# A Semi-Quantitative Physics Compiler

**Adam Farquhar**
Knowledge Systems Laboratory
Stanford University
Palo Alto, CA, USA
Ph: +001 415-723-9770
Fax: +001 415-725-5850
Adam_Farquhar@ksl.stanford.edu

**Giorgio Brajnik**
Dipartimento di Matematica e Informatica
Università di Udine
Udine, Italy
Ph: +39 (432) 272210
Fax: +39 (432) 510755
giorgio@dimi.uniud.it

**Abstract**

Incomplete information is present in many engineering domains, hindering traditional and non-traditional simulation techniques. This paper describes SQPC (semi-quantitative physics compiler), an implemented approach to modelling and simulation that can predict the behavior of incompletely specified systems, such as those that arise in the water control domain. SQPC is the first system that unifies compositional modeling techniques with semi-quantitative representations. We describe SQPC's foundations, QSIM and QPC, and how it extends them. We demonstrate SQPC using an example from the water supply domain.

## 1 Introduction

Consider the problem of water supply control. A lake has a dam with floodgates that can be opened or closed to regulate the water flow through power generating turbines, the water level (stage) of the lake, and the downstream flow. The goal of a controller is to provide adequate reservoir capacity for power generation, consumption, industrial use, and recreation, as well as downstream flow. In exceptional circumstances, the controller must also work to minimize or avoid flooding both above and below the dam. This task is both difficult and vitally important to the residents of surrounding areas. The work of controllers could be substantially eased by sound automatic modeling and simulation tools.

There are several forms of incomplete information that appear in this domain. The precise shape and capacity of lakes or reservoirs is rarely known; the outflow from opening a dam's floodgates is only crudely measured; empirical data on the level/flow-rate curve for rivers becomes less and less accurate when flood conditions approach; few quantities are measured (e.g. flow rates of minor tributaries are not measured at all); the amount of runoff to be expected from a given rainfall depends on difficult to measure surface characteristics such as saturation; the amount of rainfall that actually falls on a lake and surrounding areas is difficult to predict and is imprecisely measured. Nonetheless, both mathematical analysis and observations do provide rough bounds on the quantities involved. Often, rough accurate bounds suffice to select appropriate actions.

This domain is challenging for existing approaches to modeling and simulation. Pure qualitative reasoning techniques (Forbus 1984; Kuipers 1986) do not exploit the partial information available and consequently provide insufficiently strong predictions. Traditional numeric methods require much more precise information than is available, forcing modelers to make assumptions which may invalidate results and which may be difficult to evaluate.

Fortunately, recent advances in *semi-quantitative* simulation techniques provide a method for predicting the behavior of such systems. This work extends the purely qualitative representation (Kuipers 1986) with means for representing semi-quantitative information (Berleant & Kuipers 1988; Kuipers & Berleant 1992; Kay & Kuipers 1993). In this work, semi-quantitative information is represented in two forms: bounds on variable values and functional bounds (envelopes) on otherwise unspecified monotonic functions. This is exactly the kind of information that is available in the water supply and many engineering domains.

Several systems (Forbus & Falkenhainer 1990; Iwasaki & Low 1991; Amador, Finkelstein, & Weld 1993) have been developed that use compositional modelling techniques and exploit qualitative models to provide explanations of numeric simulations. They are unable to represent or use semi-quantitative information. In order to provide a numeric simulation, they all require complete precise initial conditions and algebraic equations.

This paper describes SQPC (semi-quantitative physics compiler), an implemented approach to modelling and simulation that uses semi-quantitative knowledge. SQPC extends QPC (qualitative physics compiler) (Crawford, Farquhar, & Kuipers 1990; Farquhar 1993; 1994) to exploit the recent advances in semi-quantitative simulation. SQPC is the first compositional modeling system to employ semi-quantitative representation and simulation.

The input to SQPC is a *domain theory* and a *scenario*. The domain theory is composed primarily of *model fragment* definitions the describe both the conditions under which physical phenomenon are active, and their consequences. The scenario specifies objects that are known to be of interest, some initial conditions, and some relations that hold throughout the scenario. Both the domain theory and scenario may include bounds on numeric values and monotonic functions. From this, SQPC generates a set of behavioral descriptions, guaranteed to cover any system trajectory consistent with the scenario and domain theory. A behavior may pass through a number of distinct operating regions, each of which is characterized by a distinct mathematical model.

# 2 Foundations

## 2.1 Semi–quantitative simulation

SQPC is built on top of the QSIM qualitative simulator (Kuipers 1986; 1993). The input to QSIM is a *qualitative differential equation* (QDE) which specifies: (i) a set of variables (continuously differentiable functions of *time*); (ii) a *quantity space* for each of these variables, specified in terms of a totally ordered set of symbolic *landmark* values; (iii) a set of constraints expressing (algebraic, differential or monotonic) relationships between variables. A QDE is an abstract description of a, perhaps infinite, set of ordinary differential equations. The output of QSIM is a set of behaviors. Each behavior is a sequence of states, where a state is a mapping of variables to qualitative values. A qualitative value represents the magnitude of the variable, which is either equal to a landmark or in the open interval specified by adjacent landmarks, and its direction of change (the sign of its time derivative: *dec, std, inc*). Each state describes either a time point or an open temporal interval.

In the semi-quantitative framework employed by SQPC, the basic qualitative representation is augmented by use of Semi–Quantitative Differential Equations (SQDE) (Berleant & Kuipers 1988; Kuipers & Berleant 1992; Kay & Kuipers 1993). Each landmark may be bounded with a precise numeric upper and lower bound. Each monotonic function constraint may be bounded with a precise functional upper and lower bound. A monotonic function constraint represents an element of an infinite set of real valued functions. Its general form is $((M\ s_1 \ldots s_n)\ x_1 \ldots x_n\ y)$ where each $s_i$ is a sign and $x_i, y$ are variables. Such a constraint denotes the set

$$\mathcal{F}_M = \left\{ f \;\middle|\; \begin{array}{l} f \text{ is a continuous} \\ \text{differentiable function} \\ f\colon \Re^n \to \Re \\ \text{and } \forall i\colon sign(\frac{\partial f}{\partial x_i}) = s_i . \end{array} \right\}.$$

An envelope for such a constraint is a pair of functions $F = <\underline{f}, \overline{f}>$ that restricts the set being characterized by the constraint. That is, an envelope $<\underline{f}, \overline{f}>$ for the above mentioned constraint characterizes the set

$$\mathcal{F}_F = \left\{ f \;\middle|\; \begin{array}{l} f \in \mathcal{F}_M \text{ and } \forall t \in T\colon \\ \underline{f}(x_1(t), \ldots, x_n(t)) \leq f(x_1(t), \ldots, x_n(t)), \\ f(x_1(t), \ldots, x_n(t)) \leq \overline{f}(x_1(t), \ldots, x_n(t)) \end{array} \right\}.$$

The semi-quantitative simulators augment behavior with the numeric bounds. They are also able to use the semi-quantitative information to rule out qualitatively possible behaviors. The first semi-quantitative techniques (Berleant & Kuipers 1988) propagated the bounds throughout each time-point state, and then used the mean-value theorem to constrain the values across time. The later dynamic envelope techniques (Kay & Kuipers 1993) construct extremal equations for the derivative of each state variable. These extremal equations are then explicitly integrated to provide bounds on variable values across time intervals. Neither technique strictly dominates the other. As a result, the bounds provided by the two methods may be intersected, yielding sometimes stronger predictions than either alone.

## 2.2 Qualitative Physics Compiler

SQPC is an extension of QPC, whose modeling language builds on Qualitative Process theory (Forbus 1984). The input to QPC is a *domain theory* and *scenario* specified in the QPC modeling language. A domain theory consists of a set of quantified definitions, called *model fragments*, each of which describes some aspect of the domain, such as physical laws (e.g. mass conservation), processes (e.g. liquid flows), devices (e.g. pumps), and objects (e.g. containers). Each definition applies whenever there exists a set of participants for whom the stated conditions are satisfied. The specific system or situation being modeled is partially described by the scenario definition, which lists a set of objects that are of interest, some of the initial conditions, relations that hold throughout the scenario, and boundary conditions.

Influences are compositional relations between variables that are particularly convenient for asserting fragments of information that can be composed into constraints. Three kinds of influences are supported. An *indirect influence* such as $(Y\ Q^+\ X)$ means that in the absence of countervailing influences, an increase in $X$ causes an increase in $Y$ and that $Y$ is functionally determined by the set of influencing variables. More precisely $(Y\ Q^+\ X)$ means that there exists $f$, a continuous function, and a set of variables $\{x_i\}_{1 \leq i \leq n}$ such that $Y = f(X, x_1, \ldots, x_n)$ and $\frac{\partial f}{\partial X} > 0$. In the case of $(Y\ Q^-\ X)$ then $\frac{\partial f}{\partial X} < 0$. The *algebraic influences* $Q_{add}$ and $Q_{sub}$ provide the constraint $\frac{\partial f}{\partial X} = 1$ and $\frac{\partial f}{\partial X} = -1$ respectively. Finally, a *direct influence* such as $(Y\ I^+\ X)$ expresses that a positive $X$ tends to increase $Y$. This is equivalent to an algebraic influence on the derivative of the influenced variable (*i.e.*, $(Y\ I^+\ X) \equiv Y' = \frac{dY}{dt}$ and $(Y'\ Q_{add}\ X)$).

QPC employs a hybrid architecture in which the model building portion is separated from the simulator. This architecture allowed SQPC to exploit semi-quantitative information without changing the overall QPC algorithm.

The input to QPC (Farquhar 1994) is a domain theory describing physical phenomena and a specification of the system to be modeled, called the scenario. The domain theory and scenario induce a set of logical axioms. QPC uses this database of logical axioms to infer the set of model fragment instances that apply during the time covered by the database (called the *active* model fragments). Inferences performed by QPC include those concerning structural relationships between objects declared in the scenario, and those aiming at computing the transitive closure of order relationships between quantities. A database with a complete set of model fragment instances defines an initial value problem which is given to the simulator in terms of equations and initial conditions. If any of the predicted behaviors cross the boundary conditions the process is repeated: a new database is constructed to describe the system as it crosses the boundaries of the current model, another complete set of active model fragments is determined, and another simulation takes place.

The output of QPC is a directed rooted graph, whose nodes are either databases or qualitative states. The root of the graph is the initial database, and a possible edge in the graph may:

- link a database to a refined database (obtained by adding more facts, either derived through inference rules or assumed by QPC when ambiguous situations are to be solved);
- link a complete database to a state (which is one of the possible initial states for the only model derivable from the database);
- link a state to a successor state (this link is computed by QSIM);
- link a state to a database (the last state of a behavior which crossed the operating region; the database describes the situation just after the transition occurred).

Each path from the root to a leaf describes one possible temporal evolution of the system being modeled. Each model in path identifies a a distinct operating region of the system.

## 3 SQPC extends QPC

SQPC extends the modeling language, the underlying representation and the inference methods employed. This section describes these extensions.

### 3.1 Modeling language

SQPC extends the QPC modeling language by adding numeric bounds on magnitudes, dimensional information, bounding envelopes on monotonic functions, and functions specified by tables.

**Numeric values.** SQPC represents numeric and qualitative magnitudes in a single framework. Both represent specific real numbers, which might be known only with uncertainty. Numeric magnitudes constrain such a number to lie within a numeric range. Note two aspects that complicate reasoning on numeric magnitudes. First, two comparable magnitudes constrained by the same range are, in general, not equal (*i.e.*, $Range(m) = [a\ b]$ and $Range(n) = [a\ b]$ do not entail that $m = n$ unless $a = b$). Secondly, range constraints on magnitudes may change during the analysis (range refinement). This may happen as an effect of the semi-quantitative simulation performed by QSIM. A model might entail $Range(m) = [a\ b]$, while a subsequent model in the behavior graph computed by SQPCmight entail $Range(m) = [a'\ b']$ where $[a'\ b'] \subseteq [a\ b]$. That is, as the analysis proceeds, SQPC may tighten the bounds on the numeric range of a magnitude.

**Dimensional information.** Variables and (symbolic or numeric) magnitudes are partitioned into dimensions. SQPC defines the seven International System dimensions as well as a *null-dimension*, which is provided to represent "pure number" quantities such as the efficiency of a turbine. A domain theory may also introduce derived dimensions specified by a list of dimension names with integer exponents (for example, `power-dimension` is $ml^2/t^3$).

Explicit representation of dimensions enables SQPC to:

1. perform dimensional analysis and verify that equations and order relations are well formed. Dimensional errors are common when writing equations and can be easily detected;

2. constrain inference about order relations. It is senseless to compare quantities that do not have the same dimension, and a reasoning mechanism not exploiting any dimensional information can produce incorrect inferences such as $x < 5 \wedge 10 < V \vdash x < V$ where a position ($x$) is being compared to a volume ($V$).

**Bounding envelopes.** An *envelope schema* is defined by a form similar to that for defining model fragments. It states a set of conditions under which a specific form of monotonic function over a tuple of variables is bounded by a functional envelope. The envelope is specified by a pair of functions. Instantiated envelope schemas are used to enrich a model with suitable envelopes. Since instantiation is automatically performed, envelopes are installed in models as needed, provided an appropriate monotonic constraint has already been included in the model.

**Tabular functions.** Tabular functions provide an important practical extension to the modeling language. A large portion of empirically collected knowledge about time-varying systems is represented and summarized in tabular form. The SQPC language permits numeric functions (used to specify envelopes) to be defined by data in a multi-dimensional table. SQPC assumes that these tables are coarse descriptions of the continuous reasonable functions that satisfy monotonic constraints. Currently SQPC provides two mechanisms for interpolating tabular data: *stepwise* functions, providing piecewise constant upper and lower bounds, or *piecewise linear* functions, providing tighter, but possibly less accurate, interpolations. In this way it is possible to define two envelope schemas from the same underlying tabular data. One envelope schema will use a linear interpolation method in a region where this approximation is known to introduce no significant error; in other regions a safer, but less precise, envelope schema using the more conservative interpolation method based on stepwise bounding functions, will be used. Of course the set of interpolation methods being used for computing tabular functions is open ended. The current version of SQPC provides the two mentioned above.

### 3.2 Reasoning

To accommodate the representational extensions described above there are several extensions that need to be made to the reasoning mechanism based on the underlying QPC architecture.

Dimensional information allows the inferences that compute order relations between variables and magnitudes to be focused. SQPC never compares two quantities with incompatible dimensions. The compatibility test is simplified by reducing all dimensions to a canonical form, represented by a vector of exponents (each position in the vector corresponds to a basic dimension).

Numeric bounds on magnitudes require a change in the computation of order relations. Except for the simple cases, in which the bounding ranges do not overlap, SQPC leaves the computation of numeric order relations to the semi-quantitative QSIM extensions. QSIM does a good job of propagating the bounds through the constraints in the SQDE. Recreating this in the SQPC knowledge base would be unnecessary, redundant, and inefficient.

SQPC needs to determine which envelopes to include in the SQDE for each model. This is non-trivial because there are several ways to describe a monotonic relationship among a set of quantities. Because each envelope that can be included is likely to strengthen the predictions, it is important to include all of the applicable ones. For instance, suppose that the model contains the constraint $(M\ (+\ -)\ X\ Y\ Z)$ but there is an envelope defined for the constraint $(M\ (+\ +)\ Y\ Z\ X)$. These two constraints are analytically equivalent, so the second constraint and its envelope should be included in the SQDE. This enables ranges for $X$ to be computed given ranges for $Y$ and $Z$, whereas an envelope for the former constraint enables computing the range for $Z$ from the ranges of the other two variables.

SQPC adds any constraint and envelope into the SQDE that is a permutation of a constraint in the SQDE. Notice that SQPC includes constraints in models after resolving influences (*i.e.*, after assuming a closed world and having determined the complete set of influencing and influenced variables). Then SQPC looks for possible envelopes (and possibly equivalent constraints) to be added to the model. This strategy makes it possible for the designer of the domain model and scenario to specify the envelopes, or envelope schemas, on the basis of the available data, independently from how influences will get resolved. In those cases where SQPC will construct models where some monotonic constraint does not have any envelope, SQPC will still be able to produce an accurate prediction, though with reduced precision.

In order to produce qualitative simulations of large systems, QPC generated attainable envisionments. Though leading to tractable simulations, this produces less detailed descriptions of the dynamics of the modeled system, for landmarks automatically generated by QSIM are crucial for identifying new critical points (and attaching numeric ranges to them). SQPC, unless appropriately instructed, does not produce an envisionment, but a more standard tree-based simulation, enabling also landmark generation. This choice, while providing more detailed results, has the drawback of requiring other methods for controlling the combinatorial explosion inherent in qualitative simulation. We are currently designing SQPC extensions based on known methods for reducing the number of spurious behaviors: *higher order derivatives* (Kuipers *et al.* 1991), *energy functions* (Fouché & Kuipers 1990), *phase-space criteria* (Lee & Kuipers 1993), and *abstraction techniques* (Clancy & Kuipers 1993).

## 4 An Example

We demonstrate SQPC on a problem from the domain of water supply control. We consider a portion of the system of lakes and rivers to be found in the scenic hill country surrounding Austin, Texas. The Colorado river flows into Lake Travis. The Mansfield Dam on Lake Travis produces hydroelectric power, controls the level of the lake, and the flow into the downstream leg of the Colorado.

The problem is to evaluate a "what if" scenario (figure 1). We are given an initial level for Lake Travis (a typical value between 690.2 and 690.3 feet) and a rough projected inflow from the Colorado river (between 791 and 950 cfs). The task is to determine what happens to the lake level and evaluate how long the hydroelectric plant can deliver power at the requested rate of 10 Mw.

In addition to the numeric bounds variables in the scenario, there are several other sources of semi-

```
(defscenario Travis-1-turbine
  "Turbine from controlled to uncontrolled regime."
  :entities  ...
  :landmarks
  ((top-of-dam :variables ((stage travis)) :range 714))
  :relations
  ((= (flow-rate colorado-up) (791 950))  ; cfs
   (= (top mansfield) top-of-dam)
   (is-open turb))
  :initial-conditions ((= (power turb) 10)                    ; Mw
                       (= (stage travis) (690.2 690.3)))      ; ft
  :envelopes
  ((stage-capacity
     :constraint (M+ (stage travis) (capacity travis))
     :upper-envelope (hi (lake-travis-stage-capacity
                          '(st cap) (list (stage travis))))
     :lower-envelope (lo (lake-travis-stage-capacity
                          '(st cap) (list (stage travis))))
     :upper-inverse (lo (lake-travis-stage-capacity
                         '(cap st) (list (capacity travis))))
     :lower-inverse (hi (lake-travis-stage-capacity
                         '(cap st) (list (capacity travis)))))))
```

Figure 1: Part of the scenario description. The initial conditions specify the desired power output of the turbine and the stage (level) of Lake Travis. The flow-rate of the upstream leg of the Colorado is set for the duration of the scenario. The `:envelopes` clause specifies the state-capacity table for Lake Travis. Any model in the scenario which includes the $M^+$ between the stage and capacity of lake travis will include this envelope.

| *Head (ft)* | *Power (Mw)* | *Discharge-rate (cfs)* |
|---|---|---|
| 120 | 8 | 1,054 |
| 120 | 9 | 1,150 |
| ... | ... | ... |
| 120 | 20 | 2,675 |
| 125 | 8 | 1,026 |
| ... | ... | ... |
| 145 | 20 | 1,940 |
| 150 | 8 | 884 |
| ... | ... | ... |
| 150 | 30 | 2,936 |

Table 1: A portion of the table describing turbine behavior. E.g. given a head of 120ft and a power setting of 8Mw, the discharge rate is expected to be 1054cfs.

quantitative information in this problem. An envelope schema (see figure 2) establishes bounds on the relation between the *head* of water above a turbine, the desired *power* output, and the *discharge rate* of water downstream. This envelope schema applies whenever the conditions and constraint portions of the envelope form are satisfied.
Lake Travis is a unique object. The relation between the *stage* (level) and *capacity* (volume) for Lake Travis is provided by an envelope specified in the scenario definition.

All of the semi-quantitative information in this domain theory is specified in the form of tables.[1] The tables reflect both observations and engineering estimates about the relationships between important variables. Table 1 shows some of the data extracted from the table describing quantitatively the behavior of the turbines in Mansfield Dam.

Solving this problem is made slightly more complex because of the behavior of the turbines. The turbines are controlled by a servo-mechanism designed to generate the desired amount of power regardless of the hydraulic pressure, which is determined by the *head* at the turbine. This is possible as long as there is sufficient *head*. When the *head* drops below the minimum threshold for a given power output, then less power is released. The domain theory captures this accurately (see figure 4). The domain theory also includes model fragments for conservation laws (e.g. of mass and energy), basic hydraulic principles (e.g. flow is proportional to head), and so on.

Figure 3 shows the SQPC output for this scenario. Under the specified conditions, the desired power level can be maintained until time `T1`, at least 45 days ($3.94 * 10^6$ seconds) after the start time. After `T1`, there will be insufficient hydraulic pressure to provide the full power output, the discharge rate from the turbine will decrease until it reaches equilibrium with the inflow at a rate between 791 and 950 cfs, and the lake

[1] The Lower Colorado River Authority (LCRA) has contributed actual tables of empirical data to the Qualitative Reasoning Group of the University of Texas for evaluation.

```
(defenvelope MANSFIELD-TURBINE-DR-E
      :comment "HEAD x POWER --> DISCHARGE-RATE of a Mansfield turbine."
      :participants  ((dam :type dams)
                      (lake :type lakes
                            :conditions ((connects dam lake river)))
                      (river :type rivers)
                      (t :type mansfield-turbines
                            :conditions ((has-valve dam t))))
      :constraint ((M - +) (head t) (power t) (discharge-rate t))
      :upper-envelope
        (hi (lake-travis-turbine-discharge-rate
                   '(hd pw dr) (list (head t) (power t))))
      :lower-envelope
        (lo (lake-travis-turbine-discharge-rate
                   '(hd pw dr) (list (head t) (power t)))))
```

Figure 2: An envelope that is applicable to any turbine of the kind found on Mansfield dam. It bounds the function that determines the discharge rate of the turbine given the head above it and the desired power output. The two functional bounds are computed by extracting the lowest and highest points from the range returned by a Lisp function (**lake-travis-turbine-discharge-rate**) defined on the basis of the "turbine rating table" shown in the text.

level will stabilize between 568' and 688'. Notice that at **T1** the lake system is entering a new operating region because the turbine is no longer servo-controlled (*i.e.*, the model fragment **NORMAL-TURBINE-MF** is no longer active). Notice also that all the variables are continuous across the transition, but some of their derivatives are not (*e.g.*, the derivative of **discharge-rate**, shown in terms of *qdir* of **discharge-rate**).

These predictions are strong enough to be useful to a system controller, even though the problem statement is very imprecise: the flow rate was very coarse; there are no semi-quantitative bounds for the relationship between *power* and *head* in the low-head situation after **t1**; the table relating *stage* and *capacity* becomes very coarse below 600'.

More precise information in the domain theory or scenario will result in more precise predictions. This is the strength of the semi-quantitative inference methods. The precision of the predictions is monotonic with the precision of the model and initial conditions.

We illustrate this by first strengthening the initial conditions of the scenario and then by strengthening the domain theory. If the upper bound on the inflow rate is reduced from 950 cfs to 800 cfs, then the upper bound on **T1**, the time that power generation drops below the desired rate, is reduced to 76 days, a 58% improvement. The domain theory can be strengthened by tightening the envelopes by using a linear interpolation for the stage-capacity curve instead of a step function. This tightens the range for **T1** to 50-58 days, an improvement of 89% from the original. Increased precision in the input or model leads to increased precision in the output.

### 4.1 Implementation status

SQPC is fully implemented in Lucid Common Lisp as an extension to QPC, which in turn uses the Algernon knowledge representation system (Crawford 1991) and QSIM. We are currently experimenting with SQPC in the water flow control domain; SQPC has been run on a dozen examples comparable to the one shown in this paper. The runtime for this example is around 4 minutes on Sun Sparc4/75. The bulk of this time is spent computing order relations with interpreted rules. Using standard rule compilation techniques or a special purpose inequality reasoner will result in a substantial (orders of magnitude) speedup.

## 5 Related work

In recent years, several research efforts have worked towards the development of self–explanatory simulators that construct numerical simulations and use a qualitative representation to help explain the results. Unlike SQPC, they do not use semi-quantitative information. Their predictions are either precise numeric ones, or purely qualitative.

SIMGEN (Forbus & Falkenhainer 1990) computes a total envisionment of the scenario and then, for each envisionment state[2] it builds a numerical simulator, monitors the simulation and, at the end of the analysis, interprets numerical results in terms of the envisionment graph. SIMGEN requires precise and complete numerical equations, initial and boundary conditions for the simulation. SIMGEN must be capable of building a numerical model for *each* envisionment state touched during the simulation; to this end it must be supplied with a library of numeric procedures for *every*

[2]corresponding to an operating region.

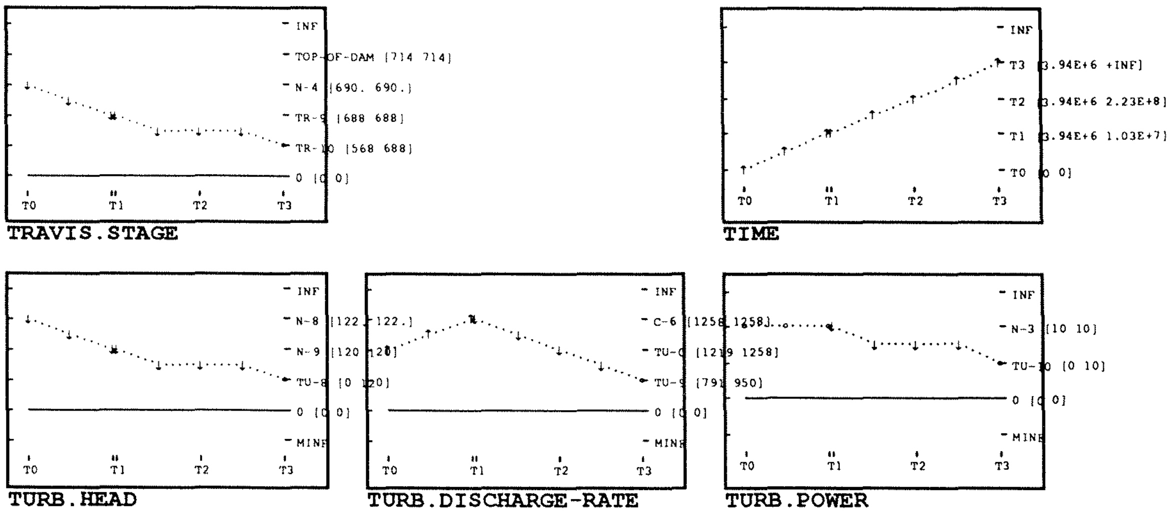


Figure 3: Behavior plot for several variables in the scenario. The power output by the turbine after **T1** is below the desired level.

possible combination of influences. SIMGEN is incapable of performing a simulation when a qualitative relation is quantitatively underspecified or when precise knowledge unavailable for any initial conditions.

DME (the Device Modeling Environment) (Iwasaki & Low 1991) is an incremental compositional modeling system capable of generating self-explanatory simulations. DME can work in two exclusive modes: qualitative or numeric. In the former case DME constructs qualitative states, and uses QSIM to generate successors; in the latter case, DME builds numerical models for simulation. In both modes, crossing an operating region triggers remodelling. DME is highly interactive and provides sophisticated explanation capabilities (Gautier & Gruber 1993; Gruber & Gautier 1993). DME requires precise numerical equations, initial and boundary conditions. Therefore, DME does not integrate qualitative and quantitative information in prediction.

Pika (Amador, Finkelstein, & Weld 1993) builds a numerical model for each operating region of the system as soon as this is needed. Pika monitors the numerical simulation and, at the end of the analysis, is capable of engaging in a simple question/answering dialogue. Pika requires precise equations, complete initial conditions (unlike the other systems), and complete specification of boundary conditions (in particular inequalities are not allowed). Compared to SQPC Pika performs limited inferences: no structural inferences are possible (this limits the expressive power of the modeling language) and influences are limited to indirect and algebraic ones: no provision is made for handling more general monotonic influences.

## 6 Conclusion

We have presented SQPC, the first system to unify compositional modeling techniques with semi-quantitative simulation. This is crucial for automatically building models of systems whose dynamics cross several operating regions. SQPC automatically constructs semi-quantitative models and produces useful predictions with imprecise knowledge. We have shown how QPC was extended to accomplish this. We argued that semi-quantitative knowledge is crucial to many applied engineering domains the one chosen for demonstrating SQPC, water supply control.

The current version of SQPC relies primarily on the Q2 static envelope (Berleant & Kuipers 1988) method which is based on the mean-value theorem. The NSIM (Kay & Kuipers 1993) dynamic envelope method, which performs explicit numeric integration of extremal bounding equations, often provides tighter bounds. Reliance on Q2 is due partly to software development schedules and partly to the extensive use of tabular information in the water supply control domain. Because NSIM performs explicit numeric integration, it requires smooth continuous functional bounds. This requires an extension to our current tabular data tools to generate smooth monotonic functional bounds for the tables. We are currently developing this extension.

The ability to use semi-quantitative information in a compositional modeler is tremendously exciting. We look forward to extending existing domain theories, such as the chemical engineering theory constructed by Catino (Catino 1993) to include semi-quantitative information and exploring the construction of substantial engineering quality models that are tractable due to greater available precision.

```
(defmodelfragment NORMAL-TURBINE-MF
    :participants ((t :type turbines)
                   (dam :type dams :conditions ((has-valve dam t)
                                                (is-open t)))
                   (lake :type lakes :conditions ((connects dam lake river)))
                   (river :type rivers))
    :operating-conditions ((> (stage lake) (base t))
                           (<= (min-head t) (head t))
                           (<= (head t) (max-head t)))
    :consequences ((I- (capacity lake) (discharge-rate t))
                   (Q- (discharge-rate t) (head t))
                   (Q+0 (discharge-rate t) (power t))
                   (Q-add (flow-rate river) (discharge-rate t))
                   (Q+0 (min-head t)(power t))
                   (Q+0 (max-head t)(power t))))
```

Figure 4: This model fragment describes normally operating turbines. An instance exists if there is an open turbine on a dam. It is active when the head above the turbine is between the variables `min-head` and `max-head` and the lake level is above the base of the turbine. Consequences specify equations that hold when an instance is active. The qualitative influence `I-` specifies that a positive `discharge-rate` decreases lake `capacity`; influences `Q-` and `Q+0` partially specify monotonic functions on the `discharge-rate` of the turbine (which is causally influenced by both `head` and `power`). The `Q-add` specifies that the `discharge-rate` sums into the river's `flow-rate`. The final influences specify that both `min-head` and `max-head` are affected by the desired `power` output.

Coupled with compositional modeling, the semi-quantitative techniques have the promise of achieving one of the major goals of qualitative reasoning: to make strong predictions about behavior, given the strongest model available.

## Acknowledgements

Adam's work has taken place in part at the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin, which is supported in part by NSF grants IRI-8904454, IRI-9017047, and IRI-9216584, and by NASA contracts NCC 2-760 and NAG 9-665. It has continued at the Stanford Knowledge Systems Laboratory, which is sponsored by the Advanced Research Projects Agency, ARPA Order 8607, monitored by NASA Ames Research Center under grant NAG 2-581; and by NASA Ames Research Center under grant NCC 2-537.

Giorgio started working on the research reported in this paper while he was visiting the UT Qualitative Reasoning Group during 1992. He would like to thank all the members of the group, and especially prof. Ben Kuipers, Bert Kay and Adam.

## References

Amador, F.; Finkelstein, A.; and Weld, D. 1993. Real-time self-explanatory simulation. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

Berleant, D., and Kuipers, B. 1988. Using incomplete quantitative knowledge in qualitative reasoning. In *Proc. of the Sixth National Conference on Artificial Intelligence*, 324–329.

Catino, C. 1993. *Automated Modeling of Chemical Plants with Application to Hazard and Operability Studies.* Ph.D. Dissertation, Department of Chemical Engeneering, University of Pennsylvania, Philadelphia, PA.

Clancy, D., and Kuipers, B. 1993. Behavior abstraction for tractable simulation. In *Proc. of the Seventh International Workshop on Qualitative Reasoning about Physical Systems*, 57–64.

Crawford, J.; Farquhar, A.; and Kuipers, B. 1990. QPC: a compiler from physical models into qualitative differential equations. In *Proc. of the Eight National Conference on Artificial Intelligence*. AAAI Press/The MIT Press.

Crawford, J. 1991. Algernon — a tractable system for knowledge representation. *SIGART Bulletin* 2(3).

Farquhar, A. 1993. *Automated Modeling of Physical Systems in the Presence of Incomplete Knowledge.* Ph.D. Dissertation, Department of Computer Sciences, the University of Texas at Austin. Available as technical report UT-AI-93-207.

Farquhar, A. 1994. A qualitative physics compiler. In *Proc. of the Twelfth National Conference on Artificial Intelligence.*

Forbus, K., and Falkenhainer, B. 1990. Self-explanatory simulations: an integration of qualitative and quantitative knowledge. In *Proc. of the Eight National Conference on Artificial Intelligence*, 380–387. AAAI Press/The MIT Press.

Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Fouché, P., and Kuipers, B. 1990. An assessment of currrent qualitative simulation techniques. In *Proc. of Fourth International Worskhop on Qualitative Reasoning about Physical Systems*, 195–205.

Gautier, P., and Gruber, T. 1993. Generating explanations of device behavior using compositional modeling and causal ordering. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

Gruber, T., and Gautier, P. 1993. Machine-generated explanations of engineering models: a compositional modeling approach. In *Proc. International Joint Conference on Artificial Intelligence*.

Iwasaki, Y., and Low, C. M. 1991. Model generation and simulation of device behavior with continuous and discrete changes. Technical Report KSL 91–69, Knowledge Systems Laboratory — Stanford University.

Kay, H., and Kuipers, B. 1993. Numerical behavior envelopes for qualitative models. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.

Kuipers, B., and Berleant, D. 1992. Combined qualitative and numerical simulation with Q3. In Faltings, B., and Struss, P., eds., *Recent advances in qualitative physics*. MIT Press. 3–16.

Kuipers, B.; Chiu, C.; Molle, D. D.; and Throop, D. 1991. Higher-order derivative constraint in qualitative simulation. *Artificial Intelligence* 51:343–379.

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

Kuipers, B. 1993. Reasoning with qualitative models. *Artificial Intelligence* 59:125–132.

Lee, W. W., and Kuipers, B. 1993. A qualitative method to construct phase portraits. In *Proc. of the Eleventh National Conference on Artificial Intelligence*. AAAI Press/MIT Press.