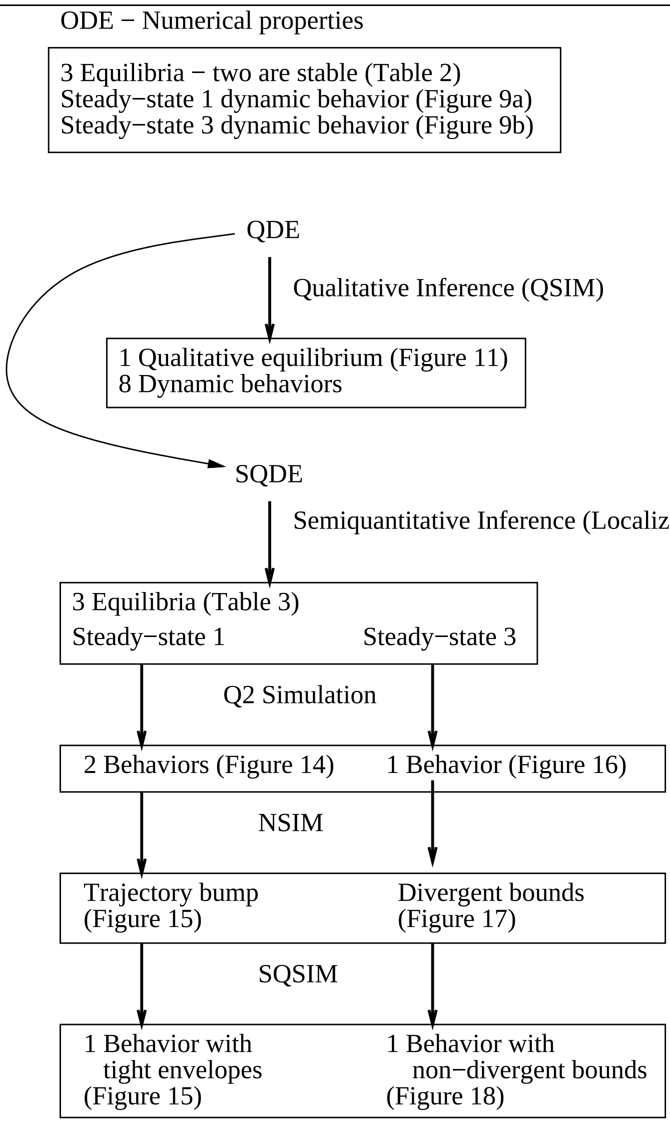


Figure 11: An overview of the analysis of the precise CSTR model using SQSIM. The user provides the SQDE (which contains the QDE) together with a (possibly-imprecise) initial condition. The prediction of equilibria and dynamic behavior is produced automatically by SQSIM.

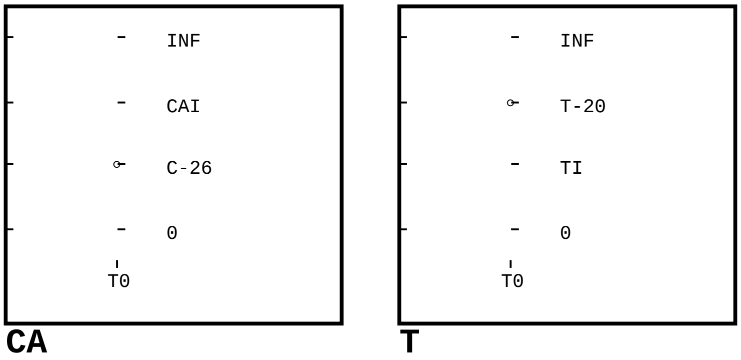


Figure 12: The qualitative equilibrium of the CSTR. Because the the three equilibria in Table 2 are qualitatively identical, QSIM generates a single equilibrium state.

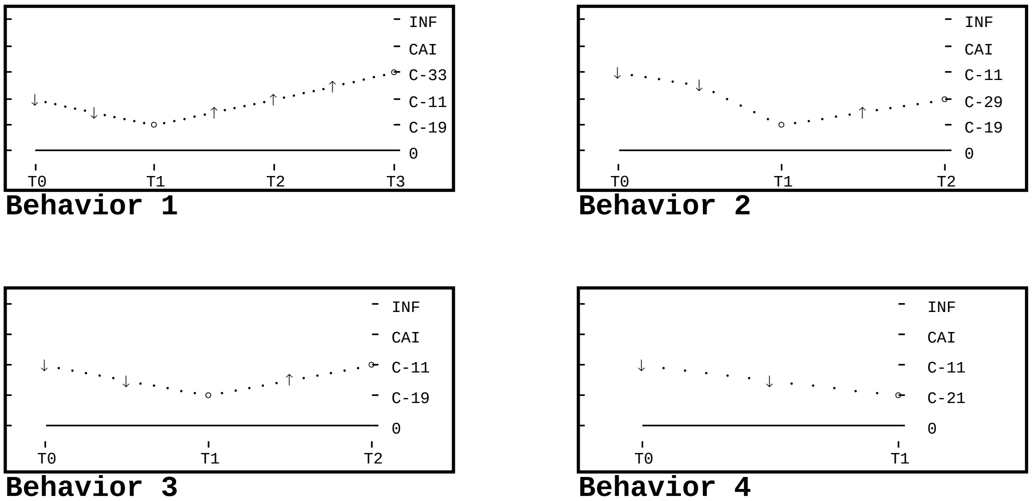


Figure 13: The four qualitative behaviors of C_A that are possible for the CSTR when decreasing C_{Ai}. Only behaviors 1 and 2 are possible when C_{Ai} decreases to 0.9.

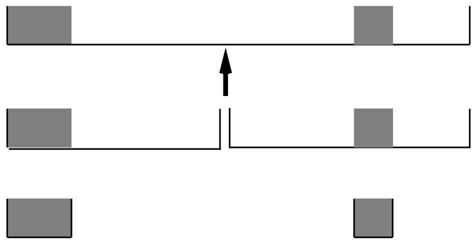


Figure 14: Localization splits an interval with a gap in it into smaller intervals by locating a small gap and then forming two intervals on either side of it. These new intervals can then be refined using Q2 to further reduce their range (gray squares). Since finding a small gap is less subject to correlation error than is examining the entire interval, Localization can quickly shrink overly-wide bounds.

the state description in two – one part where the range of the variable falls below the gap and one part where the range falls above the gap. It then applies Q2 to each of these more precise states which may result in further convergence of ranges. If not, it attempts to localize each subrange (see Figure 14). The process terminates when either the final subranges are points or after three attempts to find a gap fails[7].

For the CSTR model starting with ranges of $[0, \infty]$ for C_A and T, Localization identifies the three numerically-determined equilibria from Table 2 (see Table 3). Note that the equilibrium state descriptions have non-zero width. This is because the Q2 portion of SQSIM uses a parameter ϵ to determine when the ranges associated with landmarks reach a fixed-point so as to prevent excessive interval propagation. By decreasing ϵ, we can increase the precision with which SQSIM localizes the equilibria. Because Localization maintains the Guaranteed Coverage Property (see Section 3.1), for any non-zero ϵ the equilibrium description will contain some uncertainty. In contrast, the numerically-determined equilibria in Table 2, while precise, are not guaranteed to be correct[8]. This is because the representation (precise floating-point numbers) cannot represent the imprecision inherent in the numerical simulator. For some systems, this error could lead to incorrect results (for example, if the system is chaotic and thus sensitive to initial conditions). By simulating from an imprecise initial condition, we maintain the Guaranteed Coverage Property of SQSIM and do not "miss" potentially relevant behaviors[9].

Decreasing C_{Ai} starting near steady-state 1 Starting from the Localization-determined equilibrium interval $T = [353.358, 353.360]$ and $C_A = [0.933, 0.934]$, the dynamic simulation leads to

[7] Localization is akin to Target Interval Splitting [Berleant and Kuipers, 1992] which attempts to refute the *ends* of intervals.

[8] In fact, the values in the table are incorrect since they do not fall inside the guaranteed Q2 envelope. In this case, this is because of round-off in the table.

[9] If we *did* choose a precise initial condition, we would get a single precise behavior since the NSIM prediction of a precise SQDE and initial condition is a zero-width trajectory.

State number	C_A	T
1	$[0.933, 0.934]$	$[353.358, 353.360]$
2	$[0.609, 0.610]$	$[418.194, 418.196]$
3	$[0.156, 0.157]$	$[508.667, 508.673]$

Table 3: The equilibria of the CSTR computed using SQSIM Localization with a Q2 ϵ of 10^{-6}. By decreasing ϵ, Q2 will produce tighter bounds. The true equilibria are guaranteed to lie within the bounds.

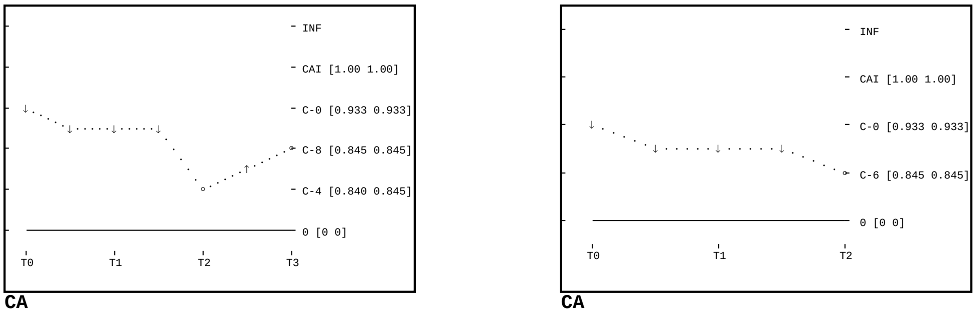


Figure 15: Q2 prediction of C_A from steady-state 1 when $C_{Ai} = 0.9$. Two behaviors are produced. The ranges of all time-points after $T0$ are $[0, \infty]$. Q2 is unable to distinguish these two cases because, in the first behavior, the ranges associated with the events $C_A(T2)$ and $C_A(T3)$ are such that the depth of the bump could possibly be zero, thus making the first and second behaviors identical.

the two behaviors in Figure 15. Q2 refutes the other 6 branches because they are not consistent with phase space location of steady-state 1. Dalle Molle determined that only the first behavior is real for the precise model. Q2 is unable to determine this because its prediction for the upper bound of the local minimum of C_A is the same as that for its final value and so the two behaviors look equivalent.

Applying NSIM to the SQDE results in the envelopes in Figure 16. Note that they fall completely inside the Q2 envelope and that there is a trajectory bump near $t = 50$. By using the SQSIM bump detector, we can eliminate the spurious behavior. Thus SQSIM predicts a single qualitative behavior and a tight dynamic envelope which suffices to closely bound any dynamic properties of interest.

Decreasing C_{Ai} starting near steady-state 3 Starting from the Localization-determined equilibrium interval $T = [508.667, 508.673]$ and $C_A = [0.156, 0.157]$, Q2 produces the single behavior shown in Figure 17. While it has refuted the other 7 behaviors, the final range of C_A ($[0.157, 0.893]$) is rather large and is bounded below by the initial concentration, so we cannot predict that the final concentration increases. By manually applying Target Interval Splitting to the final state, we find that the final value of C_A is $[0.845, 0.846]$ so the concentration has definitely increased. We

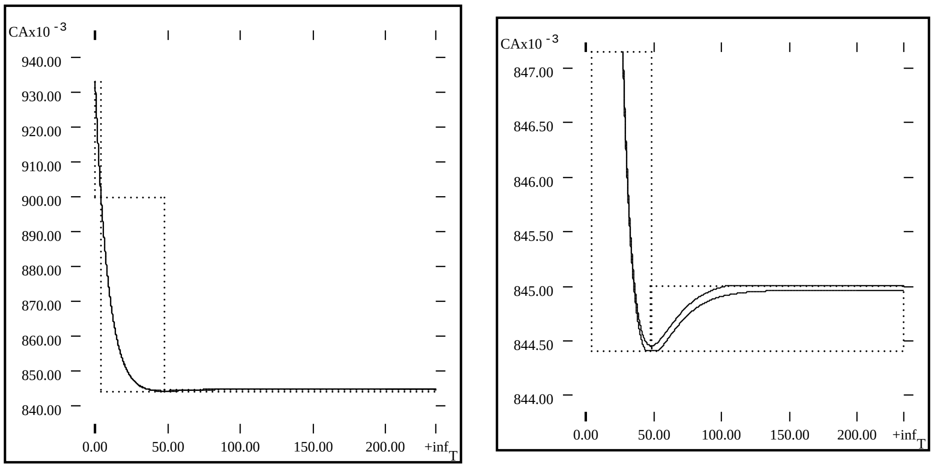


Figure 16: NSIM simulation of the example in Figure 15 together with the Q2 prediction (dashed boxes). The spurious behavior is eliminated by SQSIM due to the detection of the bump in the envelope (see blowup at right). The dynamic envelopes remain tight over the entire prediction.

still have very weak information about the time of the inverse response.

Applying NSIM only, we obtain the envelopes in Figure 18. For $C_{Ai} = 0.9$, the extremal system is quite stable up until $t \approx 20$ where it exhibits a high degree of divergence. This is due to ignoring correlations in the extremal system which turns the negative feedback in the original system into positive feedback. For other values of C_{Ai}, this divergence is not so sharp and NSIM alone produces tighter envelopes[10]. Once we reach the final state, however, it is clear from the qualitative behavior that C'_A is always positive after reaching the minimum at $T1$. Therefore, SQSIM can resimulate using order-reduction to prevent C'_A from becoming negative. This significantly shrinks the final envelopes as shown in Figure 19.

Referring back to the questions that we are interested in for this model, if we start with a SQDE of the CSTR model, SQSIM can determine the following:

- It automatically identifies the location of the equilibria.
- It produces single qualitative trajectories from the stable equilibria.
- When input concentration is perturbed from 1.0 to 0.9, SQSIM determines that the final concentration of C_A increases from steady-state 1 and decreases from steady-state 3.
- From steady-state 1 it finds a tight envelope over the the entire trajectory. From steady-state 3, the trajectory widens as time increases, but remains bounded.

[10] Experimentation reveals that $C_{Ai} \leq 0.7$ leads to tight predictions.

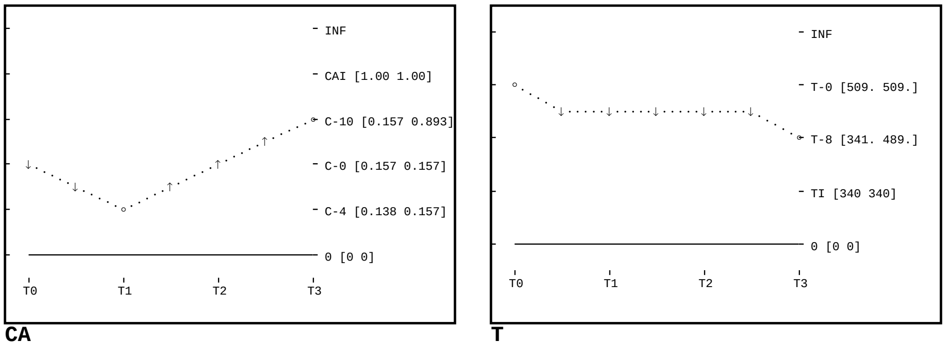


Figure 17: Q2 prediction of C_A from steady-state 3 when $C_{Ai} = 0.9$. The ranges of all time-points after $T0$ are $[0, \infty]$. One behavior is produced. Although concentration clearly increases, the final value is still rather large. By using Target Interval Splitting we can reduce this range, although dynamic properties, like the time when the minimum concentration is reached are still poorly predicted.

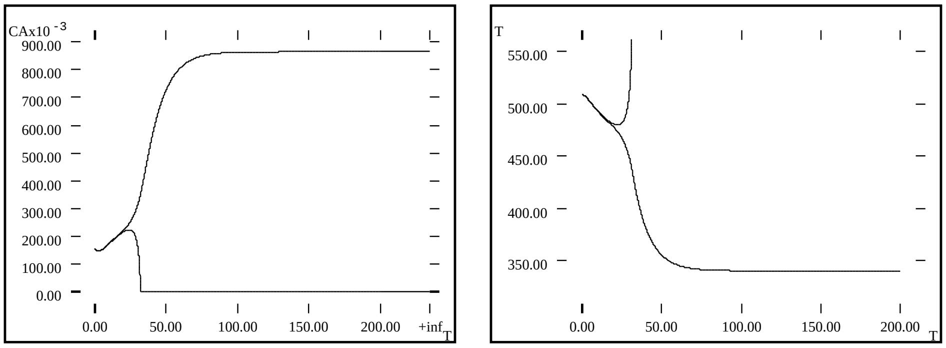


Figure 18: NSIM envelopes for the example in Figure 17. These bounds diverge after $t \approx 20$ due to the extremal system ignoring correlations between upper and lower bounds of the state variables.

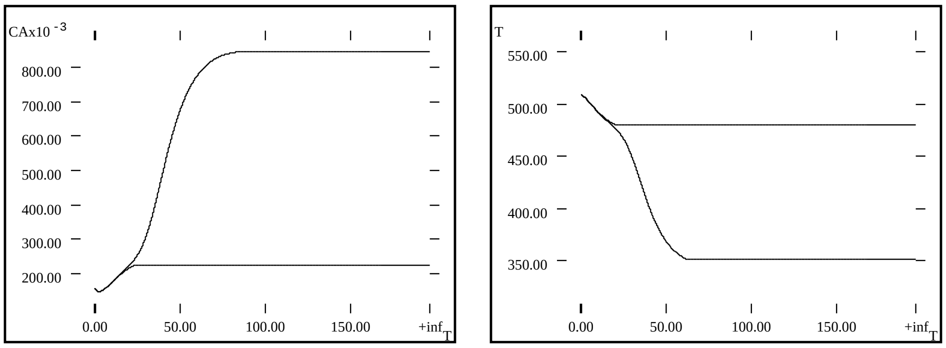


Figure 19: SQSIM envelopes for the example in Figure 17. By using order-reduction and resimulation based on and the inference at qualitative time-point $T3$ that C'_A is never negative after C_A reaches its minimum value, SQSIM can produce a tighter dynamic envelope.

- The location and deviation of the inverse responses are tightly bounded.

SQSIM is thus able to automatically infer the relevant qualitative and quantitative properties of this system, together with behavioral bounds over time, even in the presence of uncertainty in the initial condition.

4.2 The Imprecise CSTR

So far, we have considered only initial state uncertainty. In this section, we examine the more interesting case when there is uncertainty in the model by varying the values of the model parameters k_0, h_r, and E_a by $\pm1\%$ and $\pm5\%$. Varying these quantities examines cases where there is uncertainty about the reaction rate, heat of reaction, and activation energy. While SQSIM can handle the more general case of functional imprecision, for this model, parametric imprecision is sufficient to examine the properties of our method.

Decreasing C_{Ai} near steady-state 1 For 1% error in k_0 we obtain the results in Figure 20. Table 4.2 summarizes the results from varying all three variables independently by ±1 and ±5 percent. Even with model uncertainty, the behavior of the system is still tightly bounded by the dynamic envelopes and that prediction uncertainty increases with model imprecision. Target Interval Splitting yields tight initial state and final state equilibria. The increase in uncertainty prevents us from eliminating the "non-bump" behavior[11], so we have two possible qualitative behaviors. However, because the bounds are tight, the possible deviation of the inverse response is still tightly bounded. For similar amounts of uncertainty, k_0 and h_r have a similar effect on

[11]We have not done an analysis to determine if the behavior without inverse response is spurious for the uncertain model.

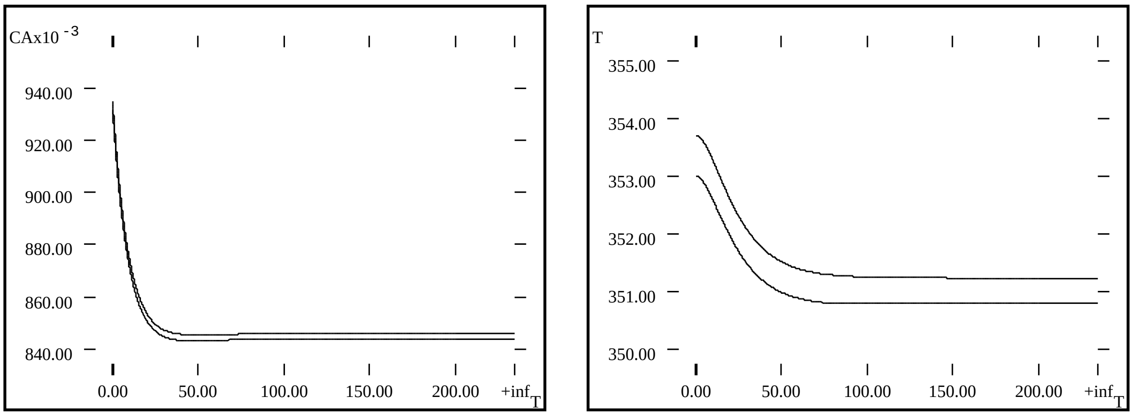


Figure 20: SQSIM simulation of the CSTR from steady-state 1 and $C_{Ai} = 0.9$ with 1% error in k_0. There are two qualitative behaviors (one with a local minimum in C_A and one without), however the dynamic envelopes are the same in each case.

prediction uncertainty. Varying E_a has a greater effect due to its position in the exponential of the differential equations.

Decreasing C_{Ai} near steady-state 3 The results for a 1% error in k_0 are shown in Figure 21. As was demonstrated in the precise case, this system exhibits divergence about this equilibrium when $C_{Ai} = 0.9$. The increase in uncertainty prevents SQSIM from precisely determining the location and time of the minimum in C_A, however, the envelopes still restrict the trajectory so we can determine the minimum possible value of this event. Table 5 summarizes the results from varying all three parameters by 1 and 5%. As was the case near steady-state 1, prediction uncertainty increases with increasing model imprecision.

We conclude this section by summarizing the strengths and limitations of our method. By combining qualitative and quantitative inference, SQSIM produces predictions in terms of both qualitative trajectories and envelopes. The qualitative prediction serves two purposes:

- It provides a basis for distinguishing trajectories that fall within the envelopes. The predictions are thus more detailed than are predictions based on envelopes alone.
- It provides information that the NSIM numerical simulator can use to further refine the dynamic envelopes.

This synergy between representations can often produce very tight bounds on the behavior of an imprecisely-defined ODE system. The degree of uncertainty in prediction is directly related to

Case		Initial State		Final State		Max Envelope Width	
var	error	C_A	T	C_A	T	C_A	T
–	0%	$[0.933, 0.934]$	$[353.358, 353.360]$	$[0.845, 0.846]$	$[350.998, 350.999]$	0.001	0.284
k_o	$\pm 1\%$	$[0.931, 0.935]$	$[353.025, 353.706]$	$[0.843, 0.847]$	$[350.795, 351.225]$	0.004	0.446
k_o	$\pm 5\%$	$[0.923, 0.942]$	$[351.797, 353.253]$	$[0.839, 0.851]$	$[349.934, 352.201]$	0.011	2.265
h_r	$\pm 1\%$	$[0.932, 0.935]$	$[353.048, 353.681]$	$[0.844, 0.846]$	$[350.794, 351.557]$	0.002	0.434
h_r	$\pm 5\%$	$[0.928, 0.938]$	$[351.904, 355.105]$	$[0.842, 0.848]$	$[350.000, 352.117]$	0.006	2.126
E_a	$\pm 1\%$	$[0.909, 0.948]$	$[350.449, 358.008]$	$[0.823, 0.859]$	$[348.326, 355.254]$	0.044	9.626
E_a	$\pm 5\%$	$[0.899, 0.979]$	$[344.294, 360.001]$	$[0.823, 0.879]$	$[344.166, 355.251]$	0.078	15.835

Table 4: Decreasing C_{Ai} to 0.9 from steady-state 1 for the imprecise CSTR for 1 and 5% error in k_0, h_r and E_a. In all cases, two qualitative behaviors are generated, however the dynamic envelopes for each behavior are identical. For each parameter, increasing the uncertainty increases the width of the initial state, final state, and envelope width.

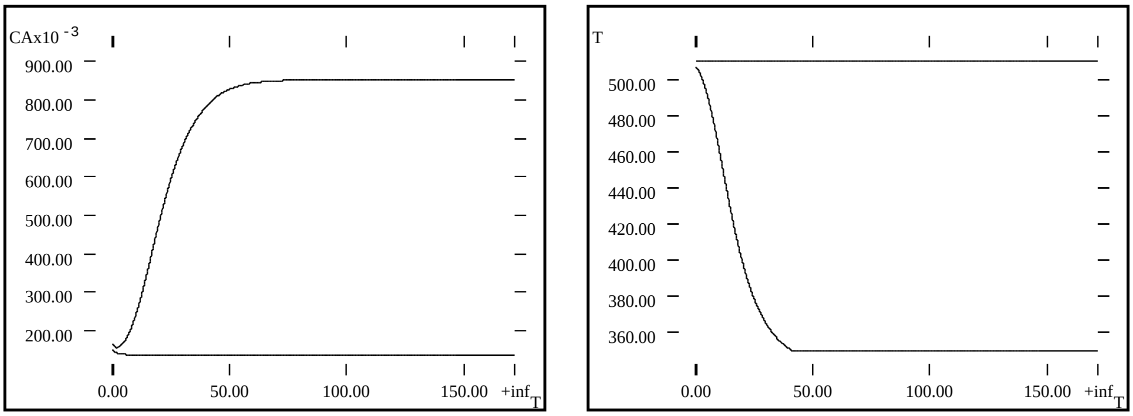


Figure 21: SQSIM envelopes for the CSTR from steady-state 3 and $C_{Ai-} = 0.9$ with 1% error in k_0. There is one qualitative behavior. The dynamic envelopes over time are very wide, however.

Case		Initial State		Final State		Max Envelope Width	
var	error	C_A	T	C_A	T	C_A	T
–	0%	$[0.156, 0.157]$	$[508.667, 508.673]$	$[0.845, 0.846]$	$[350.998, 350.999]$	0.690	148.386
k_0	$\pm 1\%$	$[0.149, 0.165]$	$[507.087, 510.116]$	$[0.843, 0.847]$	$[350.775, 351.227]$	0.697	154.847
k_0	$\pm 5\%$	$[0.143, 0.176]$	$[505.000, 511.207]$	$[0.299, 0.856]$	$[348.883, 460.074]$	0.713	161.186
h_r	$\pm 1\%$	$[0.139, 0.172]$	$[505.000, 513.859]$	$[0.844, 0.846]$	$[350.789, 351.212]$	0.707	158.612
h_r	$\pm 5\%$	$[0.120, 0.235]$	$[488.750, 524.542]$	$[0.201, 0.854]$	$[348.856, 486.789]$	0.734	169.298
E_a	$\pm 1\%$	$[0.133, 0.189]$	$[502.239, 513.269]$	$[0.224, 0.861]$	$[347.938, 513.269]$	0.728	165.331
E_a	$\pm 5\%$	$[0.074, 0.900]$	$[360.100, 525.000]$	$[0.108, 0.868]$	$[346.571, 498.245]$	0.794	178.430

Table 5: Decreasing C_{Ai} to 0.9 from steady-state 3 for the imprecise CSTR for 1 and 5% error in k_0, h_r and E_a. For each parameter, increasing the uncertainty increases the width of the initial state, final state, and envelope width.

the degree of imprecision in the ODE model and reduces to zero (and gives a single, zero-width trajectory) in the case of a precise ODE model.

Under some circumstances, however, the properties of the imprecisely-defined system are such that our inference methods produce dynamic envelopes that are overly wide given the amount of model imprecision. In Section 5, we examine related methods for clues on how to further improve bounds.

4.3 Other Examples

In addition to its application in predicting the behavior of the CSTR, NSIM has also been applied to a variety of other systems, including a model of a semiconductor vacuum system during pumpdown that was used in a monitoring system [Kay, 1991, Kay and Kuipers, 1993]. Because of the highly nonlinear nature of the model, a small error in modeling can have a large effect on the predicted behavior, and so it is very difficult to accurately predict the dynamic behavior of the system with a precise model. Thus, the typical method for diagnosing faults in these systems requires waiting until the system reaches equilibrium (which can take considerable time). By explicitly representing the modeling error in an SQDE and using SQSIM, it was possible to generate a robust dynamic prediction and monitor for faults while the pressure of the vacuum system was still changing, thus reducing fault detection time.

5 Related Work

There are very few general methods for producing bounded predictions from imprecise ODE models. In this section we examine one technique – interval ODE simulation – and describe how it compares with SQSIM.

Interval ODE simulation has been an active area of research for over thirty years. Interval ODE simulators focus on the problems of arithmetic and round-off error caused by using computer arithmetic to solve *precise* ODE systems. Interval ODE methods consider initial state and parameter uncertainty, but they normally assume that they are due primarily to machine precision and thus are small. An interval ODE simulation problem can thus be described as

> Given a precise, parameterized order-n ODE system of the form $x' = f(x, c)$ with state vector x and constant vector c and an initial bound on each component of x and c, determine a hypercube $X(t)$ such that $\forall_{i=1,n}\ x_i(t) \in X_i(t)$ for all t.

Most interval simulation methods rely on the concept of inclusion monotonicity to determine enclosures on the trajectories of state variables over time. Inclusion monotonicity is the property that if X is an interval vector and $F(X)$ is an interval function[12] then if $Y \subseteq X$ then $F(Y) \subseteq F(X)$. Typical algorithms [Moore, 1979, Corliss, 1995b] work in two stages:

1. Compute a coarse bound on $x(t)$ over some time interval $I = [t_j, t_j + h]$.
   Assume that at t_j, $X(t_j)$ bounds the trajectory. If we can find an A such that $X(t) \subseteq A$ for $t \in I$, then A will be a coarse bound for the trajectory over I. The trick is to find A (and a suitable h) so that the bound is tight and h is large. One method for finding such a bound is to use a first order Taylor expansion:

   $$A = X(t_j) + [0, h]F(t_j, X(t_j)) + \alpha$$

   where h and α are selected based on the problem.

   Because A is selected using only information at t_j, we must verify that it does indeed bound the trajectory over all of I. Inclusion monotonicity ensures that we have such a bound if, for example

   $$X(t_j) + [0, h]F([t_j, t_j + h], A)) \subseteq A$$

   If the inclusion is not met, we may either shrink h or widen A so as to maintain the inclusion.

2. Compute a tighter bound based on A.
   We can compute a tighter bound over I by expanding the Taylor Series for X and bounding the remainder with A. One such method is to use[13]

   $$X(t) = X(t_j) + \sum_{i=1}^{p} \frac{X^{(i)}(t_j)}{i!}(t - t_j)^i + \frac{A^{(p+1)}}{(p+1)!}(t - t_j)^{p+1}$$

   for $t \in I$ where the Taylor coefficients are computed recursively starting with $X(t_j)$ and A (see [Moore, 1979] for details).

[12] We say that F is an interval extension of f if $F(X_1, \ldots, X_n) = f(x_1, \ldots x_n)$ as long as the interval $X_i = [x_i, x_i]$ (i.e., it is of zero width).

[13] The symbol $Y^{(i)}$ represents the ith time derivative of Y (i.e., $\frac{d^iY}{dt^i}$).

Interval ODE simulators are also susceptible to the effects of ignoring correlations between variables, particularly the wrapping effect. Modern simulators such as AWA [Lohner, 1988] minimize this effect by using a coordinate system that that follows the trajectory of the system. AWA does this by producing an approximate solution to the ODE system and using the angle between this solution and the coordinate axes to rotate the frame of reference for computing intervals. This can greatly reduce the wrapping effect in some cases.

5.1 Comparing SQsim and Interval Simulation

In this section, we compare interval ODE methods to both the Nsim subsystem as well as the entire SQsim system. Both Nsim and interval ODE methods produce dynamic envelopes for an SQDE model. The key difference is in the focus of each method. Interval simulation methods focus on accounting for error in machine arithmetic and approximation error caused by using floating point numbers to represent real numbers in the prediction. The models and initial conditions are assumed to be precise (or very nearly so). Nsim concentrates on representing and reasoning with models that have significant imprecision and is less concerned with arithmetic error, relying on the underlying ODE solver to intelligently deal with this problem[14].

One important advantage that Nsim has is that it can simulate models that have functional uncertainty in the form of monotonic static envelopes. Functional uncertainty cannot be allowed in a method that uses Taylor expansions since computing Taylor coefficients requires a precise mathematical form.

Another Nsim advantage is that since the extremization process produces an ODE system, conventional ODE methods can be used to produce bounds. This permits the use of state-of-the-art numerical ODE solvers that may behave better with stiff systems as well as Differential Algebraic simulators that can include algebraic constraints in addition to the differential equations.

One weakness of Nsim is that extremization removes the original structure of the SQDE and replaces it with a new ODE structure. This new structure will often have very different dynamics from the original system so it is difficult to determine what correlations are being ignored in the extremal system. In particular, using coordinate transformations to combat wrapping becomes more difficult.

The main advantage of SQsim over interval ODE simulation is the integration of qualitative and quantitative inference to improve predictions. By using qualitative cues, SQsim is able to better guide Nsim into producing tighter bounds. The precise CSTR model is a good example of this since interval ODE simulators produce divergent predictions for this model (as does Nsim alone). By using qualitative information, however, SQsim is able to eliminate the divergence in the predictions. This suggests that existing interval ODE simulators could benefit from using an SQsim-like approach to simulation.

[14] Of course, since the extremal system is a precise ODE system, there is no reason that Nsim could not use an interval ODE solver to ensure that this source of error is accounted for.

5.2 Other Methods

Interval simulation methods were examined in detail in the previous section because many of the problems with reasoning about SQDEs also affect interval ODE simulation and solutions have been investigated and implemented by researchers in this field. There are, however, several other methods that provide useful perspectives on the problems of SQDE simulation. In the interest of completeness, we list some of these below.

- The NIS system [Vescovi *et al.*, 1995] is similar to NSIM in that it produces dynamic envelopes for systems describable by SQDEs. The method calculates the derivative of the SQDE system at each time point using interval arithmetic and then uses this interval bound in a custom-designed simulator to compute envelopes on the solution (current methods include Euler and Runge-Kutta). While it admits a slightly wider class of functions than NSIM, NIS is also subject to the same limitations caused by using an interval uncertainty representation, wrapping, and ignoring model correlation. Since it is a purely numerical method, it cannot exploit qualitative information about the model to prevent envelope divergence as SQSIM does.

- The system of Danès [Danès *et al.*, 1993] simulates precise models whose inputs are imprecisely-defined and satisfy certain controllability and observability requirements (which means that it can simulate only a subset of the SQDEs that SQSIM can). It is noteworthy because it casts the bounding problem as a set of optimization problems in the state space of the system. This allows it to avoid manipulating state-space uncertainty representations, thus sidestepping the correlation problems of most other methods.

- The paper by Corliss [Corliss, 1995a] describes a variety of other bounding methods for ODE systems and analyzes conditions under which they are applicable.

There is also a large body of theoretical work that provides a formal basis for many bounding methods:

- The field of differential inequalities [Walter, 1970] has relevance to bounding behaviors of imprecisely-defined ODE systems. This field analyzes conditions under which bounds exist for an ODE system. Several algorithms (e.g., [Markov and Angelov, 1986]) follow directly from this work.

- The field of differential inclusions [Aubin and Cellina, 1984] views the SQDE prediction problem from a framework where state and model uncertainty are directly represented by sets of state-space points and ODE models, respectively, and analyzes conditions under which bounds for the system exists. In [Hüllermeier, 1995], Hüllermeier examines SQDE simulation as done by NSIM in this framework and suggests methods for using alternate representations that can sometimes produce better bounds at greater computational expense.

There has also been applied work in behavior bounding in the area of VLSI simulation by Zukowski [Zukowski, 1986]. When treated as a resistive and capacitive switching network, digital VLSI circuit ODE systems are quasi-monotonic (or nearly so) and so simplified bounding methods can be used to predict circuit behavior by bounding the true circuit equations by linear ones and simulating these.

All of these methods treat the bounding problem in the quantitative domain and thus are similar in spirit to NSIM. This suggests that these methods could be improved by integrating them into a qualitative-to-quantitative framework with SQSIM. Since each method has its strengths and weaknesses, it would then be possible to select the appropriate method given the nature of the SQDE itself.

6 Summary and Future Work

This paper has described a new method for making predictions from imprecise models. The SQSIM system produces predictions from a mixed qualitative/quantitative representation of an imprecise ODE system. By separating exactly known qualitative properties from imprecisely known quantitative ones, SQSIM is able to make precise inferences from the qualitative information and use them to guide the production of quantitative predictions.

In particular, SQSIM models and predictions have the following properties:

- Behaviors are represented by a tree of qualitative behaviors, events, and dynamic trajectory envelopes. Thus, predictions are more detailed than is possible with a simple set of envelopes.
- All real behaviors of the SQDE are predicted by SQSIM.
- SQDEs can capture functional as well as parametric uncertainty. Thus a given model can cover a larger model space than can a model with parametric uncertainty alone.
- Model uncertainty is represented by ranges on parameters and static envelopes on unknown function relationships. These representations are commonly used to describe uncertainty in engineering domains.

These properties of SQSIM make it well-suited for use in design applications [Gazi *et al.*, 1993] (where guaranteed bounds are very useful) as well as for monitoring and diagnosis applications (where covering a large model space helps to reduce the number of candidate models) [Dvorak and Kuipers, 1989] .

Monitoring applications typically start with a given set of models which are matched against the data. A related area of research involves the derivation of models from a set of measurements of a physical system.

Our future plans include improving SQSIM numerical predictions in cases where ignored correlations are a problem. One possibility is to integrate AWA's coordinate transformation technique to reduce the effect of wrapping. Another possibility is to use an advanced form of Target Interval Splitting together with NSIM to rule out sections of the state space of a model.

The ultimate goal of our research is to construct a system that can identify a process directly from a stream of measurements. Such a system would first determine QDEs that are consistent with the measurements and then form SQDEs by bounding the monotonic functions with static envelopes and the constants with intervals. Initial work on the former step has been developed in [Richards *et al.*, 1992] and [Kay and Ungar, 1993] describes work on the latter step. A key requirement for integrating these tasks is the ability to produce useful predictions from an imprecise model. SQSIM helps address this need by providing a new form of inference especially suited to semiquantitative models.

A Proof that NSIM bounds all behaviors of an SQDE

Theorem 1 *Let $\mathcal{S}$ be an order-n SQDE with state vector $\mathbf{x} = [x_1, \ldots, x_n]$ and let the bounds on $\mathbf{x}$ be given by the vector $\mathbf{X} = [\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n]$. Let A:$\mathbf{x}' = f(\mathbf{x})$ be any ODE model in the space of $\mathcal{S}$ and $\alpha : \mathbf{X}' = F(\mathbf{X})$ be the extremal system for $\mathcal{S}$ with initial condition $\forall_{i=1,n}\ \underline{x}_i(t_0) \leq x_i(t_0) \leq \overline{x}_i(t_0)$. Then $\forall_{i=1,n}\ \underline{x}_i(t) \leq x_i(t) \leq \overline{x}_i(t)$ for all $t \geq t_0$.*

Proof There are two parts to the proof. The first part shows that the extremal system produced by NSIM guarantees that

$$\underline{f}_i(\mathbf{X}) \leq f_i(\mathbf{x}) \leq \overline{f}_i(\mathbf{X}) \tag{3}$$

when

$$\forall_{j \neq i}\ \underline{x}_j \leq x_j \leq \overline{x}_j \text{ and } \underline{x}_i = x_i = \overline{x}_i \tag{4}$$

The second part shows that if this relation is true, then the bound follows.

Part One Let $x_i' = f_i(\mathbf{x})$ be the ith ODE in A. Referring to Table 1, we can see that for each of the "ground" expressions $e = c$, x_i, or x_j we have that $L(e) \leq e \leq U(e)$ whenever Condition 4 holds. By induction on e, we can show that the other expressions in the table also obey $L(e) \leq e \leq U(e)$. Thus, letting $e = f_i$ we have $\underline{f}_i(\mathbf{X}) \leq f_i(\mathbf{x}) \leq \overline{f}_i(\mathbf{X})$ for each i.

Part Two Given that (3) and (4) hold, we now show that $\overline{x}_i(t) \geq x_i(t)$ for all $t \geq t_0$. The proof for the lower inequality is derived similarly. For the purposes of intuition, we prove a slightly weaker version of the theorem in which we assume $f_i(\mathbf{x}) < \overline{f}_i(\mathbf{X})$ when Condition 4 holds. A complete proof which assumes $f_i(\mathbf{x}) \leq \overline{f}_i(\mathbf{X})$ is given in [Bothe, 1992] and [Hüllermeier, 1995].

The bound holds at t_0 by assumption. For any other time t, we have either

- $\overline{x}_i(t) > x_i(t)$ in which case the result holds trivially, or

- $\overline{x}_i(t) < x_i(t)$ in which case there must be some time $t_1 < t$ such that $\overline{x}_i(t_1) = x_i(t_1)$, or
- $\overline{x}_i(t) = x_i(t)$.

Assume therefore that at time t we have $\overline{x}_i(t) = x_i(t)$ and that $\forall_{j,\tau}\ \underline{x}_j(\tau) \leq x_j(\tau) \leq \overline{x}_j(\tau)$, i.e., time t is the first time that *any* state variable might cross its bounds (if there is some state variable x_h for which the inequality does not hold, we may then take t to be the time when x_h is equal to one of its bounds and let $i = h$). We seek to determine if $\overline{x}_i$ and x_i can cross at t.

Define the difference term $d_i(t)$ and compute its Taylor Series expansion:

$$\begin{aligned} d_i(t+h) &= \overline{x}_i(t+h) - x_i(t+h) &(5)\\ &= (\overline{x}_i(t) - x_i(t)) + h(\overline{f}_i(t) - f_i(t)) + \frac{h^2}{2}(\overline{f}'_i(\zeta) - f'_i(\phi)) &(6)\end{aligned}$$

where $\zeta, \phi \in [t, t+h]$.

By assumption, the first term of this expression is zero. Defining $\epsilon_i(t) = \overline{f}_i(t) - f_i(t)$ and rearranging terms gives

$$d_i(t+h) = h(\epsilon_i(t) + \frac{h}{2}(\overline{f}'_i(\zeta) - f'_i(\phi))) \quad (7)$$

Since this is the first time at which a state variable and bounds might cross, Condition 4 must hold and so $\overline{f}_i(t) > f_i(t)$. Thus $\epsilon_i(t) = \overline{f}_i(t) - f_i(t) > 0$. Since $\epsilon_i(t) > 0$, $d_i(t+h)$ will be positive if $\epsilon_i(t) > \frac{h}{2}|\overline{f}'_i(\zeta) - f'_i(\phi)|$ which will be true for all h such that[15]

$$h < h_{min} = \frac{2\epsilon_i(t)}{|\overline{f}'_i(\zeta) - f'_i(\phi)|}$$

Thus, there is some finite interval $h \in [0, h_{min}]$ over which $\overline{x}(t+h)$ is non-decreasing, hence $\overline{x}_i$ cannot cross below x_i at t. Since this is the first point at which *any* bound could cross, we have shown that there is no time at which an upper bound trajectory could cross that of a member of $\mathcal{S}$ and so the bound must hold over all time.

References

[Aubin and Cellina, 1984] J. P. Aubin and A. Cellina. *Differential Inclusions.* Springer-Verlag, 1984.

[Berleant and Kuipers, 1992] Daniel Berleant and Benjamin Kuipers. Combined qualitative and numerical simulation with Q3. In Boi Faltings and Peter Struss, editors, *Recent Advances in Qualitative Physics.* MIT Press, 1992.

[15] If $\overline{f}'_i(\zeta) - f'_i(\phi) = 0$ then $d_i(t+h) = h\epsilon_i(t) > 0$.

[Bothe, 1992] Dieter Bothe. Multivalued differential equations on graphs. *Nonlinear Analysis, Theory, Methods, and Applications*, 18(3):245–252, 1992.

[Corliss, 1995a] George F. Corliss. Guranteed error bounds for ordinary differential equations. In *Theory of Numerics in Ordinary and Partial Differential Equations*. Oxford University Press, 1995. Also available at `ftp://boris.mscs.mu.edu/pub/corliss/ValidODE`.

[Corliss, 1995b] Geroge F. Corliss. Introduction to validated ODE solving. Technical Report 416, Department of Mathematics, Statistics and Computer Science – Marquette University, Milwaukee, WI 53233, March 1995.

[Dalle Molle, 1989] David Dalle Molle. Qualitative simulation of dynamic chemical processes. Technical Report AI89–107, Artificial Intelligence Laboratory, University of Texas at Austin, Austin, Texas 78712, May 1989.

[Danès *et al.*, 1993] P. Danès, L. Travè-Massuyès, and J. Aguilar-Martin. A generic method for computing the response of a numerically-known dynamic system to qualitative inputs. In *Proceedings of IEEE-SMC'93*, Le Touquet, France, 1993.

[Dvorak and Kuipers, 1989] Daniel Louis Dvorak and Benjamin Kuipers. Model-based monitoring of dynamic systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1238–1243, 1989.

[Gazi *et al.*, 1993] Evangeline Gazi, Herbert Kay, Benjamin Kuipers, W. D. Seider, and Lyle H. Ungar. Controller verification using qualitative reasoning (extended abstract). In *American Institute of Chemical Engineers*, 1993.

[Gelb, 1974] Arthur Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[Hüllermeier, 1995] Eyke Hüllermeier. Numerical simulation methods for uncertain dynamics. Technical Report tr-ri-95-170, Heinz Nixdorf Institut, Universität–GH–Paderborn, Fürstenaallee 11, D-33102 Paderborn, Germany, November 1995.

[Kahaner *et al.*, 1989] David Kahaner, Cleve Moler, and Stephen Nash. *Numerical Methods and Software*. Prentice-Hall, Englewood Cliffs, 1989.

[Kay and Kuipers, 1993] Herbert Kay and Benjamin Kuipers. Numerical behavior envelopes for qualitative models. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 606–613, 1993.

[Kay and Ungar, 1993] Herbert Kay and Lyle H. Ungar. Deriving monotonic function envelopes from observations. In *Working Papers from the Seventh International Workshop on Qualitative Reasoning about Physical Systems*, pages 117–123, Orcas Island, Washington, 1993.

[Kay, 1991] Herbert Kay. Monitoring and diagnosis of multi-tank flows using qualitative reasoning. Master's thesis, The University of Texas at Austin, May 1991.

[Kuipers and Berleant, 1988] Benjamin Kuipers and Daniel Berleant. Using incomplete quantitative knowledge in qualitative reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 324–329, 1988.

[Kuipers, 1984] Benjamin Kuipers. Commonsense reasoning about causality : Deriving behavior from structure. *Artificial Intelligence*, 24:169–204, 1984.

[Kuipers, 1986] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, September 1986.

[Kuipers, 1994] Benjamin Kuipers. *Qualitative Reasoning – Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA, 1994.

[Lohner, 1988] Rudolf Lohner. *Einschliessung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben un Anwendungen*. PhD thesis, Universität Karlsruhe, 1988.

[Lunze, 1989] Jan Lunze. *Robust Multivariable Feedback Control*. Prentice Hall, 1989.

[Markov and Angelov, 1986] S. Markov and R. Angelov. An interval method for systems of ODE. In *Lecture Notes in Computer Science #212 – Interval Mathematics 1985*, pages 103–108. Springer-Verlag, Berlin, 1986.

[Moore, 1979] Ramon E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

[Reckhow, 1987] K.H. Reckhow. Error analysis, first order. In Madan O. Singh, editor, *Systems and Control Encyclopedia – Theory, Technology, Applications, Volume 6*, pages 1552–1553. Oxford, 1987.

[Richards *et al.*, 1992] Bradley L. Richards, Ina Kraan, and Benjamin Kuipers. Automatic abduction of qualitative models. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 723–728, 1992.

[Vescovi *et al.*, 1995] Marcos Vescovi, Adam Farquhar, and Yumi Iwasaki. Numerical interval simulation: Combined qualitative and quantitative simulation to bound behaviors of non-monotonic systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1806–1812, 1995.

[Walter, 1970] Wolfgang Walter. *Differential and Integral Inequalities*. Springer-Verlag, Berlin York, 1970.

[Zukowski, 1986] Charles A. Zukowski. *The Bounding Approach to VLSI Circuit Simulation.* Kluwer Academic Publishers, Boston, 1986.