Machine Learning, 26, 177–211 (1993)

Learning Qualitative Models of Dynamic Systems

DAVID T. HAU dave@welchlink.welch.jhu.edu
School of Medicine, The Johns Hopkins University, Baltimore, MD 21205, USA

ENRICO W. COIERA ewc@hplb.hpl.hp.com
Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol BS12 6QZ, UK

Editors: Stephen Muggleton and David Page

Abstract. The automated construction of dynamic system models is an important application area for ILP. We describe a method that learns qualitative models from time-varying physiological signals. The goal is to understand the complexity of the learning task when faced with numerical data, what signal processing techniques are required, and how this affects learning. The qualitative representation is based on Kuipers' QSIM. The learning algorithm for model construction is based on Coiera's GENMODEL. We show that QSIM models are efficiently PAC learnable from positive examples only, and that GENMODEL is an ILP algorithm for efficiently constructing a QSIM model. We describe both GENMODEL which performs RLGG on qualitative states to learn a QSIM model, and the front-end processing and segmenting stages that transform a signal into a set of qualitative states. Next we describe results of experiments on data from six cardiac bypass patients. Useful models were obtained, representing both normal and abnormal physiological states. Model variation across time and across different levels of temporal abstraction and fault tolerance is explored. The assumption made by many previous workers that the abstraction of examples from data can be separated from the learning task is not supported by this study. Firstly, the effects of noise in the numerical data manifest themselves in the qualitative examples. Secondly, the models learned are directly dependent on the initial qualitative abstraction chosen.

Keywords: inductive logic programming, qualitative modelling, system identification, PAC learning, physiological modelling, cardiovascular system, data mining, patient monitoring

1. Introduction

Various learning algorithms have been developed in recent years to automatically construct qualitative models from system behaviors (Schut & Bredeweg, 1996). There has been a continuing drive to evolve more sophisticated approaches to this learning task, in the main using artificial data sets (Bratko, Muggleton & Varšek, 1991, Varšek, 1991, Richards, Kraan & Kuipers, 1992, Say & Kuro, 1996). As such, these efforts have essentially insulated the learning algorithm from any complexities introduced by the process of abstracting examples from raw data. The goal of this work is to understand the complexity of the learning task when faced with real-time numerical data, to understand what signal processing techniques are required to extract examples, and to understand how this affects the learning problem. To this end, we apply a standard qualitative model learning method to the task of learning models from real-time physiological signals. The qualitative representation of physiological models is adopted from Kuipers' QSIM (Kuipers, 1986). The learning algorithm is based on Coiera's GENMODEL system (Coiera, 1989), to which we have added signal processing capabilities.

The task chosen for the study is to learn models of cardiovascular physiology from patient data, gathered during surgery. Physiological models have a central role in medicine, encapsulating our understanding of experimentally observed physical processes. They act both as theories whose predictions can be used for further research, and as clinical models to assist in the delivery of therapy. However, constructing physiological models by hand is difficult and time-consuming. Further, exact quantitative models are often unavailable since many physiological systems are incompletely understood. In such circumstances, qualitative models can permit useful representations of a system to be developed in the absence of extensive knowledge.

The paper is structured as follows. Qualitative reasoning, and QSIM in particular, are introduced first. The GENMODEL algorithm is described next. We show that QSIM models are efficiently PAC learnable from positive examples only, and that GENMODEL is an ILP algorithm for efficiently constructing a QSIM model consistent with a given set of examples, if one exists. Next, we describe the front-end processing and segmenting stages that transform a signal into a set of qualitative states that are processed by GENMODEL. Finally, we describe results of experiments using the learning system on data segments obtained from six patients during cardiac bypass surgery. Our work shows that useful physiological models are efficiently learnable from physiological signals using the GENMODEL algorithm and appropriate signal processing techniques. Model variation across time and across different levels of temporal abstraction and fault tolerance is then explored. However, the assumption made by many previous workers that the task of abstracting examples from data can be separated from the learning task is not supported by this study. Firstly, the effects of noise in the numerical data manifest themselves in the qualitative examples. Secondly, the models learned are directly dependent on the initial qualitative abstraction chosen.

2. Qualitative Reasoning

A common way of modelling a dynamic system is to use a set of differential equations. The differential equations capture the structure of the system by specifying relationships among the functions of the system. From the equations and an initial state, we can derive a quantitative system behavior using analytical methods or numerical simulation.

A qualitative abstraction of the above procedure allows us to work with an incomplete specification of the system. A qualitative model can be represented by a set of qualitative constraints, or qualitative differential equations (QDEs). From the QDEs and an initial state, we can derive a qualitative system behavior using qualitative simulation. Figure 1 illustrates this (Kuipers, 1986).

Among different qualitative representations developed in past years, Kuipers' QSIM has been widely used, and has had some success in medical reasoning applications (Coiera, 1992, Ironi, 1992, Balestra & Liberati, 1992, Kuipers, 1985).

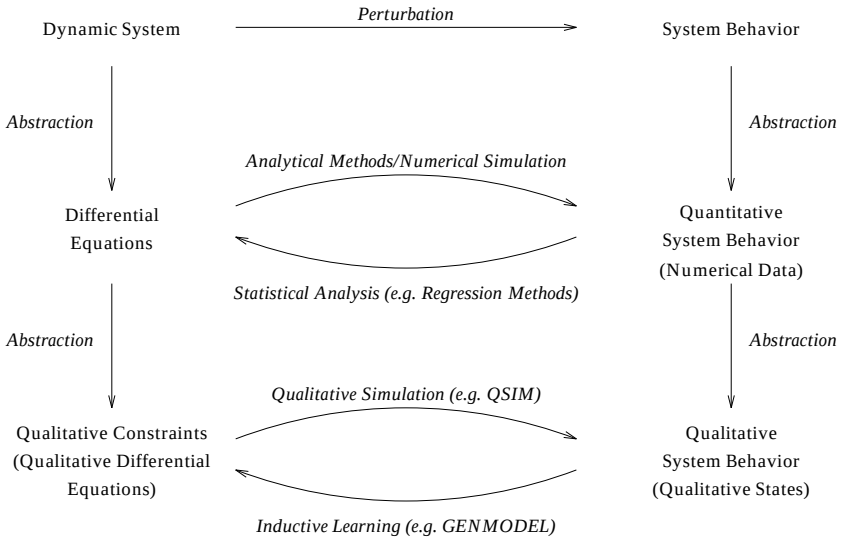

Figure 1. Qualitative reasoning is an abstraction of mathematical reasoning with differential equations and continuously differentiable functions.

2.1. Qualitative Model Constraints

QSIM (Kuipers, 1986) represents a system model through a set of qualitative constraints on the functions of the system. These include four arithmetic constraints:

1. $add(f, g, h) \iff f(t) + g(t) = h(t)$
2. $mult(f, g, h) \iff f(t) \times g(t) = h(t)$
3. $minus(f, g) \iff f(t) = -g(t)$
4. $deriv(f, g) \iff f'(t) = g(t)$

and two monotonic function constraints:

1. $M^+(f, g) \iff f(t) = H(g(t))$ where $H(x)$ is a strictly monotonically increasing function
2. $M^-(f, g) \iff f(t) = H(g(t))$ where $H(x)$ is a strictly monotonically decreasing function

2.2. Qualitative System Behavior

A system's behavior can be described in terms of a number of system functions which vary over time. Every system function $f(t)$ has associated with it a finite and totally ordered set

of *landmark values* which include all the values of interest for the function, and a finite and totally ordered set of *distinguished time points* which include all time points at which any of the system functions reaches a landmark value.

The qualitative state of a system function f at a distinguished time point t is defined as a pair $< value, direction >$. $value$ is the value of f at t, and is either a landmark value or an interval between two landmark values. $direction$ is the direction of change of f at t, and is one of *inc*, *std*, or *dec* depending on whether f is increasing, steady or decreasing at t respectively. A temporal sequence of qualitative states of f forms a qualitative behavior of f.

2.3. Qualitative Simulation: QSIM

QSIM takes a qualitative model and an initial state, and generates all possible system behaviors consistent with the constraints in the model. Starting with the initial state, QSIM repeatedly takes an active state and generates all the possible next-state transitions. These transitions are then filtered according to restrictions posed by the constraints in the system model.

3. Learning Qualitative Models

GENMODEL (Coiera, 1989) goes in the opposite direction to QSIM. It takes a system behavior and generates all the qualitative constraints that permit the system behavior. The GENMODEL algorithm works by first generating the finite set of all plausibly correct qualitative constraints with different permutations of the system functions. Then it progresses along the state history, examining subsequent system states in turn. At each state it successively prunes all constraints that are inconsistent with each state transition. The set of qualitative constraints remaining at the end represent the *most specific* model that permits the given behavior. Any subset of this model also permits the given behavior, and therefore is also a possible model of the system.

3.1. GENMODEL

GENMODEL learns qualitative models by taking a sequence of state descriptions from a dynamic behavior and performing a relative least general generalization (RLGG) (Plotkin, 1971) on them. The algorithm works in a specific to general manner, and so the models that are learned are the most specific ones that explain the observed behaviors, expressed in the more general QDE concept language (Coiera, 1992).

GENMODEL is presented with a sequence of qualitative state descriptions, describing a single time-varying behavior. The program is equipped with background knowledge of legal QDE types, and rules for deciding which value combinations satisfy those qualitative equation types. These rules are taken from (Kuipers, 1986) and represent arithmetic definitions of the qualitative constraints. GENMODEL takes the values of the time-varying

functions that compose a state description, and using the QDE definitions in its background knowledge as templates, generates an exhaustive set of possible QDEs. Each such QDE is a clause instantiated with the parameter values presented in the example state, and together these represent all the combinations of QDE types and function pairings that may potentially exist. For example, given an example state S with system functions a, b and c, at distinguished time point t_0:

$$S : a(t_0) =< 0, dec >, b(t_0) =< -3, dec >, c(t_0) =< 3, std >$$

and the QDE $add(x, y, z)$ which defines addition, GENMODEL would initially generate:

$$\begin{array}{l} add(a/0, b/-3, c/3) \\ add(a/0, c/3, b/-3) \\ add(b/-3, a/0, c/3) \\ add(b/-3, c/3, a/0) \\ add(c/3, a/0, b/-3) \\ add(c/3, b/-3, a/0) \end{array}$$

Each such clause is then tested against the mathematical QDE definitions in the background, and those that describe illegal value combinations are filtered. Successive example states repeat the filtering process by testing the surviving clauses, and only those that match the subsequent examples are retained. The result of filtering would be that GENMODEL retains:

$$\begin{array}{l} add(b/-3, c/3, a/0) \\ add(c/3, b/-3, a/0) \end{array}$$

since these satisfy the background definition of add. These clauses are retained in uninstantiated form as the generalizations:

$$\begin{array}{l} add(b, c, a) \\ add(c, b, a) \end{array}$$

Since these are mathematically identical, one of the two clauses would be deleted by a straightforward redundancy check.

3.1.1. The GENMODEL *Algorithm*

The GENMODEL algorithm is implemented in UNSW Prolog V4.2. The original implementation of GENMODEL described in (Coiera, 1989) is extended here to include dimensional analysis and fault tolerance.

Input:

- A set of system functions, $Functions$.
- A set of units for the system functions, $Units$
- An ordered set of values at distinguished time points for each system function, $Landmarks$.

- A set of qualitative states, $States$.

Output:

A qualitative model which consists of all constraints that are consistent with the state history and dimensionally correct, $Model$.

Algorithm functions used:

$search()$ A function for searching corresponding values from a set of qualitative states. The values of two functions are said to correspond if they occur at the same time point. Corresponding values are used to define legal behaviors of some qualitative constraints (Kuipers, 1986).

$dimcheck()$ A function for checking dimensional compatibility of functions within a proposed constraint.

$check()$ A function for checking validity of a constraint given a qualitative state and sets of corresponding values.

$reduce()$ A function for removing redundancy from constraints. For example, since $M^+(A, B)$ and $M^+(B, A)$ specify the same relationship, one of them can be removed.

Method:

- Search the entire state history $States$ for sets of corresponding values.
- Generate the initial search space by constructing all dimensionally correct constraints and using all permutations of system functions in $Functions$.
- Successively prune inconsistent constraints using each qualitative state in $States$.
- Remove redundancy from the remaining constraints.
- Output the result as a qualitative model.

Algorithm:

begin
$Constraints \leftarrow \emptyset$;
$Correspondings \leftarrow search(States)$;
for each f_1, f_2 in $Functions$ such that $f_1 \neq f_2$ **do**
for each $predicate2$ in $\{inv, deriv, inv\_deriv, M^+, M^-\}$ **do**
if $dimcheck(predicate2, f_1, f_2, Units)$ **then**
add $predicate2(f_1,f_2)$ to $Constraints$;
for each f_1, f_2, f_3 in $Functions$ such that $f_1 \neq f_2 \neq f_3$ **do**
for each $predicate3$ in $\{add, mult\}$ **do**
if $dimcheck(predicate3, f_1, f_2, f_3, Units)$ **then**
add $predicate3(f_1,f_2,f_3)$ to $Constraints$;
for each s in $States$ **do**
for each c in $Constraints$ **do**

if not $check(c, s, Landmarks, Correspondings)$ **then**
delete c from $Constraints$;
$reduce(Constraints)$;
$Model \leftarrow Constraints$;
end

3.1.2. Dimensional Analysis

Before generating a constraint, GENMODEL checks for compatibility of units among functions within the constraint. This approach has been used by several other learning systems, including ABACUS (Falkenhainer & Michalski, 1986), a system for quantitative discovery, and MISQ (Richards, Kraan & Kuipers, 1992), a system based upon GENMODEL.

The dimension of each function is specified at the beginning in terms of the type of quantity the function represents, e.g. $1/time$ for the heart rate (HR), $volume$ for the stroke volume (SV), and $volume/time$ for the cardiac output (CO). This allows the constraint $mult(HR, SV, CO)$ to be generated since $(1/time)\times(volume) = volume/time$, but does not allow $mult(HR, CO, SV)$ or $add(HR, SV, CO)$ to be generated since they are dimensionally incorrect. The functional constraints M^+ and M^- are not restricted by dimensions.

3.1.3. Fault Tolerance

For domains involving noisy learning data, such as hemodynamic monitoring, it is difficult to implement signal processing which filters the noise and restores the signals completely. Therefore we need to incorporate fault tolerance into GENMODEL. We adopt a simple approach by tagging a counter onto every constraint in the initial search space. This counter keeps track of how many example states the constraint has failed to match. We set a noise level η to a fraction of the total number of states in the example behavior. A constraint has to be inconsistent with this many states before it is pruned.

3.2. Probably Approximately Correct Learning

A common setting in machine learning is as follows: given a set of examples, produce a concept consistent with the examples that is likely to correctly classify future instances. The Probably Approximately Correct (PAC) model of learning introduced by Valiant (Valiant, 1984) is an attempt to make precise the notion of "learnable from examples" in such a setting. (Kearns & Vazirani, 1994) and (Rivest, 1987) describe this model in detail.

Stated informally, PAC learnability is the notion that the concept acquired by the learner should closely approximate the concept being taught, in the sense that the acquired concept should perform well on new data drawn according to the same probability distribution as the examples used for learning.

To define PAC learnability formally, we say that a concept class C is efficiently PAC learnable if there exists an algorithm A and a polynomial $s(\cdot\,,\cdot\,,\cdot)$ such that for all n, ϵ, and δ, all probability distributions P_n on X_n, and all concepts $c \in C_n$, A will with probability at least $1-\delta$, when given a set of examples of size $m = s(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ drawn according to P_n, output a $c' \in C_n$ such that $error(c') \leq \epsilon$. Further, A's running time is polynomially bounded in n and m.

3.2.1. Proving PAC Learnability

One approach of PAC learning due to Blumer *et al.* (Blumer, et al., 1987) is as follows: draw a "large enough" set of examples according to P_n, and find an algorithm which, given the examples, outputs *any* concept $c \in C_n$ consistent with all the examples in polynomial time. If there exists such an algorithm for the concept class C, C is said to be *polynomial-time identifiable*. Blumer shows that a sample size m satisfying the following lower bound is sufficient:

$$m \geq \frac{1}{\epsilon}\,(ln\,|C_n| + ln\,\frac{1}{\delta})$$

C_n is said to be *polynomial-sized* if $ln\,|C_n|$ is polynomial in n. Therefore if C_n is polynomial-sized, then m is polynomial in n, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

An algorithm that draws at least this many examples according to P_n and outputs any concept consistent with all the examples in polynomial time is a PAC learning algorithm. Therefore if C_n is polynomial-sized and polynomial-time identifiable, then it is efficiently PAC learnable.

3.2.2. An Occam Algorithm for Learning Conjunctions

In (Valiant, 1984) Valiant provides an algorithm for PAC learning conjunctions of single literals, known as monomials. The algorithm is capable of PAC learning from positive examples only. In Section 3.2.3 we will map the problem of identifying a QSIM model consistent with a given set of examples to the problem of identifying a monomial consistent with a given set of examples.

First we calculate the number of examples needed. The number of conjunctions over the Boolean variables $x_1, \ldots, x_n$ is 3^n since each variable either appears as a positive or negative literal, or is absent entirely. Applying the formula for the lower bound in the previous section, we see that a sample of size $O(\frac{n}{\epsilon} + \frac{\ln 1/\delta}{\epsilon})$ is sufficient for PAC learning.

The algorithm starts with the hypothesis conjunction which contains all the literals:

$$c' = x_1 \wedge \bar{x}_1 \wedge \cdots \wedge x_n \wedge \bar{x}_n$$

Upon each positive example x, the algorithm updates c' by deleting the literal x_i if $x_i = 0$ in the example, and deleting the literal $\bar{x}_i$ if $x_i = 1$ in the example. Thus the algorithm deletes any literal that contradicts the data.

Since the algorithm takes linear time to process each example, given m examples with m as calculated above, the running time is bounded by mn and hence is bounded by a polynomial in n, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. Therefore this is an efficient PAC learning algorithm for the class of monomials.

3.2.3. QSIM *Models are PAC Learnable*

In Section 3.2.1 we concluded that to prove that the concept class of QSIM models is PAC learnable, it suffices to prove that the class is polynomial-sized and that it is polynomial-time identifiable. The following two sections provide these proofs.

The Class of QSIM Models is Polynomial-Sized

To show that the concept class of QSIM models is polynomial-sized, we begin by noting that in the QSIM formalism there are five kinds of two-function constraints (inv, $deriv$, inv_deriv, M^+ and M^-), and two kinds of three-function constraints (add and $mult$). Therefore with n system functions, the number of non-redundant and mathematically plausible QSIM constraints N is at most:

$$N \leq 5n(n-1) + 2n(n-1)(n-2)$$

Therefore the number of possible QSIM models is at most:

$$|QSIM\text{–}Models(n)| = 2^N \leq 2^{5n(n-1)+2n(n-1)(n-2)}$$

since each QSIM constraint can either be present or absent in the model. This implies that:

$$\lg\left(|QSIM\text{–}Models(n)|\right) = N = O(n^3)$$

Therefore the concept class of QSIM models is polynomial-sized.

To PAC learn a QSIM model, we need m examples where m has the following lower bound:

$$m \geq \frac{1}{\epsilon}\left(\ln 2^N + \ln\frac{1}{\delta}\right)$$

QSIM Models are Polynomial-Time Identifiable

In this section we show that GENMODEL is an algorithm for efficiently constructing a QSIM model consistent with a given set of examples. We prove this by mapping the problem of identifying a QSIM model consistent with a given set of examples to the problem of identifying a monomial consistent with a given set of examples.

We view each QSIM model as a conjunction of QSIM constraints, and each QSIM constraint as a Boolean variable. Then learning QSIM models is equivalent to learning monotone conjunctions[1] with N Boolean variables, where N is the number of possible QSIM constraints as calculated in the previous section.

The algorithm starts with the hypothesis of a monotone conjunction which contains all N of the Boolean variables, i.e. all possible QSIM constraints:

$$c' = x_1 \wedge \cdots \wedge x_N$$

For each positive example x representing a qualitative state given to GENMODEL, the algorithm updates c' by deleting the Boolean variable x_i if the corresponding QSIM constraint is inconsistent with the example. Since each Boolean variable x_i corresponds to a QSIM constraint, the algorithm prunes any constraint that is inconsistent with each qualitative state. This is identical to the approach taken by GENMODEL.

Now it remains to show that GENMODEL takes polynomial time to perform each step in learning a QSIM model. We review these steps:

- Search the entire state history for sets of corresponding values. For m qualitative states, there are at most m sets of corresponding values, and the search takes $O(m)$ time. For n system functions, this is equivalent to $O(n^3)$ time.
- Generate the initial search space by constructing all plausibly correct constraints with different permutations of system functions. Again, for n system functions, this takes $O(n^3)$ time.
- Successively prune inconsistent constraints upon each qualitative state. Checking for consistency of a constraint with a qualitative state involves:
  - Checking landmark values and directions of change. This takes constant time.
  - Checking corresponding values. Since there are at most m sets of corresponding values, this takes $O(m)$ time, or equivalently, $O(n^3)$ time.

  Therefore, for each qualitative state, checking for consistency with all $O(n^3)$ constraints requires a total of $O(n^6)$ time. For m qualitative states, the total processing time is $O(n^9)$ for this step.
- Remove redundancy from the remaining constraints. Since we started off with m constraints, there are at most the same number of constraints remaining in the final model. Removing redundancy involves comparing constraints with one another, and requires $O(m^2)$, or equivalently, $O(n^6)$ time.
- Output the result as a qualitative model.

Therefore the total time taken by GENMODEL in learning a QSIM model is $O(n^9)$. Since this is polynomial in n, QSIM models are polynomial-time identifiable by GENMODEL.

3.2.4. Applicability of PAC Learning

How large a sample size is needed to learn a qualitative model? For our experiments, we used 8 different physiological signals. Out of the 952 constraints representing all the possible 2 and 3 signal combinations, only 99 of these represent non-redundant and mathematically plausible constraints. Therefore the sample size m has the following lower bound:

$$m \geq \frac{1}{\epsilon}(\ln 2^{99} + \ln \frac{1}{\delta})$$

For each of the training sets in our experiments (see Section 6) we used data segments containing 1000 examples. This sample size corresponds to an accuracy and a confidence level of 93% ($\epsilon = \delta = 0.07$).

There are however a number of characteristics of the qualitative model learning task that deviate from the PAC formulation.

- The quantitative to qualitative abstraction of data means that there is a large redundancy in examples. While a system's behavior may be sampled many times over a period, if the system behavior does not qualitatively change, then all the example data points during the sample period represent an identical qualitative state. Effectively this means that all these data points have been compressed into a single example. Further, the number of qualitative states drawn from a data sample increases as the temporal abstraction becomes finer grained. The abstraction mechanisms used in this work are described in Section 4.2.

- Qualitative states cannot easily be modelled as *independent* examples drawn from an underlying probability distribution. In a qualitative system behavior, the next state will always be correlated to its predecessor for two reasons. Firstly, there are mathematical constraints on the possible transitions a function can make between states (Kuipers, 1986). Further, successive states are correlated because of the presence of functional constraints limiting the values that functions can take.

  While example states from within a behavior cannot be modelled as independent examples, the selection of *example behaviors* could be. Thus it may ultimately make more sense to formulate the PAC sample size bounds not on the number of example states, but on the number of example histories.

- Signals may be corrupted by artifact and noise. The PAC learning algorithm previously developed assumes learning examples to be noise-free. In noisy domains, one would require more examples than in a noise-free one.

3.3. Comparison of GENMODEL *with Approaches to Learning Qualitative Models*

3.3.1. GENMODEL *does not require negative examples.*

GENMODEL learns from positive examples only. While modern ILP systems like PROGOL (Muggleton, 1995) are similarly able to handle only positive examples, older ILP systems

like GOLEM (Muggleton, 1992) and FOIL (Quinlan, 1990) have an additional requirement for negative examples. Most reported attempts at learning qualitative models have however, been made with these older systems.

Bratko *et al.* (Bratko, Muggleton & Varšek, 1991) report that learning the U-tube model with GOLEM requires six hand-generated negative example states, in addition to the three positive example states needed by GENMODEL. For each iteration in GOLEM, a fixed number of clauses are first generated by relative least general generalization (RLGG) (Plotkin, 1971). The clause that covers the most positive examples and none of the negative examples is chosen for propagation to the next iteration. In comparison, with dimensional analysis GENMODEL comes up with exactly the six constraints for the U-tube system. (The U-tube system is a standard reference problem in qualitative modelling (Bratko, Muggleton & Varšek, 1991, Coiera, 1989).)

In (Varšek, 1991), Varšek's genetic algorithm approach requires 17 positive example states and 78 negative example states to learn the U-tube model. In each cycle, candidate solutions are selected for "reproduction" based on a fitness function which is the sum of the fraction of positive and negative examples covered correctly and a "bonus" indicating the size of the solution.

In both approaches, it is essential for the user to give the "right" negative examples. Badly chosen negative examples or an inadequate number of them will cause an inappropriate clause to be propagated to the next iteration, which will ultimately affect the concept output in the end. However, there are no existing rules to guide the selection of negative examples. It could be argued that the dimensional analysis used in the present experiments is a form of directed negative example generation, utilizing background knowledge about the structure of the domain.

3.3.2. GENMODEL *does not require ground facts for background knowledge*

GOLEM accepts definitions of background predicates in terms of ground facts. In learning QSIM models, explicit ground facts describing QSIM constraint definitions must be generated according to functions and landmark lists relevant to the modelling problem at hand. In (Bratko, Muggleton & Varšek, 1991), Bratko *et al.* report that learning the U-tube model requires a total of 5408 ground facts as background knowledge. This is already a simplification which excludes rules regarding corresponding values in the M^+ and M^- constraints, rules regarding consistency of infinite values in the add constraint, and rules on the $mult$ constraint. In a more complex domain such as human physiology which potentially involves long landmark lists, the size of the background knowledge required can grow exceedingly large.

GENMODEL stores the QSIM constraint definitions as non-ground clauses in its background. These clauses are instantiated with values from examples, and when these clauses satisfy the constraint definitions, they are stored as active hypotheses. In this respect, GENMODEL's strategy is more closely related to recent ILP systems like PROGOL (Muggleton, 1995) than it is to GOLEM or FOIL.

3.3.3. GENMODEL *is guaranteed to produce a correct model if one exists.*

Since GENMODEL exhaustively generates all constraints consistent with an initial example and its background knowledge, and then prunes these constraints with successive examples, it is guaranteed to produce a correct model if one exists within its description language.

On the other hand, GOLEM and genetic algorithms perform heuristic searches across the concept space. GOLEM performs hill climbing with positive and negative example coverage as the heuristic guiding the search. Genetic algorithms similarly perform hill climbing with the fitness function serving as the heuristic. Since neither heuristic is a perfect quality measurement of the current model, GOLEM and genetic algorithms are not guaranteed to produce a correct model even if one exists, unless the search becomes exhaustive. Further, GOLEM is unable to learn non-deterministic concepts like add (Bratko, Muggleton & Varšek, 1991).

3.3.4. GENMODEL *does not attempt to discover new variables*

Unmeasured variables may participate in significant relationships with those variables that are included in the example data. GENMODEL as described here, does not look for such variables, but confines its search to the space of possible relationships between measured variables.

The search for new variables can be guided by postulating extensions to relationships already in the data. For example, if the data supports the relationship $M^+(A, B)$, then one could postulate a new variable C such that $add(A, C, B)$. Such approaches have featured in a number of algorithms focussed on the qualitative learning problem, including MISQ (Richards, Kraan & Kuipers, 1992) and QSI (Say & Kuro, 1996).

Since it cannot be guaranteed that all important variables have been included in a data set, such facilities are ultimately necessary in attacking the type of real-time data sets explored in this paper. However, the focus of the present work is to establish how reliably relationships amongst *known* variables can be learned. Given the difficulties associated with handling noise, and temporal abstraction that will be discussed in Section 6.3, current approaches to variable creation may need to be modified in this light.

3.3.5. Data Handling

None of the current approaches to learning qualitative models have addressed the problem of learning from raw numerical data in any significant way. The ABACUS system used simple error margins to accommodate noise in a set of static data about chemical compounds (Falkenhainer & Michalski, 1986). All the "special purpose" systems for learning qualitative models have assumed that the learning algorithm is presented with a set of example qualitative behaviors that are correct (Say & Kuro, 1996). This effectively insulates the learning system from any problems that may exist in the raw data. It also assumes that the learning stage does not need to take into account any effects from the process of abstracting qualitative examples from numerical data. There is no evidence that such assumptions are

valid, and one of the prime motivations for the present work is to explore how the qualitative abstraction stage, which generates examples, influences the learning stage.

4. System Architecture

4.1. Overview

The goal of the learning system is to generate qualitative models from physiological signals. The overall architecture of the system is illustrated in Figure 2.

Figure 2. Overall architecture of the learning system.

The physiological signal is first processed by a front-end system, which outputs a filtered signal and its derivative. These are entered into the segmenter to produce a set of qualitative states.[2] GENMODEL then uses these states to generate a qualitative model. The architecture used for front-end processing of physiological signals is shown in Figure 3.

In overview, the signal first passes through an artifact filter which removes various artifacts and linearly interpolates the intervals of the artifacts removed. The resulting signal is then processed by a median filter which removes impulsive features. A Gaussian filter then

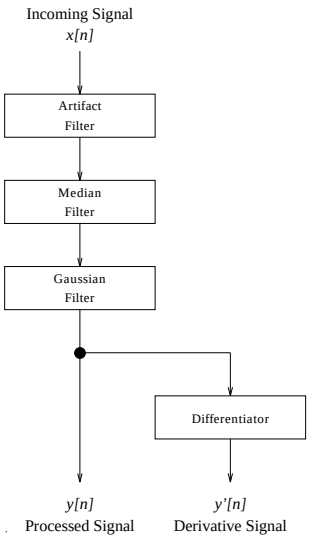

Figure 3. Architecture used for front-end processing of physiological signals.

smooths the signal to the desired level of temporal abstraction by convolving it with a Gaussian kernel of an appropriate standard deviation σ. Finally, this smoothed signal is passed through a differentiator to obtain its derivative. The smoothed signal and its derivative are passed on to the segmenter for segmentation, producing a set of qualitative states describing the system behavior represented by the signal.

4.2. Temporal Abstraction

A complex system such as the cardiovascular system involves processes operating at different time scales. From the same set of signals, depending on the particular time scale we are interested in, different sets of qualitative states and therefore different models can be obtained.

In (Kuipers, 1987) Kuipers describes a temporal abstraction relation among mechanisms operating at significantly different time scales. Processes that occur significantly faster than the time scale of a model can be considered as instantaneous with respect to the model, while those that occur much slower can be considered as constant. For example, if we look at a system on the order of hours, processes that occur within seconds can be considered as instantaneous, while those occurring over days can be viewed as constant. Therefore if we perturb a system by increasing a function A, and observe that another function B responds to this change within seconds by increasing its value, then we can still model the relationship between A and B with the functional constraint $M^+(A, B)$ even though there is a delay between the perturbation and the response, since the response within seconds is seen as occurring instantaneously at this time scale.

The temporal abstraction at which data is processed depends on two processes:

1. First, a Gaussian filter is used to remove changes lasting significantly shorter than the time scale of interest. This avoids *aliasing* (Hau, 1994).

2. Next, the segmenter determines that critical points of different functions that occur within τ sampling periods are labelled as occurring at the same distinguished time point, where τ corresponds to the time scale of interest.

4.2.1. Gaussian Filter

The idea of using a Gaussian filter to analyze changes in a signal at different scales is borrowed from the technique of *scale-space filtering* in edge detection. Scale-space filtering constructs hierarchic symbolic signal descriptions by transforming the signal into a continuum of versions of the original signal convolved with a kernel containing a scale parameter. In an image, changes of intensity take place at many spatial scales depending on their physical origin. Marr and Hildreth (Marr & Hildreth, 1980) observed that detecting zero crossings in the Laplacian of the intensity values across different scales enables a system to distinguish a physical edge from surface markings or shadows. They suggested that the original image be band-limited at several different cut-off frequencies and that an

edge detection algorithm be applied to each of the images. The resulting edge maps have edges corresponding to different scales.

In our learning system, we need to segment a set of signals at different time scales. We can do so by band-limiting our original signals at several different cut-off frequencies and segmenting the signals by detecting zero crossings in the first derivative of the signals at different scales. The segmentation then produces a set of qualitative behaviors at different time scales which can be given to GENMODEL to produce qualitative models at different scales.

To band-limit an image at different cut-off frequencies, the impulse response of the lowpass filter proposed by Marr and Hildreth is Gaussian shaped. This choice is motivated by the fact that the Gaussian function is smooth and localized in both the spatial and frequency domains.[3] A smooth impulse response is less likely to introduce any changes that are not present in the original shape. A localized impulse response is less likely to shift the location of edges. Further, Yuille and Poggio (Yuille & Poggio, 1986) and Babaud *et al.* (Babaud, et al., 1986) have separately shown that the Gaussian filter has a unique property concerning zero crossings of the first derivative of the filtered signal[4]: moving from coarse to fine scale, new zero crossings appear, but existing ones never disappear. Consequently, the extrema can be used to construct a tree describing the successive partitioning of the signal into finer subintervals as new zero crossings appear at finer scales. This partitioning of the signal by extrema moving from coarse to fine scale forms a strict hierarchy. Scale-space filtering in edge detection can be seen as a form of the more general technique of wavelet transforms in multi-resolution signal analysis, with the wavelets here being Laplacians of shifted Gaussians, and signal edges located by zero crossings of the wavelet transform (Strang, 1989).

We adopt a similar approach for segmenting our signals. The impulse response of the lowpass filter used is based on the Gaussian function, with t replaced by n to yield a discrete-time function $g[n]$:

$$g[n] = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{n^2}{2\sigma^2}}$$

for $-\infty < n < \infty$ and $\sigma > 0$. The standard deviation σ determines the cut-off frequency with a larger σ corresponding to a lower cut-off frequency. σ therefore determines the time scale we are operating at, with a smaller σ corresponding to a finer time scale and a larger σ corresponding to a coarser scale. The frequency response of the lowpass filter is the Fourier transform of $g[n]$ which is also Gaussian shaped. The infinite impulse response $g[n]$ is multiplied by the Hanning window to obtain a finite impulse response (FIR) $h_g[n]$ (Oppenheim & Schafer, 1989).

The length of the finite impulse response L is set to three standard deviations from the origin. Therefore L is proportional to the level of temporal abstraction. In our experiments, we use values of σ at 10, 20, 40, 60, 80 and 100, corresponding to values of L at 61, 121, 241, 361, 481 and 601 respectively.

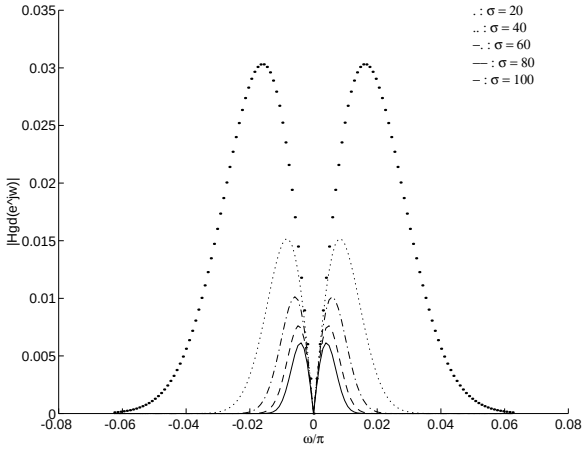

Figure 4. Equivalent frequency responses of a Gaussian filter in cascade with a bandlimited differentiator for $\sigma = 20, 40, 60, 80, 100$.

4.2.2. Differentiator

The differentiator is implemented as an FIR filter based on the frequency response of a band-limited differentiator (Oppenheim & Schafer, 1989). It is interesting to note that the lowpass filtering operation of the Gaussian filter and the derivative operation of the differentiator may be combined to obtain a single filter with the derivative of the Gaussian function as its impulse response $h_{gd}(t)$:

$$h_{gd}(t) = g'(t) = -\frac{t}{\sqrt{2\pi}\sigma^3} e^{-\frac{t^2}{2\sigma^2}}$$

The corresponding frequency response is as follows:

$$H_{gd}(\Omega) = j\Omega e^{-\frac{\Omega^2\sigma^2}{2}}$$

This frequency response is plotted in Figure 4.

From the frequency response, we note that the combined operation is equivalent to bandpass filtering where σ controls the bandwidth of the bandpass filter. Band-limiting the signals tends to reduce noise, thus reducing the noise sensitivity problem associated with detecting zero crossing points. With increasing values of σ, the bandwidth of the bandpass filter decreases and therefore more noise rejection is achieved. This agrees with our expectation since larger values of σ correspond to coarser time scales.

4.2.3. Segmenter

The segmenter consists of two parts: a *function segmenter* for each function of the system, and a *qualitative behavior generator* to coordinate the whole segmentation process.

The function segmenter segments each signal either at zero crossings of its derivative obtained from the differentiator, or when the segmenter of another function has detected a zero crossing. It then looks up its local landmark list to see if there is any existing landmark within a tolerance from the current value of the function. If so, the existing landmark becomes the qualitative value of the function in this state. If not, the segmenter creates a new landmark corresponding to the current value of the function, returns this landmark as the qualitative value of the function in this state, and stores the new landmark in the local landmark list. The direction of change of the function in the current state is obtained by observing the sign of the derivative. A positive derivative corresponds to inc (increasing). A negative derivative corresponds to dec (decreasing). A derivative within a tolerance from zero corresponds to std (steady). The qualitative value and the direction of change together form a qualitative state of the function.

The qualitative behavior generator keeps track of distinguished time points and coordinates the entire segmentation process. Each function's segmenter stores its values over the previous τ time points. When any one or more of the function segmenters detects a zero crossing in their derivatives, the generator waits for τ sampling periods to see if any other segmenters also detect a zero crossing in their derivatives. The parameter τ therefore determines the level of temporal abstraction, as discussed in Section 4.2. The generator labels all times within these τ sampling periods as the same distinguished time point. It then signals *all* segmenters to segment their signals at the time point that is the average for all functions that have had a zero crossing detected. The generator then collects a qualitative state of each function from its segmenter, and combines the qualitative states of all the functions of the system into a qualitative state of the system at the current distinguished time point. A series of such qualitative states form a qualitative behavior of the system which is input into GENMODEL.

5. Physiological Signals and Models

The cardiovascular system is a relatively well understood physiological system, is easily measured through a number of signals, and is clinically important. It thus provides us with an excellent real world domain to test the applicability of the GENMODEL approach.

5.1. Physiological Signals from Hemodynamic Monitoring

Hemodynamic monitoring provides information on many aspects of the cardiovascular system (CVS), including heart rate, arterial blood pressure, central venous pressure, skin temperature, core temperature and others. The data used in our study consists of eight such signals derived at 1 Hz. These are:

- **Heart rate (**HR**)**
- **Mean arterial blood pressure (**$ABPM$**)**
- **Mean central venous pressure (**$CVPM$**)**

- **Stroke volume (SV)** - derived from the arterial blood pressure waveform using the pulse contour method (Wesseling, et al., 1983).
- **Cardiac output (CO)** - derived from the heart rate and the stroke volume by the equation $CO = HR \times SV$.
- **Ventricular contractility (VC)** - derived from the slope of the arterial blood pressure waveform at the onset of systole (Saidman & Smith, 1984).
- **Skin-to-core temperature gradient (ΔT)** - the difference between the skin and core temperatures.
- **Rate pressure product (RPP)** - the product of the heart rate and the systolic arterial blood pressure; a good indicator of myocardial oxygen consumption (Gobel, et al., 1978).

For a detailed description of the derivation of these signals refer to (Hau, 1994), (Guyton, 1981), (Oh, 1990), (Saidman & Smith, 1984) and (Wyngaarden, Smith & Bennett, 1992).

5.2. A Qualitative Cardiovascular Model

In this section, we describe a set of possible qualitative constraints that may exist among the eight measurements in our data set. These constraints form a "gold-standard" target model of the CVS which allows us to compare our experimental results and evaluate the performance of the learning system.

Because of the enormous complexity of the CVS, formulating a model is by no means a simple task (Toal & Hunter, 1990, Weinberg, Biswas & Uckun, 1990). The constraints included in this section are not meant to be a comprehensive coverage of the system. Also, different models may exist for different clinical conditions, for example disease states. Thus a constraint may be valid only under certain circumstances.

5.2.1. Relationship Among Heart Rate, Stroke Volume and Cardiac Output

The heart rate (HR), stroke volume (SV) and cardiac output (CO) are related by the equation: $CO = HR \times SV$. This translates into the qualitative constraint: $mult(HR, SV, CO)$. Since CO is automatically derived in our data set, this relationship holds across all the training data, and should always be discovered by a learning algorithm.

5.2.2. Relationship Among Heart Rate, Arterial Blood Pressure and Rate Pressure Product

The heart rate (HR), systolic arterial blood pressure (ABPS) and rate pressure product (RPP) are related by the equation: $RPP = HR \times ABPS$. Since the behavior of the mean arterial blood pressure (ABPM) approximates that of the systolic arterial blood pressure (ABPS) well in most circumstances, the relationship translates into the qualitative constraint: $mult(HR, ABPM, RPP)$.

5.2.3. The Frank-Starling Law of the Heart

The Frank-Starling law states that within physiological limits, the heart pumps all the blood that comes to it without allowing excessive damming of blood in the veins. This translates into the qualitative constraint: $M^{+}(CVPM, CO)$.

5.2.4. Autoregulation of the Heart

When extra amounts of blood enter the heart chambers, the stretched muscle contracts with a greatly increased force, thereby automatically pumping the extra blood into the arteries. Therefore, within the physiological limit of the heart, the ventricular contractility (VC) of the heart increases with the stroke volume (SV): $M^{+}(SV, VC)$.

5.2.5. Effect of Heart Rate on Cardiac Output

An increase in heart rate can be caused by a higher oxygen demand in tissues and organs, as in physical exercise. It can also occur as a compensatory mechanism for a decreased arterial blood pressure in conditions like hypovolemia, when there is a decrease in circulating blood volume. This results in different sets of constraints which model different conditions:

- Under normal conditions, the more times the heart beats per minute, the more blood it can pump, since the stroke volume stays roughly the same. This can be seen from the equation: $CO = HR \times SV$. This can be represented by the qualitative constraint: $M^{+}(HR, CO)$. Further, a rise in heart rate increases the net influx of calcium ions per minute into the myocardial cells, and enhances ventricular contractility: $M^{+}(HR, VC)$.

- However, once the heart rate exceeds a critical level (150-170 beats per minute in normal individuals) the heart strength itself decreases, presumably because of over-utilization of metabolic substrates in the cardiac muscle: $M^{-}(HR, VC)$. This results in a significant decrease in diastolic filling time and consequently a decrease in the stroke volume: $M^{-}(HR, SV)$.

- Hypovolemia refers to an absolute and often sudden reduction in circulating blood volume relative to the capacity of the vascular system (Schlant & Alexander, 1994). When present, the body mounts a series of compensatory mechanisms including:

  - Arteriolar vasoconstriction with resultant decreased perfusion to skin and skeletal muscle. This causes an increase in the skin-to-core temperature gradient ΔT: $M^{-}(CO, \Delta T)$.
  - Increased heart rate (tachycardia): $M^{-}(CO, HR)$.
  - Increased myocardial contractility: $M^{-}(CO, VC)$.

6. Results and Interpretation

The learning system was applied to data segments obtained from six patients during cardiac bypass surgery.[5] One data segment from each of the first five patients was used to study how qualitative models learned vary across patients. Six data segments obtained from one patient (Patient 6) were used to study how qualitative models learned vary within a patient over time.

Each data segment was 1000 seconds (16.7 minutes) long, sampled at 1 Hz. The fault tolerance level in GENMODEL was set at 20% of the total number of qualitative states in each data segment. The operation performed in each case was to insert coronary artery bypass grafts, except in the case of Patient 2 which was to replace the aortic valve. Models were learned from the data segments at six different levels of temporal abstraction, represented by the six different values of L mentioned in Section 4.2.1. The results for the data segment from Patient 5 and two of the six data segments from Patient 6 are described below, at three of the six levels of temporal abstraction used ($L = 61, 241, 601$). (Hau, 1994) reports the results in full.

For each data segment, a brief overview of the patient's condition is given, followed by a plot of the original signals. Then the filtered signals at the three levels of temporal abstraction are shown followed by the model learned and an interpretation of each of the model constraints.[6]

In the following results, spurious constraints are not considered to be generally valid but are supported by the example data, i.e. they are over-specific and likely to be lost as more examples come in (see Section 6.3.4).

6.1. Patient 5

The patient was a 66-year-old gentleman with a fairly long history of angina and a proven inferior myocardial infarct 3 months before the operation. His angiogram showed severe triple vessel disease with reasonably good left ventricular function. He was hypertensive and was treated with beta-blockers (Atenolol).

The data segment was taken some time after the surgery had started. Before the period, lightness of anesthesia caused a sharp rise in ABP from 90 mmHg systolic up to 160 mmHg systolic that was sustained for several minutes. During the period the dosage of anesthetic (Enflurane), analgesic (Alfentanil) and GTN (glyceryl trinitrate or nitroglycerin, a vasodilator) were increased to bring the ABP back down.

L=61

$M^-(CVPM, \Delta T)$ (Spurious)
$M^+(SV, CO)$ Correct given that HR was constant due to beta-blockers.
$M^+(CVPM, \Delta T)$ (Spurious)

L=241

$M^+(SV, CO)$ Correct given that HR was constant due to beta-blockers.
$mult(HR, SV, CO)$ (Correct)

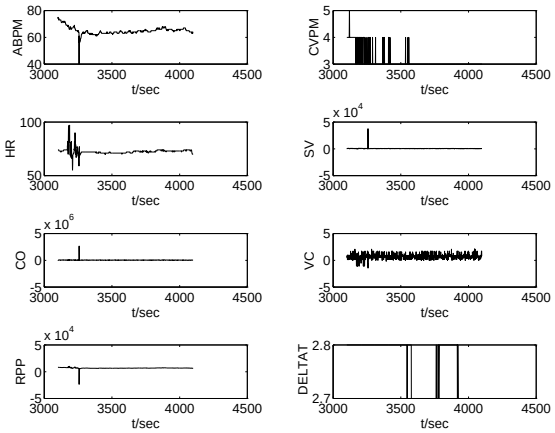

Figure 5. Patient 5: Original Signals. Note the relatively constant heart rate due to the effect of beta-blockers (see Section 6.3.4).

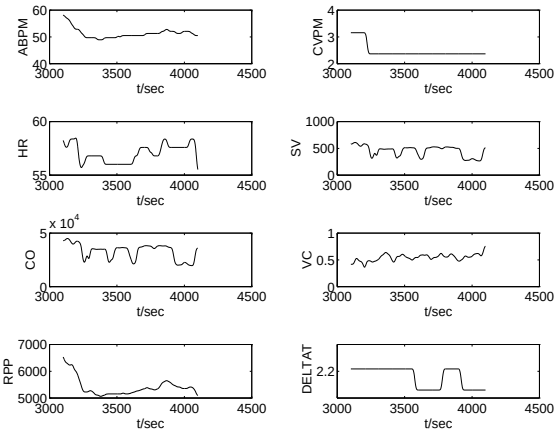

Figure 6. Patient 5: Filtered Signals ($L = 61$). Note that the trends of the relatively constant heart rate are amplified (see Section 6.3.4).

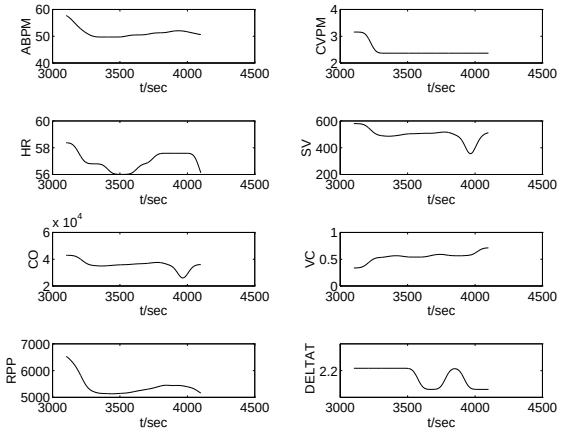

Figure 7. Patient 5: Filtered Signals ($L = 241$)

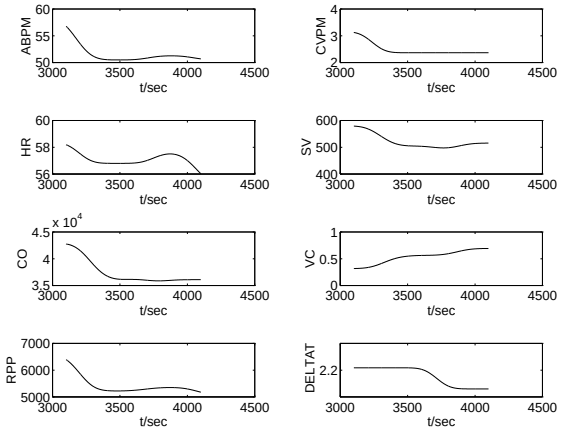

Figure 8. Patient 5: Filtered Signals ($L = 601$)

L=601

$M^+(ABPM, HR)$ (Spurious)

$M^+(ABPM, RPP)$ Correct given that HR was constant due to beta-blockers. ABPM dropped because of increased depth of anesthesia.

$M^+(CVPM, CO)$ Frank-Starling Law of the Heart.

$M^+(HR, RPP)$ (Spurious)

$mult(HR, ABPM, RPP)$ (Correct)

$mult(HR, CVPM, RPP)$ (Spurious)

$mult(HR, SV, CO)$ (Correct)

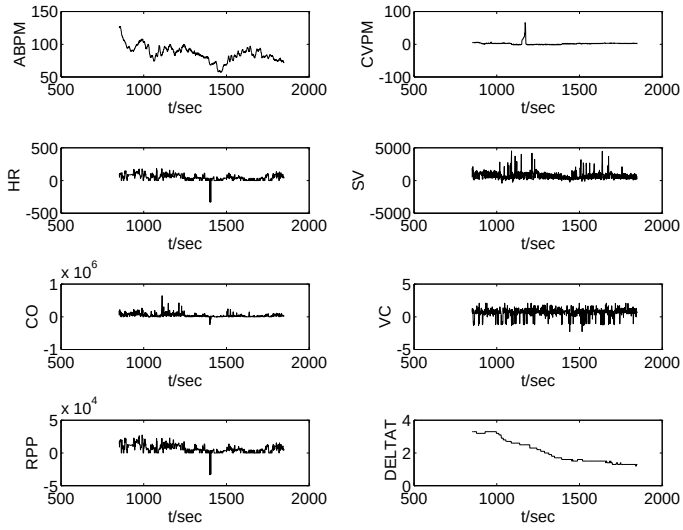

Figure 9. Patient 6, Segment 1: Original Signals

6.2. Patient 6

The patient was a 63-year-old gentleman having 1 internal mammary artery and 3 coronary artery grafts. He had a history of hypertension and angina. His angiogram showed severe disease at the origin of all left sided vessels. He was not on beta-blockers.

6.2.1. Segment 1

Prior to this segment, lightness in anesthesia caused rises in ABP (up to 180 mmHg systolic) in response to surgery. The patient then developed myocardial ischemia. In response to this, the GTN dosage was increased, which along with hypovolemia caused the ABP to drop, with the result that ischemia improved at the expense of blood pressure. The depth of anesthesia was also increased.

L=61

$mult(HR, ABPM, RPP)$ (Correct)

L=241

$M^{-}(HR, SV)$ (Spurious)
$inv\_deriv(ABPM, RPP)$ (Spurious)
$inv\_deriv(ABPM, VC)$ (Spurious)

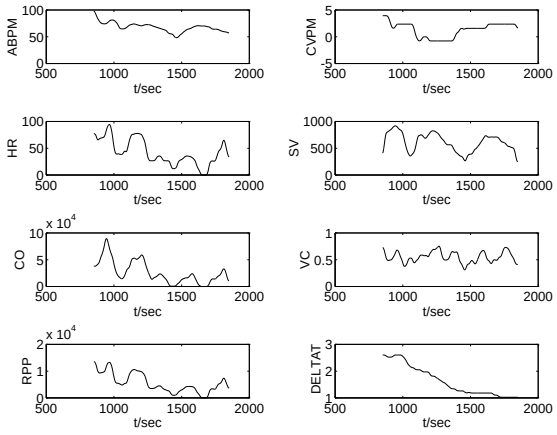

Figure 10. Patient 6, Segment 1: Filtered Signals ($L = 61$)

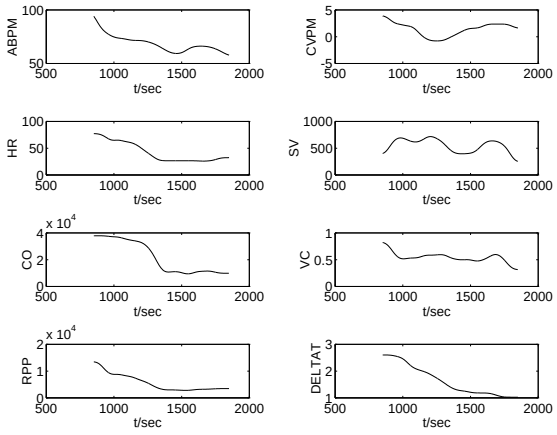

Figure 11. Patient 6, Segment 1: Filtered Signals ($L = 241$)

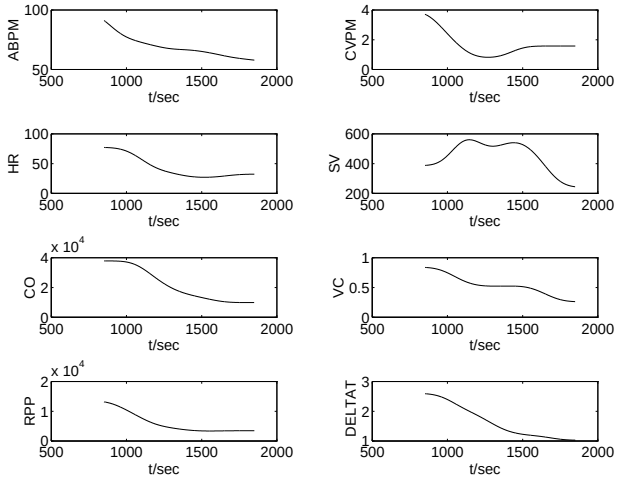

Figure 12. Patient 6, Segment 1: Filtered Signals ($L = 601$)

L=601

$M^+(ABPM, CO)$ Both dropped because of vasodilation and increased venous tone caused by increased GTN dosage.

$M^+(HR, RPP)$ Both dropped because of increased depth of anesthesia.

$M^+(ABPM, \Delta T)$ Both dropped because of vasodilating effect of GTN.

$M^+(CO, \Delta T)$ Both dropped because of vasodilation and increased venous tone caused by increased GTN dosage.

$inv\_deriv(ABPM, RPP)$ (Spurious)

$inv\_deriv(ABPM, VC)$ (Spurious)

$mult(HR, ABPM, RPP)$ (Correct)

$mult(HR, CVPM, RPP)$ (Spurious)

$mult(HR, SV, CO)$ (Correct)

6.2.2. Segment 5

The patient experienced low ABP post bypass due to poor cardiac performance secondary to a technically poor graft and possibly hypovolemia. Inotropic therapy (Dobutamine) was given and a blood transfusion was commenced, both to bring the ABP back up.

L=61

$mult(HR, ABPM, RPP)$ (Correct)

$mult(HR, SV, CO)$ (Correct)

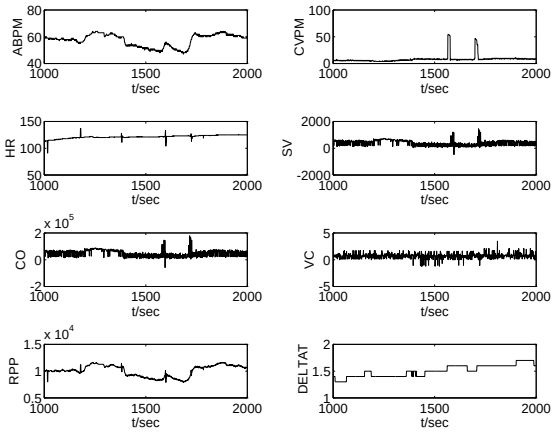

Figure 13. Patient 6, Segment 5: Original Signals

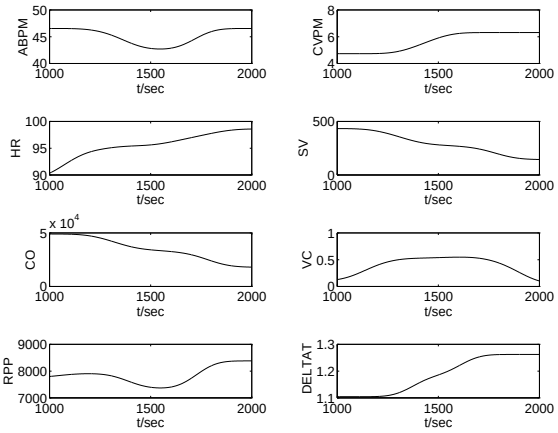

Figure 14. Patient 6, Segment 5: Filtered Signals ($L = 61$)

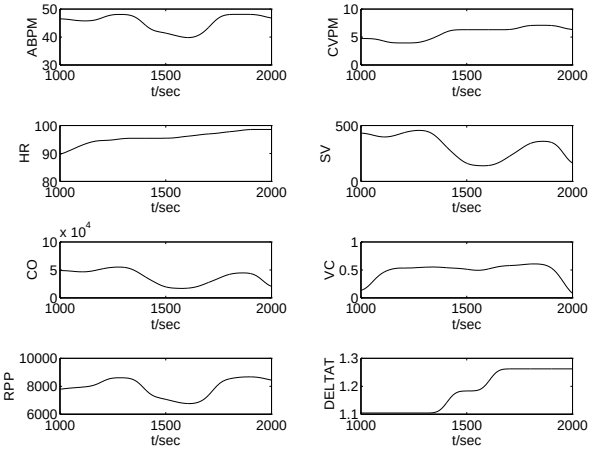

Figure 15. Patient 6, Segment 5: Filtered Signals ($L = 241$)

L=241

$M^+(ABPM, CO)$ Both dropped initially because of poor cardiac performance and hypovolemia, and started to rise following blood infusion.
$M^+(ABPM, SV)$ Both dropped initially because of poor cardiac performance and hypovolemia, and started to rise following blood infusion.
$M^+(SV, CO)$ This follows from the above two constraints.
$mult(HR, ABPM, RPP)$ (Correct)
$mult(HR, SV, CO)$ (Correct)

L=601

$M^+(SV, CO)$ Both dropped because of hypovolemia.
$M^-(HR, CO)$ HR increased both as a compensatory response to decreasing CO due to hypovolemia, and as a response to inotropic therapy.
$M^-(HR, SV)$ This follows from the above two constraints.
$inv\_deriv(SV, CO)$ (Spurious)
$mult(HR, ABPM, RPP)$ (Correct)
$mult(HR, SV, CO)$ (Correct)

6.3. Validity of Models Learned

The results in Sections 6.1 and 6.2 show that reasonable qualitative models can be learned from raw clinical data. Model constraints in our "gold standard" model constructed in Section 5.2 and other useful constraints showed up repeatedly in the models learned from our clinical data. These include:

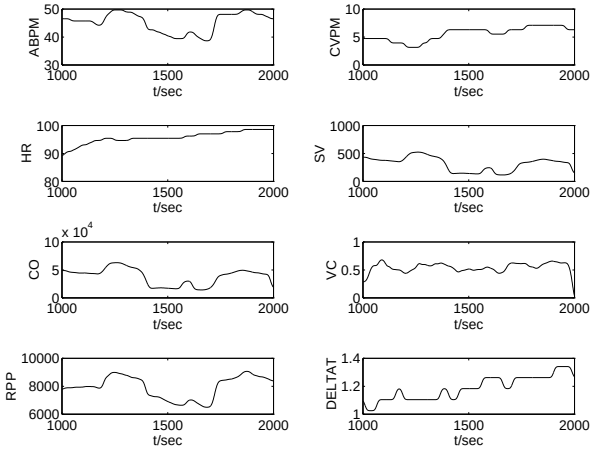

Figure 16. Patient 6, Segment 5: Filtered Signals ($L = 601$)

- constraints valid in general such as $mult(HR, SV, CO)$ and $mult(HR, ABPM, RPP)$.
- constraints valid in specific patient conditions, possibly representing compensatory mechanisms, such as $M^-(HR, CO)$ and $M^-(CO, \Delta T)$ in hypovolemia.
- constraints valid under the effect of certain drugs. For example, $M^+(SV, CO)$ showed up in patients on beta-blockers because of their steady heart rate, and $M^+(ABPM, \Delta T)$ showed up in patients with an increased dosage of GTN causing vasodilation.

6.3.1. Model Variation Across Time

As discussed in Section 3, GENMODEL learns a qualitative model by creating an initial search space of all possible QSIM constraints, and successively pruning inconsistent constraints upon each given system state. Therefore if the system changes within the modelling period (in our case 16.7 minutes), resulting in a different underlying model, neither the old model nor the new model may be obtained. Constraints in the old model are pruned because they are inconsistent with the states after the system change. Constraints in the new model are pruned before the system change because they are inconsistent with the previous system.

This may explain cases where we obtain very few or no model constraints. For example, if a patient is previously stable with an increasing relationship between the heart rate (HR) and the cardiac output (CO) ($M^+(HR, CO)$), but develops hypovolemia in the middle of a modelling process, resulting in a decreasing cardiac output and a compensatory mechanism involving an increasing heart rate, the new valid constraint is $M^-(HR, CO)$. But this will not appear in the final model because at the onset of hypovolemia, this constraint has already been pruned by GENMODEL according to states corresponding to the previously

stable condition. Further, the previously valid constraint $M^+(HR, CO)$ will be pruned because it is now inconsistent with the system states corresponding to hypovolemia.

It may be necessary to develop a mechanism for regenerating the initial search space when there is evidence that the system being modelled has undergone a change in its functional status. One may actually exploit this feature of model variation across time in the context of intelligent patient monitoring systems (see Section 7.2).

6.3.2. Model Variation Across Different Levels of Temporal Abstraction

The models learned in Sections 6.1 and 6.2 varied across different levels of temporal abstraction represented by the filter length L. For example, constraints which involve the skin-to-core temperature gradient ΔT representing the level of vasoconstriction in the body generally appeared only under large values of L, i.e. in coarser time scales. This means the response of ΔT generally lags behind the responses of other parameters.

In general, we observe that fewer model constraints were learned with decreasing L or finer time scales. This may be due to the following reasons:

- Smaller values of L and therefore smaller values of σ correspond to larger cut-off frequencies in the lowpass Gaussian filters, and larger bandwidths in the bandpass filtering operation equivalent to the cascade of the Gaussian filter with the differentiator. This reduces the amount of noise rejection achieved, and results in noise sensitivity problems in detecting zero crossing points and therefore less accurate segmentation. This additional amount of noise may have caused correct constraints to be pruned, resulting in fewer or even no constraints left in the final model.

- Smaller values of L correspond to faster processes which may have more dynamic models. A system change within a modelling period can cause constraints belonging to both the previous and the current model to be pruned, resulting in a smaller model or even one with no constraints.

6.3.3. Model Variation Across Different Levels of Fault Tolerance

We observe that in general the size of the model learned increases with increasing levels of fault tolerance. A fault tolerance level of η means that GENMODEL allows for inconsistent states up to a fraction η of the total number of states in the system behavior before pruning a constraint. Therefore with larger η, fewer constraints will be pruned and the resulting model will contain more constraints.

An indication of η being set too high is that conflicting constraints start to appear. For example, in Patient 5 with $L = 61$ (Figure 6), both $M^+(CVPM, \Delta T)$ and $M^-(CVPM, \Delta T)$ appear in the model learned. This is because both $CVPM$ and ΔT are relatively steady and contain only few inc and dec segments which distinguish between the M^+ and M^- constraints. Within a high level of tolerance, the distinction is obscured.

6.3.4. Sources of Error

False Positives: In the models learned, we observe that spurious constraints sometimes appeared in the resulting model. For example, in Segment 1 of Patient 6 ($L = 601$), we obtained the spurious constraint: $inv\_deriv(ABPM, RPP)$. This may be due to several possible reasons:

- The waveforms are relatively smooth with few critical points. This results in a system behavior with few states, corresponding to few examples for learning. With this small sample size, these examples are consistent with the incorrect constraint. For instance, if whenever ABPM decreases, RPP is positive, then the above incorrect inv_deriv constraint will be learned.
- The level of fault tolerance is set too high resulting in incorrect constraints not being pruned.

False Negatives: We observe that even constraints that are generally valid in all conditions, such as $mult(HR, SV, CO)$, did not appear in every model learned. There are several possible reasons for this:

- Since only a few states are available in a smooth data segment, they may have a disproportionate effect if they are corrupted by noise, causing correct constraints to be pruned.
- Values corrupted by noise are recorded as corresponding values by the system (Hau, 1994). This may cause correct constraints to be pruned.
- The level of fault tolerance is set too low resulting in correct constraints being pruned.

Landmark Values: Temporal abstraction refers to how close two times have to be before we label them as the same distinguished time point. Similarly, we have to decide how close two function values have to be before we label them as the same landmark value. If the tolerance is set too low, we may amplify trends of relatively steady signals. This is the case in the heart rate signals of Patient 5 (Figure 6) which are relatively steady due to the effect of beta-blockers. The fluctuations within 2-3 beats per minute are amplified into a series of inc (increasing) and dec (decreasing) segments. The whole segment could have been labeled as std (steady) with an appropriate tolerance set.

7. Further Work

The goal of this work has been to both understand the complexities that real data sets introduce into the qualitative modelling task, as well as to develop a robust method for learning qualitative models from physiological signals. Further work can be contemplated on both fronts.

7.1. Abstracting qualitative examples drawn from numerical data

The assumption made by many previous workers that the abstraction of qualitative examples from data can be separated from the learning task is not supported by the results of this study. Firstly, the effects of noise in the numerical data continue to manifest themselves up into the qualitative examples, despite significant signal processing. Secondly, the final models learned are directly dependent on the way that the initial qualitative abstraction occurs. This is demonstrated by the effects of choosing different temporal abstraction levels.

Further work is needed on both fronts. There should be a significant body of work from the signal processing literature to assist with the problems of noise. The abstraction issue however, is much deeper. The way one views data is guided by the hypotheses being pursued. There is thus no "correct" qualitative abstraction of the data. This suggests that a more interactive approach needs to be adopted for the learning task, with much more emphasis being placed on the influence that the view taken of data has on the final result.

7.2. A Learning-Based Approach to Diagnostic Patient Monitoring

From the results in Section 6, it was shown that constraints learned do track changes in patient condition over time. For example, the following changes were observed:

Compensatory mechanisms during shock e.g. the constraints $M^-(HR, CO)$ and $M^-(CO, \Delta T)$ learned when the patient experienced hypovolemia.

Effects of drugs e.g. the constraint $M^+(SV, CO)$ tracked the effect of beta-blockers because of the patient's relatively constant heart rate, and the constraint $M^+(ABPM, \Delta T)$ tracked the effect of an increased dosage of GTN causing vasodilation.

Since learned constraints can track patient condition over time, we might be able to build a diagnostic patient monitoring system based on our learning system. The patient monitoring system would continually learn models from patient data and detect changes in the models over time. Diagnoses are made based on these changes. This learning-based approach to diagnostic patient monitoring is summarized in Figure 17.

Such a system would look for stability in constraints over time, recorded as some percentage of match to incoming data. It would attempt to detect when such measures changed, indicating that the constraint was no longer valid and that a new model was being generated by an altered patient state.

In contrast, the traditional history-based approach to diagnostic patient monitoring goes in the opposite direction. It generates histories based on different models. These histories are matched with the patient data. Diagnoses are based on the histories that best match the patient data. This approach can be achieved by a hypothesize-test-refine cycle as shown in Figure 17 (Coiera, 1992, Coiera, 1993).

The learning-based approach may be more accurate since the hypothesized model is generated directly from the patient data, rather than inferred according to some matching criteria with models stored in the monitor's knowledge base.

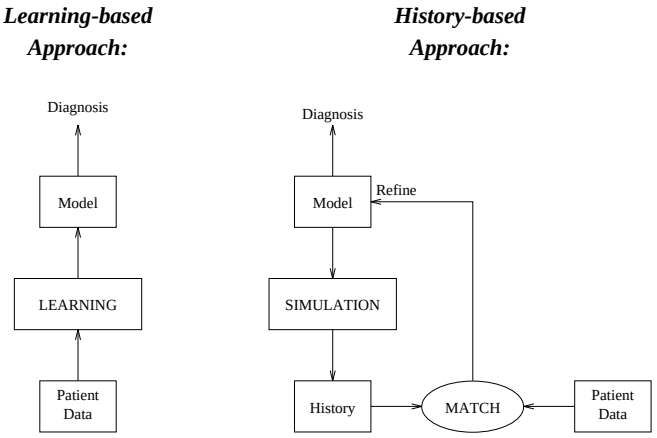

Figure 17. Two approaches to diagnostic patient monitoring. In the learning-based approach, models are continually learned from the patient data. In the history-based approach, a hypothesize-test-refine cycle is used to generate models that best match the patient data. In each approach, diagnoses are made based on the current model.

Acknowledgments

This work was supported by and carried out at the Hewlett-Packard Laboratories in Bristol, UK. The authors would like to thank Drs. Dave Reynolds and Roger Mark for helpful comments.

Notes

1. Monotone conjunctions are ones with positive literals only.
2. The front-end system and the segmenter are implemented in Objectworks\Smalltalk on the HP9000/720.
3. The Gaussian function has the smallest duration-bandwidth product with duration and bandwidth as defined in (Siebert, 1986), and is therefore optimally localized in both the spatial and frequency domains.
4. In general, this property holds true for zero crossings obtained by applying *any* linear differential operator (including the Laplacian and the first derivative) to the filtered signal (Yuille & Poggio, 1986, Babaud, et al., 1986).
5. Raw data was recorded from the Hewlett-Packard Component Monitoring System. The eight signals used for the experiments were derived from the primary measurements, described in (Hau, 1994).
6. The plots shown have arbitrary units because of constant factors omitted in the front-end processing stages, which do not affect the *qualitative* behavior.

References

Babaud, J., Witkin, A.P., Baudin, M. & Duda, R.O. 1986. Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):26–33.

Balestra, G. & Liberati, D. 1992. Qualitative simulation of urea extraction during dialysis. *IEEE Engineering in Medicine and Biology*, 11:80–842.

Blumer, A., Ehrenfeucht, A., Haussler, D. & Warmuth, M.K. 1987. Occam's razor. *Information Processing Letters*, 24(6):377–380.

Bratko, I., Muggleton, S. & Varšek, A. 1991. Learning qualitative models of dynamic systems. In *Proceedings of the International Workshop on Inductive Logic Programming*, pages 207–224.

Coiera, E. 1989. Generating qualitative models from example behaviours. DCS Report 8901, Department of Computer Science, University of New South Wales, Sydney, Australia.

Coiera, E. 1992. Monitoring diseases with empirical and model generated histories. *Artificial Intelligence in Medicine*, 2:135–147, 1990. Revised version appeared in E. Keravnou, editor, *Medical Artificial Intelligence I - Deep Models for Medical Knowledge Engineering*, pages 71-88, Elsevier, Amsterdam, The Netherlands.

Coiera, E. 1992. Qualitative superposition of unmodelled systems. Technical Report HPL-92-166, Hewlett-Packard Laboratories, Bristol, UK.

Coiera, E. 1993. Editorial: Intelligent monitoring and control of dynamic physiological systems. *Artificial Intelligence in Medicine*, 5:1–8.

Falkenhainer, B.C. & Michalski, R.S. 1986. Integrating quantitative and qualitative discovery: The ABACUS system. *Machine Learning*, 1:367–401.

Gobel, F.L., Nordstrom, L.A., Nelson, R.R., Jorgensen, C.R. & Wang, Y. 1978. Rate-pressure product as an index of myocardial oxygen consumption during exercise in patients with angina pectoris. *Circulation*, 57:549–556.

Guyton, A.C. 1981. *Textbook of Medical Physiology*. Saunders, Philadelphia, PA, sixth edition.

Hau, D. 1994. Learning qualitative models from physiological signals. Master's thesis, Department of Electrical Engineering and Computer Science, MIT.

Ironi, L., Stefanelli, M. & Lazola, G. 1992. Qualitative models in medical diagnosis. In E. Keravnou, editor, *Medical Artificial Intelligence I - Deep Models for Medical Knowledge Engineering*. Elsevier, Amsterdam, The Netherlands.

Kearns, M.J. & Vazirani, U.V. 1994. *An Introduction to Computational Learning Theory*. The MIT Press, Cambridge, MA.

Kuipers, B. 1985. Qualitative simulation in medical physiology: A progress report. Technical Report MIT/LCS/TM-280, Laboratory for Computer Science, MIT, Cambridge, MA.

Kuipers, B. 1986. Qualitative simulation. *Artificial Intelligence*, 29:289–338.

Kuipers, B. 1987. Qualitative simulation as causal explanation. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):432–444.

Marr, D. & Hildreth, E. 1980. Theory of edge detection. *Proc. Roy. Soc. London*, 207:187–217.

Muggleton, S., editor. 1992. *Inductive Logic Programming*. Academic Press, London.

Muggleton, S. 1995. Inverse entailment and Progol. *New Generation Computing*, 13:245–286.

Oh, T.E., editor. 1990. *Intensive Care Manual*. Butterworth, Sydney, Australia, third edition.

Oppenheim, A.V. & Schafer, R.W. 1989. *Discrete-time Signal Processing*. Prentice Hall, Englewood Cliffs, NJ.

Plotkin, G.D. 1971. *Automatic Methods of Inductive Inference*. PhD thesis, University of Edinburgh.

Quinlan, J.R. 1990. Learning logical definitions from relations. *Machine Learning*, 5:239–266.

Richards, B.L., Kraan, I. & Kuipers, B.J. 1992. Automatic abduction of qualitative models. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 723–728.

Rivest, R.L. 1987. Learning decision lists. *Machine Learning*, 2(3):229–246.

Saidman, L.J. & Smith, N.T., editors. 1984. *Monitoring in Anesthesia*. Butterworth, Boston, MA, second edition.

Say, R.C. Cem & Kuru, S. 1996. Qualitative system identification: deriving structure from behavior. *Artificial Intelligence*, 83:75–141.

Schlant, R.C. & Alexander, R.W., editors. 1994. *Hurst's the heart: arteries and veins*. McGraw-Hill, New York, 8th edition.

Schut, C. & Bredeweg, B. 1996. An overview of approaches to qualitative model construction. *Knowledge Engineering Review*, 11:1–25.

Siebert, W.M. 1986. *Circuits, Signals, and Systems*. The MIT Press, Cambridge, MA.

Strang, G. 1989. Wavelets and dilation equations: a brief introduction. *SIAM Review*, 31(4):614–627.

Toal, P. & Hunter. J. 1990. A qualitative model of cardiac haemodynamics. Technical Report AUCS/TR9001, Department of Computing Science, University of Aberdeen, Aberdeen, U.K.

Valiant, L.G. 1984. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Varšek, A. 1991. Qualitative model evolution. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 1311–1316, Sydney, Australia.

Weinberg, J., Biswas, G. & Uckun, S. 1990. Continuing adventures in qualitative modelling: A qualitative heart model. In *Proceedings of the Third International Conference on Industrial and Engineering Applications of AI Expert Systems (IEA/AIE 90)*, Charlestown, SC. ACM Press.

Wesseling, K.H., deWit, B., Weber, J.A.P. & Smith, N.T. 1983. A simple device for the continuous measurement of cardiac output. *Advances in Cardiovascular Physiology*, 5:16–52.

Wyngaarden, J.B., Smith, L.H. & Bennett, J.C., editors. 1992. *Cecil Textbook of Medicine*. Saunders, Philadelphia, PA, 19th edition.

Yuille, A.L. & Poggio, T.A. 1986. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25, January 1986.

Received September 8, 1995
Accepted June 28,1996
Final Manuscript August 13, 1996