Reasoning About Input-Output Modeling of Dynamical Systems

Matthew Easley and Elizabeth Bradley*

University of Colorado
Department of Computer Science
Boulder, CO 80309-0430
{easley,lizb}@cs.colorado.edu

To appear, Symposium on Intelligent Data Analysis (IDA-99)

Abstract. The goal of input-output modeling is to apply a test input to a system, analyze the results, and learn something useful from the cause-effect pair. Any automated modeling tool that takes this approach must be able to reason effectively about sensors and actuators and their interactions with the target system. Distilling qualitative information from sensor data is fairly easy, but a variety of difficult control-theoretic issues — controllability, reachability, and utility — arise during the planning and execution of experiments. This paper describes some representations and reasoning tactics, collectively termed *qualitative bifurcation analysis*, that make it possible to automate this task.

1 Input-Output Modeling

System identification (SID) is the process of inferring an internal ordinary differential equation (ODE) model from external observations of a system. The computer program PRET[5] automates the SID process, using a combination of artificial intelligence and system identification techniques to construct ODE models of lumped-parameter continuous-time nonlinear dynamic systems. As di-

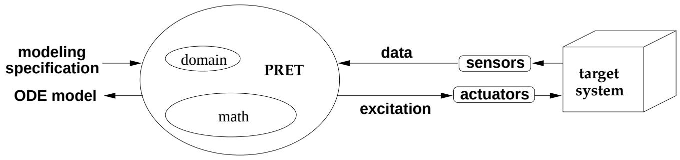

Fig. 1. PRET uses sensors and actuators to interact with target systems in an *input-output* approach to dynamical system modeling.

agrammed in Fig. 1, PRET uses domain knowledge to combine model fragments into ODEs, then employs actuators and sensors to learn more about the target system, and finally tests the ODEs against the actuator/sensor data using a body of mathematical knowledge encoded in first-order logic[20].

* Supported by NSF NYI #CCR-9357740, NSF #MIP-9403223, ONR #N00014-96-1-0720, and a Packard Fellowship in Science and Engineering from the David and Lucile Packard Foundation.

This *input-output* (I/O) approach to dynamical system modeling, which distinguishes PRET from other AI modeling tools, is very powerful and also extremely difficult. Distilling available sensor information into qualitative form is reasonably straightforward, as described in our IDA-97 paper[3], but reasoning about the information so derived is subtle and challenging. Dealing with actuators is even harder because of the nonlinear control theory that is involved. Among other things, determining what experiments one can perform from the system's present state involves complicated reasoning about *controllability* and *reachability*. In an automated framework, it is also important to reason about what can be learned from a given experiment. During the input-output modeling process, PRET must solve all three of these problems. That is, given a black-box system, a partial measurement of its current state, some knowledge about the available actuators, and some preliminary ideas about a candidate model, PRET must be able to decide what experiments are possible and useful. This is a difficult, open problem for nonlinear systems, even for human experts. The topic of this paper is a set of knowledge representation and reasoning techniques that make it possible to automate this task.

In linear systems these problems are relatively easy. Engineering approaches to linear input-output analysis are well developed; standard techniques for exciting different useful states of the system[13] include changing the type (e.g., ramp, step) or parameters (e.g., amplitude, frequency) of the input. The impulse response of a system — its transient response to a quick kick $x(t_0) = 1;\ x(t) = 0\,\forall\, t \neq t_0$ — is particularly useful. The natural resonant and anti-resonant frequencies appear as spikes and the *mode shapes* between those spikes can show whether a vibrating mechanical system is mass- or stiffness-dominated[15].

Nonlinear systems pose a far more imposing challenge to input-output modeling; their mathematics is vastly harder, and many of the analysis tools described in the previous paragraph do not apply. Almost all forms of transient analysis (e.g., step or impulse response) are useless in nonlinear problems, as is frequency response; the concept of a discrete set of spectral components simply does not make sense. Because of this, nonlinear dynamicists typically allow transients to die out and then reason about *attractors* in the *phase* or *state space*, and how the geometry and topology of those attractors change when the system parameters are varied.

Our approach targets the problems that arise in reasoning about multiple set of observations that arise in phase-portrait analysis of complex systems. In particular, we use a combined *state/parameter space* and decompose it into discrete regions, each associated with an equivalence class of dynamical behaviors, derived qualitatively using geometric reasoning. These discrete regions describe the behavior of the system in a uniquely powerful way. As each trajectory is effectively equivalent, in a well-known sense, to all the other trajectories in the same region, one can describe the behavior in that region in a much simpler way, which results in ease of analysis — and great computational savings.

The representation described in this paper — an abstraction/extension of the traditional nonlinear analysis technique termed *bifurcation analysis* — allows PRET's intelligent sensor analysis and actuator control modules to reason effectively about multiple sets of observations over a given system. Coupled with a knowledge representation and reasoning framework that adapts smoothly to how much one knows about the system (e.g., using linear analysis when appropriate), which is described in another paper[9], this representation allows PRET to reason effectively about input-output modeling of nonlinear dynamical systems.

To set the context, the following section gives a brief overview of PRET. We then focus in on the input-output modeling phase, describe our representation and reasoning framework, and show how PRET exploits that framework.

2 PRET

As outlined in the previous section, PRET[5] is an automated tool for nonlinear system identification (SID). Its inputs are a set of observations of the outputs of a black-box system, and its output is an ordinary differential equation (ODE) model of the internal dynamics of that system. PRET's architecture wraps a layer of artificial intelligence (AI) techniques around a set of traditional formal engineering methods like impulse-response analysis, nonlinear regression, etc. The AI layer combines several forms of reasoning,[1] via a special first-order logic inference system[18, 20] to intelligently assess the task at hand; it then reasons from that information to automatically choose, invoke, and interpret the results of appropriate lower-level techniques. This framework lets PRET shift fluidly back and forth between domain-specific reasoning and general mathematics to navigate efficiently through an exponential search space of possible models. This approach has met with success in a variety of simulated and real problems, ranging from textbook systems to real-world engineering applications.

PRET takes a "generate-and-test" approach to model building. It uses domain-specific knowledge to assemble combinations of user-specified and automatically generated ODE fragments into a candidate model;[2] it tests that model by performing a series of factual inferences about the ODE and the observations and then using a theorem prover[19] to search for contradictions in those sets of facts. The technical challenge here is efficiency: the search space is huge, and so PRET must identify contradictions as quickly, simply, and cheaply as possible. The key to doing so is to classify model and system behavior at an appropriate qualitative level and to exploit all available domain-specific knowledge in the most useful way. Symbolic algebra can be used to remove huge branches from the search space. If the target system is known to be chaotic, for instance, all linear ODEs can be immediately discarded, and the computation involved — calculating the Jacobian and ascertaining that all of its entries are constant — requires only simple, inexpensive symbolic reasoning. In other situations, prun-

[1] qualitative reasoning, qualitative simulation, numerical simulation, geometric reasoning, constraint propagation, resolution, reasoning with abstraction levels, declarative meta-control, and a simple form of truth maintenance.

[2] In mechanics, for instance, PRET uses Newton's laws to combine force terms; in electronics, it uses Kirchhoff's laws to sum voltages in a loop or currents in a cutset.

ing a single leaf off the tree of possible models can be extremely expensive (e.g., estimating parameter values for a nonlinear ODE prior to a final corroborative simulation/comparison run, which is a complicated global optimization problem[4]). Some analysis methods, such as phase-portrait analysis, apply to all ODEs, whereas others are only meaningful in specific domains (e.g., creep tests in viscoelastic systems). Orchestrating this complex reasoning process is a very difficult problem; its solution requires carefully crafted knowledge representation frameworks[9] that allow for an elegant formalization of the essential building blocks of an engineer's knowledge and reasoning, and powerful automated machinery[20] that uses the formalized knowledge to reason flexibly about a variety of modeling problems.

The input-output modeling strategies that are the topic of this paper play important roles in both the generate and the test phase. The "input" half of PRET's *intelligent sensor/actuator analysis and control* module — which is reviewed briefly in the following sections and covered in detail in [3] — uses geometric reasoning and delay-coordinate embedding to distill abstract, useful qualitative information from a highly specific numeric sensor data set. The "output" part, described in the following sections, reasons about multiple sets of observations about a given system using a new knowledge representation called the *qualitative state/parameter space* and an associated reasoning strategy termed *qualitative bifurcation analysis*, both of which are AI-adapted versions of well-known nonlinear dynamics techniques. For more details on the rest of PRET — issues, solutions, internal representations, encoded knowledge bases, examples solved, etc. — please consult the papers cited in the previous two paragraphs.

3 Qualitative Bifurcation Analysis

One of the goals of the qualitative reasoning (QR) community[12] is to abstract specific instances of behavior into more-general descriptions of a system. An 80kg adult bouncing on the end of a bungee cord, for instance, will produce a different time series from a 50kg child, but both produce similar damped oscillatory responses. Reasoning about these two behaviors in their time-series form can be difficult, as it requires detailed examination of the amplitude decay rate of and the phase shift between two decaying sinusoids. The state-space representation, which suppresses the time variable and plots position versus velocity, brings out the similarity between these two behaviors in a very clear way. Both bungee jumps, for example, manifest on a state-space plot as similar decaying spirals. Automated phase-portrait analysis techniques[2, 21, 22], which combine ideas from dynamical systems, discrete mathematics, and artificial intelligence, generate qualitative descriptions that capture this information.

A discretized version of the state-space representation can abstract away many low-level details about the dynamics of a system while preserving its important qualitative properties. The cell-to-cell-mapping formalism[14], for instance, discretizes a set of n-dimensional state vectors onto an n-dimensional mesh of uniform boxes or *cells*. The circular state-space trajectory in Fig. 2(a), for example, — a sequence of two-vectors of floating-point numbers — can be represented

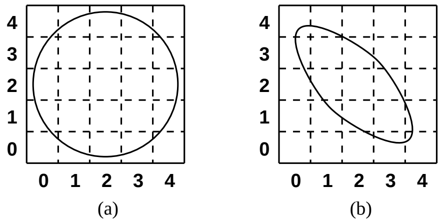

Fig. 2. Identifying a *limit cycle* using the cell-dynamics method.

by the *cell sequence* [... (0,0) (1,0) (2,0) (3,0) (4,0) (4,1) (4,2) (4,3) ...]. Because multiple trajectory points are mapped into each cell, this discretized representation of the dynamics is significantly more compact than the original series of floating-point numbers and therefore much easier to work with. Using this representation, the dynamics of a trajectory can be quickly and qualitatively classified using simple geometric heuristics — in this case as a *limit cycle*. PRET's intelligent sensor analysis procedures use this type of discretized geometric reasoning to "distill" out the qualitative features of a given state-space portrait, allowing PRET to reason about these features at a much higher (and cheaper) abstraction level. This scheme is covered in detail in [3].

This is only, however, a very small part of the power of the qualitative phase-portrait representation. Dynamical systems can be extremely complicated; attempting to understand one by analyzing a single behavior instance — e.g., system evolution from *one* initial condition at *one* parameter value, like Fig. 2(a) — is generally inadequate. Rather, one must vary a system's inputs and control parameters and study the change in the response. Even in one-parameter systems, however, this procedure can be difficult; as the parameter is varied, the behavior may vary smoothly in some ranges and then change abruptly ("bifurcate") at critical parameter values. A thorough representation of this behavior, then, requires a "stack" of state-space portraits: at least one for each interesting and distinct range of values. Constructing such a stack requires automatic recognition of the boundaries between these ranges, and the cell dynamics representation makes this very easy. Fig. 2(b), for example, shows another limit cycle trajectory — one with different geometry but identical topology. The key concept here is that a set of geometrically different and yet qualitatively similar trajectories — an "equivalence class" with respect to some important dynamical property — can be classified as a single coherent group of state-space portraits. This is the basis of the power of the techniques described in this paper.

Consider, for example, a driven pendulum model described by the ODE

$$\ddot{\theta}(t) + \frac{\beta}{m}\dot{\theta}(t) + \frac{g}{l}\sin\theta(t) = \frac{\gamma}{ml}\sin\alpha t$$

with mass (m), arm length (l), gravity constant (g), damping factor (β), drive amplitude (γ) and drive frequency (α). m, l, g and β are constants; the state variables of this system are θ and $\omega = \dot{\theta}$. In many experiments, the drive amplitude and/or frequency are controllable: these are the "control parameters" of the system. The behavior of this apparently simple device is really quite complicated and interesting. For low drive frequencies, it has a single stable fixed point; as the frequency is raised, the attractor undergoes a series of bifurcations between chaotic and periodic behavior. These bifurcations do not, however, necessarily cause the attractor to *move*. That is, the qualitative behavior of the system changes and the operating regime (in state space) does not. Traditional analysis of this system would involve constructing state-space portraits, like the ones shown in Fig. 2, at closely spaced control parameter values across some interesting range; this is the *bifurcation analysis* procedure introduced in the previous paragraph. Traditional AI/hybrid representations do not handle this smoothly, as the operating regimes involved are not distinct. If, however, one

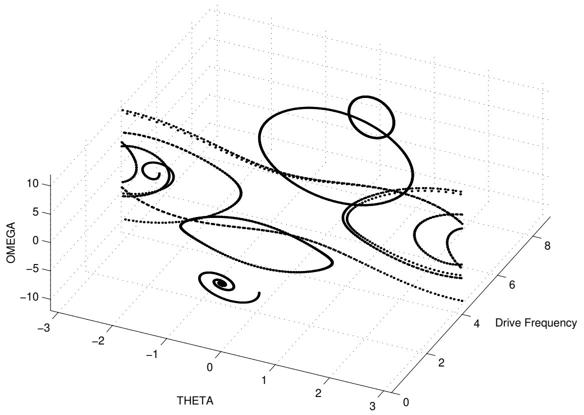

Fig. 3. A state/parameter (S/P) space portrait of the driven pendulum: a parameterized collection of state-space portraits of the device at various Drive Frequencies.

adds a parameter axis to the state space, most of these problems vanish. Fig. 3 describes the behavior of the driven pendulum in this new *state/parameter-space* (S/P-space) representation. Each θ, ω slice of this plot is a state-space portrait, and the control parameter varies along the Drive Frequency axis.

Our final step is to combine this state/parameter-space idea with the qualitative abstraction of cell dynamics, producing the *qualitative state/parameter space* (QS/P-space) representation that is the basis of the KRR framework that is the topic of this paper. A QS/P-space portrait of the driven pendulum is shown in Fig. 4. This representation is similar to the S/P-space portrait shown in Fig. 3, but it groups similar behaviors into equivalence classes, and then uses those groupings to define the boundaries of qualitatively distinct regions.

This qualitative state/parameter-space representation is an extremely powerful modeling tool. One can use it to identify the individual operating regimes, then create a separate model in each, and perhaps use a finite-state machine to model transitions between them. More importantly, however, the QS/P-space representation lets the model builder leverage the knowledge that its regions — e.g., the five slabs in Fig. 4 — all describe the behavior of the *same system*, at different parameter values. This is exactly the type of knowledge that one needs

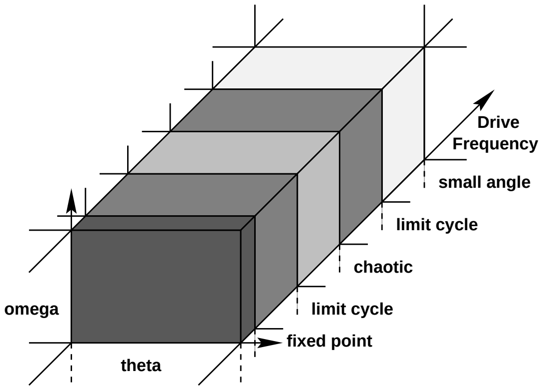

Fig. 4. A *qualitative state/parameter-space* (QS/P-space) portrait of the driven pendulum. This is an abstraction of the state/parameter space portrait shown in Fig. 3; it groups qualitatively similar behaviors into equivalence classes to define the boundaries of qualitatively distinct regions of state/parameter space.

in order to plan how to learn more about a system by changing its inputs and observing the results. The remainder of this paper expands upon these ideas, describing how the QS/P-space representation helps PRET perform input-output modeling of dynamical systems.

4 Input-Output Modeling in PRET

The goal of input-output modeling is to apply a test input to a system, analyze the results, and learn something useful from the cause/effect pair. In this section, we describe how PRET reasons about this process using the QS/P representation introduced in the previous section.

As described in Sect. 2, PRET takes a generate-and-test approach, using a small, powerful domain theory to build ODE models and a larger, more-general ODE theory to test those models against the known behavior of the system. I/O modeling using the QS/P representation contributes to this process in a variety of ways. Firstly, it allows PRET to reason effectively about test inputs; a good test input excites the behavior in a useful but not overwhelming way, and choosing such an input is nontrivial. The representation described in the previous section and pictured in Fig. 4 also allows PRET to reason about sensible hypothesis combinations — a process without which the generate phase would be reduced to blind enumeration of an exponential number of candidate models. Finally, qualitative I/O modeling techniques help PRET reason about state variables and observations — information whose sole source would otherwise be the user.

The "input" part of PRET's input-output reasoning takes place in the *intelligent sensor data analyzer*[3]. This module first reconstructs any hidden dynamics from the sensor data and then analyzes the results using geometric reasoning. The first of these two steps is necessary because fully *observable* systems, in which all of a system's state variables can be measured, are rare in normal engineering practice. Often, some of the state variables are either physically inaccessible or cannot be measured with available sensors. This is control theory's *observer* problem: the task of inferring the internal state of a system solely from observations of its outputs. Delay-coordinate embedding[1], PRET's solution to this problem, creates an m-dimensional *reconstruction-space* vector from m time-delayed samples of data from a single sensor. The central idea is that the reconstruction-space dynamics and the true (unobserved) state-space dynamics are topologically identical. This provides a partial solution to the observer problem, as a state-space portrait reconstructed from a single sensor is qualitatively identical to the true multidimensional dynamics of the system.[3] Given a reconstructed state-space portrait of the system's dynamics, the intelligent sensor data analyzer's second phase distills out its qualitative properties using the cell dynamics paradigm discussed in Sect. 3. The results of reconstructing and analyzing the sensor data are a set of qualitative observations similar to those a human engineer would make about the system, such as "the system is oscillating." This information is useful as it not only raises the abstraction level of PRET's reasoning about models but also is critical to the mechanics of the qualitative bifurcation analysis process, as described later in this section.

Reasoning about actuators is much more difficult, so the development of PRET's *intelligent actuator controller* has been slow. The problem lies in the inherent difference between passive and active modeling. It is easy to recognize damped oscillations in sensor data without knowing anything about the system or the sensor, but using an actuator requires a lot of knowledge about both. Different actuators affect different system properties (e.g., the half dozen knobs on the front of a stereo receiver). They also have very different characteristics (range, resolution, response time, etc.); consider the different dynamics of cooking on campfires, gas/electric stoves, or blast furnaces. Identical actuators can affect systems in radically different ways; a gear shift lever in a car, for instance, invokes very different responses, if it is moved into "first" or "reverse." When the sensor and the system are linear, there are some useful standard procedures for choosing test inputs, codifying the results, and reasoning about their implications — e.g., step and impulse response — but these kinds of drive signals elicit tremendously complicated responses from nonlinear systems, making output analysis very difficult. In nonlinear systems analysis, one typically applies *constant* inputs, ignores any transients, and reasons about the resulting *attractors* in the state-space representation, as described in Sect. 3. Deciding how to use an actuator is only the first part of the problem. Any planning about experiments must also consider the set of possible states of the system — those that are reachable from the existing state with the available control input. Finally, effective input-output modeling requires reasoning about *useful* experiments: those that increase one's knowledge about the target system in a productive way. The ultimate goal of PRET's intelligent sensor/actuator control module is to find and exploit the overlap between these sets of *useful* and *possible* experiments.

[3] This property also allows PRET to estimate an upper bound on the number of state variables in a system.

To solve these difficult problems — controllability, reachability, and utility — PRET must reason about multiple sets of observations of a system, each made under a different actuator condition. It must also *plan* those actuator conditions, which involves modeling not only the actuator itself but also the behavior of the actuator-system interface. Our current solution assumes that PRET knows the actuator input range — a reasonable assumption because the actuator normally exists as an external device, unlike the internal workings of an unknown physical system. Using the QS/P paradigm developed earlier, coupled with the cell dynamics technique and a simple binary search strategy, PRET first performs a qualitative bifurcation analysis. It begins at the lower end of the actuator range, setting the drive signal to a constant value, letting the transient die out, and then using cell dynamics to classify the behavior. It then increments the actuator input and repeats the process. When the attractor bifurcates, PRET zeroes in on the bifurcation point by successively bisecting the actuator input interval. The result of this procedure is a QS/P-space portrait of the system, complete with regime boundaries and behavioral descriptions in each regime, such as:

> "in the temperature range from 0 to $50°C$, the system undergoes a damped oscillation to a fixed point at $(x, y) = (1.4, -8)$; when $T > 50°C$, it follows a period-two limit cycle located at..."[4]

PRET then invokes the model-building process in each regime, and finally attempts to unify these models into a single ODE.

In the driven pendulum example, this procedure works as follows. Qualitative bifurcation analysis identifies five separate qualitative state/parameter-space regions, as shown in Fig. 4. PRET then builds an ODE model for each regime using procedures described in Sect. 2. These ODEs are shown in Table 1. Note that four of these five ODEs are different, but all five are, in reality, instances of a single ODE that accounts for the physical behavior across the whole parameter range. PRET's goal is to find that globally valid model, so it must *unify* these ODEs. Unification is reasonably straightforward if it is correctly interleaved with the model-building process. In the driven pendulum, for example, PRET analyzes the

Drive Frequency	ODE	Description
None	$\ddot{\theta}(t) = -\frac{\beta}{m}\dot{\theta}(t) - \frac{g}{l}\sin\theta(t)$	damped oscillator
Low	$\ddot{\theta}(t) = -\frac{g}{l}\sin\theta(t)$	nonlinear solution
Medium	$\ddot{\theta}(t) + \frac{\beta}{m}\dot{\theta}(t) + \frac{g}{l}\sin\theta(t) = \frac{\gamma}{ml}\sin\alpha t$	"true" (full) solution
High	$\ddot{\theta}(t) = -\frac{g}{l}\sin\theta(t)$	nonlinear solution
Very High	$\ddot{\theta}(t) = -\frac{g}{l}\theta(t)$	linear (small angle) solution

Table 1. Valid models of the driven pendulum in different behavioral regimes.

system in the small-angle regime,[5] producing the model $\ddot{\theta}(t) = -\frac{g}{l}\theta(t)$. When the actuator moves the system to the neighboring limit cycle regime, where

[4] PRET's syntax is much more cryptic; it has no natural language capabilities.
[5] where $\sin\theta \approx \theta$ and the system acts like a simple harmonic oscillator

larger-angle behavior dominates, the small-angle solution no longer holds, forcing a new model search, which yields the model $\ddot{\theta}(t) = -\frac{g}{l}\sin\theta(t)$. PRET then tries to reconcile the two models, applying both of them in both regimes. Since $\ddot{\theta}(t) = -\frac{g}{l}\theta(t)$ is a special case of $\ddot{\theta}(t) = -\frac{g}{l}\sin\theta(t)$, the former holds in only one of the two, whereas the latter holds in both, so PRET discards the $\ddot{\theta}(t) = -\frac{g}{l}\theta(t)$ model and goes on to the next regime, repeating the model building/unification process. Once PRET finds a single model that accounts for all observed behavior in all regimes across the range of interest, its task is complete. Such a model may not, of course, exist; a system may be governed by completely different physics in different regimes, and no single ODE may be able to account for this kind of behavior. In this case, the models in the different regimes would be mutually exclusive, and PRET would be unable to unify them into a single ODE, and so it would simply return the list of regimes, models, and transitions. This is exactly the form of a traditional hybrid model[6] of a multi-regime system.

As is true of automated modeling in general, evaluating the results of this approach can be difficult because the question "How is this model better?" is hard to formalize. From an engineering standpoint, a successful model is one that matches observed behavior to within predefined specifications; PRET is designed to be an engineer's tool, so its judgment of what constitutes success or failure is exactly that. *Parsimony* is another desirable attribute in a model: one wishes to account for the observed behavior using as few — and as simple — ODE terms as possible. Finally, the speed with which PRET produces such a model is another important metric, particularly as we work with more-complex systems and search spaces. Ultimately, the best form of evaluation will consist of whether or not PRET's models are useful for control system design — that is, whether the ODE that PRET constructs of a radio-controlled car can actually be used as the heart of a controller designed to direct that car to perform some prescribed action. We are in the process of evaluating models of real-world systems in several domains — ranging from robotics to hydrology — in this manner.

5 Relationship to Related Work

Most of the work in the AI/QR modeling community builds qualitative models by combining a set of descriptions of state into higher-level abstractions or *qualitative states*[8, 11]. Many tools also reason about equations at varying levels of abstraction, from *qualitative* differential equations (QDEs) in QSIM[16] to ODEs in PRET. PRET's approach differs from many of these tools in that it works with noisy, incomplete sensor data from real-world systems, and attempts not to "discover" the underlying physics, but rather to find the simplest ODE that can account for the given observation. In the QR research that is most closely related to PRET, ODE models are built by evaluating time series using qualitative reasoning techniques and then using a parameter estimator to match the resulting model with a given observed system[7]. This modeling tool selects models from a set of pre-enumerated solutions in a very specific domain (linear visco-elastics). PRET is much more general; it works on linear *and nonlinear* lumped-parameter continuous-time ODEs in a variety of domains and uses *dynamic* model generation to handle arbitrary devices and connection topologies.

PRET shares goals and techniques with several other fields. It solves the same problems as traditional system identification[15], but in an automated fashion, and it relies upon many of the standard methods and ideas found in basic control theory texts such as controllability and reachability[17]. Finally, PRET includes many of the same concepts that appear in the data analysis literature[10], but it adds a layer of AI techniques, such as symbolic data representation and logical inference, on top of these.

6 Conclusion

The goal of the work described in this paper is to automate the type of input-output analysis that expert scientists and engineers apply to modeling problems, and to use that technology to improve the PRET modeling tool, which automatically constructs ODE models of nonlinear dynamical systems. The challenges involved are significant; the nonlinear control-theoretic issues involved in planning and executing experiments routinely stymie human experts. First, PRET must autonomously manipulate a control parameter in order to analyze the system and find behaviorally distinct regimes. Then, it must use knowledge about the behavior and the regime boundaries to reason about what experiments are useful and possible. Finally, PRET must use this information to perform the experiments and analyze the results.

The *qualitative state/parameter-space* representation described in this paper solves some of the problems that arise in phase-portrait analysis of complex systems by combining a state/parameter-space representation with the qualitative abstraction of cell dynamics. This QS/P-space representation, wherein a system's dynamics are classified into discrete regions of qualitatively identical behavior, supports a set of reasoning tactics, collectively termed *qualitative bifurcation analysis*, which allows PRET to reason about multiple sets of observations over a given system.

PRET's sensor-related reasoning is essentially complete, but its reasoning about the relationship between models and excitation sources — as well as final design decisions about how to treat actuator knowledge in an explicit way — are still under development. PRET currently uses very little domain knowledge about its target systems; instead, it relies upon *general* mathematics and physics — principles that are broadly applicable and supported by a well-developed, highly formalized body of mathematical knowledge that applies in *any domain*. The point of this decision was to make PRET easily extensible to other domains; because of this choice, refitting PRET for some new domain is simply a matter of a few lines of Scheme code. However, as we extend PRET into more network-oriented domains, such as electrical circuits, we are discovering that effective use of domain theory may be critical to streamlining PRET's generate phase[9]. A network-oriented modeling approach will also help PRET reason about actuators in a more-intelligent fashion, as the actuator itself, with its various, non-ideal properties, may be represented directly as part of the network. For example, a sinusoidal current source often has an associate impedance that creates a loading effect on the rest of an electrical circuit. With a network approach, these effects naturally become part of the model — just as they do in real systems.

Acknowledgments: Apollo Hogan, Joe Iwanski, Brian LaMacchia, and Reinhard Stolle also contributed code and/or ideas to this project.

References

1. H. Abarbanel. *Analysis of Observed Chaotic Data*. Springer, 1995.
2. E. Bradley. Autonomous exploration and control of chaotic systems. *Cybernetics and Systems*, 26:299–319, 1995.
3. E. Bradley and M. Easley. Reasoning about sensor data for automated system identification. *Intelligent Data Analysis*, 2(2), 1998.
4. E. Bradley, A. O'Gallagher, and J. Rogers. Global solutions for nonlinear systems using qualitative reasoning. *Annals of Mathematics and Artificial Intelligence*, 23:211–228, 1998.
5. E. Bradley and R. Stolle. Automatic construction of accurate models of physical systems. *Annals of Mathematics and Artificial Intelligence*, 17:1–28, 1996.
6. M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control. In *Proceedings of the 33rd IEEE Conference on Decision & Control*, pages 4228–4234, December 1994. Lake Buena Vista, FL.
7. A. C. Capelo, L. Ironi, and S. Tentoni. Automated mathematical modeling from experimental data: An application to material science. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, 28(3):356–370, 1998.
8. J. de Kleer and J. S. Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24:7–83, 1984.
9. M. Easley and E. Bradley. Generalized physical networks for automated model building. In *Proceedings of IJCAI-99*, 1999. To appear.
10. A. Famili, W.-M. Shen, R. Weber, and E. Simoudis. Data preprocessing and intelligent data analysis. *Intelligent Data Analysis*, 1(1), 1997.
11. K. D. Forbus. Interpreting observations of physical systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):350–359, 1987.
12. K. D. Forbus. Qualitative reasoning. In J. A. Tucker, editor, *CRC Computer Science and Engineering Handbook*. CRC Press, Boca Raton, FL, 1997.
13. R. Haber and H. Unbehauen. Structural identification of nonlinear dynamic systems — A survey on input/output approaches. *Automatica*, 26(4):651–677, 1990.
14. C. S. Hsu. *Cell-to-Cell Mapping*. Springer, New York, 1987.
15. J.-N. Juang. *Applied System Identification*. Prentice Hall, Englewood Cliffs, 1994.
16. B. J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–338, 1986.
17. B. C. Kuo. *Automatic Control Systems*. Prentice Hall, seventh edition, 1995.
18. R. Stolle. *Integrated Multimodal Reasoning for Modeling of Physical Systems*. PhD thesis, University of Colorado at Boulder, 1998.
19. R. Stolle and E. Bradley. A customized logic paradigm for reasoning about models. In *Proceedings of the QR-96*, 1996.
20. R. Stolle and E. Bradley. Multimodal reasoning for automatic model construction. In *Proceedings of AAAI-98*, pages 181–188, July 1998. Madison, WI.
21. K. Yip. *KAM: A System for Intelligently Guiding Numerical Experimentation by Computer*. Artificial Intelligence Series. MIT Press, 1991.
22. F. Zhao. Computational dynamics: Modeling and visualizing trajectory flows in phase space. *Annals of Mathematics and Artificial Intelligence*, 8:285–300, 1993.