Journal of Artificial Intelligence Research 32 (2008) 825–877 Submitted 06/07; published 08/08

Qualitative System Identification from Imperfect Data

George M. Coghill G.COGHILL@ABDN.AC.UK
School of Natural and Computing Sciences
University of Aberdeen, Aberdeen, AB24 3UE. UK.

Ashwin Srinivasan ASHWIN.SRINIVASAN@IN.IBM.COM
IBM India Research Laboratory
4, Block C, Institutional Area
Vasant Kunj Phase II, New Delhi 110070, India.
and
Department of CSE and Centre for Health Informatics
University of New South Wales, Kensington
Sydney, Australia.

Ross D. King RDK@ABER.AC.UK
Deptartment of Computer Science
University of Wales, Aberystwyth, SY23 3DB. UK.

Abstract

Experience in the physical sciences suggests that the only realistic means of understanding complex systems is through the use of mathematical models. Typically, this has come to mean the identification of quantitative models expressed as differential equations. Quantitative modelling works best when the structure of the model (i.e., the form of the equations) is known; and the primary concern is one of estimating the values of the parameters in the model. For complex biological systems, the model-structure is rarely known and the modeler has to deal with both model-identification and parameter-estimation. In this paper we are concerned with providing automated assistance to the first of these problems. Specifically, we examine the identification by machine of the structural relationships between experimentally observed variables. These relationship will be expressed in the form of qualitative abstractions of a quantitative model. Such qualitative models may not only provide clues to the precise quantitative model, but also assist in understanding the essence of that model. Our position in this paper is that background knowledge incorporating system modelling principles can be used to constrain effectively the set of good qualitative models. Utilising the model-identification framework provided by Inductive Logic Programming (ILP) we present empirical support for this position using a series of increasingly complex artificial datasets. The results are obtained with qualitative and quantitative data subject to varying amounts of noise and different degrees of sparsity. The results also point to the presence of a set of qualitative states, which we term *kernel subsets*, that may be necessary for a qualitative model-learner to learn correct models. We demonstrate scalability of the method to biological system modelling by identification of the glycolysis metabolic pathway from data.

1. Introduction

There is a growing recognition that research in the life sciences will increasingly be concerned with ways of relating large amounts of biological and physical data to the structure and function of higher-level biological systems. Experience in the physical sciences suggests that the only realistic means of understanding such complex systems is through the use of mathematical models. A topical example is provided by the Physiome Project which seeks to utilise data obtained from sequencing the human genome to understand and describe the human organism using models that: "...include everything from diagrammatic schema, suggesting relationships among elements composing a system, to fully quantitative, computational models describing the behaviour of the physiological systems and an organism's response to environmental change" (see `http://www.physiome.org/`). This paper is concerned with a computational tool that aims to assist in the identification of mathematical models for such complex systems.

Broadly speaking, system identification can be viewed as "the field of modelling dynamic systems from experimental data" (Soderstrom & Stoica, 1989). We can distinguish here between: (a) "classical" system identification techniques, developed by control engineers and econometricians; and (b) machine learning techniques, developed by computer scientists. There are two main aspects to this activity. First, an appropriate structure has to be determined (the model-identification problem). Second, acceptably accurate values for parameters in the model are to be obtained (the parameter-estimation problem). Classical system identification is usually (but not always) used when the possible model structure is known *a priori*. Machine learning methods, on the other hand, are of interest when little or nothing is known about the model structure. The tool described here is a machine learning technique that identifies *qualitative* models from observational data. Qualitative models are non-parametric; therefore all the computational effort is focussed on model-identification (there are no parameters to be estimated). The task is therefore somewhat easier than more ambitious machine learning programs that attempt to identify parametric quantitative models (Bradley, Easley, & Stolle, 2000; Džeroski, 1992; Džeroski & Todorovski, 1995; Todorovski, Srinivasan, Whiteley, & Gavaghan, 2000). Qualitative model-learning has a number of other advantages: the models are quite comprehensible; system-dynamics can be obtained relatively easily; the space of possible models is finite; and noise-resistance is fairly high. On the down-side, qualtitative model-learners have often produced models that are under- or over-constrained; the models can only provide clues to the precise mathematical structure; and the models are largely restricted to being abstractions of ordinary differential equations (ODEs). We attempt to mitigate the first of these shortcomings by adopting the framework of Inductive Logic Programming (ILP) (see Bergadano & Gunetti, 1996; Muggleton & Raedt, 1994). Properly constrained models are identified using a library of syntactic and semantic constraints—part of the *background knowledge* in the ILP system—on physically meaningful models. Like all ILP systems, this library is relatively easily extendable. Our position in this paper is that:

> *Background knowledge incorporating physical (and later, biological) system modelling principles can be used to constrain the set of good qualitative models.*

Using some some classical physical systems as test-beds we demonstrate empirically that:

- A small set of constraints, in conjunction with a Bayesian scoring function, is sufficient to identify correct models.
- Correct models can be identified from qualitative or quantitative data which need not contain measurements for all variables in the model; and they can be learned with

sparse data with large amounts of noise. That is, the correct models can be identified when the input data are incomplete, incorrect, or both.

A closer examination of the performance on these test systems has led to the discovery of what we term *kernel subsets*: minimal qualitative states that when present guarantee our implementation will identify a model correctly. This concept may be of value to other model identification systems.

Our primary interests, as made clear at the outset, lie in biological system identification. The completion of the sequencing of the key model genomes and the rise of new technologies have opened up the prospect of modelling cells *in silico* in unprecedented detail. Such models will be essential to integrate the ever-increasing store of biological knowledge, and have the potential to transform medicine and biotechnology. A key task in this emerging field of *systems biology* is to identify cellular models directly from experimental data. In applying qualitative system identification to systems biology we focus on models of metabolism: the interaction of small molecules with enzymes (the domain of classical biochemistry). Such models are the best established in systems biology. To this end, we demonstrate that the approach scales up to identify the core of a well-known, complex biological system (glycolysis) from qualitative data. This system is modelled here by a set of 25 qualitative relations, with several unmeasured variables. The scale-up is achieved by augmenting the background knowledge to incorporate general chemical and biological constraints on enzymes and metabolites.

The rest of the paper is organised as follows. In the next section we present the learning approach ILP-QSI by means of an example: the u-tube. We also describe the details of the learning algorithm in this section. In Section 3 we apply the learning experiments to a number of other systems in the same class as the u-tube, present the results obtained, and discuss the results for all the experiments reported thus far. Section 4 extends the work from learning from qualitative data to a set of proof-of-concept experiments to assess the ability of ILP-QSI to learn from quantitative data. The scalability of the *method* is tested in Section 5 by application to a large scale metabolic pathway: glycolysis. In Section 6 we discuss related work; and finally in Section 7 we provide a general discussion of the research and draw some general conclusions.

2. Qualitative System Identification Using ILP

In order to aid understanding of the method presented in this paper we will first present a detailed description of the process as applied to an illustrative system: the u-tube. The u-tube has been chosen because it is a well understood system, and is one that has been used in the literature (Muggleton & Feng, 1990; Say & Kuru, 1996). The results emerging from this set of experiments will allow us to draw some tentative conclusions regarding qualitative systems identification.

In subsequent sections we will present the results of applying the method described in this section to further examples from the same class of system; this will enable us to generalise our tentative conclusions. We will also apply the method to a large scale biological system to demonstrate the scalability of the *method*.

State	h_1	h_2	q_x	$-q_x$
1	$< 0, std >$	$< 0, std >$	$< 0, std >$	$< 0, std >$
2	$< 0, inc >$	$< (0, \infty), dec >$	$< (-\infty, 0), inc >$	$< (0, \infty), dec >$
3	$< (0, \infty), dec >$	$< 0, inc >$	$< (0, \infty), dec >$	$< (-\infty, 0), inc >$
4	$< (0, \infty), dec >$	$< (0, \infty), inc >$	$< (0, \infty), dec >$	$< (-\infty, 0), inc >$
5	$< (0, \infty), std >$	$< (0, \infty), std >$	$< 0, std >$	$< 0, std >$
6	$< (0, \infty), inc >$	$< (0, \infty), dec >$	$< (-\infty, 0), inc >$	$< (0, \infty), dec >$

Table 1: The envisionment states used for the u-tube experiments. The qualitative values are in the standard form used by QSIM. Positive values for the magnitude are represented by the interval $(0, \infty)$, negative values by the interval $(-\infty, 0)$ and zero by 0. The directions of change are self explanatory with increasing represented by *inc*, decreasing by *dec* and steady by *std*.

2.1 An Illustrative System: The U-tube

The u-tube system (Fig. 1) is a closed system consisting of two tanks containing (or potentially containing) fluid, joined together at their base by a pipe. Assuming there is fluid in the system it passes from one tank to the other via the pipe – from the tank with the higher fluid level to the tank with the lower fluid level (as a function of the difference in height). If the height of fluid is the same in both tanks then the system is in equilibrium and there is no fluid flow.

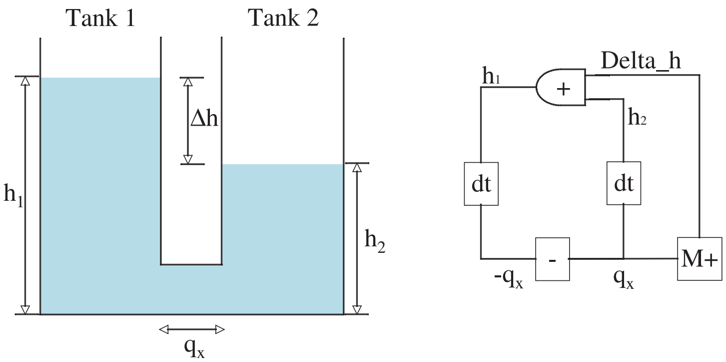
The u-tube can be represented by a system of ordinary differential equations as follows:

$$\left.\begin{array}{l} \frac{dh_1}{dt} = k \cdot (h_1 - h_2) \\ \\ \frac{dh_2}{dt} = k \cdot (h_2 - h_1) \end{array}\right\} \tag{1}$$

A qualitative model may be obtained simply by abstracting from the real numbers, which would normally be associated with Equation 1, into the quantity space of the signs. A common formalism used to represent qualitative models is QSIM (Kuipers, 1994). In this representation models are conjunctions of constraints, each of which are two or three place predicates representing abstractions of real valued arithmetic and functional operations. All variables in a model have values represented by two element vectors consisting of (in the most abstract case) the sign of both the magnitude and direction of change of the variable. In order to accommodate this restriction on the number of variables in a constraint we may rewrite Equation 1 as follows:

$$\left.\begin{array}{l} \Delta h = (h_1 - h_2) \\ q_x = k \cdot \Delta h \\ \frac{dh_1}{dt} = q_x \\ \frac{dh_2}{dt} = -q_x \end{array}\right\} \tag{2}$$

where h_1 and h_2 are the height of fluid in Tank 1 and Tank 2 respectively; Δh is the difference in the height of fluid in the tanks; and q_x is the flow of fluid between the tanks. This can be converted directly to QSIM constraints as shown in Fig. 1.



DERIV(h_1,q_x),
DERIV(h_2,$-q_x$),
ADD(h_2,$Delta\_h$, h_1),
M$^+$($Delta\_h$,q_x),
MINUS(q_x,$-q_x$).

Figure 1: The u-tube: (left) physical; (middle) QSIM diagrammatic; (right) QSIM constraints. In the QSIM version of the model `Delta_h` corresponds to Δh in the physical model. In QSIM, M$^+(\cdot,\cdot)$ is the qualitative version of a functional relation which indicates that there is a monotonically increasing relation between the two variables which are its arguments. The M$^+(\cdot,\cdot)$ constraint represents a family of functions that includes both linear and non-linear relations.

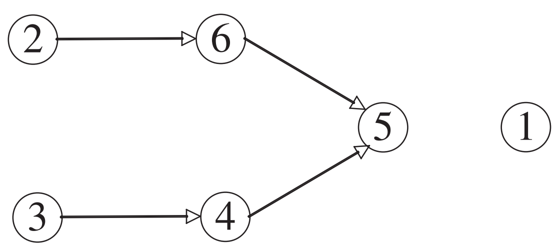


Figure 2: The u-tube envisionment graph.

Appropriate qualitative analysis of the u-tube will produce the *states* shown in Table 1, which are the states of the envisionment. These represent all the distinct qualitative states in which the u-tube may exist and Fig. 2 depicts all the possible behaviours (in terms of transitions between states)[1]. This figure represents a *complete envisionment* of the system, which is the graph containing all the qualitative states of the system and all the transitions between them for a particular input value. In the case of the u-tube presented here there is no input (which is equivalent to a value of zero). On the other hand the behaviours of a u-tube may be observed under a number of experimental (initial) conditions, with measurements being taken of the height of fluid in each tank and the flow between the tanks. These can be converted (by means of a quantitative-to-qualitative converter) into a set of qualitative observations (or states). If sufficient temporal information is available to enable the calculation of qualitative derivatives, each observation will be a tuple stating the magnitude and direction of change of the measured variable. These observations will also contain the states in the complete envisionment of Table 1 (or some subset thereof).

1. State 1 represents the situation where there is no fluid in the system, so nothing happens and it is not interesting.

The u-tube is a member of a large class of dynamic systems which are defined by their *states*: state systems. In such systems the values of the variables at all future times are defined by the current state of the system regardless of how that state was achieved (Healey, 1975). This means that for simulation, any system state can act as an initial state. In the current context it means that in order to learn the structure of such systems we need only focus on the states themselves and may ignore the transitions between states. This enables us to explore the power set of the envisionment to ascertain the conditions under which system identification is possible. Given these qualitative observations as examples, background knowledge consisting of constraints on models (described later) and QSIM relations, the learning system (which we name ILP-QSI) performs a search for acceptable models. To a suitable first approximation, the basic task can be viewed as a discrete optimisation problem of finding the lowest cost elements amongst a finite set of alternatives. That is, given a finite discrete set of models S and a real-valued cost-function $f : \mathcal{S} \rightarrow \Re$, find a subset $\mathcal{H} \subseteq \mathcal{S}$ such that $\mathcal{H} = \{H | H \in \mathcal{S} \text{ and } f(H) = \min_{H_i \in \mathcal{S}} f(H_i)\}$. This problem may be solved by employing a procedure that searches through a directed acyclic graph representation of possible models. In this representation, a pair of models are connected in the graph if one can be transformed into another by an operation called *refinement*. Fig. 3 shows some parts of a graph for the u-tube in which a model is refined to another by the addition of a qualitative constraint. An optimal search procedure (the branch-and-bound procedure) traverses this graph in some order, at all times keeping the cost of the best nodes so far. Whenever a node is reached where it is certain that it and all its descendents have a cost higher than that of the best nodes, then the node and its descendents are removed from the search.

There are a number of features apparent in the u-tube model that are relevant to the learning method utilised in this work (and discussed in Section 2.3) that will be described here since they regard general modelling issues relevant to the learning of qualitative models of dynamic systems.

The first thing that may be noted in this regard is that the expressions in Equation 2 and the resulting qualitative constraints are *ordered*; that is, given the values for the exogenous variables and the magnitude of the state variables (the height of fluid in the tanks in this case) the equations can be placed in an order such that the variables on the left hand side all may have their values calculated before they appear on the right hand side of an equation[2]. This particular form of ordering in known as *causal ordering* (Iwasaki & Simon, 1986). A causally ordered system can be depicted graphically as shown in Fig. 4.

A causally ordered model contains no *algebraic loops*. In quantitative systems one tries to avoid algebraic loops because they are hard to simulate, requiring additional simultaneous equation solvers to be used.

A qualitative model combined with a Qualitative Reasoning (QR) inference engine will provide an envisionment of the system of interest. That is, it will generate all the qualitatively distinct states in which the system may exist. In the case of the u-tube there are six such states as given in Table 1. Example behaviours resulting from these states are shown in Fig. 2.

2. This ordering is not required by QSIM in order to preform qualitative simulation. However, the ability to order equations in this manner can be utilised as a filter in the learning system in order to eliminate models containing algebraic loops.

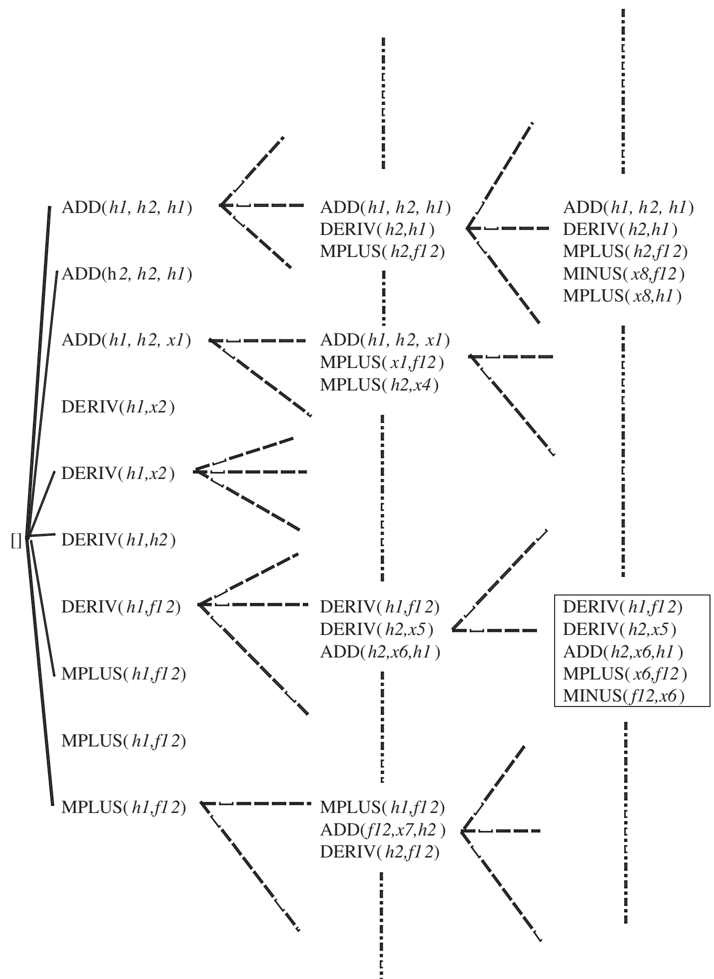


Figure 3: Some portions of the u-tube lattice (with the target model in the box).

It may be noted that the differential equation model captures the essence of the explanation given in the first paragraph of this section. It is sufficient to *explain* the operation of such a system, as well as to predict the way it will behave, and it contains only those variables and constants necessary to achieve this task - i.e. the model is ***parsimonious.***[3]

Furthermore, examination of the causal diagram in Fig. 4 indicates that the causal ordering is in a particular direction – from the magnitudes of the state variables to their derivatives. The link between the derivatives and the magnitudes of the state variables is through an integration over time. This is ***integral causality*** and is the preferred kind

3. It is possible that for didactic purposes we may want to include more detail, for example a relation between the intertank flow and the pressure difference, or between the height of fluid and the pressure. There is no reason why we would expect such relations to be found; although in the context of an adequate theoretical framework into which the model fits, the model provides pointers in that direction. On the other hand, one can envisage simpler models existing which may be suitable for prediction but inadequate for the required kind of explanation. See Section 6 for more on this.

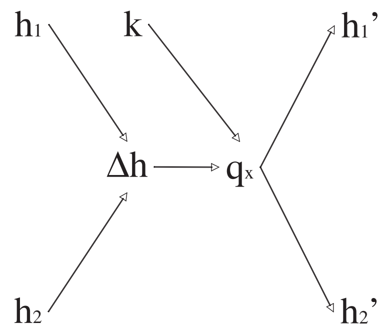


Figure 4: A causal ordering of the u-tube model given in Equation 2.

of causality in systems engineering modelling; and simulation generally. This is because integration smooths out noise whereas differentiation introduces it.

All variables are either endogenous or exogenous. Exogenous variables influence the system but are not influenced by it. ***Well posed*** models do not have any *flapping* variables; that is, endogenous variables that appear in only one constraint. Because QSIM includes a `DERIV` constraint linking the state variables directly to their derivatives, and all the systems in which we are interested are regulatory, containing feedback paths, all endogenous variables must appear in at least two constraints.

Well posedness and parsimony are mandatory properties of the model, the other properties are desirable but not always achievable and so may have to be relaxed. However, for all the systems examined in this paper each of these properties holds.

A final feature of the u-tube model is that it represents a single system. It is an assumption implicit in all the learning experiments described in this paper that the data measured belongs to a single coherent system. This is in keeping with general experimental approaches where it is assumed that the measurements are related in some way by being part of the same system. Of course we may get this wrong and have to relax the requirement because we discover that what we thought were related cannot actually be brought together in a single model. This generalises the requirement for parsimony in line with Einstein's adage that a model should be "as simple as possible and no simpler". In this case it translates to minimising the number of disjoint sub-systems identified.

2.2 A Qualitative Solution Space

In Section 2.3 we shall present an algorithm for automatically constructing models from data. With this method we utilise background knowledge consisting of QSIM modelling primitives combined with systems theory meta-knowledge (e.g. parsimony and causality). Later we shall also provide an analysis of the models learned and the states utilised to learn them in order to ascertain which, if any, states are more important for successful learning. One way to facilitate this analysis is to make use of a *solution space* to relate the qualitative states to the critical points of the relevant class of systems (via the isoclines of the system)[4]

4. The *critical points* of a dynamic system are points where one or more of the derivatives of the state variables is zero. The *isoclines* are contours of critical points.

(Coghill, 2003; Coghill, Asbury, van Rijsbergen, & Gray, 1992). As stated previously, a qualitative analysis of the u-tube will generate an envisionment containing six states, as shown in Table 1, and depicted in the envisionment graph given in Fig. 2. Continuing with

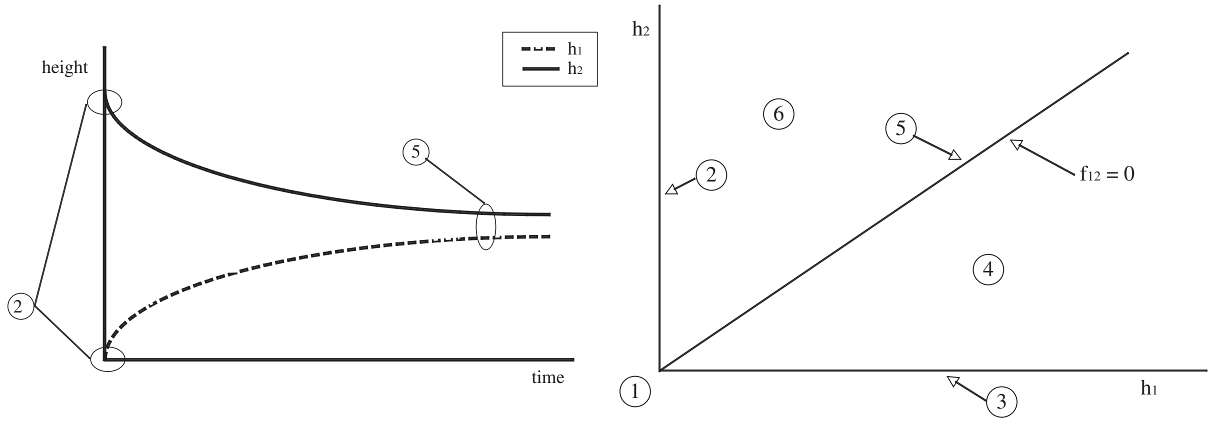


Figure 5: The qualitative states of the u-tube system presented on representative time courses (left) and on the solution space (right). The state numbers refer to the states of the u-tube described above. (State 5 represents the steady state which is strictly speaking only reached at $t = \infty$, but is in practice taken to occur when the two trajectories are "sufficiently close", as shown here.)

the u-tube; there are two ways it can behave (ignoring state 1), captured in Fig. 5. Either the head of fluid in tank 1 is greater than that in tank 2 (state 4) (in the extreme tank 1 is empty – state 3), or the head is greater in tank 2 than tank 1 (state 6). Fig. 5 (left) shows the transient behaviour for the extreme case where tank 1 is empty (state 2); it can be seen from this diagram that while the head starts in this condition its eventual end is equilibrium (state 5). In this state Equation 1 can be rewritten as:

$$\left.\begin{aligned} 0 &= k \cdot (h_1 - h_2) \\ 0 &= k \cdot (h_2 - h_1) \end{aligned}\right\} \tag{3}$$

By definition k must be non-zero; so the only solution to this pair of equations is:

$$h_2 = h_1$$

This relation can be plotted on a graph as shown on the right hand side of Fig. 5. Now the qualitative states of the u-tube may be placed on this solution space graph in relation to the equilibrium line. This representation (similar in form to a phase space diagram) is useful because it provides a global picture of the location of the qualitative states of an envisionment relative to the equilibria or critical points of the system. It has also been utilised in the construction of diagnostic expert systems (Warren, Coghill, & Johnstone, 2004). For further details of this means of analysing envisionments see the work of Coghill (2003) and Coghill et al. (1992).

$bb(i, \rho, f)$: Given an initial element i from a discrete set S; a successor function $\rho : S \rightarrow 2^S$; and a cost function $f : S \rightarrow \Re$, return $H \subseteq S$ such that H contains the set of cost-minimal models. That is for all $h_{i,j} \in H, f(h_i) = f(h_j) = f_{min}$ and for all $s' \in S \backslash H$ $f(s') > f_{min}$.

1. $Active := \langle (i, -\infty) \rangle$.
2. $worst := \infty$
3. $selected := \emptyset$
4. while $Active \neq \langle\rangle$
5. begin
   (a) remove element $(k, cost_k)$ from $Active$
   (b) if $cost_k < worst$
   (c) begin
      i. $worst := cost_k$
      ii. $selected := \{k\}$
      iii. let $Prune_1 \subseteq Active$ s.t. for each $j \in Prune_1$, $\underline{f}(j) > worst$ where $\underline{f}(j)$ is the lowest cost possible from j or its successors
      iv. remove elements of $Prune_1$ from $Active$
   (d) end
   (e) elseif $cost_k = worst$
      i. $selected := selected \cup \{k\}$
   (f) $Branch := \rho(k)$
   (g) let $Prune_2 \subseteq Branch$ s.t. for each $j \in Prune_2$, $f_{min}(j) > best$ where $f_{min}(j)$ is the lowest cost possible from j or its successors
   (h) $Bound := Branch \backslash Prune_2$
   (i) for $x \in Bound$
      i. add $(x, f(x))$ to $Active$
6. end
7. return $selected$

Figure 6: A basic branch-and-bound algorithm. The type of *Active* determines specialised variants: if *Active* is a stack (elements are added and removed from the front) then depth-first branch-and-bound results; if *Active* is a queue (elements added to the end and removed from the front) then breadth-first branch-and-bound results; if *Active* is a prioritised queue then best-first branch-and-bound results.

2.3 The Algorithm

The ILP learner used in this research is a multistage procedure, each of which addresses a discrete optimisation problem. In general terms, this is posed as follows: given a finite discrete set S and a cost-function $f : S \rightarrow \Re$, find a subset $H \subseteq S$ such that $H = \{s | s \in S \text{ and } f(s) = \min_{s_i \in S} f(s_i)\}$. An optimal algorithm for solving such problems is the "branch-and-bound" algorithm, shown in Fig. 6 (the correctness, complexity and optimality properties of this algorithm are presented in a paper by Papadimitriou & Steiglitz, 1982). A specific variant of this algorithm is available within the software environment comprising Aleph (Srinivasan, 1999). The modified procedure is in Fig. 7. The principal differences from Fig. 6 are:

1. The procedure is given a set of starting points H_0, instead of a single one (i in Fig. 6);

2. A limitation on the number of nodes explored (n in Fig. 7);
3. The use of a boolean function *acceptable* : $\mathcal{H} \times \mathcal{B} \times \mathcal{E} \rightarrow \{FALSE, TRUE\}$. acceptable(k,B,E) is TRUE, if and only if: (a) Hypothesis k "explains" the examples E, given B in the usual sense understood in ILP (that is, $B \wedge k \models E$ in the absence of noise); and (b) Hypothesis k is consistent with any constraints I contained in the background knowledge (that is $B \wedge k \wedge I \not\models \Box$). In practice, it is possible to merge these requirements by encoding the requirement for entailing some or all of the examples as a constraint in B;
4. Inclusion of background knowledge and examples (B and E in Fig. 7). These are arguments to both the refinement operator ρ (the reason for this will become apparent shortly) and the cost function f.

The following points are relevant for the implementation used here:

- Each qualitative model is represented as a single definite clause. Given a definite clause C, the qualitative constraints in the model (the size of the model) are obtained by counting the number of qualitative constraints in C. This will also be called the "size of C".
- Constraints, such as the restriction to well-posed models (described below), are assumed to be encoded in the background knowledge;
- The initial set H_0 in Fig. 7 consists of the empty clause denoted here as $\emptyset$. That is, $H_0 = \{\emptyset\}$;
- $acceptable(C, B, E)$ is $TRUE$ for any qualitative model C that is consistent with the constraints in B, given E.
- *Active* is a prioritised queue sorted by f;
- The successor function used is ρ_A. This is defined as follows. Let S be the size of an acceptable model and C be a qualitative model of size S' with $n = S - S'$. We assume B contains a set of mode declarations in the form described by (Muggleton, 1995). Then, given a definite clause C, obtain a definite $C' \in \rho_A(C, B, E)$ where $\rho_A = \rho_A^n = \langle D' | \ \exists D \in \rho_A^{n-1}(C, B, E) \text{ s.t. } D' \in \rho_A^1(D, B, E)\rangle, (n \geq 2)$. $C' \in \rho_A^1(C, B, E)$ is obtained by adding a literal L to C, such that:
  - Each argument with mode $+t$ in L is substituted with any input variable of type t that appears in the positive literal in C or with any variable of type t that occurs in a negative literal in C;
  - Each argument with mode $-t$ in L is substituted with any variable in C of type t that appears before that argument or by a new variable of type t;
  - Each argument with mode $\#t$ in L is substituted with a ground term of type t. This assumes the availability of a generator of elements of the Herbrand universe of terms; and
  - $acceptable(C', B, E)$ is $TRUE$.

$bb_A(B, E, H_0, \rho, f, n)$: Given background knowledge $B \in \mathcal{B}$; examples $E \in \mathcal{E}$; a non-empty set of initial elements H_0 from a discrete set of possible hypotheses $\mathcal{H}$; a successor function $\rho : \mathcal{H} \times \mathcal{B} \times \mathcal{E} \rightarrow 2^{\mathcal{H}}$; a cost function $f : \mathcal{H} \times \mathcal{B} \times \mathcal{E} \rightarrow \Re$; and a maximum number of nodes $n \in \mathcal{N}$ $(n \geq 0)$ to be explored, return $H \subseteq \mathcal{H}$ such that H contains the set of cost-minimal models of the models explored.

1. $Active = \langle\rangle$
2. for $i \in H_0$
   (a) add $(i, -\infty)$ to $Active$
3. $worst := \infty$
4. $selected := \emptyset$
5. $explored := 0$
6. while ($explored < n$ and $Active \neq \langle\rangle$)
7. begin
   (a) remove element $(k, cost_k)$ from $Active$
   (b) increment $explored$
   (c) if $acceptable(k, B, E)$
   (d) begin
      i. if $cost_k < worst$
      ii. begin
         A. $worst := cost$
         B. $selected := \{k\}$
         C. let $Prune_1 \subseteq Active$ s.t. for each $j \in Prune_1$, $\underline{f}(j, B, E) > worst$ where $\underline{f}(j, B, E)$ is the lowest cost possible from j or its successors
         D. remove elements of $Prune_1$ from $Active$
      iii. end:
      iv. elseif $cost_k = worst$
         A. $selected := selected \cup \{k\}$
   (e) end
   (f) $Branch := \rho(k, B, E)$
   (g) let $Prune_2 \subseteq Branch$ s.t. for each $j \in Prune_2$, $\underline{f}(j, B, E) > worst$ where $\underline{f}(j, B, E)$ is the lowest cost possible from j or its successors
   (h) $Bound := Branch \backslash Prune_2$
   (i) for $x \in Bound$
      i. add $(x, f(x, B, E))$ to $Active$
8. end
9. return $selected$

Figure 7: A variant of the basic branch-and-bound algorithm, implemented within the Aleph system. Here $\mathcal{B}$ and $\mathcal{E}$ are sets of logic programs; and $\mathcal{N}$ the set of natural numbers.

The following properties of ρ_A^1 (and, in turn of ρ_A) can be shown to hold (Riguzzi, 2005):

- It is locally finite. That is, $\rho_A^1(C, B, E)$ is finite and computable (assuming the constraints in B are computable);
- It is weakly complete. That is, any clause containing n literals can be obtained in n refinement steps from the empty clause;

  - It is not proper. That is, C' can be equivalent to C;
  - It is not optimal. That is, C' can be obtained multiply by refining different clauses.

  In addition, it is clear by definition that given a qualitative model C, $acceptable(C', B, E)$ is $TRUE$ for any model $C' \in \rho^1_A(C, B, E)$. In turn, it follows that $acceptable(C', B, E)$ is $TRUE$ for any $C' \in \rho_A(C, B, E)$.

- The cost function used (following Muggleton, 1996) is $f_{Bayes}(C, B, E) = -\mathrm{P}(C|B, E)$ where $\mathrm{P}(C|B, E)$ is the Bayesian posterior probability estimate of clause C, given background knowledge B and positive examples E. Finding the model with the maximal posterior probability (that is, lowest cost) involves maximising the function (McCreath, 1999):

$$Q(C) = \log D_{\mathcal{H}}(C) + p \log \frac{1}{g(C)}$$

  where $D_{\mathcal{H}}$ is a prior probability measure over the space of possible models; p is the number of positive examples (that is, $p = |E|$); and g is the generality of a model. We use the approach used in the ILP system C-Progol to obtain values for these two functions. That is, the prior probability is related to the complexity of models (more complex models are taken to be less probable, *a priori*); and the generality of a model is estimated using the number of random examples entailed by the model, given the background knowledge B (the details of this are presented by Muggleton in his paper of 1996).

  We have selected this Bayesian function to score hypotheses since it represents, to the best of our knowledge, the only one in the ILP literature explicitly developed for the case where data consist of positive examples only (as is the situation in this paper, where examples are observations of system behaviour: system identification from "non-behaviour" does not represent the usual understanding of the task we are attempting here).

It is evident that these choices make the branch-and-bound procedure a simple "generate-and-score" approach. Clearly, the approach is only scalable if the constraints encoding well-posed models are sufficient to restrict acceptable models to some reasonable number: we describe a set of such constraints that are sufficient for the models examined in this paper. In the rest of the paper, the term ILP-QSI will be taken to mean the Aleph branch-and-bound algorithm with the specific choices above.

2.3.1 Well-posed models

Well-posed models were introduced in Section 2.1; in the current implementation they are defined as satisfying at least the following syntactic constraints:

1. *Size.* The model must be of a particular size (measured by the number of qualitative relations for physical models in Sections 2.4 and 3 or the number of metabolites for the biological model in Section 5). This size is pre-specified.

2. *Complete.* The model must contain all the measured variables.

3. *Determinate.* The model must contain as many relations as variables (a basic principle of systems theory—the reader may recall a version from school algebra, where a system of equations contains as many equations as unknowns).

4. *Language.* The number of instances of any qualitative relation in the model must be below some pre-specified limit. This kind of restriction has been studied in greater detail in the work of Camacho (2000).

and at least the following semantic constraints:

5. *Sufficient.* The model must adequately explain the observed data. By "adequate", we intend to acknowledge here that due to noise in the measurements, not all observations may be logical consequences of the model[5]. The percentage of observations that must be explainable in this sense is a user-defined value.

6. *Redundant.* The model must not contain relations that are redundant. For example, the relation $\texttt{ADD}(inflow, outflow, x1)$ is redundant if the model already has $\texttt{ADD}(outflow, inflow, x1)$.

7. *Contradictory.* The model must not contain relations that are contradictory given other relations present in the model.

8. *Dimensional.* The model must contain relations that respect dimensional constraints. This prevents, for example, addition of relations like $\texttt{ADD}(inflow, outflow, amount)$ that perform arithmetic on variables that have different units of measurement.

The following additional constraints which are here incorporated in the algorithm could be ignored (because they are preferences rather than absolute rules), but all results presented in this paper require them to be satisfied:

9. *Single.* The model must not contain two or more disjoint models. The assumption is that if a set of measurements are being made within a particular context then the user desires a single model that includes those measurement variables.

10. *Connected.* All intermediate variables should appear in at least two relations.

11. *Causal.* The model must be causally ordered (Iwasaki & Simon, 1986) with an integral causality (Gawthrop & Smith, 1996). That is, the causality runs through the algebraic constraints of the model from the magnitudes of the state variables to their derivatives; and from the derivatives to the magnitudes through a `DERIV` constraint only.

This list is not intended to be exhaustive: we fully expect that they would need to be augmented by other domain-specific constraints (the biological system identification problem described in Section 5 provides an instance of this). The advantage of using ILP is that such augmentation is possible in a relatively straightforward manner.

5. Strictly speaking, the model in conjunction with the background knowledge.

2.4 Experimental Investigation of Learning the U-tube System

In this section we present a comprehensive experimental test of the learning algorithm described in the previous section. We again focus on the u-tube to illustrate the approach and explain the results obtained. In a subsequent section we will present the results of applying ILP-QSI to learning the structure of a number of different systems of a similar kind. The data utilised in these experiments is qualitative. It is assumed that either the measurements themselves yield qualitative values or that they are quantitative time series that have been converted to qualitative values. This latter may be necessary in situations where the quantitative time series data are not available in sufficient quantity to permit quantitative system identification to be performed.

The following is the general method applied to learning all the systems studied for this paper.

2.4.1 Experimental Aim

Using the u-tube system, investigate the model identification capabilities of ILP-QSI using qualitative data that are subject to increasing amounts of noise and are made increasingly sparse in order to ascertain the circumstances under which the target system may be accurately identified, as a function of the number of qualitative observations available.

2.4.2 Materials and Method

The model learning system ILP-QSI seeks to learn qualitative structural models from qualitative data; therefore the focus of the experiments is on learning from qualitative data.

Data There are no inputs (exogenous variables) to this system. The data required for learning are combinations of the qualitative states (of which there are 6) from the envisionment shown in Table 1.

Method There are two distinct sets of experiments reported here: those based on **noise free** data and those based on **noisy** data. The former assume that the data provided are correct and are used to test the capability of ILP-QSI in handling sparse data. The latter set of experiments captures the situation where the qualitative data may be incorrect because of measurement errors due to noise in the original signal, or through errors introduced in a quantitative to qualitative transformation (which may occur in cases where the original data is numerical).

Noise-free data. We use the following method for evaluating ILP-QSI's system-identification performance from noise-free data:
For the system under investigation:

1. Obtain the complete envisionment from specific values of exogeneous variables.

   (In the particular case of the u-tube discussed in this section there are exogenous variables and the envisionment states are as shown in Table 1, as stated above.)

2. With non-empty subsets of states in the envisionment as training data construct a set of models using ILP-QSI and record the precision of the result.[6] The number of possible non-empty sets of states for the different test scenarios for the u-tube is 63. ($2^N - 1$, where N is the number of states in the complete envisionment)

3. Plot learning curves showing average precision versus size of training data.

Noisy data. We use the following method for evaluating ILP-QSI's system-identification performance from noisy qualitative data:
For the system under investigation:

1. Obtain the complete envisionment from specific values of exogeneous variables.

2. Replace non-empty subsets of states in the envisionment with randomly generated noise states. With each such combination of correct and random states, as training data construct a set of models using ILP-QSI and record the precision of the result.[7] Given a complete envisionment of N states, replacing a random subset $k > 0$ of these with random states will result in a "noisy" envisionment consisting of $N - k$ noise-free states and k random states. As with Step 2 for noise-free data, an exhaustive replacement of all possible subsets of the complete envisionment with random states will result in $2^N - 1$ noisy test sets.

3. Plot learning curves showing average precision versus size of training data.

2.4.3 Results

The results of performing these experiments, showing the precision of learning the target model versus the number of states used (for both noise-free and noisy data) are shown in Fig. 8. It is evident that for both situations precision improves with the number of states used and that the results from the experiments with noisy data have lower precision than those with the noise-free data (though the curves have the same general shape). Both these results are as one would expect.

With noise-free data we find that it was not possible to identify the target model using just one state as data. However it was possible to identify the target model using pairs of states in 53% of cases. These states are:

$$[2, 3], [2, 4], [2, 5], [3, 5], [3, 6], [4, 5], [4, 6], [5, 6]$$

We refer to these as *Kernel sets*. For the time being we merely report this finding and delay a discussion of its significance until after reporting the results for the experiments on the other systems in the class.

6. This is the proportion of the models in the result that are equivalent to the correct model. Thus, for each training data set, the result returned by ILP-QSI will have a precision between 0.0 and 1.0. The term *precision* as used here has the meaning usually associated with it in the Machine Learning community rather than that familiar in Qualitative Reasoning.
7. As with the non-noisy data, for each training data set, the result returned by ILP-QSI will have a precision between 0.0 and 1.0.

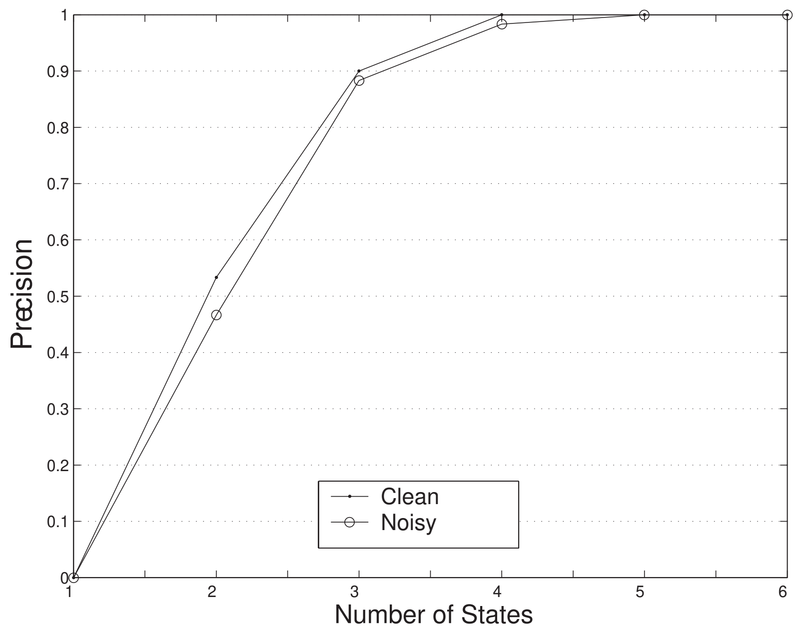


Figure 8: Precision of models obtained for the u-tube.

3. Experiments on Other Systems

In this section we present the same experimental setup applied to a number of other systems: coupled tanks, cascaded tanks and a mass spring damper. These systems are representative of a class of system appearing in industrial contexts (e.g. the cascaded tanks system has been used as a model for diagnosis of an industrial Ammonia Washer system by Warren et al., 2004) as well as being useful analogs to metabolic and compartmental systems.

In each case the experimental method is identical to that utilised for the u-tube as described in Section 2.4. For each system we give a description of the system and the target model, the envisionment associated with the system, a statement of the data used in the experiments, and a summary of the results obtained from the experiments.

3.1 Experimental Aim

For three physical systems: coupled tanks, cascaded tanks and mass-spring-damper (a well known example of a servomechanism), investigate the model identification capabilities of ILP-QSI using qualitative data that are subject to increasing amounts of noise and are made increasingly sparse.

3.2 Materials and Method

Data Qualitative data available consist of the complete envisionment arising from specific values for input variables. The precise details of the data are given with each experiment.

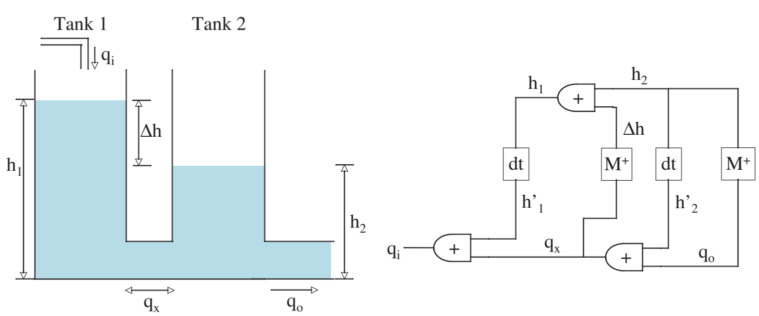
Method The method used is the same as that for the u-tube and described in Section 2.4.

State	h_1	h_2	q_x	q_o
1	$< 0, std >$	$< 0, std >$	$< 0, std >$	$< 0, std >$
2	$< 0, inc >$	$< (0, \infty), dec >$	$< (-\infty, 0), inc >$	$< (0, \infty), dec >$
3	$< (0, \infty), dec >$	$< 0, inc >$	$< (0, \infty), dec >$	$< 0, inc >$
4	$< (0, \infty), dec >$	$< (0, \infty), inc >$	$< (0, \infty), dec >$	$< (0, \infty), inc >$
6	$< (0, \infty), inc >$	$< (0, \infty), dec >$	$< (-\infty, 0), inc >$	$< (0, \infty), dec >$
7	$< (0, \infty), dec >$	$< (0, \infty), std >$	$< (0, \infty), dec >$	$< (0, \infty), std >$
8	$< (0, \infty), std >$	$< (0, \infty), dec >$	$< 0, inc >$	$< (0, \infty), dec >$
9	$< (0, \infty), dec >$	$< (0, \infty), dec >$	$< (0, \infty), dec >$	$< (0, \infty), dec >$
10	$< (0, \infty), dec >$	$< (0, \infty), dec >$	$< (0, \infty), std >$	$< (0, \infty), dec >$
11	$< (0, \infty), dec >$	$< (0, \infty), dec >$	$< (0, \infty), inc >$	$< (0, \infty), dec >$

Table 2: The envisionment states used for the coupled tanks experiments. (The states are labeled to be in accord with those for the u-tube; since state 5 in the u-tube does not appear in the coupled tanks envisionment there is no state labeled ‘5’ in this table.)

3.3 The Coupled Tanks

This is an open system consisting of two reservoirs as shown in Fig. 9. Essentially, it can be seen as a u-tube with an input and an output. The input, q_i, flows into the top of tank 1 and the output, q_o, flows out of the base of tank 2 (see Fig. 9).



DERIV(h_1,h_1'),
DERIV(h_2,h_2'),
ADD(h_2,$Delta\_h$, h_1),
M$^+$($Delta\_h$,q_x),
M$^+$(h_2,q_o),
ADD(h_2',q_o,q_x),
ADD(q_x,h_1',q_i).

Figure 9: The coupled tanks: (left) physical; (middle) QSIM diagram; (right) QSIM relations.

In these experiments we assume that we can observe: q_i, q_x, h_1, h_2, and q_o. Thus system identification must discover a model with three intermediate variables, h_1', h_2' and Δh.

Data There is one exogenous variable, namely the flow of liquid into tank 1 (q_i). In the experiments described here the input flow is kept at zero (that is, $q_i = \langle 0, std \rangle$), making the system for this particular case just moderately more complex than the u-tube. The complete envisionment consists of 10 states, as shown in Table 2 and Fig. 10, which means there are 1024 experiments in this set.

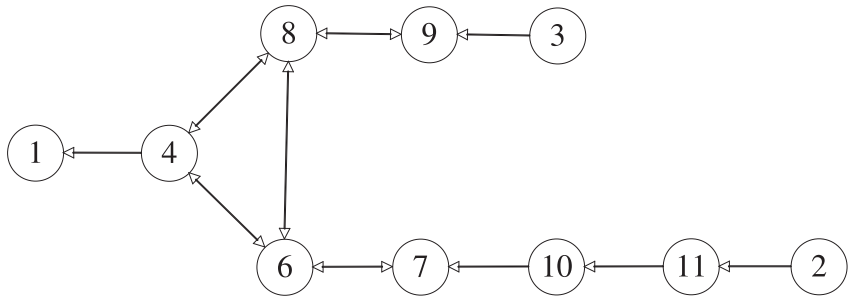


Figure 10: Coupled tanks envisionment graph.

3.3.1 Results

The precision graphs for the coupled tanks experiments are shown in Fig. 11. Here again results show the improvement in precision as the number states used increases and also the deterioration in precision when noise is added. The effect of noise is worse when fewer states are used than was the case for the u-tube, though its effect is nullified when all states are used.

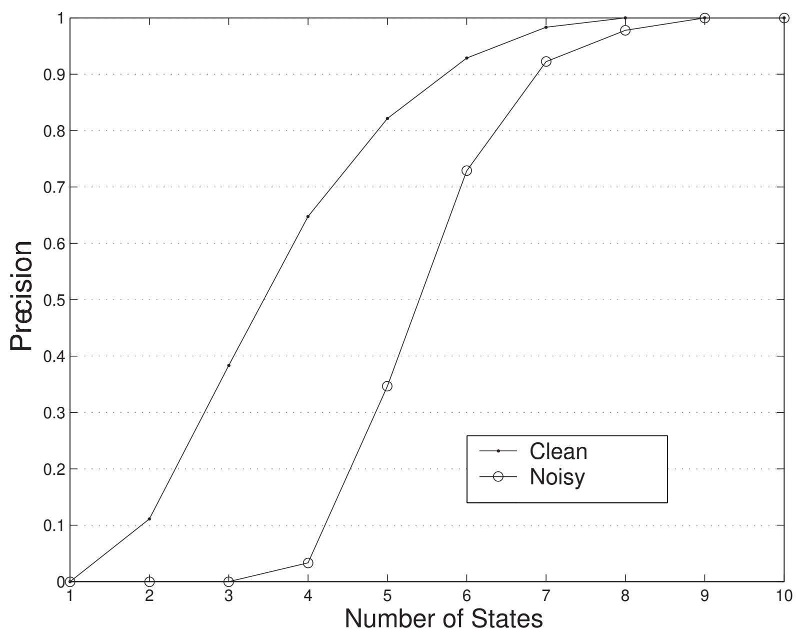


Figure 11: Coupled tanks precision graphs.

For the noise free data it was again not possible to identify any models using a single datum but utilising pairs of states yielded the target model in 11% of cases. The relevant pairs of states (kernel sets) are:

$$[2,7], [3,8], [4,8], [6,7], [7,8]$$

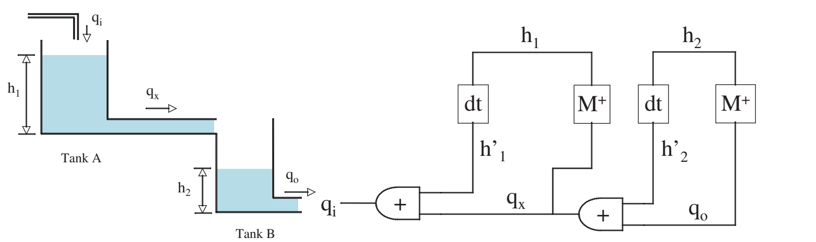
Whereas in the u-tube experiments all the states in which the target model was successfully learned were supersets of the pairs, in the coupled tanks case there are sets of three states (which are not supersets of the pairs listed above) that result in successful identification of the target model:

$$\begin{gathered}[2,3,9],[2,3,10],[2,3,11]\\ [2,4,9],[2,4,10],[2,4,11]\\ [3,6,9],[3,6,10],[3,6,11]\\ [4,6,9],[4,6,10],[4,6,11]\end{gathered}$$

3.4 Cascaded Tanks

This system is also an open system. However, flow through the system is always uni-directional (unlike the coupled tanks system). In principle, the system can be broken into two sub-systems each containing one reservoir, each with their own input and output.

An example of the system is shown in Fig. 12. Liquid flows into tank 1, and then uni-directionally from tank 1 into tank 2. As is apparent from the figure, the flow is into the top of tank 1 and out of the base of tank 2.



DERIV(h_1,h_1'),
DERIV(h_2,h_2'),
M$^+$(h_1,q_x),
M$^+$(h_2,q_o),
ADD(h_2',q_o,q_x),
ADD(q_x,h_1',q_i).

Figure 12: The cascaded tanks: (left) physical; (middle) QSIM diagrammatic; (right) QSIM relations.

We assume that we can observe: q_i, h_1, h_2, and q_x. Thus system identification must discover a model with two intermediate variables, h_1' and h_2'. The numbered list of states (or complete envisionment) for this case is shown in Fig. 13 and Table 3.

Data There is one exogenous variable, namely the flow of liquid into tank 1 (q_i). We increase the complexity by allowing a steady positive input flow (that is, $q_i = \langle(0,\infty), std\rangle$). The complete envisionment consists of 14 states, as shown in Fig. 13 and Table 3 which means 16,383 experiments are required .

3.4.1 Results

The precision graphs for the cascaded tanks are shown in Fig. 14. The graphs are similar in shape to the coupled systems, but showing generally lower precision with noisy-data. Further examination shows that we are unable to identify the target model from fewer than three states. The subset triples (which form the kernel sets in this case) from which the

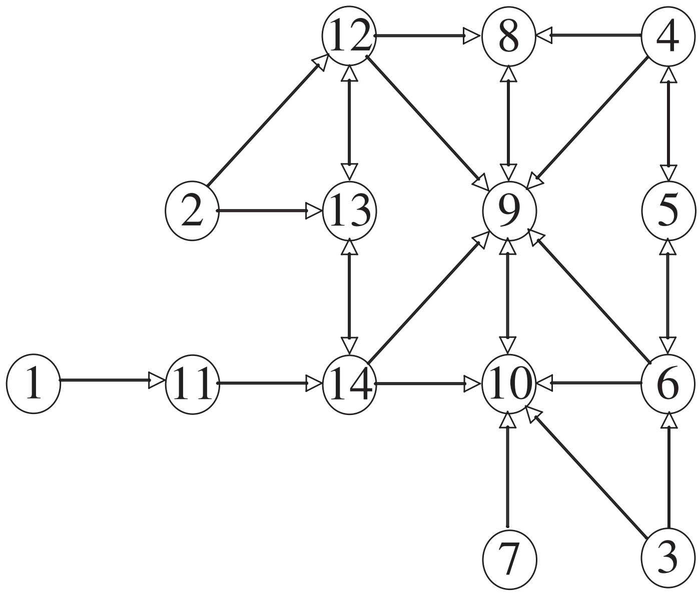


Figure 13: Cascaded tanks envisionment graph.

State	h_1	h_2	q_x	q_o
1	$< 0, inc >$	$< 0, std >$	$< 0, inc >$	$< 0, std >$
2	$< 0, inc >$	$< (0, \infty), dec >$	$< 0, inc >$	$< (0, \infty), dec >$
3	$< (0, \infty), dec >$	$< 0, inc >$	$< (0, \infty), dec >$	$< 0, inc >$
4	$< (0, \infty), dec >$	$< (0, \infty), dec >$	$< (0, \infty), dec >$	$< (0, \infty), dec >$
5	$< (0, \infty), dec >$	$< (0, \infty), std >$	$< (0, \infty), dec >$	$< (0, \infty), std >$
6	$< (0, \infty), dec >$	$< (0, \infty), inc >$	$< (0, \infty), dec >$	$< (0, \infty), inc >$
7	$< (0, \infty), std >$	$< 0, inc >$	$< (0, \infty), std >$	$< 0, inc >$
8	$< (0, \infty), std >$	$< (0, \infty), dec >$	$< (0, \infty), std >$	$< (0, \infty), dec >$
9	$< (0, \infty), std >$	$< (0, \infty), std >$	$< (0, \infty), std >$	$< (0, \infty), std >$
10	$< (0, \infty), std >$	$< (0, \infty), inc >$	$< (0, \infty), std >$	$< (0, \infty), inc >$
11	$< (0, \infty), inc >$	$< 0, inc >$	$< (0, \infty), inc >$	$< 0, inc >$
12	$< (0, \infty), inc >$	$< (0, \infty), dec >$	$< (0, \infty), inc >$	$< (0, \infty), dec >$
13	$< (0, \infty), inc >$	$< (0, \infty), std >$	$< (0, \infty), inc >$	$< (0, \infty), std >$
14	$< (0, \infty), inc >$	$< (0, \infty), inc >$	$< (0, \infty), inc >$	$< (0, \infty), inc >$

Table 3: The envisionment states used for the cascaded tanks experiments.

target model was identified are:

$$[1, 3, 4], [1, 3, 5], [1, 3, 8], [1, 3, 9], [1, 7, 4], [1, 7, 5], [1, 7, 8], [1, 7, 9]$$

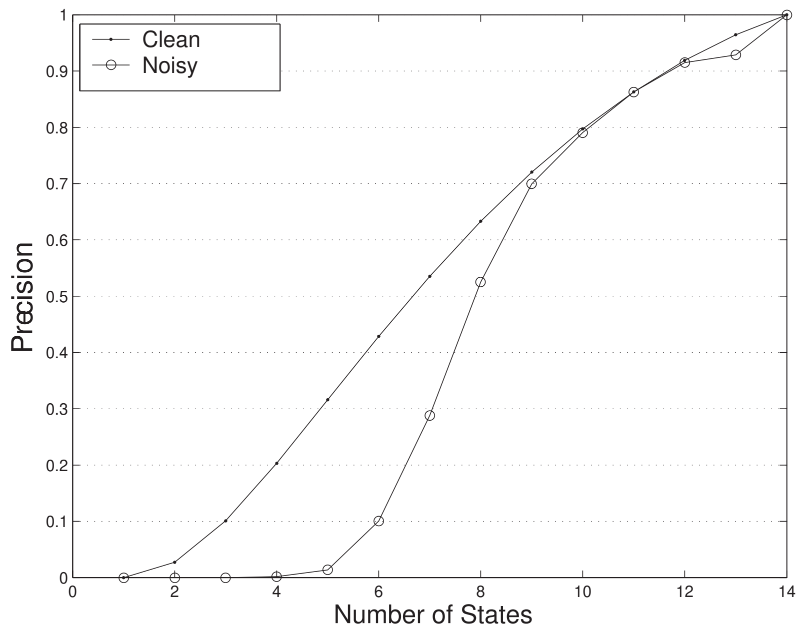
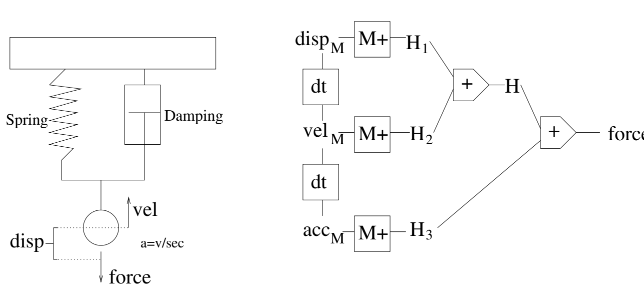


Figure 14: Cascaded tanks precision graph.

3.5 Mass-Spring-Damper

The final physical system considered is an abstraction of a wide variety of servomechanisms with a displacing force. An example of the system is shown in Fig. 15. In this situation, a mass is held in equilibrium between two forces. If the equilibrium is disturbed, oscillatory behaviour is observed. The motion of the mass is damped so that the oscillations do not continue indefinitely, and will eventually return to the original equilibrium position. If an external force is applied (for example pulling the mass down) its final resting place will be displaced from the natural equilibrium point (see Fig. 15). The mass M has displacement $disp_M$ from its rest position, and at any time, t, it is moving with velocity vel_M and accelerating at rate acc_M. We assume that we can observe the variables: $disp_M$, vel_M, acc_M,



DERIV($disp_M$,vel_M),
DERIV(vel_M,acc_M),
M$^+$($disp_M$,H_1),
M$^+$(vel_M,H_2),
M$^+$(acc_M,H_3),
ADD(H_1,H_2,H_4),
ADD(H_3,H_4,$force$).

Figure 15: The spring system (a) physical; (b) QSIM diagrammatic; (c) QSIM relations

and $force$. Qualitative system identification must now find a model with four intermediate variables, H_1, H_2, H_3 and H_4; as well as a intermediate relation ADD(H_1,H_2,H_4), between three of these variables. The input force, *force*, is exogenous. In the experiments here, we only consider the case where there is a steady force being applied to the system (that is, $Force_A = \langle (0, \infty), std \rangle$). The complete envisionment for this case is shown in Fig. 16, where the equilibrium point is represented by state 2. The precision graphs are shown in

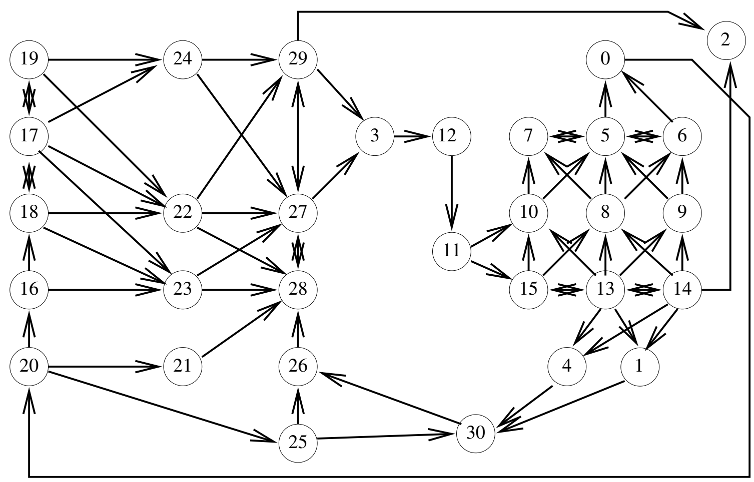


Figure 16: Mass-spring-damper envisionment graph

Fig. 17. For this system the envisionment contains 31 states, which makes exhaustive testing unrealistic. Instead sets of clean and noisy states were randomly selected from the space of possible experiments. Nonetheless it can be observed that the average precision graphs are in-line with those obtained for the tanks experiments. However, the actual precision values suggest that both sparse data and noise have less of an effect here than the other systems. This may be due to the tight relationship between the two derivatives in the spring model, making the system extremely constrained.

3.6 Discussion of Results

An inspection of the experimental results reveals an expected pattern: in all cases the precision curves (for both noisy and noise free experiments) have the same general shape. Experiments which utilise fewer states identify the target model less often than when a greater number of states are used. However, a closer examination of the results reveals that even when few states are used (pairs or triples) the target model may be consistently found when particular combinations of states are used. In order to understand why this is so requires us to look at the solution spaces for the systems concerned.[8]

We will examine the u-tube and coupled tanks together because they are very closely related systems and both had zero input. The cascaded tanks system is slightly different and had a non-zero input and so will be discussed later in the section.

8. We do not discuss the spring system here because of its complexity.

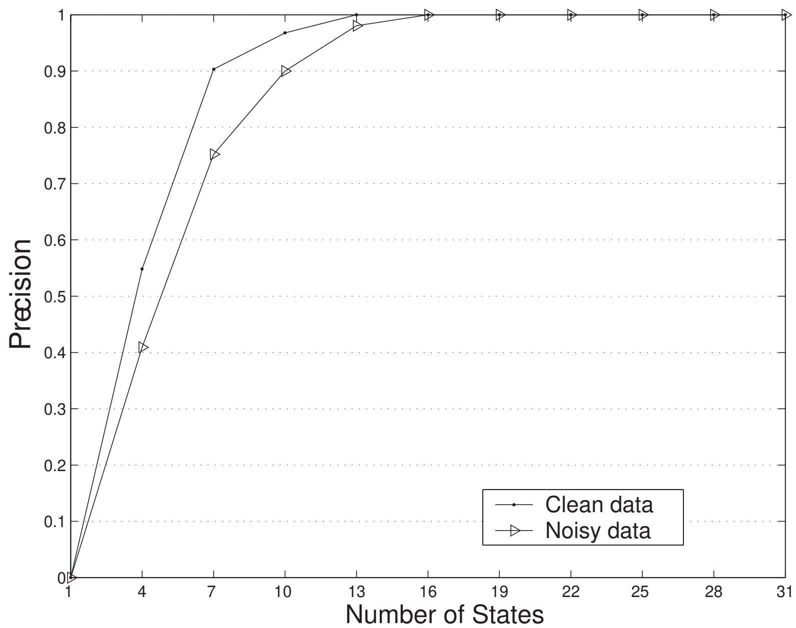


Figure 17: Mass-spring-damper precision graph

3.6.1 The U-tube and Coupled Tanks

The bare results, while interesting, do not give any indication of why the particular pairs or triples highlighted should precisely identify the target model. In order to ascertain 'why?' we must examine the envisionment states given in Tables 1 and 2, from these we can itemise the relevant features of the sets of states as follows:

- For both the u-tube and coupled tanks there is at least one critical point in each pair.
- For both these systems each pair of states contains one state for each branch in the envisionment graph (Fig. 2 & Fig. 10); and of these at least one is at the extreme of its branch. That is, states where one tank is either empty or the state immediately succeeding this, and the other tank is relatively full so that that the derivatives for that height in each tank have opposite signs.
- For both systems all supersets of these minimal sets will precisely learn the target model.

These observations lead us to suggest that for coupled systems the ability of the learning system to identify the structure of a model is dependent on the data used including the critical points and on having data that covers all the different types of starting point that the system behaviours can have. This is in keeping with what systems theory would lead us to expect (Gawthrop & Smith, 1996).

In order properly to appreciate what is indicated by these kernel sets and the relation of the systems to each other we need to look at the solution spaces (Coghill et al., 1992;

Coghill, 2003) for the two systems. These are shown in Fig. 18 (and their derivation is similar to that given in Section 2.2 and detailed in Appendix A). From these we can get a clear picture of where the kernel pairs and triples lie with respect to the critical points of the system.

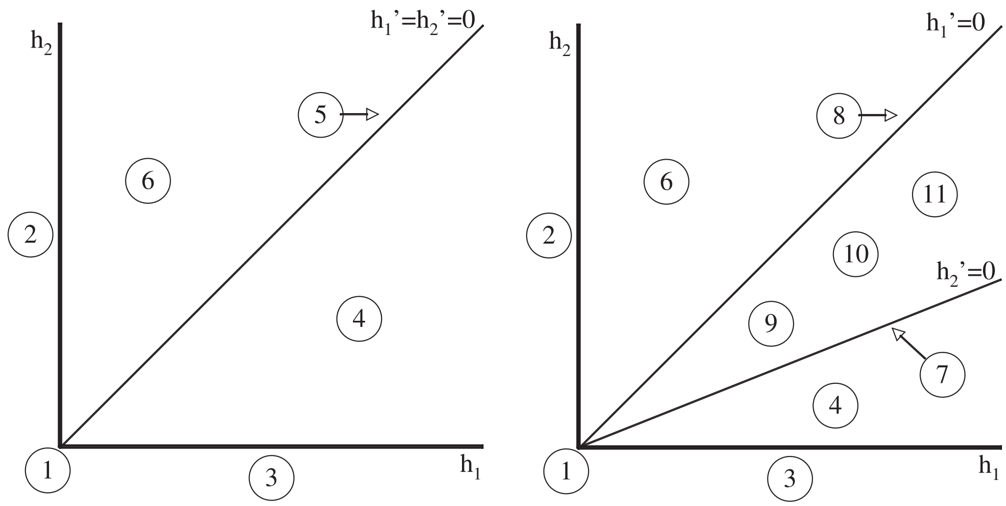


Figure 18: The solution spaces for the u-tube and coupled tanks systems.

From the system diagrams provided in Fig. 1 and Fig. 9 it can be seen that the u-tube and coupled tanks systems differ only in the fact that the coupled tanks has an outlet orifice, whereas the u-tube does not. This accounts for the major difference in their solution spaces; namely that the coupled tanks has two critical points (states 7 and 9) whereas the u-tube has only one (state 5) – which is actually the steady state. This gives rise to the additional states: 9, 10 and 11 which lie between the critical points.[9] It can be observed that as the outlet orifice from tank 2 in the coupled tanks system decreases in size the space between the isoclines in the solution space will become narrower until it disappears when the orifice closes. This can be seen more formally by comparing equations 6 and 10 in Appendix A. There it is clear that as k_2 approaches zero, equation 6 approximates equation 10 (and when $k_2 = 0$ the two equations are the same).

If we now look again at the sets of pairs we can observe that they are related in ways that reflect the relationship between the two coupled systems. Firstly, looking at the pairs. For the u-tube there are 4 pairs which include the critical point (steady state), state 5: [2, 5], [3, 5], [4, 5], and [5, 6]. Now noting from the discussion above that state 5 in the u-tube relates to either of states 7 or 8 in the coupled tanks then we find that the analogous pairs exist in the kernel set for the coupled tanks: [2, 7], [3,8], [4, 8], and [6, 7]. This leaves one pair from the coupled tanks pairs unaccounted for: [7, 8]. However, this is no surprise since that pair is taken to map to state 5 in the u-tube; and it is the consistent finding that no singleton state is sufficient to learn a model of the system.

9. There are three states here because they differ only in the magnitude of the q_x or $qdir$ of h'_1 and h'_2, neither of which appear explicitly in the solution space. Readers may convince themselves of this by comparing Table 1 with the envisionment in Table 2.

There are still 4 pairs in the u-tube experiments from which we are able to learn reliably the target model that do not have a corresponding coupled tanks pair. These are: [2, 3], [2, 4], [3, 6], and [4, 6]. A comparison with the triples for the learning of the coupled tanks model reveals that these states are the pairs which are conjoined with either state 9, 10 or 11 to make up the triples. The inclusion of these states warrants further explanation since they are the states which distinguish the closed u-tube from the open coupled tanks. In all three of these states the state variables both have the value $\langle(0, \infty), dec)\rangle$; a situation that cannot occur in the u-tube. Combining this with the fact that the four pairs listed above do not contain a critical point and are qualitatively identical in both systems leads one to the conclusion that the additional information contained in these triple kernel sets enables one to distinguish between the u-tube and coupled tanks in such a case.

These results extend, strengthen and deepen those reported in Coghill et al. (2004) and Garrett et al. (2007).

3.6.2 The Cascaded Tanks

The cascaded tanks system is asymmetrical with the flow only being possible in one direction. The fact that the input is a positive steady flow makes the setup marginally more complex in that regard than for the coupled systems, where there was no input flow.

The kernel sets from which a model of this system may be learned (presented in Section 3.4.1 are depicted schematically in Fig. 19 in order to explain the results obtained. If we look at the first and middle columns of this diagram and ignore, for the time being, the downstream tank, we can see that what is represented are two pairs of states: the tank empty combined with the tank being at steady state, or the tank empty combined with the state where the amount of fluid in the tank is greater than steady state. We have confirmed experimentally that these are kernel sets from which a single tank model can be learned.

If we now ignore the upstream tank (apart from its outflow) and examine the middle and third columns of the diagram we can see that these divide into two groups according to whether the input to the downstream tank is steady or decaying (positive and decreasing). For each of these there are two pairs of states, which are the same as for the upstream tank: the tank empty combined with the tank being at steady state, or the tank empty combined with the state where the amount of fluid in the tank is greater than steady state. In the case of this tank it can be seen that the cross product of states appear in the kernel sets because each case represents a valid possible situation.

These results lead to two major conclusions with regard to the cascaded tanks system:

1. ILP-QSI effectively identifies the individual components of the cascade and combines them through the cascade point.

2. The situation with the downstream tank, where the input was either a steady flow or a decreasing flow, indicates that utilising a variety of inputs can aid in the identification process.

The former conclusion may serve as a pointer to the possibility of incremental learning of cascaded systems.

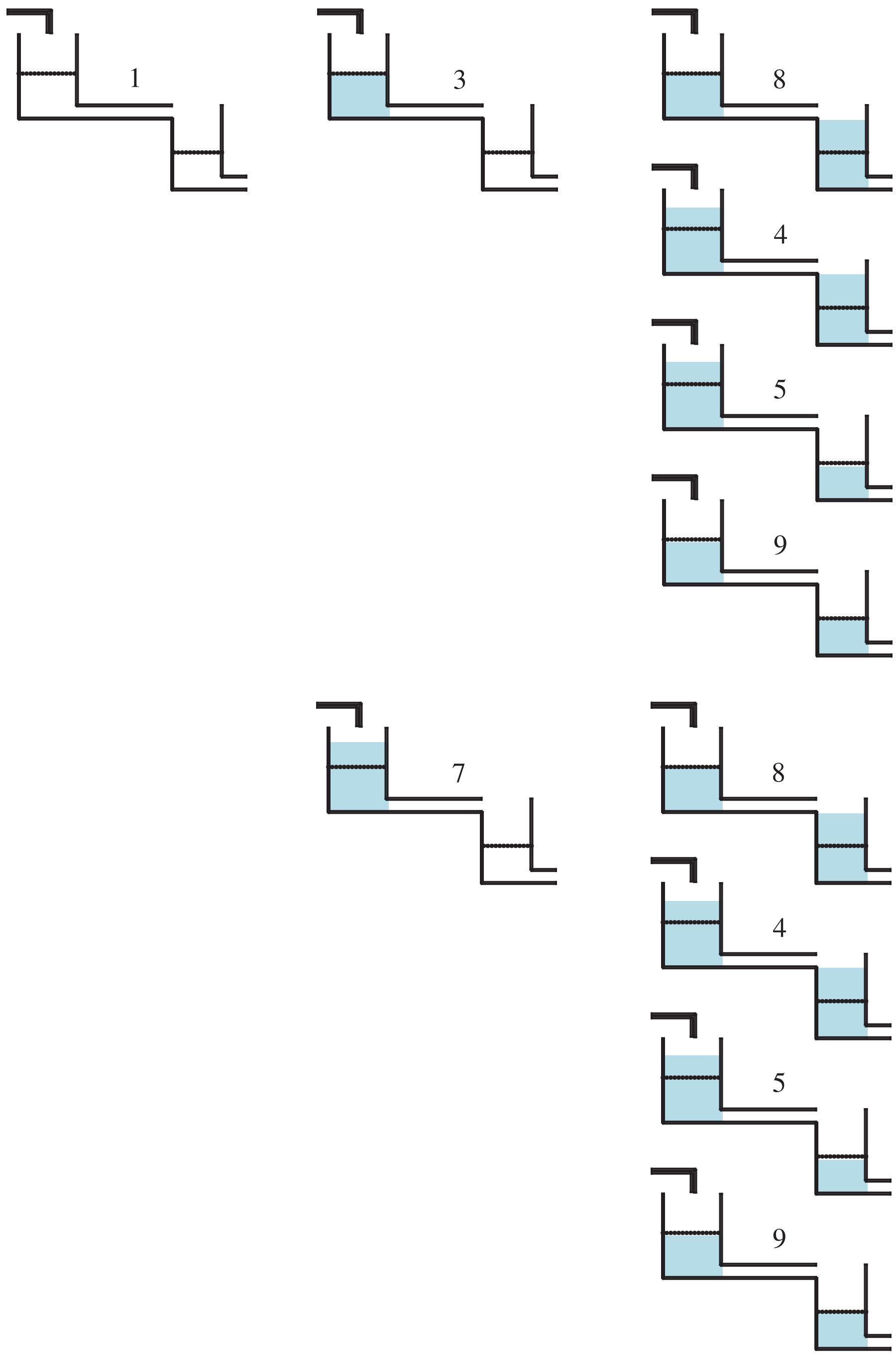


Figure 19: A schematic representation of the triples of states from which the target model for the cascaded tanks systems is learned.

4. Experiments with Quantitative Data

This part of the experimental testing of our system is a " proof-of-concept" test. As has been stated our system has been designed to learn qualitative models from qualitative data. As such it is assumed that the conversion of any quantitative data has already been performed, not least because the needed qualitative data analysis would require another research project and is beyond the scope of this paper. However, in order to test the usability of our system with quantitative data and to test its ability to go through the whole process from receiving the data to producing the model, we have implemented a rudimentary data analysis package to facilitate this. Of course this is not exhaustive, but it will permit us to test the results produced via such a process for consistency with those produced from the experiments with qualitative data.

4.1 Experimental Aim

Using the four physical systems, investigate the model identification capabilities of ILP-QSI using numeric traces of system behaviour that are subject to increasing amounts of noise.

4.1.1 Quantitative to Qualitative Conversion

Before proceeding to describe the experiments carried out, we present the method used to convert numerical data into the qualitative form required by ILP-QSI because this is utilised in each set of experiments.

We have adopted a straightforward and simple approach to performing the conversion. For a quantitative variable x, values at N real-valued time series steps were numerically differentiated by means of a central difference approach (Shoup, 1979) such that,

$$\left.\begin{array}{rcl} \frac{dx_i}{dt} & = & \frac{(x_i - x_{i-1}) + (x_{i+1} - x_i)}{2} \\ \frac{d^2x_i}{dt^2} & = & (x_i - x_{i-1}) - (x_{i+1} - x_i) \end{array}\right\} i = 2 \cdots N-1$$

A quantitative variable x is converted into a qualitative variable $q = \langle qmag, qdir \rangle$, where $qmag \in \{$(-∞,0), 0, (0,∞)$\}$ is generated from x, and $qdir \in \{dec, std, inc\}$ is generated from dx/dt. The qualitative derivative of q, $\dot{q}$, is obtained in a similar manner but is generated from from dx/dt and d^2x/dt^2 respectively.

The data are typically noisy—either inherently, or because of the process of differentiation—and we perform some simple smoothing of the first and second derivatives using a Blackman filter (a relative of the moving average filter – see Blackman & Tukey, 1958). In each case, the filter is actually applied to the result of a Fast Fourier Transform (FFT) and the result obtained by taking the real part of the inverse FFT. We note here that this form of smoothing is appropriate only when a sufficient number of time steps are present.

Having obtained a (smoothed) numerical value x_i for variable x at instant i, its qualitative magnitude $qmag(x_i)$ is, in principle, simply obtained by the following:

$$qmag(x_i) = \begin{cases} (-\infty, 0) & \text{if } x_i < 0 \\ 0 & \text{if } x_i = 0 \\ (0, +\infty) & \text{otherwise} \end{cases}$$

In practice, since floating-point values are unlikely to be exactly zero, we have found it advantageous to re-apply the filtering process to data straddling zero to eliminate small fluctuations around this value. Despite these measures, in addition to generating correct qualitative states (true positives) the conversion can produce errors: states generated may not correspond to true states (false positives); and some true states may not be generated (false negatives). Fig. 20 shows an example of this (the problem is, of course, exacerbated further if the original quantitative data are noisy). The reason for these imperfect results from noise free quantitative data are twofold: one is the smoothing process on small fluctuations around zero; but the main reason is that, as discussed above, creating a full qualitative state involves numerical differentiation which introduces noise into the data for the derivatives that affects the ability of the process to convert from quantitative to qualitative with absolute accuracy.

System	True States	Generated States	True Positives	False Positives	False Negatives
u-tube	6	6	4	2	0
Coupled	10	14	6	8	0
Cascaded	14	8	5	3	6
Spring	33	33	20	13	0

Figure 20: An example of the errors resulting from generating qualitative states from traces of system behaviour. Here, the traces were generated by the following initial conditions: $h_1 = 2.0, h_2 = 0.0$ for all three tank systems; and $disp_M = 2.0$, $vel_M = 0.0$ for the spring.

4.2 Materials and Method

Numerical simulations of the four physical systems were constructed using the same general relations as the qualitative models. Once again the experiments were carried out utilising both noise free and noisy data, as described in the rest of this section.

4.2.1 Data

The models used for the numerical simulations had the same structure as the qualitative models, but with the substitution of a real valued parameter for each monotonic function relation. This gives a linear relation between the two variables; more complex, non-linear functions might have been used, but linear functions provided a suitable approximation of the known behavior of these systems, as shown graphically in Fig. 21 (a)–(d); which is as much as is required for this proof-of-concept study.

For a given set of function parameter values, initial conditions, and input value, a quantitative model produces a single quantitative behaviour (this contrasts with qualitative models that can produce a list of all possible behaviours for the model). Parameter values were chosen so that the models approached a steady state during the time period of the test. The models were implemented in Matlab 5.3 using the ODE15s ODE solver. Each

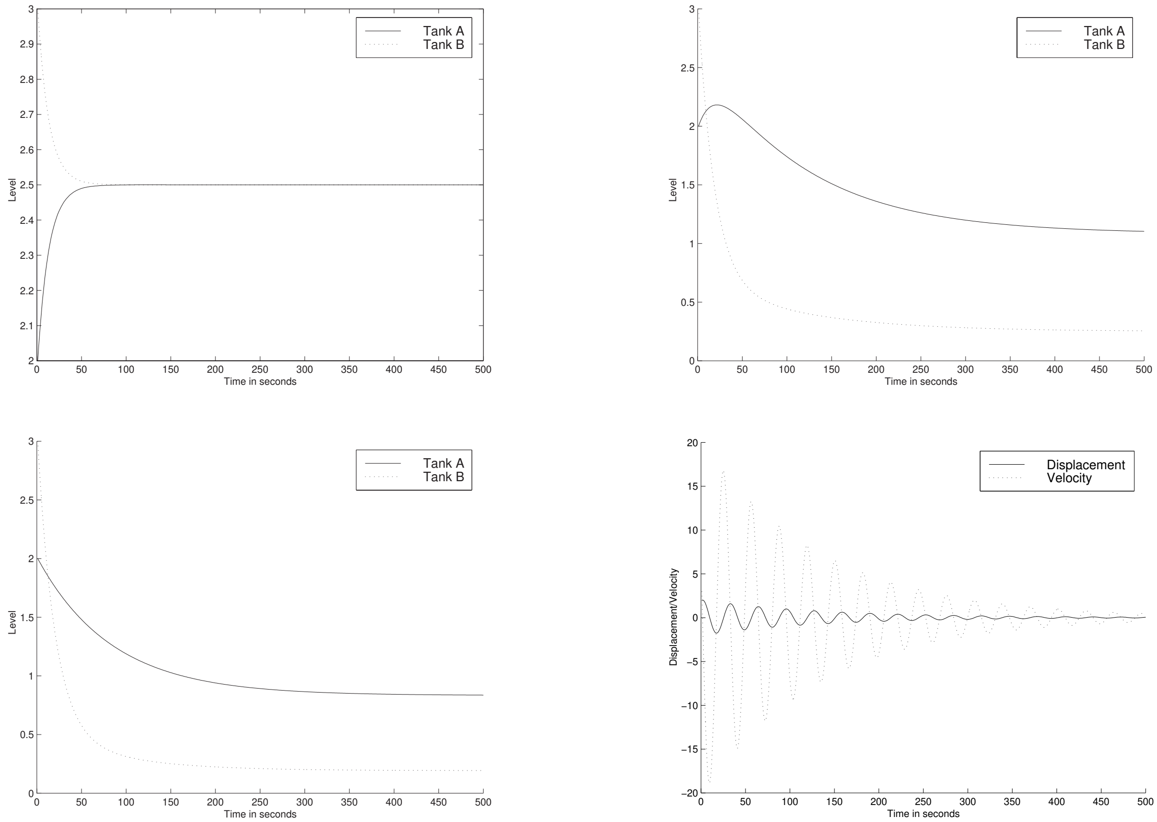


Figure 21: A graph of example numeric behavior of (a) the u-tube, top left; (b) the coupled tanks, top right; (c) the cascaded tanks, bottom left, and (d) damped spring, bottom right.

time point generated by the simulation was made available as part of the sampled data. This ensures that the sampling rate is suitably fast with respect to the Nyquist criterion. It also guarantees that a sufficient number of data points are available as required by the Beckman filter.

4.2.2 Method

Noise-free data. We use the following method for evaluating ILP-QSI's system-identification performance from noise-free data:

For each of the four test-systems:

(a) Obtain the system behaviour of the test system with a number of different initial conditions and input values. Convert each of these into qualitative states using the procedure in Section 4.1.1.

(b) Using *all* the qualitative states obtained as training data construct a set of models using ILP-QSI and record the precision of the result (this is the proportion of the models in the result that are equivalent to the correct model). Thus, for each training data set, the result returned by ILP-QSI will have a precision between 0.0 and 1.0.

The following details are relevant: (a) The quantitative models were each put into three separate initial conditions, so that the magnitude of the two state variables were set to $(2, 0)$, $(0, 3)$ and $(2, 3)$. Specifically, these were the initial values of the two tank levels for all three tank systems, and the displacement and velocity for the spring. These values were not crucial but were chosen for the initial conditions because they caused the numerical models to converge on the steady state for each system in a reasonable number of iterations; (b) Each initial condition gave rise to a system behaviour and hence a set of qualitative states. In the second step above, qualitative states from *all* behaviours is used as training data. This is because kernel subsets necessary for correct system identification usually contain qualitative states from multiple quantitative behaviours. (c) The conversion process results in erroneous qualitative states (see Section 4.1.1). Thus, the training data used here contain both false positives and false negatives.

Noisy data. We use the following method for evaluating ILP-QSI's system-identification performance from noisy quantitative data:

For each of the four test-systems:

(a) Obtain the system behaviour of the test system with a number of different initial conditions and input values.

(b) Corrupt each system behaviour with additive noise;

(c) Convert each corrupted behaviour into qualitative states using the procedure in Section 4.1.1.

(d) Using *all* the qualitative states obtained as training data construct a set of models using ILP-QSI and record the precision of the result (this is the proportion of the models in the result that are equivalent to the correct model). Thus, for each training data set, the result returned by ILP-QSI will have a precision between 0.0 and 1.0.

In the second step noise was added to the numerical data sets as follows. A Gaussian noise signal (with a μ of 0.0 and σ of 1.0) was generated using by the built-in Matlab function `normrnd` and scaled to three orders of magnitude of the original noise, namely 0.01, 0.1 and 1.0 (representing "low", "medium" and "high" amounts of noise respectively). These scaled noise variants were added to the numerical values of the system behaviour obtained from each initial condition.

4.3 Quantitative Experimental Results

The process of converting from quantitative to qualitative states introduces errors, even for noise free data. Table 4 shows the proportion of correct qualitative states to the total number of qualitative states that were obtained from the numerical signal, including noisy states. The table shows this proportion for all four systems, under the different degrees

		Initial States		
Model	Noise level	(2,0)	(0,3)	(2,3)
u-tube	0	4/6	4/6	3/5
	0.01	2/8	2/8	2/9
	0.1	2/10	2/10	2/15
	1	2/37	2/37	2/53
Coupled	0	6/14	5/13	5/14
	0.01	6/16	4/14	4/12
	0.1	6/16	5/25	4/15
	1	6/58	6/61	4/46
Cascaded	0	5/8	5/8	3/8
	0.01	4/10	4/12	2/9
	0.1	4/17	4/21	2/13
	1	4/39	4/49	4/38
Spring	0	20/33	19/35	22/39
	0.01	23/48	20/36	18/41
	0.1	23/49	20/38	18/44
	1	20/65	20/52	19/53

Table 4: The input data for the numeric experiments, described as the proportion of the number of clean states / the total number of converted states for different systems and degrees of noise.

of noise. The numerical simulations were not intended to be exhaustive and do not cover every possible such behaviour; so it is not surprising to observe that there is no case where all the states of the complete envisionment are generated.

The results of the qualitative experiments detailed in the previous section indicate that in order successfully to learn the target model data from all branches of the envisionment are required, and the greater the number of such states used the greater the liklihood of learning the target model structure. Therefore in these experiments we utilised all the states generated from the numerical simulations.

The results from the numerical data experiments are shown in Fig. 22. These experiments show that it is possible to learn models from clean and noisy numerical data even when the qualitative states generated from the clean numerical data contain a number of unavoidable data transformation errors. The results for each of the systems used are as follows:

The spring system This system has 31 states in the complete envisionment and Table 4 shows that the quantitative to qualitative conversion process yields around 20 of those states. It can be seen from Fig. 17 that learning from 20 states gives 100% precision in learning this target model, even in the presence of noise. It is not surprising, therefore, to find that the learning precision is perfect up to the highest noise level. Since the qualitative experiments were done by sampling, the slight downturn at the highest noise level could be due to the large number of noisy states generated in this

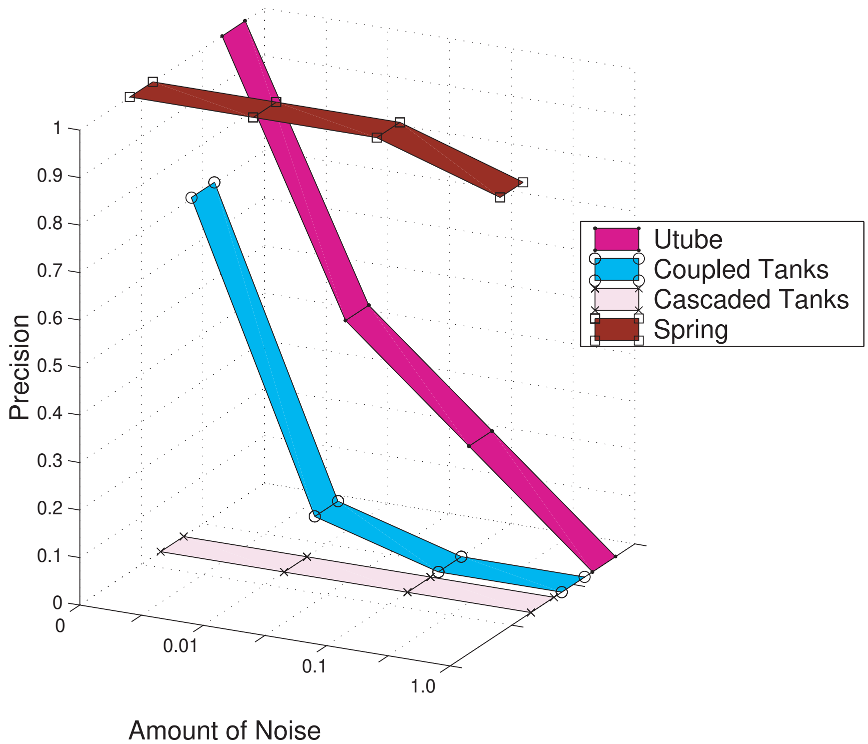


Figure 22: A comparison of the results from the numerical learning tests, averaged over all three initial conditions. Tests that attempted to learn from few states, such as the cascaded tanks, are more likely to fail than those that have large numbers of states, such as the spring. This is consistent with the kernel subset principle introduced in Section 2.4 since model-learning is not precise without the presence of certain key states in the input data.

experiment. Hence we can say that these results are in keeping with the qualitative experiments.

u-tube & coupled tanks The complete envisonments for these systems contain 6 and 10 states respectively. Table 4 shows that the number of true states generated is less than the complete envisionment, and significantly less than the number of noisy states in each case. As one would expect from the results presented in Fig. 8 the u-tube gives better results than the coupled tanks (having a higher proportion of the envisionment states present). Ultimately, the ability to learn the model is completely curtailed by the noise; though sooner in the case of the coupled tanks (which the qualitative experiments show to be more sensitive to the presence of noise). This is consistent with the results of the qualitative experiments.

The cascaded tanks In the qualitative learning experiments all the kernel subset triples had state 0 included. This is the state representing the situation where both tanks are empty to begin with, and is not one of the initial states included in the numerical simulations. Also, a perusal of Fig. 14 reveals that the introduction of noise radically reduces the learning precision, and for 4 states (the average number of true states generated by the qualitative to quantitative conversion process) it is zero. Taking account of all these facts it was to be expected that the cascaded tanks model would not be successfully identified by these experiments. This is again consistent with the findings of the qualitative experiments.

5. Application to Biological System Identification

The work reported thus far has been aimed at demonstrating the viability of ILP-QSI and of identifying the bounds of operation of the approach. In this section we examine scalability of the *method* to identify a complex real world biological network. We use the glycolysis pathway as a test case for identification.

5.1 The Test System: Glycolysis

We chose to study the metabolic pathway of glycolysis as a test case. Glycolysis is one of the most important and ubiquitous in biology, it was also historically one of the first to be discovered, and still presents a challenge to model accurately.

The QSIM primitives are sufficient to model adequately the qualitative behaviour of the glycolysis pathway. There are, however, two problems. First, few biologists would understand such a model, as he or she would reason at a much higher level of abstraction. Second, the computational complexity of the corresponding system identification task for glycolsis (a qualitative model with 100 or more QSIM relations) is, at least currently, intractable. We address both these by modelling metabolic pathways in a more abstract manner using biologically meaningful *metabolic components* (MC) (a similar approach to constructing complex qualitative models of the human heart was used in Bratko, Mozetic, & Lavrac, 1989). Specifically, we note that in metabolic pathways, there are essentially two types of object: metabolites (small molecules) and enzymes (larger molecules that catalyze reactions). We use component models of each of these objects as described below (King, Garrett, & Coghill, 2005).

5.1.1 Modelling Metabolites and Enzymes

The concentrations of metabolites vary over time as they are synthesised or utilised by enzymatically catalysed reactions. As a result their concentration at any given time-point is a function of: (a) their concentration at the previous time-point; and (b) the degree to which they are used or created by various enzyme reactions.

When modelling enzymes, each enzyme is assumed to have at most two substrates and at most two products. If there are two substrates or products these are considered to form a substrate or product complex, such that the amount of the complex is proportional to the amount of the substrates or products multiplied together. This models the probability that both substrates (or products) will collide with the enzyme with sufficient timeliness to be catalysed into the product complex (or substrate complex). The substrate complex is converted into the product complex, which then disassociates into the product metabolites, and vice versa. We shall use the phrase "flow through the enzyme" to denote the amount of substrate complex formed minus the amount of product complex formed. (See the work of Voit and Radivoyevitch (2000) for details of enzyme kinetics.)

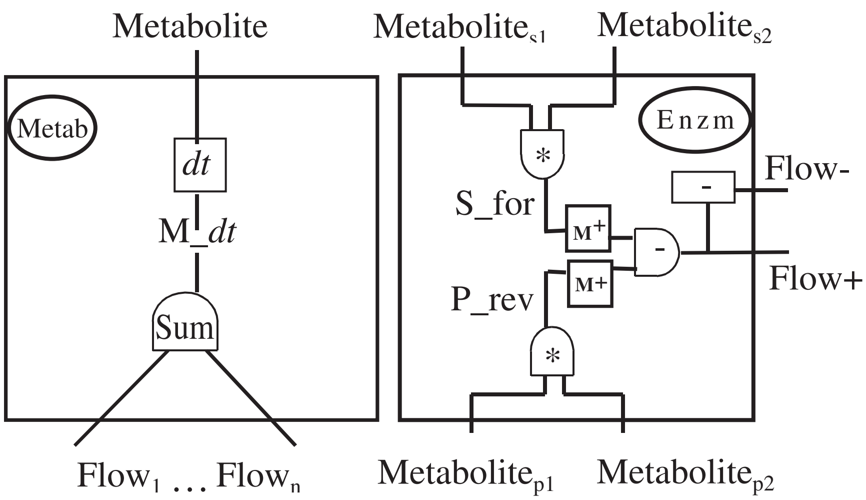


Figure 23: The Metabolic Components (MCs) used in the biological experiments, with the underlying QSIM primitives.

Quantitative, and corresponding qualitative representations of the metabolite and enzymes using QSIM relations, are therefore:

Metabolites:

$$\frac{dM}{dt} = \sum_{i=1}^{n} Flow_i \qquad (4)$$

DERIV($Metabolite$,M_dt),
SUM($Flow_1$, ..., $Flow_n$,
M_dt).

Enzymes:

$$Flow_i = f(\prod_{s=1}^{S} Metabolite_s) - g(\prod_{p=1}^{P} Metabolite_p) \qquad (5)$$

PROD($Metabolite_1$, ..., $Metabolite_s$, *S-for*),
PROD($Metabolite_1$, ..., $Metabolite_p$, *P-rev*),
M+(*S-for*, Ds),
M+(*P-rev*, Dp),
SUB(Ds, Dp, *Flow*),
MINUS($Flow$,$Flow_{minus}$).

Here, S refers to the input metabolites to an enzymatic reaction, or its *substrates*, and P refers to the products of an enzymatic reaction. The `SUM()` and `PROD()` predicates are simply extensions of the `ADD()` and `MULT()` predicates, over any number of inputs. Fig. 23 shows how these constraints are grouped together as metabolic components (MCs). This permits us to create more general constraints representing the *metabolite* and *enzyme* components as follows:

ENZYME((S_1, S_2) (P_1, P_2) $enzymeFlow$)

METABOLITE($metaboliteConc\ metaboliteFlow\ (enzymeFlow_1 \ldots enzymeFlow_n)$)

1.	(*Hexokinase*):	Glc + ATP $\rightarrow$ G6P + ADP.
2.	(*Phosphoglucose isomerase*):	G6P $\rightarrow$ F6P.
3.	(*Phosphofructokinase*):	F6P + ATP $\rightarrow$ F16BP + ADP
4.	(*Aldolase*):	F16BP $\rightarrow$ DHAP + G3P
5.	(*Triose phosphate isomerase*):	DHAP $\rightarrow$ G3P
6.	(*Glyceraldehyde 3-phosphate dehydrogenase*):	G3P + NAD $\rightarrow$ 13BP + NADH.
7.	(*Phosphoglycerate kinase*):	13BP + ADP $\rightarrow$ 3PG + ATP.
8.	(*Phosphoglycerate mutase*):	3PG $\rightarrow$ 2PG.
9.	(*Enolase*):	2PG $\rightarrow$ PEP
10.	(*Pyruvate kinase*):	PEP + ADP $\rightarrow$ Pyr + ATP.

Figure 24: The reactions included in our qualitative model of glycolysis. The reactions that consume ATP and NADH are not explicitly included.

Here the `ENZYME` predicate identifies the substrates and products (the first argument) and returns a single variable representing the flow through the enzyme (the second argument). The `METABOLITE` predicate relates the level and flow of metabolites (the first and second arguments) with the flow through enzymes (the third argument).

5.1.2 Modelling Glycolysis

Using qualitative components representing metabolites and enzymes, we construct a qualitative model of glycolysis. Our model uses 15 metabolites, namely: pyruvate (Pyr), glucose (Glc), phosphoenolpyruvate (PEP), fructose 6-phosphate (F6P), glucose 6-phosphate (G6P), dihydroxyacetone phosphate (DHAP), 3-phosphoglycerate (3PG), 1,3-bisphosphoglycerate (13BP), fructose 1,6-biphosphate (F16BP), 2-phosphoglycerate (2PG), glyceraldehyde 3-phosphate (G3P), ADP, ATP, NAD, and NADH. We have not included H+, H_2O, or Orthophosphate as they are assumed to be ubiquitous (in addition, the restriction of substrates and products to being at most three in number prevents their inclusion).

The qualitative state of glycolysis is defined by the set of qualitative states of the 15 metabolites. Table 5 is a representation of one such qualitative state. To understand this state consider the first entry intended to represent the qualitative state of NAD (that is, NAD concentration: $< 0, \infty), dec >$, and NAD flow: $< (-\infty, 0), dec >$). The meaning of this is that the concentration of NAD is positive $(0, \infty)$ and decreasing (dec), and the rate of change of the concentration of NAD (in analogy to the physical systems, the "flow" of NAD) is negative $(-\infty, 0)$ and decreasing (dec). Similar meanings apply to the other metabolites. Note that metabolic concentrations must be between 0 and ∞; it cannot be negative, and the 0 state is uninteresting.

Using this representation, a possible model for glycolysis is shown in Fig. 25. The model describes constraints on the levels and "flows" of metabolites. Thus, the constraint `enzyme((G3Pc, NADc), (13BPc, NADHc), Enz6f)` states that the flow through enzyme 6 (Enz6f) controls the transformation of the concentrations G3Pc and NADc into the levels 13BPc and NADHc; whereas the constraint `metabolite(NADc, NADc, (Enz6f, -))` states that the

Metabolite	Concentration	Flow
NAD	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
NADH	$<(0,\infty),inc>$	$<(0,\infty),inc>$
ATP	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
ADP	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
Pyr	$<(0,\infty),inc>$	$<(0,\infty),dec>$
Glc	$<(0,\infty),dec>$	$<(-\infty,0),inc>$
PEP	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
F6P	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
G6P	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
DHAP	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
3PG	$<(0,\infty),inc>$	$<(0,\infty),std>$
13BP	$<(0,\infty),std>$	$<0,inc>$
F16PB	$<(0,\infty),inc>$	$<(0,\infty),dec>$
2PG	$<(0,\infty),dec>$	$<(-\infty,0),dec>$
G3P	$<(0,\infty),inc>$	$<(0,\infty),inc>$

Table 5: A legal qualitative state of the 15 metabolites observed during glycolysis.

```
ENZYME((Glcc, ATPc),(G6Pc,ADPc),Enz1f),
ENZYME((G6Pc),(F6Pc),Enz2f),
ENZYME((F6Pc,ATPc),(F16BPc,ADPc),Enz3f),
ENZYME((F16BPc),(G3Pc,DHAPc),Enz4f),
ENZYME((DHAPc),(G3Pc),Enz5f),
ENZYME((G3Pc,NADc),(13BPc,NADHc),Enz6f),
ENZYME((13BPc,ADPc),(3PGc,ATPc),Enz7f),
ENZYME((3PGc),(2PGc),Enz8f),
ENZYME((2PGc),(PEPc),Enz9f),
ENZYME((PEPc,ADPc),(Pyrc,ATPc),Enz10f),
METABOLITE(ATPc,ATPf, (Enz10f +), (Enz7f, +),(Enz1f, -),(Enz3f, -)),
METABOLITE(ADPc,ADPf,(Enz1f, +),(Enz3f, +),(Enz10f, -)(Enz7f, -)),
METABOLITE(NADc,NADf,(Enz6f, -)),
METABOLITE(NADHc,NADHf,(Enz6f, +)),
METABOLITE(Pyrc,Pyrf,(Enz10f, +)),
METABOLITE(Glcc,Glcf,(Enz1f, -)),
METABOLITE(PEPc,PEPf,(Enz9f, +),(Enz10f, -)),
METABOLITE(F6Pc,F6Pf,(Enz2f, +),(Enz3f, -)),
METABOLITE(G6Pc,G6Pf,(Enz1f, +),(Enz2f, -)),
METABOLITE(DHAPc,DHAPf,(Enz4f, +),(Enz5f, -)),
METABOLITE(3PGc,3PGf,(Enz7f, +),(Enz8f, -)),
METABOLITE(13BPc,13BPf,(Enz6f, +),(Enz7f, -)),
METABOLITE(F16BPc,F16BPf,(Enz3f, +), (Enz4f, -)),
METABOLITE(2PGc,2PGf,(Enz8f, +),(Enz9f, -)),
METABOLITE(G3Pc,G3Pf,(Enz5f, +),(Enz4f, +),(Enz6f, -)).
```

Figure 25: A representation of a qualitative model of glycolysis (see text for details).

concentration (NADc) and flow (NADf) of the metabolite NAD is controlled by flow through the single enzyme number 6 (Enz6f : Glyceraldehyde 3-phosphate dehydrogenase), and that this enzyme removes (signified by the '-') NAD ('+' would mean the enzyme flow adds the corresponding metabolite).

5.2 Experimental Aim

The specific system identification task we were interested in is: Given qualitative observations of metabolic states, can ILP-QSI identify a correct qualitative model for glycolysis?

5.3 Materials and Method

Our methodology is depicted in Fig. 26, where we describe two separate ways of identifying biochemical pathways. We make the following assumptions:

1. The data are sparse and not necessarily measured as part of a continuous time series. This is realistic given current experimental limitations in metabolomics, and rules out the possibility of numerical system identification approaches.

2. Only metabolites of known structure are involved in the model. The reason for this is that we employ a chemoinformatic heuristic to decrease the number of possible reactions. The heuristic is based on the reasonable assumption that any chemical reaction catalysed by an enzyme only breaks a few chemical bonds. Full details are in the paper by King et al. (2005). This is the strongest assumption we make. Even given the rapid advance of metabolomics (NMR, mass-spectroscopy, etc.), it is not currently realistic to assume that all the relevant metabolites in a pathway are observed and their structure determined.

3. Only metabolites of known structure are involved in a particular pathway. This is a restriction because current metabolomics technology can observe more compounds than can be structurally identified.

4. All reactions involve at most three substrates and three products.

5. For the qualitative states: we can measure the direction of change in metabolite level and first-derivative. This requires sampling the level at least three times in succession.

5.3.1 Logical/Graph-based Constraints

We first considered the logical/graph-based (LG) nature of the problem. The specific domain of metabolism imposes strong constraints on possible LG based models. We used these constraints in the following way:

1. Chemical reactions conserve matter and atom type (Valdes-Perez, 1994). For glycolysis we generated all possible ways of combining the 18 metabolites to form matter and atom type balance reactions (≤ 3 reactants and ≤ 3 products). This produced 172 possible reactions where the substrates balanced the products in the number and type of each element. The number compares well with the 2,300,000 possible reactions which would naively be possible.

2. Typical biochemical reactions only make/break a few bonds, and cannot arbitrarily rearrange atoms to make new compounds. A reaction was considered plausible if it broke 1 bond per reactant. This analysis was done originally by hand, and we have subsequently developed a general computer program that can automate this task.

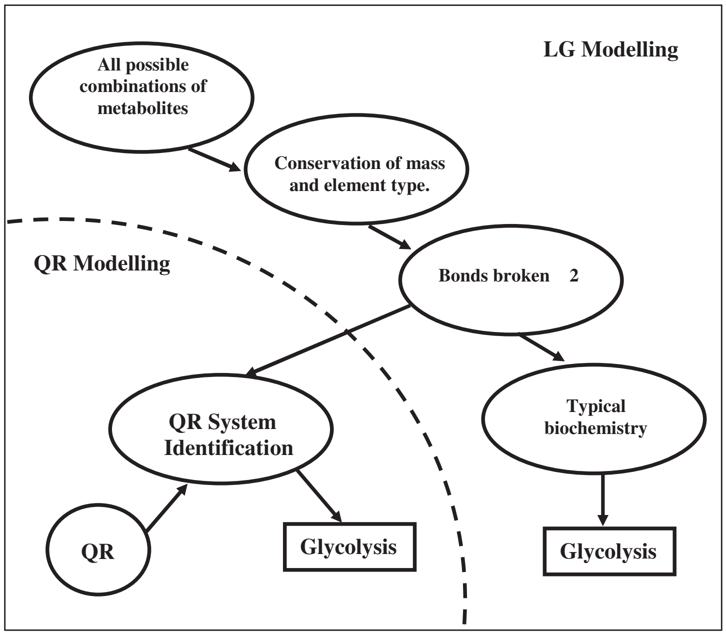


Figure 26: Our Metabolic System Identification methodology.

Of the 172 balanced reactions 18 were considered chemically plausible. Of these 18 reactions, 10 are the actual reactions of glycolysis and 8 are decoy reactions.

5.3.2 Qualitative Reasoning Constraints

We used a simple generate and test approach to learning. For the first computational experiment we used the 10 reactions of glycolysis and the 8 decoy reactions that were considered chemically feasible, see Fig. 24. All these reactions, in the absence of evidence to the contrary, are considered to be irreversible. We first generated all possible ways of combining the 18 reactions which connected all the 15 main substrates in glycolysis (models are non-disjoint). This generated 27,254 possible models with $\leq$ 10 reactions - it was not necessary to look for models with more reactions than that of the target (parsimony), as the models can be generated in size order. The smallest number of reactions necessary to include all 15 metabolites was of size 5. All of the 27,254 models involved the reaction: glyceraldehyde 3-phosphate + NAD $\Leftrightarrow$ 1,3-bisphosphoglycerate + NADH (reaction 6); so we could immediately conclude that this reaction occurred in glycolysis.

We formed example qualitative states of glycolysis using our QR simulator (in a pseudo random manner) to test these models. The states thus generated did not contain any noise. The 27,254 possible models were then tested against these states, and if a model could not generate a particular state it was removed from consideration (accuracy constraint). Note that the flows of the metabolites through each enzyme are not observed - they are intermediate variables. All we observe are the overall levels and flows of the metabolites. This makes the system identification task much harder.

For efficiency, we used the fast YAP Prolog compiler. We also formed compiled down versions of the enzyme and metabolite MCs (input/output look-up tables), and compiled down parts of QSIM. We also adopted a resource allocation method that employed increasingly computationally expensive tests: i.e. forming filter tests with exponentially increasing numbers of example states.

5.3.3 Results

After several months of compute time on a 65 node Beowulf cluster we reduced the 27,254 possible models down to 35 (a 736 fold reduction). These models included the target model (glycolysis), plus 34 other models that could not be qualitatively distinguished from it. All 35 models included the following six reactions (see Fig. 24):

3. F6P + ATP → F16BP + ADP
4. F16BP → DHAP + G3P
5. DHAP → G3P
6. G3P + NAD → 13BP + NADH
8. 3PG → 2PG
9. 2PG → PEP

These reactions form the core of glycolysis.

Examining the 35 models also revealed that the correct model had the fewest cycles, however we do not know if this is a general phenomenon.

We attempted to use the Progol positive only compression measure to distinguish between these models. This is based on comparing the models on randomly generated states. However, this was unsuccessful as no model covered any of the 100,000 random states we generated! We believe this is due to the extremely large state space. However, a simple modification of this approach does work. If we produce random states from glucogenesis (glycolysis driven in the reverse direction), then the true model of glycolysis covers fewer examples than any of the 34 alternatives, and so is identified as the true model. Note that this approach, unlike the use of Progol positive only compression measure, requires that new experimental data is obtained.

Thus we have demonstrated that the *method* for learning qualitative models of dynamic systems is scalable to handle a relatively large metabolic system. We have achieved this by means of MCs that represent meaningful units in the domain, and which map directly to the QSIM constraints from which they are abstracted. They also enable us to present these more complex models in a more “user friendly” manner, removing the need to understand the structure of high order differential equations.

6. Related Work

System identification has a long history within machine learning: we present below some of the important signposts that were directly relevent to the work here. *We focus on the strand of research which deals with learning qualitative models of dynamic systems.*

The earliest description of which we are aware concerning a computer program identifying a quantitative model to explain experimental data is the work by Collins (1968). There a procedure is described that heuristically searches through equation structures, which are linear combination of functions of the observed variables. Better known is the Bacon system (Langley, 1981), early versions of which largely concentrated on the parameter estimation problem, in particular selecting the most appropriate values for any exponents in the equations. For example, given a class of algebraic equation structures Bacon.1 was able to reconstruct Kepler's model for planetary motion from data. While later work (for example the work of Nordhausen & Langley, 1993) attempted to extend this work to deal with identifying both the algebraic structure and the relevant parameters, Bacon highlighted the importance of *bias* (Mitchell, Keller, & Kedar-Cabelli, 1986) in machine learning, both in constraining the possible model structures and in the space of possible models conforming to those structures. Other quantitative equation discovery systems in this lineage are: Coper (Kokar, 1985), that uses dimensional analysis to restrict the space of equations; Fahrenheit/EF (Langley & Zytkow, 1989) and E* that only examine the space of bivariate equations; Abacus (Falkenhainer & Michalski, 1986) that can identify piecewise equations; Sds that uses type and dimensionality restrictions to constrain the space of equations; the Lagrange family of equation finders (Džeroski & Todorovski, 1993; Todorovski & Džeroski, 1997; Todorovski et al., 2000; Todorovski, 2003) which attempt to identify models in the form of ordinary and partial differential equations; and IPM (Langley, George, Bay, & Saito, 2003) with its extensions and developments, Prometheus/RPM (Bridewell, Sandy, & Langley, 2004; Asgharbeygi, Bay, Langley, & Arrigo, 2006), which incorporate process descriptions (Forbus, 1984) to aid the construction and revision of quantitative dynamic models.

Focussing specifically on non-classical system identification for metabolic models, perhaps the most notable work on identification is that of Arkin, Shen, and Ross (1997) who identified a graphical model of the reactions in a part of glycolysis from experimental data. The work of Reiser, King, Kell, Muggleton, Bryant, and Oliver (2001) presents a unified logical approach to simulation (deduction) and system identification (induction and abduction). An interesting recent approach, presented by Koza, Mydlowec, Lanza, Yu, and Keane (2000), examines the identification of metabolic ODE models using genetic programming techniques. In this, the cellular system is viewed as an electrical circuit and the space of possible circuits is searched by means of a genetic programming approach.

The earliest reported work on the identification of qualitative models is that of Mozetic, 1987, and colleagues, who identified a model of the electrical activity of the heart. This work, reported more fully in (Bratko et al., 1989) remains a landmark effort in the qualitative modelling of a complex biological system. However, as other researchers have noted (Bratko, Muggleton, & Varsek, 1992), these results were obtained only for static models and did not provide insight into how models of dynamic systems should be identified.

The first machine learning system for learning qualitative models of dynamic systems was Genmodel (Coiera, 1989a, 1989b). Genmodel did not need any negative examples of system behaviour and models learned were restricted to qualitative relationships amongst the observed variables only (that is, no intermediate, or hidden, variables were hypothesized). The model, obtained using the notion of a most specific generalization of observed variables (in the sense of Plotkin, 1971), was usually over-constrained. That is, it contained more constraints than necessary to characterize fully the dynamics of the system being modeled. An updated version of Genmodel developed by Hau and Coiera (1993) showed that dimensional analysis (Bhaskhar & Nigam, 1990) could be used as a form of directed negative example generation. The new version could learn from from real-valued experimental data (which were converted internally into a qualitative form), but still required all variables to be known and measured from the outset. The system MISQ, entirely similar in complexity and abilities to the earlier version of Genmodel was developed by Kraan, Richards, and Kuipers (1991). This was later re-implemented in a general-purpose relational learning program Forte (Richards & Mooney, 1995), which allowed the hypothesis of intermediate variables (Richards, Kraan, & Kuipers, 1992). The relational pathfinding approach used by MISQ (through the auspices of Forte) is a special form of Inductive Logic Programming, the general framework of which is much more powerful

Bratko and colleagues were the first to view the problem of learning dynamic qualitative models explicitly as an exercise in Inductive Logic Programming (ILP) and first demonstrated the possibility of introducing intermediate (unobserved) variables in the models. They used the ILP system Golem (Muggleton & Feng, 1990) along with the QSIM representation to produce a model of the u-tube system. The model identified was equivalent to the accepted model (in the sense that it *predicted* the same behaviour) but the structure generated was not in a form that could help *explain* the behaviour (Coghill & Shen, 2001). Like Genmodel, the model produced was over constrained. Unlike Genmodel, Golem required both positive and negative examples of system behaviour and was shown by Hau and Coiera (1993) to be sensitive to the actual negative examples used.

Say and Kuru (1996) describe a program for system identification from qualitative data called QSI. QSI first finds correlations between variables, and then iteratively introduces new relations (and intermediate variables), building a model and comparing the output of that model with the known states until a satisfactory model is found. Say and Kuru characterized this approach as one of "diminishing oscillation" as it approaches the correct model. Like Genmodel and MISQ, QSI does not require "negative" observations of system behaviour. Unlike those systems, it does not use dimensional analysis and there does not appear to be any mechanism of incorportating such constraints easily within the program. The importance of dimensional analysis is recognised though: the authors suggest that it should be central to the search procedure.

Thus, while the identification of quantitative models has had a longer history in machine learning, learning qualitative models has also been the subject of notable research efforts. In our view, MISQ (the version as implemented within Forte) and QSI probably represent the state-of-the-art in this area. Their primary shortcomings are these:

- It is not apparent from the description or experimental evaluation of MISQ whether or not it is able to handle imperfect data (the correctness theorem presented only applies for complete, noise-free data).
- MISQ seeks the most constrained model that is consistent with the data. Often, exactly the opposite is sought (that is, we want the most parsiomonius model).
- QSI only deals with qualitative data and does not appear to include any easy mechanism for the incorporation of new constraints to guide its search.

7. General Discussion

In this paper we have presented a method for learning qualitative models of dynamic systems from time-series data (both qualitative and quantitative). In this section we discuss the general findings and limitations, as well as suggesting a number of directions for developing this research theme.

7.1 Computational Limitations

A major limitation of the ILP-QSI system in identifying glycolysis was the time taken (several months on a Beowulf cluster) to reduce the models from the 27,000 possible ones generated using chemoinformatic constraints, to the single correct one using the qualitative state constraints. While it would be preferable for this process to be faster, it is important to note that identifying a system with 10 reactions and 15 metabolites from scratch is an extremely hard identification task. We doubt that any human could achieve it, and we believe it would be a challenge for all the system identification methods we are aware of. It is difficult to compare system identification methods and we believe there is a need for competitions such as those run by KDD to compare methods.

The computational time of identification is dominated by the time taken to test if a particular model can produce certain observed states: examining 27,000 models is not unusual for a machine learning program, but it is unusual for a program to take hours to test if individual examples are covered. The slow speed of our identification method is therefore not a problem with what is normally considered the learning method (i.e. how the search of the space of possible models is done), but rather, is intrinsic to the complex relationship between a model and the states it defines. The cover-test method is, in the worst case, exponential in the maximum size of the model. Note that our lack of an efficient, i.e. polynomial algorithm, to determine cover is not because we are using qualitative states. We believe that the inherent difficulty of this task applies to both quantitative and qualitative models. In some areas of mathematics moving from the discrete to the real domain can simplify problems - this is the basis of much of the power of analysis. However, there is currently little evidence that this is the case in system identification, and quantitative models would seem to aggravate the problem. As cover tests are essentially deductions: can a set of axioms and rules (computer program/model) produce a particular logical sentence (observed state); they are in general non-computable. However, in real scientific systems, as they are bounded in space and time, non-computability is not a problem, however we expect all system identification methods to struggle with the task (Sadot, Fisher, Barak, Admanit, Stern, Hubbard, & Harel, 2008).

7.2 Kernel Subsets

From our presentation of the results of the experimentation it is clear that certain subsets of states (termed the *kernel subsets*) guarantee that the target model will be learned. From the analysis of these kernel subsets of state sets, we hypothesise that the kernel sets reflect the qualitative structure of the system of interest.

For a coupled system, in order to learn the structure of a system with a high degree of precision, the data used should come from tests yielding qualitatively different behaviors: i.e. behaviors which would appear as distinct branches in an envisionment graph. However, this hypothesis only provides a necessary, not a sufficient, condition for learning because it does not identify which states in each branch are suitable starting points for an experiment. For example consider the coupled tanks system. One could select states 9 and 11; these are from different branches yet they do not form a kernel subset. On the other hand, it was noted in presenting the results for this system that the key states in these kernel subsets were states 7 and 8. These states are in different branches and represent the *critical points* of the first derivative of the state variables of the system. This indicates the importance of these states to the definition of a system.

If a test were set up in which all the state variables were at their critical points then the test could be run for a very short time and the correct model structure identified. However, it is probably impossible in practice to set up such a test; especially in the situation where the structure of the system is completely unknown. An alternative is to set up multiple tests with the state variables set to their extrema: from which initial conditions all the states of the envisionment will eventually be passed through. However it still may be difficult to set up such tests, and they could take a long time to complete. These two scenarios form the ends of a spectrum within which the most practical experimental setting will lie. The identification of the best strategies is an important area of research to which the present work is clearly relevant.

On the other hand, for cascaded systems the kernel sets capture the asymmetry in the structure. Here again the extrema and critical points play an important role; but in this case it is subordinate to the fact that ILP-QSI automatically decomposes the system into its constituent parts for learning. This fact points to an important conclusion for learning larger scale complex systems; namely that the learning can be facilitated by, where possible, decomposing the system into cascaded subsystems.

7.3 Future Work

Having validated ILP-QSI on data derived from real biological systems, the next step is to explore how successful it can be at modelling real experimental data. It would be relatively straightforward to obtain data from water tanks and springs, but it would be much more interesting to work on real biological data. For such work to be successful it is likely that the quantitative to qualitative conversion process will need to be improved. Although not the focus of the work here, developing a more rigorous approach would be crucial in using ILP-QSI in a laboratory setting (Narasimhan, Mosterman, & Biswas, 1998). Once this has been done it will be much easier to use real experimental data for analysis by ILP-QSI. Specifically, the improvements required are the ability to extract all the qualitative states passed through during a numerical simulation, whilst minimizing noise. Nevertheless, this is not a direct limitation of our ILP-QSI method.

The following possibilities would benefit from further investigation:

- The QR representation used could be changed from QSIM to a more detailed and flexible one such as Morven (Coghill & Chantler, 1994; Coghill, 1996).
- The hypotheses presented about kernel subsets, such as why they are formed from some states and not others, need to be confirmed and analyzed further.
- The ability to map and explore the features of the model space would be of great use in planning further enhancements and, alongside kernel subsets, will help give an understanding of exactly which states will allow reliable learning.
- Large scale complex systems are generally identified piece by piece. The results from the cascaded tanks experiments indicate some circumstances under which this may be most easily facilitatied. Further investigation of this is warranted.
- As an alternative to the methods described in this paper, an incremental approach that identifies subsystems of the complete system is an interesting avenue of investigation (Srinivasan & King, 2008).

8. Summary and Conclusions

In this paper we have presented a novel system, named ILP-QSI, which learns qualitative models of dynamic processes. This system stands squarely in a strand of research that integrates Machine Learning with Qualitative Reasoning and extends the work in that area in the following ways:

The ILP-QSI algorithm itself extends the work; it is a branch and bound algorithm that makes use of background knowledge of (at least) three kinds in order to focus and guide the search for well posed models of dynamic processes.

Syntactic Constraints: The model size is prespecified; models must be complete and determinate; and must not proliferate instances of qualitative relations.

Semantic Constraints: The model must adequately explain the data; it must not contain relations that are redundant or contradictory; and the relations in the model must respect dimensional constraints.

System Theoretic Constraints: The model should be singular and not disjoint; all endogenous variables must appear in at least two relations; and the model should be causally ordered.

We have thoroughly tested the system on a number of well known dynamic processes. This has enabled us to ascertain that ILP-QSI is capable of learning under a variety of conditions of noisy and noise free data. This testing has also allowed us to identify some conditions under which it is possible to learn an appropriate model of a dynamic system. The conclusions from this aspect of the work are:

- Learning precision is related to the richness (or sparcity) and noisiness of the data from which the learning is performed.
- The target model is precisely learnt if the data used is a kernel subset.
- These kernels are made up of states from different branches in the envisionment graph.
- The system critical points play an important role in identifying the model structure.
- There is a spectrum of possibilities with regard to the setting up of suitable experiments to garner data from which to learn models of the physical or biological systems of interest.
- Cascaded parts of systems help to identify suitable points of decomposition for model learning.

While ILP-QSI is designed to learn a qualitative structural model from qualitative data, it is sometimes the case that the original measurements are quantitative (albeit sparse and possibly noisy). In order to ascertain how ILP-QSI would cope with qualitative data generated from quantitative measurements we carried out a “proof-of-concept” set of experiments from each of the physical process models previously utilised. The results from these were in keeping with the results obtained from the qualitative experiments. This adds weight to the conclusions regarding the viability of our approach to learning structural models of dynamic systems under adverse conditions.

Finally, in order to test the scalability of the method, we applied ILP-QSI to a large scale metabolic pathway: glycolysis. In this case the search space was deemed too large to attempt learning the QSIM primitives alone. However, knowledge of the domain enabled us to group these primitives into a set of *Metabolic Components* from which models of metabolic pathways can more easily be constructed. Also, for this part of the research Logical graph based models were used to represent background domain knowledge. Utilising these, we were able to identify 35 possible structures for the glycolysis pathway (out of a possible 27,254); of these the target model had the fewest cycles (though we do not know if this is a general phenomenon) and minimally covered the data generated from the reverse pathway of glucogenesis.

The overall conclusions of the this work are that qualitative reasoning methods combined with machine learning (specifically ILP) can successfully learn qualitative structural models of systems of high complexity under a number of adverse circumstances. However, the work reported herein constitutes a step in a line of research that has only recently begun; and, as with all interesting lines of research, it raises in its turn interesting questions that need to be addressed.

Acknowledgments

This work was supported in part by BBSRC/EPSRC grant BIO10479. The authors would like to thank Stephen Oliver and Douglas Kell for their helpful discussions on the biological aspects of this paper. We would also like to thank Simon Garrett for many interesting and fruitful interactions.

Appendix A. The Derivation of the Solution Space for the Tanks Systems

In this appendix we provide a summary of whence the solution spaces for the tanks systems utilised in this project are constructed. Further details regarding envisionments and their associated solution spaces may be found in the work of Coghill et al. (1992) and Coghill (2003).

In order to facilitate this analysis we will need to make use of a quantitative version of the system models. For ease of exposition we will make the additional assumption that the systems are linear.[10]

A.1 The U-tube

A quantitative model of the u-tube system is

$$\frac{dh_1}{dt} = k(h_1 - h_2)$$
$$\frac{dh_2}{dt} = k(h_2 - h_1)$$

By inspection of these two equations it is easy to see that (ignoring the trivial case where $k = 0$) the derivatives in these two equations are both zero when:

$$h_1 = h_2 \tag{6}$$

That is:

$$h_1 = h_2 \Rightarrow \frac{dh_1}{dt} = \frac{dh_2}{dt} = 0$$

This accounts for the relationship, depicted in Fig. 18, between h_1 and h_2 when the derivatives are zero. From the envisionment table for the u-tube (Table 1 in Section 2.2) we see that the only state with zero derivatives is state 5; hence it is represented by this line.

A.2 The Coupled Tanks

A quantitative model of the coupled tanks system is

$$\frac{dh_1}{dt} = q_i - k_1(h_2 - h_1) \tag{7}$$
$$\frac{dh_2}{dt} = k_1(h_2 - h_1) - k_2 \cdot h_2 \tag{8}$$

When $\frac{dh_1}{dt} = 0$ Equation 7 can be rewritten as:

$$0 = q_i - k_1(h_2 - h_1)$$
$$= q_i - k_1 h_2 - k_1 h_1$$

10. In fact for the types of non-linearity normally associated with systems of this kind the solution spaces are qualitatively identical to those described here, although the analysis required to construct them is slightly more complicated.

which can be re-arranged to give

$$h_2 = \frac{q_i}{k_1} - h_1$$

When q_i is zero this reduces to

$$h_2 = h_1 \tag{9}$$

When $\frac{dh_2}{dt} = 0$ Equation 8 can be rewritten as:

$$\begin{aligned} 0 &= k_1(h_2 - h_1) - k_2h_2 \\ &= k_1h_1 - k_1h_1 - k_2h_2 \\ &= (k_1 - k_2)h_2 - k_1h_1 \end{aligned}$$

so

$$(k_1 - k_2)h_2 = k_1h_1$$

Re-arranging gives

$$h_2 = \frac{k_1}{(k_1 - k_2)}h_1 \tag{10}$$

This accounts for the relations between h_1 and h_2 depicted in the solution space of Fig. 18.

A.3 The Cascaded Tanks

A quantitative model of the cascaded tanks system is:

$$\frac{dh_1}{dt} = q_i - k_1h_1 \tag{11}$$

$$\frac{dh_2}{dt} = k_1h_1 - k_2h_2 \tag{12}$$

When $\frac{dh_1}{dt} = 0$ Equation 11 can be re-arranged as:

$$q_i = k_1h_1$$

or

$$h_1 = \frac{q_i}{k_1}$$

When $\frac{dh_2}{dt} = 0$ Equation 12 can be rewritten as:

$$k_2h_2 = k_1h_1$$

or

$$h_2 = \frac{k_1}{k_2} h_1$$

This accounts for the relations between h_1 and h_2 depicted in the solution space of Fig. 27.

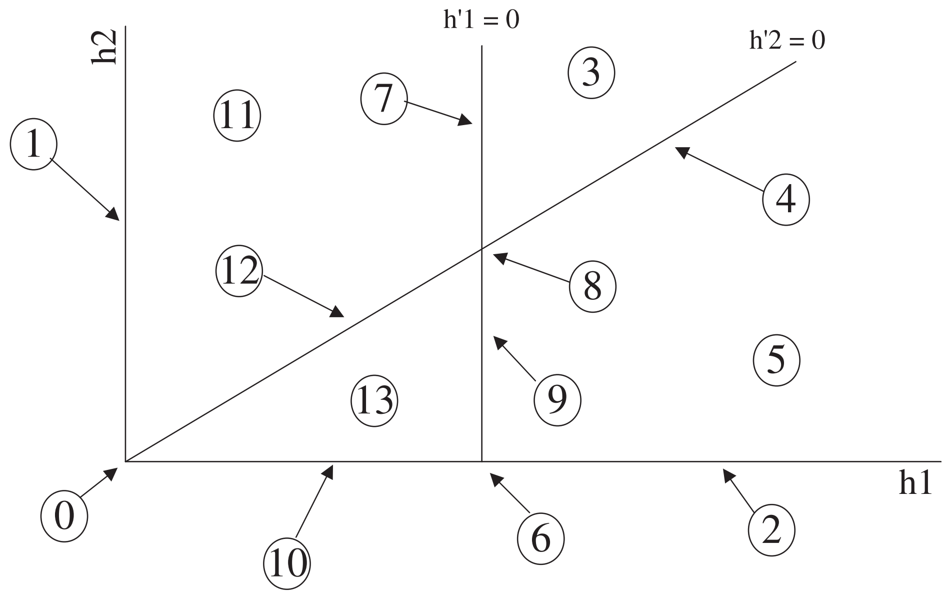


Figure 27: The solution space for the cascaded tanks system.

References

Arkin, A., Shen, P., & Ross, J. (1997). A test case of correlation metric construction of a reaction pathway from measurements. *Science*, *277*, 1275–1279.

Asgharbeygi, N., Bay, S., Langley, P., & Arrigo, K. (2006). Inductive revision of quantitative process models. *Ecological modelling*, *194*, 70–79.

Bergadano, F., & Gunetti, D. (1996). *Inductive Logic Programming: From Machine Learning to Software Engineering*. MIT Press.

Bhaskhar, R., & Nigam, A. (1990). Qualitative physics using dimensional analysis. *Artificial Intelligence*, *45*, 73–111.

Blackman, R. B., & Tukey, J. W. (1958). *The Measurement of Power Spectra*. John Wiley and Sons, New York.

Bradley, E., Easley, M., & Stolle, R. (2000). Reasoning about nonlinear system identification. Tech. rep. CU-CS-894-99, University of Colorado.

Bratko, I., Mozetic, I., & Lavrac, N. (1989). *KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems*. MIT Press, Cambridge, Massachusetts.

Bratko, I., Muggleton, S., & Varsek, A. (1992). Learning qualitative models of dynamic systems. In Muggleton, S. (Ed.), *Inductive Logic Programming*, pp. 437–452. Academic Press.

Bridewell, W., Sandy, J., & Langley, P. (2004). An interactive environment for the modeling and discovery of scientific knowledge.. Tech. rep., Institute for the Study of Learning and Expertise, Palo Alto, CA.

Camacho, R. (2000). *Inducing Models of Human Control Skills using Machine Learning Algorithms*. Ph.D. thesis, University of Porto.

Coghill, G. M. (1996). *Mycroft: A Framework for Constraint-Based Fuzzy Qualitative Reasoning*. Ph.D. thesis, Heriot-Watt University.

Coghill, G. M. (2003). Fuzzy envisionment. In *Proc. of the Third International Workshop on Hybrid Methods for Adaptive Systems*, Oulu, Finland.

Coghill, G. M., Asbury, A. J., van Rijsbergen, C. J., & Gray, W. M. (1992). The application of vector envisionment to compartmental systems.. In *Proceedings of the first international conference on Intelligent Systems Engineering*, pp. 123–128, Edinburgh, Scotland.

Coghill, G. M., & Chantler, M. J. (1994). Mycroft: a framework for qualitative reasoning. In *Proceedings of the Second International Conference on Intelligent Systems Engineering*, pp. 49–55, Hamburg, Germany.

Coghill, G. M., Garret, S. M., & King, R. D. (2004). Learning qualitative models of metabolic systems. In *Proceedings of the European Conference on Artificial Intelligence ECAI-04*, Valencia, Spain.

Coghill, G. M., & Shen, Q. (2001). On the specification of multiple models for diagnosis of dynamic systems. *AI Communications*, *14*(2), 93–104.

Coiera, E. W. (1989a). Generating qualitative models from example behaviours. Tech. rep. 8901, University of New South Wales, Deptartment of Computer Science.

Coiera, E. W. (1989b). Learning qualitative models from example behaviours. In *Proc. Third Workshop on Qualitative Physics*, Stanford.

Collins, J. (1968). A regression analysis program incorporating heuristic term selection. In Dale, E., & Michie, D. (Eds.), *Machine Intelligence 2*. Oliver and Boyd.

Džeroski, S. (1992). Learning qualitative models with inductive logic programming. *Informatica*, *16*(4), 30–41.

Džeroski, S., & Todorovski, L. (1993). Discovering dynamics. In *International Conference on Machine Learning*, pp. 97–103.

Džeroski, S., & Todorovski, L. (1995). Discovering dynamics: from inductive logic programming to machine discovery. *J. Intell. Information Syst.*, *4*, 89–108.

Falkenhainer, B., & Michalski, R. S. (1986). Integrating quantitative and qualitative discovery: The abacus system. *Machine Learning*, *1*(4), 367–401.

Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence*, *24*, 169–204.

Garrett, S. M., Coghill, G. M., Srinivasan, A., & King, R. D. (2007). Learning qualitative models of physical and biological systems. In Džeroski, S., & Todorovski, L. (Eds.), *Computational discovery of communicable knowledge*, pp. 248–272. Springer.

Gawthrop, P. J., & Smith, L. P. S. (1996). *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall, Hemel Hempstead, Herts, England.

Hau, D. T., & Coiera, E. W. (1993). Learning qualitative models of dynamic systems. *Machine Learning*, *26*, 177–211.

Healey, M. (1975). *Principles of Automatic Control*. Hodder and Stoughton.

Iwasaki, Y., & Simon, H. A. (1986). Causality in device behavior. *Artificial Intelligence*, *29*, 3–32. See also De Kleer and Brown's rebuttal and Iwasaki and Simon's reply to their rebuttal in the same volume of this journal.

King, R. D., Garrett, S. M., & Coghill, G. M. (2005). On the use of qualitative reasoning to simulate and identify metabolic pathways.. *Bioinformatics*, *21*(9), 2017 – 2026.

Kokar, M. M. (1985). Coper: A methodology for learning invariant functional descriptions. In Mitchell, T., Carbonell, J., & Michalski, R. (Eds.), *Machine Learning: A Guide to Current Research*, pp. 151–154. Kluwer Academic Press.

Koza, J. R., Mydlowec, W., Lanza, G., Yu, J., & Keane, M. A. (2000). Reverse engineering and automatic synthesis of metabolic pathways from observed data using genetic programming.. Tech. rep. SMI-2000-0851, Stanford University.

Kraan, I. C., Richards, B. L., & Kuipers, B. J. (1991). Automatic abduction of qualitative models. In *Proceedings of Qualitative Reasoning 1991 (QR'91)*.

Kuipers, B. (1994). *Qualitative Reasoning*. MIT Press.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science*, *5*, 31–54.

Langley, P., George, D., Bay, S., & Saito, K. (2003). Robust induction of process models from time series data.. In *Proc. twentieth International Conference on Machine Learning*, pp. 432–439, Washington, DC. AAAI Press.

Langley, P., & Zytkow, J. (1989). Data-driven approaches to empirical discovery. *Artificial Intelligence*, *40*, 283–312.

McCreath, E. (1999). *Induction in first order logic from noisy training examples and fixed example set sizes*. Ph.D. thesis, University of New South Wales.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, *1*, 47–80.

Mozetic, I. (1987). Learning of qualitative models. In Bratko, I., & Lavrac, N. (Eds.), *Progress in Machine Learning: Proceedings of EWSL '87: 2nd European Working Session on Learning*, pp. 201–217. Sigma Press.

Muggleton, S. (1995). Inverse Entailment and Progol. *New Gen. Comput.*, *13*, 245–286.

Muggleton, S. (1996). Learning from positive data. *Lecture Notes in AI*, *1314*, 358–376.

Muggleton, S., & Feng, C. (1990). Efficient induction of logic programs. In *Proc. of the First Conf. on Algorithmic Learning Theory*. OHMSHA, Tokyo.

Muggleton, S., & Raedt, L. D. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, *19,20*, 629–679.

Narasimhan, S., Mosterman, P., & Biswas, G. (1998). A systematic analysis of measurement selection algorithms for fault isolation in dynamic systems. In *Proc. Ninth Intl. Workshop on Principles of Diagnosis (DX-98)*, pp. 94–101, Cape Cod, MA.

Nordhausen, B., & Langley, P. (1993). An integrated framework for empirical discovery. *Machine Learning*, *12*, 17–47.

Papadimitriou, C., & Steiglitz, K. (1982). *Combinatorial Optimisation*. Prentice-Hall, Edgewood-Cliffs, NJ.

Plotkin, G. (1971). *Automatic Methods of Inductive Inference*. Ph.D. thesis, Edinburgh University.

Reiser, P., King, R., Kell, D., Muggleton, S., Bryant, C., & Oliver, S. (2001). Developing a logical model of yeast metabolism. *Electronic Transactions on Artificial Intelligence*, *5*, 233–244.

Richards, B. L., Kraan, I., & Kuipers, B. J. (1992). Automatic abduction of qualitative models. In *Proc. of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pp. 723–728. MIT Press.

Richards, B. L., & Mooney, R. J. (1995). Automated refinement of first-order horn-clause domain theories. *Machine Learning*, *19*(2), 95–131.

Riguzzi, F. (2005). Two results regarding refinement operators. In Kramer, S., & Pfahringer, B. (Eds.), *Late Breaking Papers, 15th International Workshop on Inductive Logic Programming (ILP05), August 10–13, 2005*, pp. 53–58, Münich, Germany.

Sadot, A., Fisher, J., Barak, D., Admanit, Y., Stern, M. J., Hubbard, E. J. A., & Harel, D. (2008). Towards verified biological models.. *IEEE/ACM Trans. Comput. Biology and Bioinformatics.*, *5(2)*, 1–12.

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

Shoup, T. E. (1979). *A Practical Guide to Computer Methods for Engineers*. Prentice-Hall Inc., Englewood Cliffs, N. J. 07632.

Soderstrom, T., & Stoica, P. (1989). *System Identification*. Prentice Hall.

Srinivasan, A. (1999). The Aleph Manual. Available at http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/.

Srinivasan, A., & King, R. D. (2008). Incremental identification of qualitative models of biological systems using inductive logic programming. *J. Machine Learning Research* to appear.

Todorovski, L. (2003). *Using domain knowledge for automated modeling of dynamic systems with equation discovery*. Ph.D. thesis, Faculty of Electrical Engineering and Computer Science, University of Ljubljana, Slovenia.

Todorovski, L., Srinivasan, A., Whiteley, J., & Gavaghan, D. (2000). Discovering the structure of partial differential equations from example behavior. In *Proceedings of the*

Seventeenth International Conference on Machine Learning, pp. 991–998, San Francisco.

Todorovski, L., & Džeroski, S. (1997). Declarative bias in equation discovery. In *Proc. 14th International Conference on Machine Learning*, pp. 376–384. Morgan Kaufmann.

Valdes-Perez, R. E. (1994). Heuristics for systematic elucidation of reaction pathways.. *J. Chem. Informat. Comput. Sci.*, *34*, 976–983.

Voit, E. O., & Radivoyevitch, T. (2000). Biochemical systems analysis of genome-wide expression data. *Bioinformatics*, *16*(11), 1023–1037.

Warren, P., Coghill, G. M., & Johnstone, A. (2004). Top down and botton up development of a fuzzy rule-based diagnostic system. In *Proc. of the Fourth International Workshop on Hybrid Methods for Adaptive Systems*, Aachen, Germany.