ROBUST IDENTIFICATION USING SEMIQUANTITATIVE METHODS

Herbert Kay [1]

Caelum Research / NASA Ames Research Center
Moffett Field, CA 94035
bert@ptolemy.arc.nasa.gov

Abstract: While model-based process monitoring and system identification share the common goal of describing the behavior of a physical system based on a mathematical model, the assumptions made by system identification are typically too strong to apply it directly to monitoring problems. This article describes SQUID, a new system identification method that uses refutation rather than search to identify a model of a physical system. By ruling out implausible models rather than searching for the best model that fits the data, SQUID is more robust in the face of uninformative data and structural model uncertainty than are traditional identification methods.

Keywords: System identification, Monitoring, Diagnosis, Qualitative simulation, Numerical simulation

1. INTRODUCTION

This article describes SQUID (for SemiQUantitative IDentifier), a new system identification (SI) method that uses refutation rather than search to refine a model space of Ordinary Differential Equations (ODEs). SQUID was designed to meet the needs of monitoring under conditions where traditional SI methods have difficulty due to severe model imprecision and uninformative observations.

System identification can be viewed as a process that consists of a *model selector* that chooses a particular model space from a set of model spaces and a *refinement operator* that uses the observations to reduce the model space. If the model space cannot be refined, then the model selector chooses a new model space and the process continues. Typical system identification methods for state-space modeling describe model spaces with parametric ODE models (Ljung, 1987). The refinement operator consists of a parameter estimation method (which generates an estimate and bound for each parameter) and a validation method that determines if the refined model is acceptable.

Model-based monitoring (Dvorak and Kuipers, 1989) can also be viewed in terms of a model selector (the *hypothesis generator*, which selects a model space) and a refinement operator (the *tracker*, which simulates the fault model and matches it to the observation stream thus reducing the model space). Thus, monitoring can be seen as a form of identification, and, in particular parameter estimation can be seen as a tracking method. Unfortunately, there are several aspects to the monitoring problem that make tracking with parameter estimation difficult to apply:

- Parametric models do not easily capture structural uncertainty. Often, failures result in structural changes to the physical system that may change the functional relationships

[1] 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.

between modeled variables. While structural change can be represented using multiple parameterizations, this approach complicates the model selection phase.

- Parameter estimation does not work well with uninformative data. If the dynamics of the process are not adequately displayed in the observation stream, the parameter estimator may converge to the wrong model. Since physical systems that are in continuous operation do not always exercise their full dynamics over a fixed period of time, they will often generate uninformative data, resulting in refined models that may falsely signal a discrepancy with observations.
- Validation is difficult with uninformative data. Thus, it may be difficult to tell if the refined model is correct.
- Parameter estimation is expensive for highly-parameterized ODE models since the ODE must be solved at every search step with new parameter values. Furthermore, gradient estimates are more error-prone since they must be calculated using finite differences.

While there are existing identification-based monitoring systems (see (Isermann, 1984)), they assume that these difficulties are not present.

SQUID overcomes these difficulties in two ways. By using a more descriptive model representation that includes both parametric and structural uncertainty, it can provide better coverage of faults. By using refutation to rule out models rather than searching for a best model, SQUID is robust in the face of uninformative observations and is applicable in situations where parametric estimation is too expensive.

Section 2 describes the representation for model spaces and their predictions as well as the simulation method. Section 3 details the design of SQUID and Section 4 demonstrates SQUID on two problems related to monitoring. Section 5 compares SQUID and traditional system identification and Section 6 discusses the impact of SQUID and future work in improving model-based monitoring with semiquantitative models.

2. REPRESENTING AND SIMULATING SEMIQUANTITATIVE (SQ) MODELS

SI methods operate in two domains – the *model space* which is the set of models that may describe the physical process and the *trajectory space* which is the set of all trajectories produced by members of the model space. This section describes the SQUID representation for each of these spaces.

The size of the initial model space for an SI problem is determined by the amount of uncertainty about the physical process that it represents. To describe this uncertainty, 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. SQUID makes use of this distinction by using a multi-level representation based on the QSIM (Kuipers, 1984; Kuipers, 1994) representation for ODE systems. The levels are:

- The structural level (SDE).
  At this level, the form of the ODE system is described 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.
- The qualitative level (QDE).
  The qualitative level adds information about each model variable by breaking its domain into and ordered list of *landmarks* that represent important values of the variable. The QDE also adds information about the shape of the functional constraints (i.e., monotonic, parabolic, sigmoidal, etc.).
- The quantitative level (SQDE).
  This level captures the uncertainty in the model. Parametric imprecision is represented with numerical intervals that bound the uncertainty while functional imprecision is represented by *static envelopes* within which the functional constraint must lie.

SQUID uses the SQDE as its model space representation. Note that since each level includes the levels above it, the SQDE contains both QDE and SDE descriptions. Figure 1 shows the representation of a simple one-tank system with a constant inflow. Note that the SQDE can represent a single ODE system by restricting the width of landmark intervals and static envelopes to zero. The SQDE therefore covers a space of ODE systems.

The SQsim simulator (Kay, 1996) generates a trajectory space in the form of *SQ behaviors* from an SQDE and initial condition. An SQ behavior consists of three components:

- The *qualitative trajectory* describes the shape of the behavior in terms of regions of monotonic behavior with respect to time. This description is produced by applying the QSIM qualitative simulator to the QDE portion of the SQDE.
- The *semiquantitative events* describe the uncertainty about the location of events (e.g.,

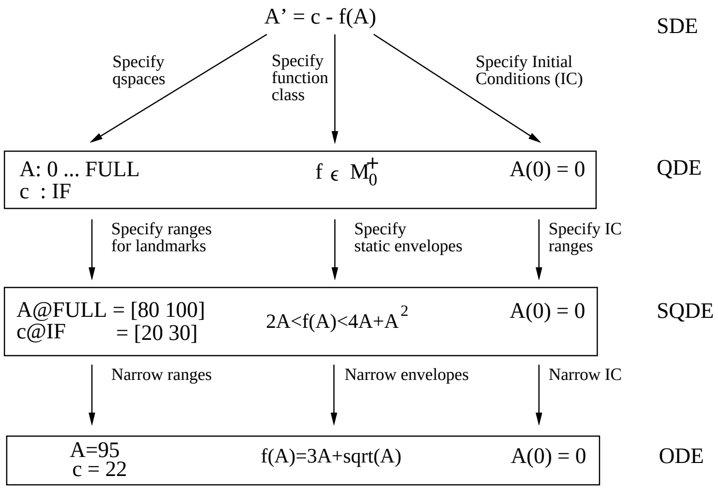


Fig. 1. Multi-level model of an imprecisely-known system. At each level, the representation covers a space of ODEs. At the top is the purely structural SDE. The QDE adds qualitative information to the SDE, and the SQDE adds imprecise numerical information to the QDE. Moving downward specifies more information, thus reducing the size of the model space.

the value and time of an extremum) by providing interval bounds on the value and time of the event. This description is produced by using the Q2 interval propagator (Kuipers and Berleant, 1988), which propagates the ranges associated with events through the SDE portion of the SQDE.

- The *dynamic envelopes* define the overall bounds of the trajectory of the system with a pair of functions that bound all trajectories of the system. This description is produced by the NSIM (Kay and Kuipers, 1993) simulator which forms an ODE system whose behavior is guaranteed to bound all behaviors of ODEs covered by the SQDE.

The resulting SQ behaviors carry the guarantee that all real behaviors of the ODEs covered by the SQDE are included in the SQ behavior set generated by SQSIM. This guaranteed coverage property is essential to SQUID's refinement operator.

3. DETAILS OF SQUID

SQUID refutes portions of a model space by matching predictions to observations and then mapping the results back to the model space (see Figure 2). The portion of the model space that does not match is eliminated, resulting in a better model of the process. The mapping from model space to trajectory space is done via a simulator and the reverse mapping is done by an *inverse simulator*.

In traditional identification, only one model at a time from the model space is matched against the measurements. While this approach simplifies both the simulator (which can use standard

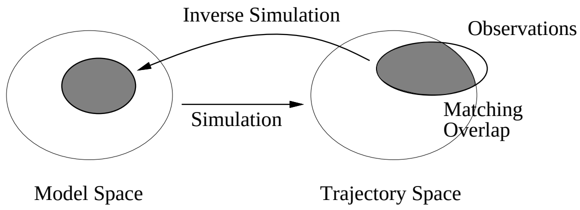


Fig. 2. Refinement matches observations to the trajectory space of a corresponding model space. It then infers a smaller model space that is consistent with the smaller trajectory space.

numerical methods) and the inverse simulator (which is trivial since there is a one-to-one mapping between a precise model and its behavior), it forces one to view the model space as a collection of independent models, ignoring any similarity between them.

In contrast, SQUID views the model space as a collection of sets of models that are related by abstract properties of their associated trajectories (monotonicity, location of extrema, etc.). Matching is then performed between these abstract properties of the trajectory space and the corresponding properties of the observations (see Figure 3). This process is called *trend matching* to distinguish it from the *raw data matching* of traditional methods.

Trend matching has several advantages over raw data matching:

- Abstract properties describe sets of models, thus trend matching can rule out multiple behaviors at once.
- Since abstract properties are simpler to match than precise functional forms, trend

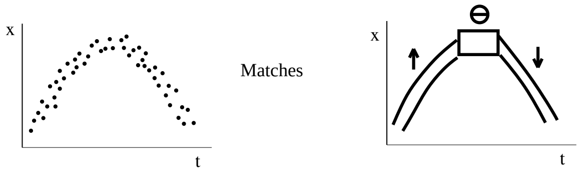


Fig. 3. Trend matching uses abstract properties of the trend (monotonicity, location of extrema, and bounding envelope) to compare data to models. This provides a trajectory representation suitable for comparison with SQSIM predictions.

matching can be cheaper than raw data matching.

- The trend description for a set of observations is model-independent. Thus, the same trend description can be used to match against multiple models.

By describing the trend in the same way as SQSIM describes trajectories (i.e., using qualitative trajectory, event, and dynamic envelope descriptions), it is easy to compare the two and see where they overlap. This overlap can then be used by the inverse simulator to refute models whose predictions do not fall in the overlap region.

SQUID takes as input an SQ behavior generated by SQSIM together with a set of noisy observations. There are three steps to SQUID:

(1) Form an *SQ trend* for each measured variable. This trend represents the set of all possible trajectories that could have generated the observations (within some statistical confidence level). The SQ trend consists of trajectory, event, and dynamic envelope information for each measured variable.
(2) Map the SQ trends to the SQ behavior.
(3) If no match is found, the SQDE is refuted. Otherwise, refine the SQDE by using the inverse simulator to rule out those portions of the model space that could not have generated the match.

These steps are discussed more fully in the following sections.

3.1 *Forming the Semiquantitative Trend*

Forming the SQ trend requires producing qualitative, event, and dynamic envelope representations of the observation streams for each measured variable. This is done in three steps:

- Qualitative filtering produces the qualitative trend by determining regions of monotonicity and intervening extrema in the observation stream. It uses a filtering kernel that detects changes in slope across the filtering window and uses these to locate extrema.

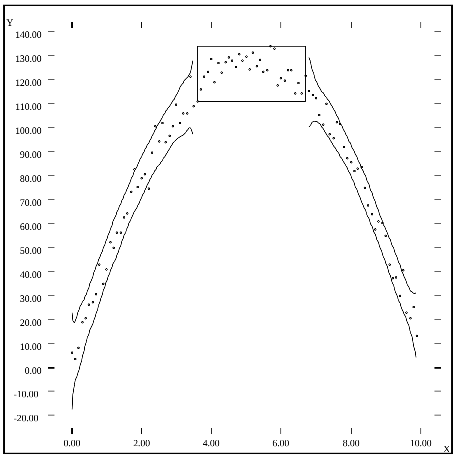


Fig. 4. Producing an SQ trend from noisy data. The qualitative filter partitions the data into an extremum (box) and the surrounding monotonic regions which are then bounded using MSQUID.

- The extent of events is determined by bounding the extrema detected by the qualitative filter.
- The dynamic envelope for each monotonic region is defined by using MSQUID (Kay and Ungar, 1993), a neural-network function approximator that derives an estimate and bounds on monotonic data.

Figure 4 shows the result of SQ trend formation.

3.2 *Mapping SQ Trends to SQ Behaviors*

Once the SQ trend of the measurement stream has been computed, it can be mapped to the SQ behavior generated from the SQDE by SQSIM. Since both descriptions presumably describe the same physical system, they should overlap. This overlap represents the section of the trajectory space that is consistent with both descriptions and will normally be smaller than the trajectory space defined by the SQ behavior alone.

The SQ trend and SQ behavior are compared by mapping each of their components separately. For each component, the mapper seeks to reduce the size of the SQ behavior so as to yield a smaller trajectory space for the model.

3.3 *Refining the Model*

Model refinement is the process of mapping a trajectory space back to the model space that generated it. In the case of SQUID, this mapping takes an SQ behavior and determines the SQDE that covers the smallest set of ODEs that could have produced it while preserving the guarantee

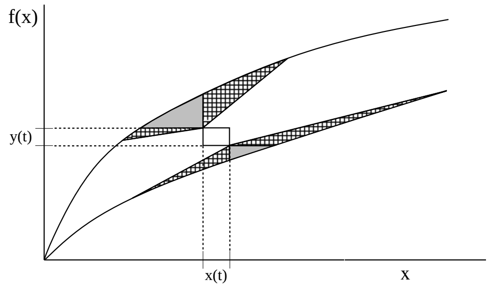


Fig. 5. Refinement of static envelopes for $y = f(x)$. Observations imply that at time t, the system lies in the rectangle $[x(t), y(t)]$. Since f is monotonic, the gray regions of the envelope are unreachable. By utilizing curvature information, the envelope can be reduced further (crossed regions).

that no ODE is excluded unless it is genuinely impossible. This process is referred to as *inverse simulation*. Note that there are two sources of imprecision in the SQDE – parameter uncertainty and static envelope uncertainty. The inverse simulator must therefore refine these sources to refine the model.

Parametric uncertainty is reduced by using the Q2 interval propagator to refine bounds on model parameters given bounds on observed variables at specific time-points. Once propagation has reduced the parametric uncertainty, the static envelopes can be refined by excluding regions that do not obey the monotonicity requirements given the uncertainty at a time-point (see Figure 5).

4. EXAMPLES

In this section SQUID is applied to the task of identifying a two-tank cascade. The SQDE is defined by [2]:

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

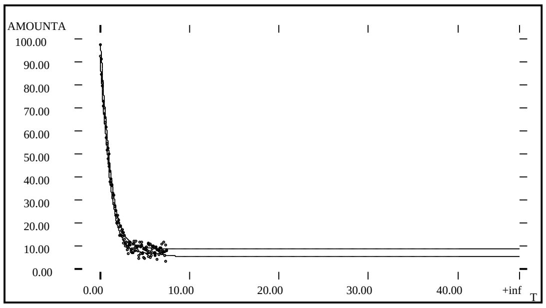
with $c \in [25, 25]$ and static envelopes for both f and g of $[2x, 15\sqrt{x}]$ for $x < 15$ and $[0.1x + 30, 15\sqrt{x}]$ for $x \geq 15$. Note that this model contains considerable functional uncertainty in both f and g and is hence unrepresentable using a parametric model.

Observational data was generated from the model:

$$
\begin{aligned}
A' &= 25 - 9\sqrt{A} \\
B' &= 9\sqrt{A} - 8\sqrt{B}
\end{aligned}
$$

$$A(0) = 95 \quad B(0) = 0$$

[2] The prime ($\prime$) symbol denotes the time derivative of a variable.



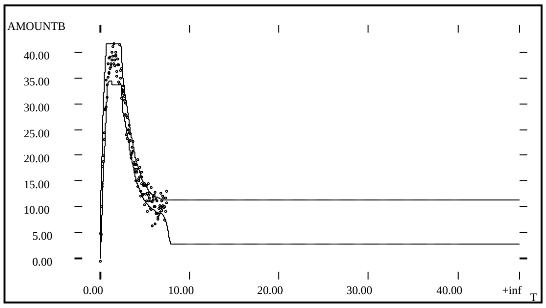


Fig. 6. Dynamic envelopes for A and B together with measurements for the two-tank cascade using the envelopes in Figure 7.

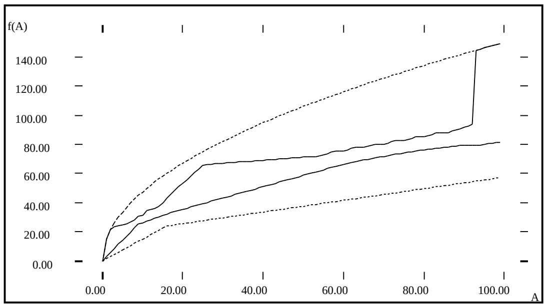
with additive noise of variance 4 added to the observed terms A, B, $f(A)$, and $g(B)$. Figure 6 shows the predicted behavior of the system in terms of A and B while Figure 7 shows the improvement in the static envelopes (which represent the improvement in the definitions of f and g in the SQDE) as a result of applying SQUID to the observations.

SQUID has also been applied to the detection of temperature drift within an adiabatic CSTR as well as to a number of other single and double compartmental models.

5. COMPARISON TO TRADITIONAL IDENTIFICATION

The previous example demonstrates the key differences between parameter-based identification and SQUID:

- SQUID can refine non-parametric models by reducing the bounds on static envelopes (see Figure 7).
- Uninformative data does not eliminate plausible models. In the example, if data had been collected only after the system had reached equilibrium (e.g., after $t = 10$), SQUID would have been able to refine those portions of the model related to equilibrium behavior (such as the maximum value of $f(A)$) without refining the dynamics. Since they must find a best model, however, traditional



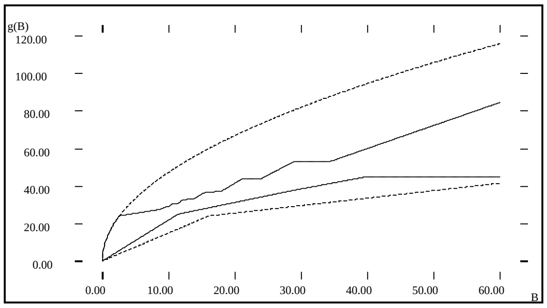


Fig. 7. Refinement of f and g in the two-tank cascade when A, B, $f(A)$, and $g(B)$ are measured. The outer envelopes are the initial ones provided by the SQDE.

identification methods would have selected a dynamics even though the observations did not exhibit any.

- Refutation is a conservative method. Since only implausible models are removed, the resulting model space can be larger than when using traditional estimation techniques which are based on finding a best model. For monitoring tasks, a larger model space translates into the need for more observations to detect a model-observation mismatch. Thus, SQUID gains robustness at the expense of potentially slower fault detection.

6. CONCLUSIONS

This article has described SQUID, a new method for the identification of of state-space models that uses refutation rather than search to refine the model space. This feature makes the method robust with respect to uninformative observations. SQUID uses an ODE model space representation that permits both parametric and structural uncertainty which permits a more flexible description of the model space. These two features make SQUID a good candidate for use in a monitoring system since uninformative data and structural uncertainty under faults are common.

SQUID shares with the MIMIC (Dvorak and Kuipers, 1989) and TrenDx (Haimowitz, 1994) systems the idea that model-based tracking and matching of observations to predictions is an effective method for monitoring. It's focus, however, is on methods for tracking rather than on methods for selecting among possible fault hypotheses.

Future work involves combining SQUID with a model-based monitoring system and applying it to more complex systems. Additionally, the use of SQUID as a guide for a traditional parameter identification method is being investigated. This latter system would combine the precision of traditional estimation methods with the robustness of semiquantitative methods.

7. REFERENCES

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

Haimowitz, Ira Joseph (1994). Knowledge-Based Trend Detection and Diagnosis. PhD thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Isermann, Rolf (1984). Process fault detection based on modeling and estimation methods – A survey. *Automatica* **20**(4), 387–404.

Kay, Herbert (1996). SQSIM: A simulator for imprecise ode models. Technical Report AI96–247. Artificial Intelligence Laboratory, University of Texas at Austin. Austin, Texas 78712.

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

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

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

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

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

Ljung, Lennart (1987). *System Identification: Theory for the User*. Prentice Hall. Englewood Cliffs, New Jersey.