Introducing boundary conditions in semi–quantitative simulation

Giorgio Brajnik
Dipartimento di Matematica e Informatica
Università di Udine
Udine — Italy
Ph: +39 (432) 272.210
Fax: +39 (432) 510.755
giorgio@dimi.uniud.it

Abstract: Boundary value problems specifying how external influences on dynamic systems vary over time greatly extend the scope of qualitative reasoning techniques, enabling them to achieve a much wider applicability. This paper discusses conceptual and practical aspects that underlie the problem of handling boundary conditions in SQPC, a sound program for modeling and simulating dynamic systems in the presence of incomplete knowledge. Issues concerning the ontology (actions *vs.* measurements), the temporal scale (instantaneous *vs.* extended changes), the impact of discontinuity on model structure and the consequences of incompleteness in predictions are discussed. On the basis of the experimentation done so far it is claimed that given the generality of the assumptions underlying the techniques presented in the paper, and given the relatively low computational cost that is often required to solve a boundary value problem, they are viable and can be utilized to widen the applicability spectrum of Qualitative Reasoning.

1 Introduction

Though qualitative simulation [Kuipers, 1994; Bobrow, 1993] plays a crucial role in many Qualitative Reasoning (QR) tasks (such as control, diagnosis or design), few QR tools are able to deal with boundary conditions which specify how external influences on systems vary over time. In fact, except for a few cases (like [Forbus, 1989]), no qualitative simulator takes as input a description of how certain variables evolve over time, and lets them affect the simulation. These tools solve more or less sophisticated initial value problems where initial conditions of *autonomous* systems are given. Dealing with non–autonomous systems greatly extends the scope of QR techniques, enabling them to achieve a much wider applicability. In fact, they could encompass capabilities such as:

- simulating, monitoring and diagnosing systems in realistic situations, where they are affected by time–varying controls and environmental parameters;
- evaluating the effects of control laws (*i.e.* sequences of actions) applied to specific systems in dynamically changing situations;
- evaluating consistency of models of dynamic systems with respect to sequences of measurements of observable variables (for data interpretation or theory validation).

Consider for example 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. Careful evaluation of the effect of actions in critical and dynamically changing situations is crucial for decision making, and sound modeling and simulation tools could be extremely useful to support this activity. They could also be used to evaluate empirically derived models and parameters, or to forewarn of undesired possible future situations. This domain is challenging for existing approaches to modeling and simulation, for it poses many requirements. Several forms of incomplete information appear in this domain: for example, 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. Nonetheless, rough bounds on quantities are usually accurate enough to support decision. Pure qualitative reasoning techniques do not exploit the partial information available and consequently provide too weak 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. New models need to be constructed to cope with changes in relevant entities, operating modes, and modeling assumptions. Accurate results (instead of approximate ones) are needed to perform an adequate risk evaluation and forewarning.

Considering boundary conditions in qualitative simulation poses a number of basic questions that are independent from the specific framework adopted:

- *Ontology*: which ontology better suits the aim? Some approaches already known in literature exploit the concept of *action*, while others don't represent actions at all but focus on *measurements*. What is the relationship between the two concepts?
- *Temporal scale*: shall *instantaneous* or *extended actions* be allowed? The former may be adequate in certain situations but they introduce discontinuities difficult to handle, while the latter may impose a too detailed analysis.
- *Model structure*: how do boundary conditions affect the model? Changes in boundary conditions may call for changes in the model to cope with varying modeling assumptions. Do these changes require the same mechanism for revising the model as the ones required when operating regions are crossed? How do these changes interact with the chosen temporal scale?
- *Incomplete knowledge and data*: how will incompleteness in models and incompleteness in boundary conditions affect predictions? What is the sensitivity of predictions with respect to such kinds of incompleteness? What is needed to control the additional ambiguity of predictions caused by considering boundary conditions? How does qualitative time used in simulation correspond to "real" time used in observing and acting upon the system?

This paper discusses the main conceptual and practical aspects that underlie the problem of handling boundary conditions in SQPC (Semi–Quantitative Physics Compiler), an implemented program fulfilling the above mentioned requirements for modeling and simulating dynamic systems.

2 Semi–Quantitative Physics Compiler

SQPC [Farquhar and Brajnik, 1995] performs *self-monitoring simulations* of incompletely known, dynamic, piecewise–continuous systems. It monitors the simulation in order to detect violations of model assumptions. When this happens it modifies the model and resumes the simulation.

SQPC is built on top of the QSIM qualitative simulator [Kuipers, 1986; Kuipers, 1994] and extends QPC [Farquhar, 1994]. The input to SQPC is a *domain theory* and *scenario* specified in the SQPC 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. SQPC smoothly integrates symbolic with numeric information, and is able to provide useful results even when only part of the knowledge is numerically bounded. The domain theory includes *symbolic* or *numeric magnitudes* which represent specific real numbers known with uncertainty (numeric magnitudes constrain such numbers to lie within given ranges); *dimensional information*; *envelope schemas* (they state the conditions under which a specific monotonic function over a tuple of variables is bounded by a pair of numeric functions) and *tabular functions* (numeric functions defined automatically by interpolating multi–dimensional data tables). The specific system or situation being modeled is described by the scenario definition, which lists objects that are of interest, some of the initial conditions and relations that hold throughout the scenario.

SQPC employs (inheriting it from QPC) a hybrid architecture in which the model building portion is separated from the simulator. The domain theory and scenario induce a set of logical axioms. SQPC 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 SQPC concern structural relationships between objects declared in the scenario, and the computation of 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 QSIM in terms of equations and initial conditions. If any of the predicted behaviors crosses the operating region conditions the process is repeated. A new database is constructed to describe the system as it crosses the boundaries of the current model, then another complete set of active model fragments is determined and another simulation takes place.

The output of SQPC 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: (i) link a database to a refined database (obtained by adding more facts, either derived through inference rules or assumed by SQPC when ambiguous situations are to be solved); (ii) link a complete database to a state (which is one of the possible initial states for the only model derivable from the database); (iii) link a state to a successor state (this link is computed by QSIM); and (iv) link a state to a database (the last state of a behavior that has crossed the operating region to the database which 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 and each model in such paths identifies a distinct operating region of the system. SQPC is proven to construct all possible sequences of initial value problems that are entailed by the domain theory and scenario. Thanks to QSIM correctness, it produces also all possible trajectories.

3 Boundary conditions and automated modeling

The problem of performing a self-monitored simulation is extended by providing as input also a *stream of measurements* and by requiring that the output consists of *all possible* trajectories that are *compatible* with measurements.

A *measurement* is a time-tagged mapping of values to a set of variables, which can be either *exogenous*, (*i.e.* representing quantities that can be affected by external influences), or *non-exogenous*. The stream of measurements considered in a simulation must satisfy the following two assumptions: all critical points of all exogenous variables should be measured (*sampling assumption*); and measurements should be chronologically ordered.

For generality, we don't require other properties on measurements. In particular, they need not include *all* variables of the system; they need not concern each time the *same* set of variables; they need not be the result of a *periodic* sampling process, and their time tags and measured values may be expressed as intervals over the real numbers to cope with imprecise data and noise processing.

The following interdependent standpoints provide a rationale for these assumptions and are tentative answers to some of the questions raised in the introduction.

Ontology. An *action* is an activity done by some agent affecting some exogenous variable, while a *change* in such variables is the effect of an action. We prefer to explicitly represent changes and introduce only implicitly actions because appropriate treatment of changes is needed even in case actions are explicitly represented. Explicit representation of actions (like the one adopted in [Forbus, 1989]) could be useful in applications requiring the *generation* of control laws (*i.e.* deciding when to apply a certain action), an issue not tackled in this paper.

Measurements may or may not yield evidence of some action: they do it if they concern exogenous variables (the measured value may reveal that a change occurred or is occurring); they don't if they concern only non-exogenous ones.

Temporal Scales. We envision two kinds of actions (hence of changes): those with a finite duration (*extended changes*) and those occurring instantaneously (*instantaneous changes*). Both are worth considering: instantaneous changes may be used when the time-scale of the action is much smaller than the system's one and limited knowledge is available for modeling the transient during which the action takes place, or the transient is not interesting enough. For example, given a medium-term analysis (days or weeks), an in-depth investigation of the transient occurring on a dam-lake system *during* a control action of opening a gate is uninteresting. Such a change can therefore be conceptualized as instantaneous. Similarly if no knowledge is at hand for modeling the dynamics during the transient, the effects of operating an electrical switch can be conceptualized again as instantaneous. On the other hand, extended changes could be profitably used when the actual duration is known and predictions of events occurring during the action are wanted; for example, to predict what actually happens inside a servo-controlled turbine when an operator changes the power level requested to the turbine.

While actions (and changes) may be instantaneous or not, measurements are assumed to be instantaneous events. The *sampling assumption* implies that the beginning and end of an extended change are marked by measurements, whereas the occurrence of an instantaneous change is marked by a single measurement. Therefore, during a *segment* (the time interval between two consecutive measurements of the same variable) an exogenous variable may be either constant or strictly monotonic. Of course, some knowledge is required to correctly interpret a measurement (whether it marks an instantaneous change or not) since by itself a measurement does not provide this information. This knowledge derives (in the proposed framework) from properties of measured exogenous variables declared in the scenario description. Such variables may be subject either to extended changes or to instantaneous ones, but not both.

Continuity. Continuity is a fundamental assumption for qualitative reasoning techniques used to constrain the possible intra/inter-model changes that can occur in a system. In order to manage instantaneous changes, we assume that:

1. *state* variables (variables whose time derivative is included in the model) are *piecewise-C^1* (*i.e.* continuous anywhere, and differentiable everywhere but in a set of isolated points);

2. *non-state* variables are at least *piecewise-C^0* (*i.e.* continuous anywhere but in a set of isolated points);

If an instantaneous change occurs on exogenous variables $\Delta = \{V_1 \dots V_n\}$, in order to correctly deal with the transient, one needs to determine how the discontinuity propagates from Δ onto other variables of the model. Fortunately, the abovementioned continuity assumptions suffice to support a sound and effective criterion (termed *continuity suspension*) for identifying all the variables that are potentially affected by the discontinuity of variables in Δ.

Given a model $\mathcal{M}$, let us say that a variable Z is *totally dependent* on a set of variables $\mathcal{A}$ iff the model includes a non–dynamic, continuous functional relation $R(X_1, \dots X_i, Z, X_{i+1}, \dots X_n)$ with $n \geq 1$ such that $\forall i: (X_i \in \mathcal{A}$ or X_i is totally dependent on $\mathcal{A})$. For example, if the model includes the constraint *((M (+ +)) X Y Z)* then X is totally dependent on $\{Y, Z\}$. Furthermore, let $TD(\mathcal{A}) = \{X | X \text{ is totally dependent on } \mathcal{A}\}$.

Let $\mathcal{E}$ be the set of exogenous variables and $\mathcal{S}$ the set of state variables of $\mathcal{M}$. Then define $\mathcal{PD}_\Delta$ (the set of variables that are potentially affected by the discontinuity of variables in Δ) as the maximum set of variables of $\mathcal{M}$ that satisfies:

1. $\Delta \subseteq \mathcal{PD}_\Delta$ (since variables in Δ are affected by the discontinuity);
2. $\mathcal{S} \cap \mathcal{PD}_\Delta = \emptyset$ (by continuity assumption, $\mathcal{PD}_\Delta$ cannot contain any state variable);
3. $\mathcal{E} \cap \mathcal{PD}_\Delta = \Delta$ (by the sampling and continuity assumptions, unmeasured exogenous variables must be continuous);
4. $TD(\mathcal{S} \cup \mathcal{E} - \Delta) \cap \mathcal{PD}_\Delta = \emptyset$ (by definition of total dependency, if Z totally depends on a set of necessarily continuous variables, then Z must be continuous too and cannot belong to $\mathcal{PD}_\Delta$).

Continuity suspension handles discontinuous changes of variables in Δ by computing the set $\mathcal{PD}_\Delta$ so that, during a transient, variables in $\mathcal{PD}_\Delta$ are unconstrained and can therefore get any new value, whereas those not in $\mathcal{PD}_\Delta$ will keep their previous value.

Correctness of continuity suspension is easy to prove: if $\mathcal{PD}_\Delta$ were equal to the set of all the variables in the model, then no restriction would be in effect during the transient, yielding all possible value changes, including the "true" ones. Since conditions 2, 3 and 4 would remove from $\mathcal{PD}_\Delta$ only necessarily continuous variables, no variable affected by Δ will be ever removed from $\mathcal{PD}_\Delta$.

Unfortunately, continuity suspension is not complete, for the set $\mathcal{PD}_\Delta$ may include also variables that are not affected by Δ (for example, if $y = \frac{dx}{dt}$ belongs to the model and $y \notin TD(\mathcal{S} \cup \mathcal{E})$, then $y \in \mathcal{PD}_\Delta$). Rules 1–4 are not sufficiently strong to exclude certain variables from $\mathcal{PD}_\Delta$. They exclude only variables that are *necessarily continuous*, leaving in $\mathcal{PD}_\Delta$ those that are *necessarily discontinuous* (like those in Δ) plus those that are *possibly discontinuous* (like y). On the other hand, since $\mathcal{PD}_\Delta$ is determined on the basis of the model holding before the transient takes place, and nothing is known about what happens during the transient, soundness demands that only necessarily continuous variables are removed from $\mathcal{PD}_\Delta$.

Non–exogenous variables can be measured too, but unlike exogenous ones their behavior during a segment is not known in advance and they do not introduce discontinuities. Such measurements greatly refine predictions (by restricting predicted ranges or by rejecting predictions that are inconsistent with measured values), if they simultaneously involve several variables.

Model structure. Actions may affect the model structure in two ways.

First, they may affect the set of modeling assumptions, calling for a revision of model structure. Model revision may occur either during an extended change (*e.g.* when a valve is being opened the flow regime of the fluid may change from laminar to turbulent), or during the transient of a discontinuous change (*e.g.* if opening a valve is an instantaneous action, then a discontinuous change propagates onto other variables, and new models need to be defined to accurately cover the possible consequences of such a quick action). In the former case no discontinuity is introduced, reducing model revision to the "normal" revision triggered by the crossing of an operating region (in the previously mentioned example, the region being crossed refers to the variable *Reynolds–number* becoming greater than a certain threshold). In the latter case (model revision occurring during an instantaneous change) the discontinuity in $\mathcal{PD}_\Delta$ weakens the process of determining the next model(s): referring to the previous example, the discontinuous change in valve section affects other variables (like fluid flow, speed, etc.) whose "next" value will not be constrained by continuity, making it difficult to ascertain whether the flow, after the change, will still be laminar or will became turbulent. In fact, though conceptually being determined by state variables, variables in $\mathcal{PD}_\Delta - \Delta$ usually cannot be given a unique new value if continuity is relaxed because of the inherent ambiguity of the qualitative algebra of signs.

Second, two modeling decisions may be inconsistent. The decision of determining the set of exogenous variables and the decision of determining the set of state variables may lead to two kinds of conflicts: (i) if some state variables are treated as exogenous the resulting model may be overconstrained. Analytically this would lead, in general, to a badly defined model whereas qualitatively this is not necessarily true, since the incomplete knowledge used in the model and state may supply additional degrees of freedom; (ii) state variables may get values which are incompatible with those measured for exogenous variables. Such discrepancies are an indication that the model is clearly a wrong description of the system under study. Both kinds of conflicts are easily identified, though their automatic resolution is far from being trivial since it requires a modeling choice.

4 Semi–Quantitative Boundary Problems

In order to perform a simulation guided by measurements the user has to declare which are the exogenous variables, which are their properties and how to acquire their measurements. This is done in the scenario declaration form (see figure 1). The property of being piecewise–constant or piecewise–monotonic is invariant in a scenario.

Including a new measurement in a simulation may lead to a model revision and/or a state change. SQPC handles each measurement as a transition (called measurement–transition, or *M–transition*) between two models. When building a new database SQPC adds measured values in the database and recognizes ongoing actions by looking ahead in the measurement stream for each piecewise–monotonic variable[1]. The new model will include appropriate constraints: `constant` for piecewise–constant variables; `constant`, `increasing` or `decreasing` for piecewise–monotonic ones, according to the difference of measured values at the ends of the segment.

Two decisions are critical when performing a measurement–guided simulation: realizing when an M–transition occurs and deciding how to revise the model and its initial state.

Recognizing M–transitions. An M–transition occurs when simulation time T_s (the time of the last state being simulated, S) and the time T_m of the next measurement are the same. Unfortunately, unless predictions are very precise, this comparison is usually ambiguous, for time ranges might be overlapping. Even if measured values were extremely precise (*i.e.* singleton ranges), as long as predicted ranges for time have positive length, they would be a source of ambiguity. In the worst case the three possible orderings between T_s and T_m need to be generated.

Two situations may occur when deciding whether to fire an M–transition: the measurement is taken while some action is ongoing (*i.e.* some exogenous variable is moving towards its final — with respect to the ongoing action — value) or not. In the former case, information of the value of such variables in state S can be used to reduce the ambiguity in T_s and T_m: for example, if such variables reach their values in S and their values are measured at time T_m, then it follows that $T_s = T_m$. In the latter case (only piecewise–constant or non–exogenous variables are involved in the measurement), or when ambiguity is not completely resolved, all three possibilities are explicitly represented (the non–overlap situation is straightforward, and subsumed by the overlap one):

- $T_m = T_s$, and S is indeed the state involved with the measurement; if T_m and T_s overlap, $T_m = T_s$ is asserted in S (usually restricting T_s).
- $T_m < T_s$, which means that the simulation advanced too much. Since the M–transition check is performed at each point state, S must be the first point state whose T_s is greater than T_m. A new state S' is generated by copying it from the predecessor of S (an interval state) and $T_s = T_m$ is asserted on S'. S is discarded.
- $T_m > T_s$, meaning that we should keep on simulating. No M–transition occurs from S, and $T_m > T_s$ is asserted on S.

Revising the model and generating an initial state. When a model has to be revised on the basis of a measurement, the specific details on how it does change depend on which variables are measured and if there are ongoing actions. There are four cases:

1. the next measurement includes only *non exogenous variables*. In this case the model does not change, qualitative values inherited by variables across the M–transition do not change either, and the only thing that changes is their new ranges (*i.e.* measured and predicted ranges are intersected in the initial state);
2. the next measurement includes only *piecewise–constant exogenous variables* Δ. Continuity suspension is applied across the M–transition by (i) assigning measured values to variables in Δ, (ii) inheriting previous values for variables not in $\mathcal{PD}_\Delta$, and (iii) leaving variables in $\mathcal{PD}_\Delta - \Delta$ unspecified. When SQPC constructs a database from the model and the state originating the M–transition, the usual SQPC refinement mechanisms (including QSIM's state completion) will be used to deduce appropriate initial values for variables in $\mathcal{PD}_\Delta - \Delta$.
3. a set of *piecewise–monotonic exogenous variables* $\mathcal{M}$ are affected by some ongoing actions. In order to revise the model, SQPC does a lookahead searching for the next measured value for

[1] The depth of such a lookahead is user–defined, and may range from the next absolute measurement to the measurement ending the next segment of each piecewise–monotonic variable.

```
(DefScenario LakeTravis
  :entities ((travis        :type lakes)
             (colorado-dn :type rivers)
             (colorado-up :type rivers)
             (mansfield   :type dams)
             (turbine-1   :type mansfield-turbines))
  :structural-relations ((flows-into colorado-up travis)
                         (connects mansfield travis colorado-dn)
                         (has-valve mansfield turbine-1))
  :landmarks ((top-of-dam :variables ((stage travis)) :value 714)) ; ft
  :initial-conditions ((= (power turbine-1) 20)                   ; Mw
                       (= (stage travis) (690.25 690.3))          ; ft
                       (= (flow-rate colorado-up) (900 950))      ; cfs
                       (= (base turbine-1) 564))                  ; ft
  :exogenous-variables
     (((power turbine-1)         :type :pw-constant)
      ((flow-rate colorado-up) :type :pw-monotonic))
  :measurements (((7.0e5 7.01e5)          ; sec
                  ((power turbine-1) 10))  ; Mw
                 ((4.32e6 4.33e6)          ; sec
                  ((flow-rate colorado-up) (400 420))))          ; cfs
  ... )
```

Figure 1: Declaration of exogenous variables in scenario definition (clause `:exogenous-variables`): `(power turbine-1)` is declared *piecewise-constant* while `(flow-rate colorado-up)` is *piecewise-monotonic*. Two measurements are given (clause `:measurements`): one after approx. 8 days (between 7.0e5 and 7.01e5 sec.) regarding an instantaneous action which brings `(power turbine-1)` to the value of 10 Mw, the other regarding an action lasting approx. 50 days (4.32e6 and 4.33e6 sec.) specifying a decrease of `(flow-rate colorado-up)` from its initial value to a value comprised between 400 and 420 cfs.

each variable in $\mathcal{M}$. By comparing their current values with measured ones, appropriate time-dependent constraints (saying that a variable is either *increasing*, *decreasing* or *constant* on its next segment) are added to the model (if no next measurement is available the variable is assumed `constant`). The new model is then initialized with values inherited from the transition state, since all variables are continuous across the M–transition.

4. *any combination of previous cases (1, 2 and 3).* This is dealt with by a straightforward combination of respective operations, since there is no complex interaction between the effects of simultaneous measurements of variables having different properties.

4.1 Implementation issues

The solution outlined above leads to two pragmatic issues. First, SQPC performs a model revision step for each considered measurement. Since model revision steps are expensive in terms of computing resources (empirically, they consume up to 75% of the time required by a simulation), it is worth investigating whether this activity can be made more efficient. Fortunately, it turns out that model revision triggered by M–transition is limited and well defined. On one hand, if no piecewise–constant variables are involved in the measurement, the only part of the model that is subject to change are the constraints on exogenous variables and their quantity spaces. No complex reasoning is needed to generate the new model nor its initial state: both can be directly derived from previous ones. On the other hand, if the measurement involves some piecewise–constant variables, propagating their discontinuities onto other variables may cause ambiguous evaluation of operating conditions of model fragments, leading to expensive branching in simulation. Even in this case, however, there is a simple syntactic criterion that can be used to detect whether the discontinuity affects the set of active model fragments. In fact, if variables in $\mathcal{PD}_\Delta$ are not used in conditions of any model fragment, then no model fragment depends on them, the model structure does not change and continuity suspension suffices to compute the next state. This criterion has a dramatic effect on run–times: an activity which requires few minutes is performed in just a few seconds.

Second, measurements introduce a number of distinctions that would go unnoticed in a non–guided simulation.
First, each measured value is normally associated to a landmark, which needs to be totally ordered in respective quantity space. In general, increasing the cardinality of quantity spaces increases the number of distinctions that the qualitative simulator does. In SQPC landmark creation can be disabled across M–transitions, reducing the resolution of the output (since variables' values across M–transitions are

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

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

not represented as landmarks labeled with numeric ranges), but reducing also the ambiguity that can occur when suspending continuity.

Second, a three–way branch occurs if simulation time overlaps with measurement time. One branch is marked with the assumption $T_m = T_s$, where T_s is the time of a qualitative event (*e.g.* some variable reaching a landmark). Though theoretically sound, the probability that a measurement — an instantaneous event — is taken at the same time of an independent, instantaneous qualitative event (*e.g.* measuring a gate opening exactly when the lake stage reaches a threshold) is infinitesimal. This is another sort of distinctions that can be neglected without much loss of information.

Third, another sort of ambiguity is caused by distinctions made on order relationships between overlapping ranges of consecutive measurements of an exogenous variable. Special purpose user–defined predicates can be used by SQPC for comparing two overlapping ranges in order to reduce ambiguity.

5 An example

We will demonstrate SQPC on a problem regarding the domain of water supply control. 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 the effects of some actions in a "what–if" scenario (figure 1). We are given an initial level for Lake Travis (a value between 690.2 and 690.3 ft), a rough initial inflow from the Colorado river (between 900 and 950 cfs) and an initial requested rate of 20 Mw for the power delivered by the hydroelectric plant. In addition it is known that the input flow is decreasing — its minimum rate has been estimated between 400 and 420 cfs after 50 days. The task is to determine what happens to the lake level and evaluate the effect of reducing the requested power from 20 to 10 Mw after 8 days.

Several model fragments describe the behavior of lakes, rivers, dams, turbine, etc. and envelope schemas provide numeric bounds on relations between quantities. Most envelopes are derived from tabular data resulting from engineering estimates. Table 1 partially describes the behavior of turbines in Mansfield Dam.[2] In this example, tables are interpolated stepwise by SQPC to provide piecewise-constant (rather imprecise, but accurate) upper and lower bounds. Turbines are controlled by servo-mechanisms 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 it drops below the minimum threshold for a given power output then less power is released. Different sets of model fragments capture these operating modes accurately.

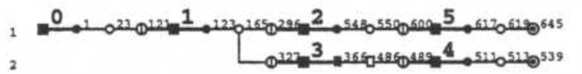

Figure 2: Two behaviors are predicted for the scenario, ending both in quiescent states. Each involves four models (black squares), two M–transitions and one transition (from model 2 to 5 for the first behavior, from 1 to 3 for the second one) from a servo-controlled to a non controlled operating regime of the turbine.

Figure 2 shows the two predicted behaviors. They are generated because of the time–ambiguity between the second measurement and the transition of the turbine to a new operating region (the latter event occurring between 9 and 87 days). Figure 3 shows the time plot of some of the variables in the first behavior. Under the specified boundary conditions the power level of 20 Mw will surely be maintained until time `T1`, the time of the first measurement (8 days); then, though reducing the requested power, eventually there will be insufficient hydraulic pressure to supply the requested power. This will happen for the first behavior (figure 3) at time `T3`, between 50 and 76 days (*i.e.* after the measurement of the input flow rate is taken). For the second behavior, not shown, after at least 8 days and not beyond 50 days (*i.e.* before the measurement). Finally, the lake system reaches equilibrium with the lake level stabilized between 568 and 588 ft.

Notice the instantaneous change occurring on variable `TURBINE-1.POWER` at time `T1` which affects other variables like `TURBINE-1.DISCHARGE-RATE`. Continuity suspension is applied to these variables and their

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

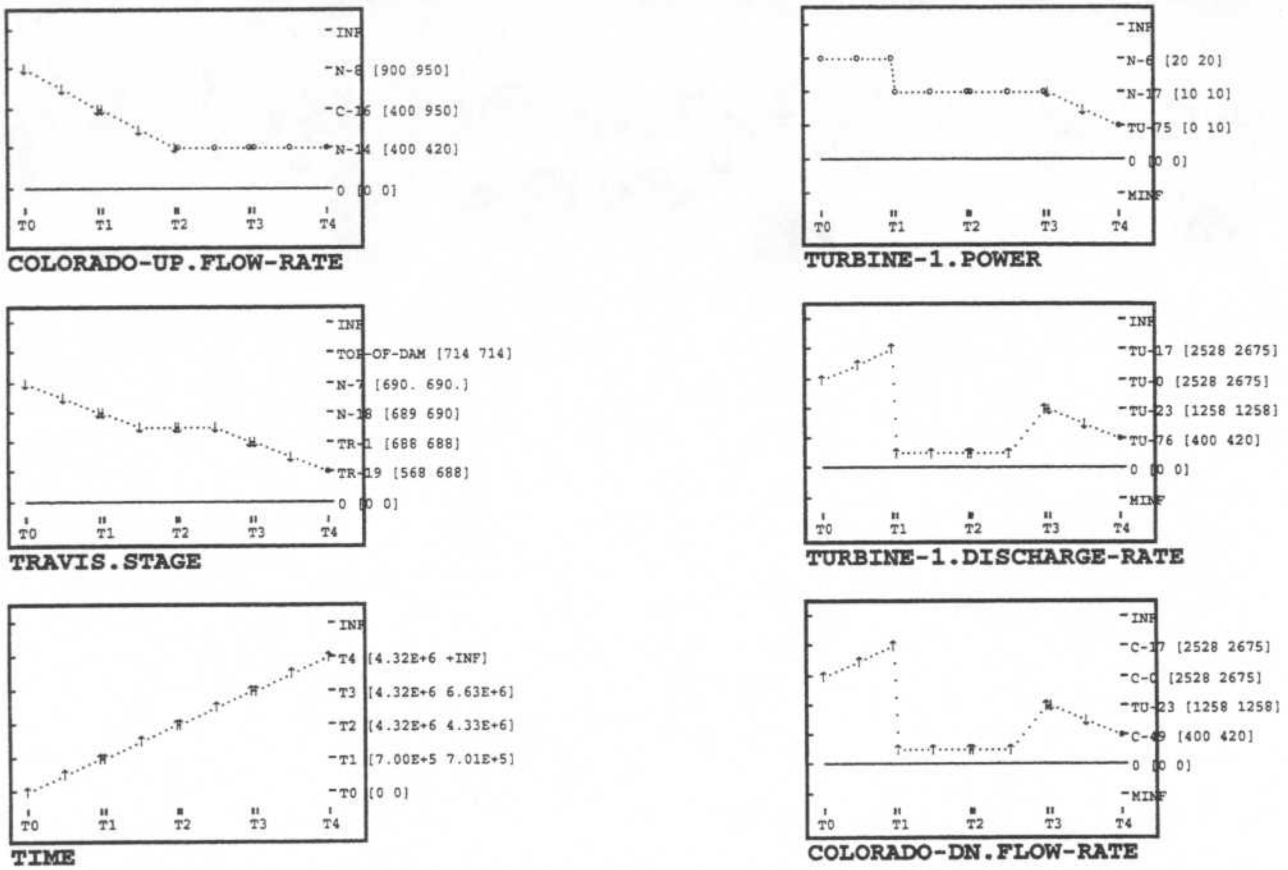

Figure 3: Plot of some of the variables for the first of two behaviors predicted for the scenario. The first M–transition occurs at time `T1` (notice the sudden drop of `TURBINE-1.POWER` and affected variables `TURBINE-1.DISCHARGE-RATE` and `COLORADO-DN.FLOW-RATE`). The second one at `T2`, where `COLORADO-UP.FLOW-RATE` becomes constant. Finally, at `T3` a transition occurs to a region where the turbine is no longer servo–controlled (`TRAVIS.STAGE` reaches the threshold 688 ft). `TURBINE-1.POWER` is no longer treated as an exogenous variables (since the servo–mechanism does not operate any more) and it becomes a dependent variable.

behavior across the M–transition occurring at `T1` is not constrained. On the other hand, the second M–transition occurs when the input flow rate reaches its lowest value and since it involves a piecewise–monotonic variable, all variables are continuous across the transition.

If measurements included observations for other (non–exogenous) variables, then the ambiguity in times could disappear and certain ranges shrink. For example, if the second measurement were

```
((4.32e6 4.33e6)
 ((flow-rate colorado-up) (400 420))
 ((stage travis) (688.5 688.7)))
```

(*i.e.* of the same input flow–rate, taken at the same time [4.32e6 4.33e6] but involving in addition the non–exogenous variable `stage travis`), then only the first behavior would be consistent with the observed value of the lake stage. In fact, only in the first behavior the ordering of the events "head reaching the minimum threshold (when *stage*= 688 *ft*)" and "measurement at time [4.32e6 4.33e6]" is compatible.

5.1 Implementation status

SQPC is fully implemented in Lucid Common Lisp as an extension to QPC, which in turn uses QSIM. We are currently experimenting SQPC in the water supply control domain and in economics. It has been run on several examples comparable to the one shown in this paper.

The runtime for this example is around 8 minutes on a Sun 20. The bulk of this time is spent computing order relations with interpreted rules during the three full–fledged modeling steps. Using a special purpose inequality reasoner, whose implementation is underway, will result in a substantial (orders of magnitude) speedup.

6 Related work

Several efforts facing the issues discussed in this paper have been reported in literature, but none of them covers the whole problem or provides viable and sound solutions.

[Kuipers and Shults, 1994] and [Forbus, 1989] provide some means to represent external influences on a system and to implement a guided simulation. *Expressive Behavior Tree Logic* [Kuipers and Shults, 1994] is a temporal logic (integrated in QSIM) that can be used to specify, in logical statements, the qualitative behavior of variables and have QSIM generate a simulation compatible with them. This method, still under development, is complementary with respect to the one presented in this paper since it does not handle model revisions caused by external influences nor quantitative information.

Forbus [Forbus, 1989] explicitly introduces the concept of action, with pre and post–conditions. The purely qualitative total envisionment that is generated includes all possible instantiation of known actions. Forbus allows only instantaneous actions and adopts heuristic criteria to handle discontinuities. No provision is made to handle quantitative information, nor to focus the envisionment process.

One work that centers on discontinuities either caused by external influences or autonomous, is that of [Nishida and Doshita, 1987]. Nishida and Doshita describe two methods for handling discontinuities: (i) approximating a discontinuous change by a quick continuous change and (ii) introducing mythical states to describe how a system is supposed to go through during a discontinuous change. The former requires a complex machinery to compute the limit of the quick change, whereas the second is based on heuristic criteria for selecting appropriate states.

Many other approaches have been described which aim to interpret measurements of dynamic systems. Some of them do not perform a simulation, like DATMI [DeCoste, 1991] which interprets measurements with respect to a total envisionment. Others, like MIMIC [Dvorak, 1992], though performing a semi–quantitative simulation and refining predictions with measured data, do not cope with model revisions nor with guided simulations. (Indeed, some of the ideas presented in SQPC descend from techniques first applied in MIMIC, *e.g.* for integrating measurements into simulations.)

7 Conclusion

The main issues arising from considering measurements in a self–monitoring simulation have been discussed. Boundary conditions expressed in terms of instantaneous or extended changes of exogenous variables are used to guide and refine an online or offline (depending on the depth of the lookahead) incremental simulation of incompletely known lumped–parameters systems.

From the conceptual analysis and from the experimental activity done so far it appears that considering boundary conditions by itself does not aggravate the uncertainty of predictions. If measurements are added to a scenario of an incompletely known situation, the precision of the output does not change significantly. Nor does it change if measured values become less precise. It does worsen considerably though if uncertainty affects the time of events, because of range overlap, which is dealt with by representing the different orderings of events. If inter–dependent variables are simultaneously measured, however, the output precision increases since ranges can be restricted and inconsistent behaviors refuted. Furthermore, it would be straightforward to extend SQPC in such a way to suggest to the user when some additional measurement would be needed to reduce the ambiguity.

Discontinuous changes are comparatively more difficult to handle. The adopted criterion to handle the transient, continuity suspension, limits the combinatorial growth of possible trajectories taking place during the transient by restricting the number of variables that could be affected by discontinuities. The method is correct and, though being incomplete, it has not proven yet to be a bottleneck. Furthermore, though being used only on M–transitions, continuity suspension is a general criterion that could be used also to handle other kinds of transitions imposing discontinuous changes on variables (for example to model abrupt faults).

Computationally, the cost of handling non–autonomous systems is often relatively low (even in cases where a limited model revision is needed). It may well happen, however, that dealing with instantaneous changes requires a complex modeling activity. Even though appropriate precautions are taken to limit the number of such activities, a substantial number of measurements with ambiguous events quickly leads to intractable problems.

In conclusion, we believe that given the generality of the assumptions underlying the techniques presented in the paper, and given the relatively low computational cost that is often required to solve a boundary value problem, it seems worthwhile employing them to widen the applicability spectrum of Qualitative Reasoning.

Acknowledgments

Part of the research reported in this paper took place while I was visiting the UT Qualitative Reasoning Group, at Austin, TX during 1992. I'm indebted with Ben Kuipers for many illuminating discussions, and with Adam Farquhar for letting me use his QPC program. Many thanks to Dan Clancy and Bert Kay for making me understand several parts of QSIM and to Franco Ceotto for his help in implementing SQPC.

References

[Bobrow, 1993] D. Bobrow. Special volume: AI in perspective. *Artificial Intelligence*, 59(1–2):103–146, 1993.

[DeCoste, 1991] D. DeCoste. Dynamic across–time measurement interpretation. *Artificial Intelligence*, 51, 1991.

[Dvorak, 1992] Daniel L. Dvorak. Monitoring and diagnosis of continuous dynamic systems using semi-quantitative simulation. Technical Report AI 92-170, Artificial Intelligence Laboratory, The University of Texas at Austin, 1992.

[Farquhar and Brajnik, 1995] Adam Farquhar and Giorgio Brajnik. A semi–quantitative physics compiler. In *Tenth International Conference on Applications of Artificial Intelligence in Engineering*, Udine, Italy, July 1995. Presented also at the Eighth International Workshop on Qualitative Reasoning on Physical Systems, 1994, Nara, Japan.

[Farquhar, 1994] A. Farquhar. A qualitative physics compiler. In *Proc. of the 12th National Conference on Artificial Intelligence*, pages 1168–1174. AAAI Press / The MIT Press, 1994.

[Forbus, 1989] K. Forbus. Introducing actions into qualitative simulation. In *IJCAI–89*, pages 1273–1278, 1989.

[Kuipers and Shults, 1994] B. Kuipers and B. Shults. Reasoning in logic about continuous systems. In *8th International Workshop on Qualitative Reasoning about physical systems*, pages 164–175, Nara, Japan, 1994.

[Kuipers, 1986] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.

[Kuipers, 1994] B. Kuipers. *Qualitative Reasoning: modeling and simulation with incomplete knowledge.* MIT Press, Cambridge, Massachusetts, 1994.

[Nishida and Doshita, 1987] T. Nishida and S. Doshita. Reasoning about discontinuous change. In *AAAI87*, pages 643–648, 1987.