

Research Article

Qualitative Reasoning for Quantitative Simulation

Mehmet Fatih Hocaoglu 

Department of Industrial Engineering, Faculty of Engineering and Natural Sciences, Istanbul Medeniyet University, Istanbul, Turkey

Correspondence should be addressed to Mehmet Fatih Hocaoglu; hocaoglumf@gmail.com

Received 11 May 2017; Revised 5 August 2017; Accepted 5 September 2017; Published 1 January 2018

Academic Editor: Luis Carlos Rabelo

Copyright © 2018 Mehmet Fatih Hocaoglu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Qualitative simulation is a well-known reasoning technique that involves the use of simulation technologies. Reasoning is made to determine qualitative values and change directions of system variables, and it is done for each time point and time interval following the time point. Qualitative variables possess continuous qualitative value sets that are discretized by landmark points. Qualitative simulation uses qualitative time representation and its quantitative value is of no interest. The main purpose of this study was to develop a technique to determine time steps for a quantitative simulation under guidance of qualitative information. The proposed technique determined time advances using qualitative and quantitative information together to obtain a robust time step as wide as possible for simulation time advances. For this purpose, sign algebraic properties and derivation roots of quantitative equations and qualitative variable values with their change directions were used to compute time advances. In the approach, qualitative simulation determined landmark points to be advanced, and quantitative simulation calculated the duration required. Using the proposed algorithm, the simulation is advanced instead of iterating simulation time for a predefined time step and checking whether or not there is any activity in the interval, directly to the time points that are qualitatively different.

1. Introduction

Qualitative simulation (QS) is defined as a reasoning technique using simulation, and it generates possible value and their change directions that are defined as state on time axis for continuous event systems, using qualitative information of the system being simulated, and the states following each other in time point and time interval sequences constitute behavior trees [1, 2]. Qualitative differential equations (QDEs) utilized by QS represent a function family, not a specific function; they accommodate a set of functions beneath. Time axis in QS is constituted by time points and interval between time points. Qualitative state descriptions are computed for both time points and intervals, and they constitute a simulation trajectory. Because of the incomplete information used by QS, each qualitative variable yields next value set rather than a single value. As a result of this, simulation execution gives possible alternative trajectories.

The main motivation of this study is to develop a simulation time management technique that advances simulation steps to system discontinuity points that are discovered by

qualitative reasoning and sign algebra. In contrast to the fixed time advance mechanism for continuous event simulation, this approach maps continuous event simulation into discrete event simulation by discovering events and discontinuity points using qualitative information with quantitative functional analysis. It is known that every time step in continuous event time management may possibly not compute a qualitatively distinct state vector. As seen in the bathtub example given in Section 5, although different numerical values for each time point in a time interval $[t, t + \Delta t]$ are calculated, the time interval represents the same qualitative state vector, because all numerical values between t and $t + \Delta t$ are mapped to the same qualitative interval. For example, qualitative meaning is that water level is between zero and full and it is increasing ($((0, \text{full}), \text{inc}))$ (different cases for tub problem are given in Table 2). Until a qualitative landmark point is reached, qualitative interpretations and mathematical properties (sign of derivatives, change direction) for the time interval remain the same.

In this study, a time advance algorithm is developed based on qualitative state vectors and qualitative algebra. The

TABLE 1: Qualitative value (QV) transition rules.

P-Successors $QV(v, t_i) \Rightarrow QV(v, t_{i+1})$			I-Successors $QV(v, t_i, t_{i+1}) \Rightarrow QV(v, t_{i+1})$		
P.1	$\langle l_j, \text{std} \rangle$	$\langle l_j, \text{std} \rangle$	I.1	$\langle l_j, \text{std} \rangle$	$\langle l_j, \text{std} \rangle$
P.2	$\langle l_j, \text{std} \rangle$	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	I.2	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	$\langle l_{j+1}, \text{std} \rangle$
P.3	$\langle l_j, \text{std} \rangle$	$\langle (l_{j-1}, l_j), \text{dec} \rangle$	I.3	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	$\langle l_{j+1}, \text{inc} \rangle$
P.4	$\langle l_j, \text{inc} \rangle$	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	I.4	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	$\langle (l_j, l_{j+1}), \text{inc} \rangle$
P.5	$\langle l_j, \text{dec} \rangle$	$\langle (l_{j-1}, l_j), \text{dec} \rangle$	I.5	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	$\langle (l^*)^1, \text{std} \rangle$
P.6	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	I.6	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	$\langle l_j, \text{std} \rangle$
P.7	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	I.7	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	$\langle l_j, \text{dec} \rangle$
P.8	$\langle (l_j, l_{j+1}), \text{std} \rangle$	$\langle (l_j, l_{j+1}), \text{std} \rangle$	I.8	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	$\langle (l_j, l_{j+1}), \text{dec} \rangle$
P.9	$\langle (l_j, l_{j+1}), \text{std} \rangle$	$\langle (l_j, l_{j+1}), \text{inc} \rangle$	I.9	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	$\langle (l_j, l_{j+1}), \text{std} \rangle$
P.10	$\langle (l_j, l_{j+1}), \text{std} \rangle$	$\langle (l_j, l_{j+1}), \text{dec} \rangle$	I.10	$\langle (l_j, l_{j+1}), \text{std} \rangle$	$\langle (l_j, l_{j+1}), \text{std} \rangle$

¹ l^* is a value between l_j and l_{j+1} .

TABLE 2: Differences between cases.

	Case 1	Case 2	Case 3
Inflow	Constant	Constant	Function dependent
Outflow	Constant because of constant inflow	Constant because of constant inflow	Function dependent
Interaction with the system	No interaction	Inflow/outflow valve clearance is changed exogenously	No interaction

main idea is to calculate a time step as wide as possible to reach the next qualitative state vector point (named as landmark points) that is close to the current state value of simulation model depending on its change in direction instead of reaching the qualitative value by many small time steps that the quantitative equations are solved for. In some sense, this involves mapping a continuous space into a discrete one and allowing it to reach a state point directly, instead of traversing from one state to another by small time steps.

In Section 2, qualitative simulation is summarized and Section 3 gives brief information about the solution procedures for continuous event simulation with a time management taxonomy. While Section 4 gives fundamentals of the proposed qualitative time management approach, the approach is clarified by an example in Section 5. In the last section, advantages and disadvantages of the proposed solution are discussed.

2. Qualitative Simulation

Qualitative simulation (QS) is accepted as one of the qualitative reasoning (QR) techniques which are utilized to model human reasoning approach. The aim of QS is to generate possible future qualitative behaviors of any kind of system. During behavior generation, initial states, qualitative constraint set, and qualitative value set that are associated with qualitative state variables of the system are handled as an input set.

Several approaches and ontologies are proposed to accomplish this purpose by researchers. Device centered ontology [3], process centered ontology [4], and constraint based approach [2] are well-known methods. QSIM algorithm, as a constraint based approach, is one of the most widely used algorithms in the developed techniques [1]. In constraint based qualitative simulation, qualitative variable values are represented as a discrete value set consisting of landmark points. A landmark point represents a qualitatively distinct model state. QS is performed from time point to time interval and continued. In Table 1, qualitative value transition rules from time point to time interval (P) and time interval to time point (I) are depicted. As seen in the table, a qualitative value of variable v is represented by a value or an interval with a change direction. The QSIM representation was developed, in part, to make the abstraction relationship between the qualitative representations and the theory of differential equations explicit and precise. Qualitative differential equation (QDE) is a tuple of four elements, $\langle V, Q, C, T \rangle$, each of which will be defined below [2], and they are used to filter out spurious behaviors.

- (i) V is a set of variables, each of which is a “reasonable” function of time.
- (ii) Q is a set of quantity spaces, one for each variable in V .
- (iii) C is a set of constraints applied to the variables in V . Each variable in V must appear in some constraint.
- (iv) T is a set of transitions which are rules defining the boundary of the domain of applicability of the QDE.

The variables in a QDE represent time-varying quantities. In order to be possible for qualitative reasoning, variables must be restricted to correspond to functions of time whose behavior is reasonable. First, a reasonable function must be continuously differentiable. Transitions between domains will make it possible to represent isolated discontinuities between regions of continuous behavior. Second, it is convenient to consider each variable, including time, to range over the extended real number line, R^* , which includes the endpoints $-\infty$ and $+\infty$. The function $v : [0, \infty] \rightarrow R^*$ is defined to be continuous at ∞ exactly if $\lim_{t \rightarrow \infty} v(t)$

exist. Using extended real number line allows the use of the invariant that the value of a variable always lies in a closed interval defined by landmark values. Third, there will be a need to avoid functions of time whose qualitative properties change infinitely often in a finite interval. All these properties define a reasonable function.

Definition 1. Where $[a, b] \subseteq R^*$, the function $f : [a, b] \rightarrow R^*$ is a reasonable function over $[a, b]$ if

- (1) f is continuous on $[a, b]$,
- (2) f is continuously differentiable on $[a, b]$,
- (3) f has only finitely many critical points in any bounded interval,
- (4) the one-sided limits $\lim_{t \rightarrow a^+} f'(t)$ and $\lim_{t \rightarrow b^-} f'(t)$ exist in R^* . $f'(a)$ and $f'(b)$ are defined to be equal to these limits.

Proposition 2. Any reasonable function $f : [a, b] \rightarrow R^*$ defined over a bounded interval $[a, b]$ has a finite set of landmark time points, $a = t_0 < t_1 < \dots < t_n = b$, in its domain, and a finite set of landmark values, $l_1 < l_2 < \dots < l_k$, in its range. A qualitative variable v and its quantity space $l_1 < l_2 < \dots < l_k$ define a symbolic language, a finite set of meaningful distinctions, for describing the values of a reasonable function $f(t) : [a, b] \rightarrow R^*$. At any time t , the qualitative value of $f(t)$ can be defined in terms of its ordinal relationships with the landmarks in its quantity space and its direction of change.

Definition 3. The qualitative value of $f(t)$, $QV(f, t)$, with respect to the quantity space, $l_1 < l_2 < \dots < l_k$, is the pair $\langle qmag, qdir \rangle$, where

$$qmag = \begin{cases} l_j, & \text{if } f(t) = l_j, \text{ a landmark value,} \\ (l_j, l_{j+1}), & \text{if } (ft) \in (l_j, l_{j+1}), \end{cases}$$

$$qdir = \begin{cases} \text{inc,} & \text{if } f'(t) > 0, \\ \text{std,} & \text{if } f'(t) = 0, \\ \text{dec,} & \text{if } f'(t) < 0. \end{cases} \quad (1)$$

Qualitative reasoning by simulation, in other words, qualitative simulation is achieved beginning from a time point and inferring qualitative variable values for the time interval following and continuing so. The inference about qualitative values of variables at each simulation step is done using the rules shown in Table 1. P-Successors rule set defines time point to time interval transitions and carries out I-Successors rule set and vice versa. As an example, P.1, P.2, and P.3 rules show a variable value, which is equal to a landmark value and being stable on that point, it keeps its value and change direction, increases through the next landmark point, and decreases through the previous landmark point, respectively, in time interval following that time point.

3. Simulation Strategies and Time Management Techniques

There are three available strategy types of discrete event simulation which use discrete event simulation language. (1) Event-scheduling strategy deals with the prescheduling task of the events. Activating events with the help of test on the global state is out of the context of this strategy. (2) Activity scanning strategy is different from event-scheduling strategy because of the fact that the conditions of events are arranged by contingency test. Besides that, the scheduling process is carried out in time as well. (3) Process-oriented strategy is a combination of event-scheduling and activity scanning strategy mentioned above. There is always a distinction between the continuous time of real world and simulation time in the simulation environment. Simulation time can be established either using discrete time of continuous events or discrete events. In a discrete time simulation system, time variable is increased at the level of Δt from the point of t so the last time period is $t + \Delta t$. Changes in states are calculated between $[t, t + \Delta t]$ points at the $t + \Delta t$ time. In brief, discrete event simulation progress is computed from the occurrence time of one event until the occurrence time of the next event, in other words, from one state change to the next state. That is why discrete event driven simulation is more advantageous due to the fact that simulation model advances from one state to the another state. During this process, computations between inactive periods are not performed [5]. State transition which occurs by an event saves the model features the same state till either an internal transition happens or another event receives [6].

In Agent driven Simulation Framework (AdSiF) which is solution based model for the proposed approach, discrete/continuous simulation approach and continuous event simulation approach are developed by scheduling a pseudo-event for each time interval and each time point (Hocaoglu, 2005, 2011). It means that execution is succeeded in the same way with discrete event simulation. By the way, the pseudoevent can be defined as a description of discretization point on a continuous function trajectory. Mentioned points are candidates for event publishing. Time step Δt is accepted as a dynamic parameter and it changes for each kind of simulation models. Every simulation model can define its own time step according to its time requirement that are based on ODEs or equation set.

The related works related combination of qualitative reasoning with quantitative simulation are commonly studied using interval algebra enveloping qualitative state variables with numerical intervals. First studies in this field are performed by means of a specific tool given as follows:

- (i) Nsim (H. Kay and Kuipers, 1993; Herbert Kay, 1996)
- (ii) Q2 (B. J. Kuipers and Berleant, 1988)
- (iii) Q3 (Berleant, 1991)
- (iv) SQSIM and MSQUID (Herbert Kay and Ungar, 2000).

The essential solution is carried out by integrating landmarks and monotonic functions. Qualitative behaviors created by

QSIM perform as a structure in order to represent qualitative information. These behaviors serve by marking both landmark values with real intervals and monotonic function constraints with real functions. The quantitative limits can be enlarged across constraints in order to derive narrower limits or in order to realize a contradiction and separate out the existing behaviors (Benjamin J. Kuipers, 1993). NSim (Numerical Simulator Using Interval Method) is developed by improving some features of QSIM simulation tool which generates a numerical bounding envelope for each parameter in the QSIM model extended with partial quantitative information.

The concept of qualitative discrete event simulation was explored by Ingalls [7–9] who developed a simulation methodology that combines discrete event simulation with qualitative simulation using temporal interval as simulation time specification. Temporal interval allows the user to define time as an interval in the simulation. This means that the current state of the simulation can occur at any time during the interval defined by $t = [t^-, t^+]$. Temporal intervals are represented by the modeling of their endpoints, assuming, for any interval t , the lesser endpoint is denoted by t^- and the greater by t^+ .

The approach allows the modelers to express times symbolically for both situations when timing of events is unknown and when timing is known, though this can vary. The time advance values could be expressed as linear polynomials such as $1, 2, s, t, 2s, t + s, 2t - s + 9$, which must be evaluated to nonnegative real numbers [10]. The time advance in this approach highly depends on time resolution. The interval given is discretized by minimum time step and each step yields a simulation trajectory branch. From this respect, if the interval $[t^-, t^+]$ is spanned by Δt , it gives the number of $(t^+ - t^-)/\Delta t$ trajectory branches. It is assumed that there is a state transition or an event occurrence at related time step.

QDE and activity-tracking paradigm offer solutions to advance simulation time close to discrete event simulation. QDE solutions are based on discretizing time step by Δt minimum possible time steps and computing the likelihood for each step and finally generating parallel simulation trajectories [9]. Activity-tracking paradigm offers a quantization approach of change of the parameters in the temporal and spatial dimension [5].

It is known that the symbolic discrete event system uses the polynomial representation. Since QDEs define neither any quantitative time based relationships nor numerical relationships between qualitative variables, ordinary (quantitative) differential equations (ODEs) are associated with QDEs. Association is achieved by both defining quantitative differential equation of qualitative ones and defining quantitative counterparts of landmark points. In this matching, the first derivation of quantitative equations gives qualitatively distinct time points and the points are taken into consideration as landmark points. QDEs determine which qualitative variable requires time advance and ODEs determine their quantitative values. In this respect, the simulation model has two equation sets. The first one is quantitative differential equations and the other one is QDEs, which guide ODEs. Most studies on qualitative discrete event simulation are

based on uncertainty of event occurrence time points. The uncertainty of the event time is represented in a closed time interval in R , which is also known as temporal interval [8]. Because of the uncertain order of events, there would be ties on the future event calendar. If there is a tie, QDEs would not assume a tie breaking strategy. Instead, it creates threads that make up all of the possible ordering of ties. Thus, the future event calendar in QDEs collects all the event notices whose execution order is uncertain and groups them in a set, called the nondeterministically ordered set (NOS). Each of these event notices will be executed in turn and results in a set of threads that will include all of the possible ordering of event sequences. The capability of generating all possible scenarios is achieved with the thread generation algorithm. This distinctive characteristic of QDEs of generating all possible ordering of event sequences is known as coverage [11]. A solution based on event occurrence probability is developed by [9]. The solution provides a means to tie breaking in qualitative discrete event simulation.

As another qualitative discrete simulation tool, QMTOOL, which is based on QSIM, has been extended with object-oriented features. The variables used with the important elements in a system or subsystems are input, state, output, and connection. The connection variable describes how the variables in an element are related. Variables can be assigned: an identifier (name), magnitude, sign, maximum, and minimum values (operating range) [12].

In the approach proposed here, time advance points are determined by event occurrences. Any landmark point of a system variable and derivation points are taken into consideration as event occurrences and an event occurrence is discovered in the following given cases:

- (i) Event occurrences. A time point where a set of events are received is determined as time point to be advanced
- (ii) Qualitative landmark points. A qualitative landmark point is determined for system variables. Each landmark point represents a meaningful distinction for the variable. For example, empty and full are two landmark points for a gas tank. Any value between two landmark points represents the same qualitative meaning such as a value between empty and full meaning “*there is some gas in the tank.*” As shown earlier, a system variable is given as a pair consisting of a value and a direction change. Direction change is used to determine what landmark point is to be reached and how much time is necessary
- (iii) Function derivatives. First and second derivatives of a function give distinct time points on simulation trajectory and it is assumed that an event is created at related time points. If the first derivative of a variable is equal to zero at t_i , it is interpreted as the fact that the variable reaches a critical point and at the adjacent time interval $[t_i, t_{i+1}]$, its direction of change could be inc, dec, or std. If the second derivation of the variable is negative, the variable changes its change direction to dec after std. If the first and the second derivatives are equal to zero and the third derivative is positive or

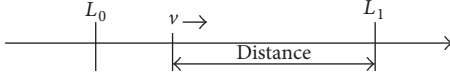


FIGURE 1: Time step calculation for a variable.

negative, the variable keeps its direction as inc or dec, respectively. If a function is the derivative between value couples defined as $[a, b]$, a and b are defined as landmark points.

System variable change directions are driving factors, especially in value movement between landmark points, leaving or approaching from/to a landmark point.

4. Managing Simulation by Qualitative Reasoning

The relationship between QS and quantitative simulation is examined in [13]. In this paper, it is contended giving some related issues with time management. While QS enhances quantitative simulation by reasoning techniques and allowing usage of qualitative characteristics of system variables such as derivative signs and dependencies between variables, quantitative simulation alleviates incompleteness and the ambiguity that qualitative models have. Moreover, qualitative information enriches quantitative simulation with deep knowledge that keeps simulation production rules in structure rather than “if-then” rule sets.

Qualitative reasoning in simulation is accommodated to manage time and events. QS simulation advances as a reasoning process from time point to time interval, time interval to time point, and so on. The basic idea of QS is to infer values and change of directions that system variables take at next time interval and at time point. Qualitative values taken by system variables are landmark values. In this sense, while QS determines qualitative magnitudes that consist of value and change of direction, quantitative simulation determines when it happens, in other words, duration to be granted.

4.1. QR in Time Management. Qualitative reasoning in time management is used to compute the time to be granted to pass from one time point to the following spanning time interval between these two time points for a continuous event system. Time step is calculated using variable values at two adjacent time points, not at time interval defined by qualitative transition rules. The main purpose to do this is to get as big as and as accurate as time step that is enough to reach next landmark point depending on variable value and direction of change. As seen in Figure 1, the variable X has a value v that is between landmark L_0 and L_1 and its change of direction is toward L_1 . We aim to find a time step that the variable needs to reach value represented by landmark L_1 . The change amount on the variable value is equal to distance $= L_1 - V$.

Having an accurate time advance is important for a faster simulation execution and minimized round-off error

and truncation error in computations [14]. It is clear that smaller time step size than necessary yields double precision error and bigger time step size than necessary yields two important problems. The first one is truncation error and the other one is ignoring qualitatively important function points, such as first-order derivative, second order derivative, and qualitatively distinguished points.

Quantitative time advance is computed under the guidance of qualitative simulation. To be able to calculate time advance size, time requirement for each state variable is found and the minimum value is set as time to advance. Qualitative analysis by discovering state transitions of state variables using both landmark point transitions and sign algebra gives qualitatively distinct points to advance. Quantitative analysis computes the time advance size to reach these points.

State variables start at time t_0 with values and change of directions constituting initial state. All values at time t_0 are known as quantitative and qualitative. The qualitative variable values for time interval $[t_0, t_1]$ following t_0 are generated using qualitative simulation (Table 1 rule set). Filtering applied to the result is computed using rule set given in Table 1. In quantitative simulation, interest is in time points and time interval is used to reach the next time point. In this respect, $[t_i, t_{i+1}]$ time interval is seen as time step with size $(t_{i+1} - t_i)$ to advance time point t_{i+1} from t_i . The five fundamental rules of time inference depending on value and change direction of a qualitative variable are given below.

Time Rule 1. A variable insists keeping its change direction the same for infinite time unless receiving an event interrupting time interval or reaching a landmark point.

Time Rule 2. A received event causes an external transition and it creates a new critical (landmark) point in value set of related qualitative variable and the time that the event is received is a candidate time point to be advanced.

Time Rule 3. A dependent variable time request is determined by minimum time request of independent variables that it depends on.

Time Rule 4. A variable with increased or decreased change direction requests a duration enough to reach first landmark point at its direction path.

Time Rule 5. A variable with std change direction (stable in time interval) requires infinite time.

Under these general rules, durations necessary to succeed in qualitative value transitions given in Table 1 are interpreted as follows. Quantitative time advance is determined based on interval to point transitions. The following interpretations show why reasoning is done on time intervals and not on time points. I.1 says a value $\langle l_j, \text{std} \rangle$ at $[t_i, t_{i+1}]$ is kept at time t_{i+1} . Because of the fact that it does not change its value, the variable aims to keep its value for infinite time. When a value increasing in a landmark interval (I.2) reaches landmark upper bound of the interval it is in and changes its change direction to std at time point following the time interval, the variable requests a limited duration (computable using

quantitative equations) and it changes its change direction to std because of either an event received or the success of a condition that disturbs it. Time request in I.3 gives the same result with I.2 but the change direction of the variable remains the same because there is no received event and successful condition. I.4 does not request any time step. A system variable that has a value given in I.4 consumes time step determined by any other variables. I.5 and I.9 are an external transition case and it happens if and only if an event is received or a condition that changes its value succeeds. Time step is defined as the time that the event is received. I.6 and I.7 are computed in a similar way with I.2 and I.3, respectively. A dual interpretation of I.4 and I.1 is valid for I.8 and I.10, respectively.

External transition defines state transition forced by an event received. In this case, differently from internal transition in which the simulation model transits its state by consuming the time assigned to the state it is in, external transition interrupts the current state at the event received time without consuming the whole state time. The duration consumed is equal to the difference between the event received time and state entry time.

4.2. QR in Event Handling and Detecting Event Occurrences. A continuous system receives and sends events. A received event may affect change directions of qualitative values and this determines the time point to advance. A qualitative variable reaching one of its landmark points makes an internal state transition and sends events attached to the state. In this sense, QSim representation is extended by event attachments to the landmark points. The rules regarding receiving an event are summarized below.

Event Rule 1. A qualitative variable may change its change direction if and only if the system receives an event interrupting it (related rules are I.2, I.5, and I.9).

Event Rule 2. A received event determines the time point to advance and causes an external transition.

The rule regarding sending an event is summarized below.

Event Rule 3. First and second derivation time points are candidate event of creation points.

Event Rule 4. An event may be created at a landmark point discovered by an external transition.

4.3. Transitions. QDE and related ODE are defined for a specific region. The transitions associated with a QDE define the limits of the region of applicability of the QDE and optionally specify a transition to a new QDE if that limit is reached.

Definition 4. A transition is a rule of the form condition \rightarrow transition function, where

- (i) the condition is a pattern of the form ($\langle \text{variable} \rangle (\langle qmag \rangle, \langle qdir \rangle)$), or a Boolean combination of such patterns,

- (ii) the transition function is applied to the current state if the condition succeeds. It returns a new qualitative state, perhaps defined with respect to a new QDE, from which simulation can resume.

Qualitative time management is done using currently active QDEs. The algorithms and rules developed are generic and valid for all QDEs and there is no domain dependency.

4.4. Abstracting Structure from ODE to QDE. To be able to use qualitative simulation for time management, two folded models are required: quantitative model (Ordinary Differential Equations (ODE)) and qualitative model (QDEs). It is possible to abstract QDE set using ODE set, given that a suitable ODE is decomposed into an equivalent set of simultaneous equations by introducing terms for each subexpression. Abstraction gives a weaker QDE set than ODE. "Weaker" means that any behavior that satisfies the ODE must satisfy the QDE, but not necessarily vice versa. Qualitative reasoning dictates the point qualitatively distinct to be reached in mathematical space of system variables. In this sense, qualitative simulation serves what state vector landmark points are to be reached and quantitative simulation computes when they are reached.

5. Example: Bathtub Problem

Bathtub problem is a simple but clear enough step to show both quantitative and qualitative simulation procedures and time management and event handling using qualitative reasoning. The system basically has two valves to fill and drain. The qualitative model of problem is given below.

$$\begin{array}{ll}
 M + (\text{amount}, \text{outflow}) & \text{amount} = \text{netflow} * \Delta t \\
 \text{Add} (\text{netflow}, \text{outflow}, \text{inflow}) & \text{dinflow}/dt = 0 \\
 \text{Deriv} (\text{amount}, \text{netflow}) & \text{Transitions} \\
 \text{Constant} (\text{inflow}) & \text{amount} (\text{full}, \text{inc}) \rightarrow t
 \end{array} \quad (2)$$

Graphical representation of the system and qualitative equations set are depicted in Figures 2(a) and 2(b), respectively. Figure 2(c) shows the relation between components. Valve-F and Valve-D have relation with tub named *Fills* and *Drains*, respectively.

$M+$ represents a monotonic increasing function and it is bidirectional function but the causality flows in one direction. In model representation, higher water amount is correlated with higher outflow.

$$\begin{array}{l}
 \text{Add} (X, Y, Z) \text{ is equal to } X + Y = Z, \\
 \text{Deriv} (X, Y) : \frac{dX}{dt} = Y.
 \end{array} \quad (3)$$

Transitions section in qualitative model represents model transition. That means the simulation uses another set of equations representing the system. In this case, after transition the pouring amount of water is calculated, amount of water in the tub is equal to full, and it does not change.

TABLE 3: Qualitative behavior trajectory for Case 1.

	t_0	P-Successors ($[t_0, t_1]$)	I-Successors (t_1)
Amount	$\langle 0, \uparrow \rangle$	P.4: $\langle (0, \text{full}), \uparrow \rangle$	I.2: $\langle \text{full}, \Theta \rangle$, I.3: $\langle \text{full}, \uparrow \rangle$, I.4: $\langle (0, \text{full}), \uparrow \rangle$, I.5: $\langle f^*, \Theta \rangle$
Outflow	$\langle 0, \uparrow \rangle$	P.4: $\langle (0, \infty), \uparrow \rangle$	I.2: $\langle \infty, \Theta \rangle$, I.3: $\langle \infty, \uparrow \rangle$, I.4: $\langle (0, \infty), \uparrow \rangle$, I.5: $\langle \text{of}, \Theta \rangle$
Inflow	$\langle \text{if}, \Theta \rangle$	P.1: $\langle \text{if}, \Theta \rangle$	I.1: $\langle \text{if}, \Theta \rangle$
Netflow	$\langle (0, \infty), \downarrow \rangle$	P.7: $\langle (0, \infty), \downarrow \rangle$	I.6: $\langle 0, \Theta \rangle$, I.7: $\langle 0, \downarrow \rangle$, I.8: $\langle (0, \infty), \downarrow \rangle$, I.9: $\langle (0, \infty), \Theta \rangle$

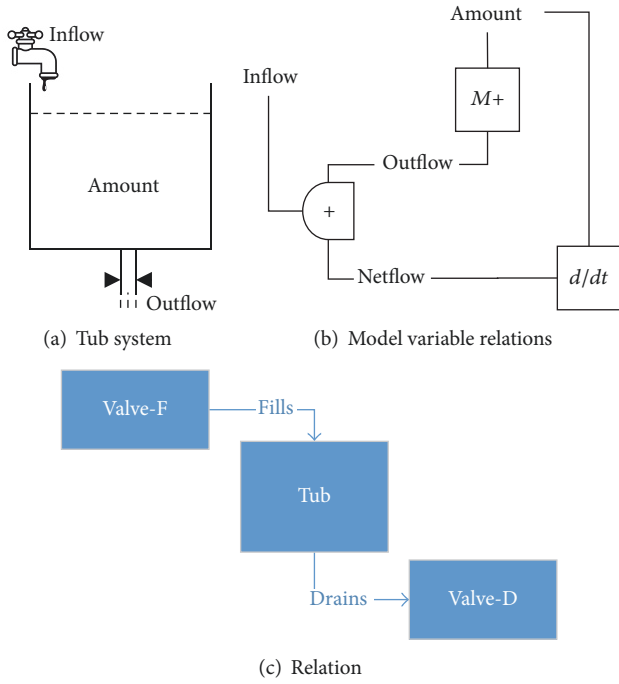


FIGURE 2: Bathtub problem graphical representation.

The cases given below are not alternative qualitative behaviors of the system being simulated. Rather, they are alternative scenario designs. At each case, qualitative simulation generates total envision (whole alternative behaviors) of the system being simulated. While each behavior consists of current qualitative values, landmarks to be reached, and change of direction at time points and time intervals, quantitative simulation calculates the time span to reach those landmark points if it is possible. This allows us to advance simulation over qualitative landmark points. A landmark point is reached at each simulation step or a landmark point is created at the time point that an event is received and sent by any other simulation components. This makes simulation execution faster than stepping by Δt time step and calculating a state vector and more accurate because qualitatively distinct landmark points are calculated directly instead of approaching the values by slicing time.

Case 1 (constant flow). In this type, inflow variable has a constant numeric value and maintains the same qualitative value. The value is determined as “*if*” that is placed between

zero and ∞ . It does not change during simulation. Qualitative value set for qualitative variables is given below:

$$\begin{aligned}
 \text{amount} &= \{0, \text{full}, \infty\}, \\
 \text{inflow} &= \{0, \text{if}, \infty\}, \\
 \text{outflow} &= \{0, \infty\}, \\
 \text{netflow} &= \{-\infty, 0, \infty\}.
 \end{aligned} \tag{4}$$

Initial state (t_0) is seen in Table 3. The only variable that is known is inflow and it is equal to *if*. The other variables are determined by propagating its value using qualitative equations. After completing qualitative variables at t_0 , the prediction of its successor state describing the open interval $[t_0, t_1]$ is given in Table 3.

From $[t_0, t_1]$ interval to t_1 point, for amount, the rules I.2, I.3, I.4, and I.5 are applied (in Table 3 the rules and time point values for each rule are given for the variable amount, resp. For I.5 a value f^* is discovered). It is possible to calculate a duration for I.2 and I.3 to find an answer for the question “How much time is required to reach the full level starting from the value at t_0 ?” The variable amount, which has zero value at t_0 , reaches full at t_1 according to both rules. The quantitative value change is equal to $(\text{full} - 0)$ and the duration it takes is calculated as $t_{\text{amount}} = (\text{full} - 0) / \text{netflow}$. The rule P.7 is the only rule to be applied for the variable *netflow* and it gives the same qualitative value for the time interval following t_0 . For the other variables, the rules applied and their qualitative values in the time interval $[t_0, t_1]$ are given in Table 3.

The answer to the question at t_1 is sought whether or not the amount will keep its direction as inc or it will be stable at *full* value or a value in the interval $[0, \text{full}]$. The answer is given by the variables to which the variable amount is connected. There are inflow and outflow variables that determine the variable *netflow*. If *netflow* is greater than zero, then the change direction of the variable amount is known as increase (inc) and if it is equal to zero, its change direction is known as std. It is necessary to check if there is an event scheduled for t_1 to change to inflow or outflow or both of the valves, so that *netflow* is equal to zero. If there is no event scheduled like these (it is given), the solution accepting amount equal to $\langle \text{full}, \text{while } \Theta \rangle$ is accepted as invalid. Similarly, the value $\langle (0, \text{full}), \text{std} \rangle$ or $\langle f^*, \text{std} \rangle$ represented by I.5 is also not applicable because of the same reason (no event scheduled to change inflow and outflow).

At time t_1 , the value of amount equal to full and direction of change is *inc*, and this is a transition condition (amount

TABLE 4: The behaviors at t_1 .

	1	2	3
Amount	$\langle \text{full}, \uparrow \rangle$	$\langle \text{full}, \Theta \rangle$	$\langle (0, \text{full}), \Theta \rangle$
Outflow	$\langle (0, \infty), \uparrow \rangle$	$\langle \text{of}, \Theta \rangle$	$\langle \text{of}, \Theta \rangle$
Inflow	$\langle \text{if}, \Theta \rangle$	$\langle \text{if}, \Theta \rangle$	$\langle \text{if}, \Theta \rangle$
Netflow	$\langle (0, \infty), \downarrow \rangle$	$\langle 0, \Theta \rangle$	$\langle 0, \Theta \rangle$

(full, inc) $\rightarrow t$). If the simulation is terminated, pouring level of water is calculated depending on passing time after reaching the level *full* and the value of *netflow*. Simulation execution requires infinite duration $([t_1, \infty])$ until scheduling an event. If any event is scheduled to change inflow and/or outflow level between t_0 and t_1 , the time advance is determined as the time that the event is scheduled. Rule I.4 keeps the variable value interval and change direction for t_1 the same with $[t_0, t_1]$ interval. Since it does not give any landmark points, there is no other variable that reached a landmark point in its qualitative value set, and because we aim to calculate a time step I.4 is accepted as not suitable for our purpose. The rules I.2 and I.3 give results that suitable to calculate a time step and both give the same time step and it is a time step necessary to reach the level *full* from the level zero. It is possible to filter out spurious qualitative behaviors using the qualitative model constraints [1]. The process is given in Appendix. After filtering qualitative behaviors three behaviors remain. These are shown in Table 4. As mentioned earlier, there is no event scheduled to change inflow or outflow; behavior (2) and behavior (3) are eliminated. If there is an event scheduled to change flows with a time label smaller than the time step necessary to reach level full, the event time is granted as simulation time.

As seen here, qualitative model gives an answer to where time advance is to be searched and quantitative model gives an answer to how big the time span is. Instead of writing an “if-then” rule to control whether or not the variable amount reaches its maximum full value to switch to pouring model, the control rule, in other words, production rules, is presented as structural information.

Case 2 (constant flow and flow change event scheduled). Inflow has constant values and it is set to zero at time t_e by *changeFlow* event. Time interval $[t_0, t_1]$ is calculated as in Case 1. If t_e is in the interval, time request is determined as t_e ; otherwise it is set as t_1 . When tub model receives *changeFlow* event between t_0 and t_1 , it calculates the amount of water filled until the time event is received. t_e seems as a new landmark point situated between t_0 and t_1 . The variables have new values at time t_e . The value set at time t_e is the same in previous case at time t_1 . The values of *inflow* and *outflow* are changed by the event *changeFlow*. A new value for inflow (if*) and outflow (of*) is set. The next step is a point to interval transition (from t_e to $[t_e, t_1]$). The next value of *amount* and the duration required is calculated using new *netflow* value. If *netflow* is zero, infinite duration is required to reach maximum level. If the value is positive it is calculated as before but if it is a negative value (which means outflow is greater than inflow), because of *netflow* direction of change,

the amount decreases to zero value. The duration to reach zero water level is calculated depending on new *netflow* value.

Simulation execution is completed in two steps. The first step is determined by the event scheduled and the second step is from the point event received to the point that the variable amount reaches its *full* value or zero level.

Case 3 (gradually changing flows). In this case, *inflow* and *outflow* change their values by a function of time ($\text{inflow} = f(\Delta t)$ and $\text{outflow} = g(\Delta t)$). It is assumed that an automated valve controller changes exogenous clearance levels of inflow valve and outflow valve. It tries to make a valve clearance level maximum if it is closed or vice versa if the valve is open. Clearance levels for inflow and outflow are determined by a time dependent $\cos(t)^2$ function and $(\cos(t) - \sin(t))^2/2$ function, respectively. At the initial state t_0 , the inflow valve is opened *full* and *outflow* valve is closed and qualitative value set of variables is given below.

$$\begin{aligned} \text{amount} &= \{0, \text{full}, \infty\}, \\ \text{inflow} &= \{0, \text{max if}\}, \\ \text{outflow} &= \{0, \text{max of}\}, \\ \text{netflow} &= \{-\text{max of}, 0, \text{max if}\}. \end{aligned} \quad (5)$$

The fact that while in Case 1, inflow is set as a constant, in Case 2 inflow and outflow valve clearances are changed by time dependent functions, and this is the main difference.

Outflow obtains its clearance level to full clearance (open) by time from closed state. Since each variable has a time advance request to reach time t_1 , after $[t_0, t_1]$ interval computation, transition from the interval to time point t_1 is computed using Table 1 as in the earlier cases. At this stage, time advance requirements are computed starting by independent variables and they are *inflow* and *outflow*.

The first and the second derivations of *inflow* function give

$$-2 * \sin(t) * \cos(t), \quad (6)$$

$$2 * (\sin^2(x) - \cos^2(x)) \quad (7)$$

functions. Roots of the first derivation in $[0, 2 * \pi]$ are $[0, \pi/2, \pi, 3 * \pi/2, 2 * \pi]$ (using (6)). While these points give minimum and maximum clearance for valves, the roots of $\cos(t)^2$ function give minimum clearance for valves and these are $\pi/2$ and $3 * \pi/2$ (using (7)). Transition from initial time t_0 to the time point t_1 passes through time interval $[t_0, t_1]$. As seen in Table 3, by applying the rules from rule set at Table 1, the variable values at the interval $[t_0, t_1]$ are calculated. Inflow reaches its minimum value at $\pi/2$ (one of the roots of the first derivative) and at this time point, since the second derivative has a positive value, *inflow* changes its direction of change from dec to inc (see (8)).

$$\begin{aligned} \text{inflow}' &= -2 * \sin(t) * \cos(t) = 0 \rightarrow t = \frac{\pi}{2}, \\ \text{Inflow}'' &= 2 * \left(\sin^2\left(\frac{\pi}{2}\right) - \cos^2\left(\frac{\pi}{2}\right) \right) = [+]. \end{aligned} \quad (8)$$

TABLE 5: Qualitative behavior trajectories.

	t_0	P-Successors ($[t_0, t_1]$)	I-Successors (t_1)
Amount	$\langle 0, \uparrow \rangle$	P.4: $\langle (0, \text{full}), \uparrow \rangle$	I.2: $\langle \text{full}, \Theta \rangle$, I.3: $\langle \text{full}, \uparrow \rangle$, I.4: $\langle (0, \text{full}), \uparrow \rangle$
Outflow	$\langle 0, \uparrow \rangle$	P.4: $\langle (0, \text{max Of}), \uparrow \rangle$	I.2: $\langle \text{max Of}, \Theta \rangle$, I.3: $\langle \text{max Of}, \uparrow \rangle$
Inflow	$\langle \text{max If}, \downarrow \rangle$	P.5: $\langle (0, \text{max If}), \downarrow \rangle$	I.6: $\langle \text{max If}, \Theta \rangle$, I.7: $\langle \text{max If}, \downarrow \rangle$
Netflow	$\langle (0, \text{max If}), \downarrow \rangle$	P.7: $\langle (0, \text{max If}), \downarrow \rangle$	I.6: $\langle \text{max Of}, \Theta \rangle$, I.7: $\langle \text{max Of}, \downarrow \rangle$

Similarly, the calculation for outflow is done. $\text{outflow}' = \sin^2(x) - \cos^2(x) = 0 \rightarrow \{\pi/4, 3 * \pi/4, 5 * \pi/4, 7 * \pi/4\}$, since the closest time point is $\pi/4$; the sign of the second derivation at this point is checked. At time points $\pi/4$ and $5 * \pi/4$, the second derivation of outflow has positive values ($\text{outflow}'' = 4 * \cos(x) * \sin(x) = 4 * \cos(\pi/4) * \sin(\pi/4) = [+]$ and it is $[-]$ for $3 * \pi/4$ and $7 * \pi/4$). Since the second derivative has a positive value, *outflow* changes its direction of change from dec to inc at time point $\pi/4$ after having a std change of direction. From time zero to $\pi/4$, outflow keeps its direction of change as decrease and between $\pi/4$ and $5 * \pi/4$ (the root of outflow function) it changes its direction of change to inc after std at $\pi/2$ because of its positive second derivative at π . The smallest time step is determined as $t_1 = \pi/2$. Since the amount at time step t (amount_t) is equal to $\text{amount}_{t-1} + \text{newflow}_{[t-1, t]}$, its quantitative value is calculated using integral value of outflow and inflow for intervals $[0, \pi/2]$ and $[\pi/2, 0]$, respectively (9). The simulation advances directly to time $\pi/2$. Simulation execution continues stepping through qualitatively distinct time points that are landmark points and roots of functions and their derivatives.

$$\begin{aligned}
 \text{outflow} &= \int_0^{\pi/2} \frac{(\cos(t) - \sin(t))^2}{2} dt = \frac{\cos^2(t) + t}{2} \\
 &= \frac{\pi - 2}{4}, \\
 \text{inflow} &= \int_0^{\pi/2} \cos^2(t) dt = \frac{\cos(t) \sin(t) + t}{2} = \frac{\pi}{2}, \quad (9) \\
 \text{amount} &= \text{amount}_0 + \text{inflow} - \text{outflow} \\
 &= 0 + \left(\frac{\pi}{2} - \frac{\pi - 2}{4} \right) = \frac{\pi - 2}{4}.
 \end{aligned}$$

In this case, if the value of amount is greater than the landmark *full*, it is thought to be subtracted from the value of amount to calculate the amount of pouring water. But the better way is to determine how much time is necessary to reach the landmark *full*, as seen below. Because, at further iteration of the simulation, the smallest time step may come from the variable *amount*, this is not the case in the approach to pass any variable landmark value.

$$\text{full} = \text{amount}_0 + \text{inflow} - \text{outflow}. \quad (10)$$

If there is no solution, the time step is assumed to be infinite (at the beginning of the simulation (at time 0) the amount is equal to zero).

$$\text{full} = 0 + \frac{\cos(t) \sin(t) + t}{2} - \frac{\cos^2(t) + t}{2}. \quad (11)$$

Let us assume *full* is equal to five (5) and initial value of amount is equal to zero. In this case, there is no real solution for these value sets. It is assumed that the variable amount requires infinite time. For later simulation execution, *amount* reaches a value of 4.80. The next step size is calculated by the amount required to reach the landmark *full* (it is equal to 5) because of the fact that it gives a smaller time step than the time step required by the valve clearances.

We know that any solution for the variable amount is an exact value and it is when the amount reaches its *full* value. While there is a possibility of passing the landmark value of a system variable, when the slicing time advance mechanism is used, in this case, since the time is advanced directly to the landmarks points as stepping stones, it ensures that each time advance calculates the exact value of the system variable at the exact time point where it has the value. As seen in Table 5, because of the small valve steps, the variable amount may have a value between 0 and full (I-Successors I.4). The time required for the amount to reach $\langle \text{full}, \Theta \rangle$ or $\langle \text{full}, \uparrow \rangle$ is the time granted to the time step by the variable amount. Until this happens, it is seen as a behavior filtering rule and $\langle \text{full}, \Theta \rangle$ and $\langle \text{full}, \uparrow \rangle$ are filtered.

Simulation behavior state diagrams covering all cases given here are seen in Figures 3 and 4. The behavior diagrams are developed based on AdSiF. Figure 3 shows the behavior of a valve. As seen in the figure, the event “close” closes the valve and the event “open” opens it. If a valve is closed it stays in closed state for infinite duration and it is interpreted as the same way for the event “open” and state “active.” If a valve is opened, it starts at maximum open clearance and activates the behavior *Fsa_SetClearance* at “Out” phase of the state active. If a valve is for draining, its initial behavior is set as *Fsa_Close* and the clearance is set to zero. The behavior calculates the next clearance level using (6)–(8) and the duration necessary to reach the calculated clearance is computed by the function *ValveDuration* that is the duration calculator function of the state *Calculate Clearance*. It sends an event name *setClearance* to the tub where the valve is related to the relation known as *Fills* or *Drains*.

The tub is in active state for infinite time because of its initial behavior *Fsa_Initial*. Depending on qualitative value of the variable *amount*, one of the behaviors *Fsa_FullLevelRequest*, *Fsa_TimeToZero*, and *Fsa_StdLevel* is activated because of their activation conditions. *Fsa_FullLevelRequest* is activated if the variable *amount* has an increasing direction (activationCond: $\langle \text{val}, \text{inc} \rangle$) and activation of other behaviors is interpreted as similar. At any time during simulation, there is an active “Request Duration” state. Anytime the event named *setClearance* is received, the tub activates the behavior *Fsa_LevelCalculator* and it

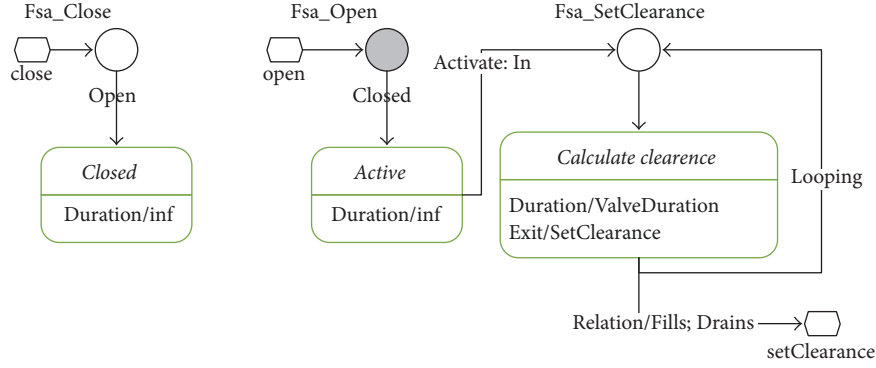


FIGURE 3: Valve behavior diagram.

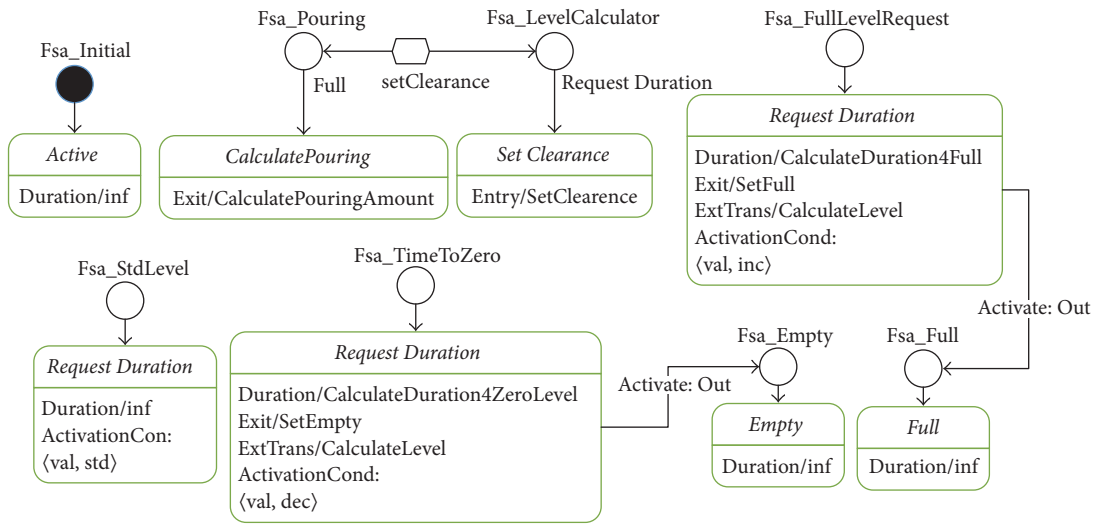


FIGURE 4: Tub behavior diagram.

sets clearance level. Since the behaviors that consist of the state “Request Duration” that is entry state of the behavior *Fsa_LevelCalculator* make external transition, they execute *ExTrans/CalculateLevel* functions for the duration elapsed to calculate the water level in the tub. The state *Request Duration* at *Fsa_FullLevelRequest* calculates how much time is required to reach the level *full* using *CalculateDuration4Full* (10). If it reaches this level, it activates the behavior *Fsa_Full* at the state out phase. *Fsa_Full* stays in *Full* state for infinite time. Similarly, the state same state at *Fsa_TimeToZero* calculates the duration necessary to reach landmark level zero. If the tub is in *full* state and it receives *setClearance* event, it activates *Fsa_Pouring*. State *full* calculates the pouring water amount up to the event received time.

It is seen that the qualitative model supports a quantitative discrete model in two ways: (1) the qualitative model guides quantitative model to find which variable gives time advance and (2) the landmark points work as structural rules, and derivation points work as an inner event creation mechanism for managing execution flow. If the whole system is modeled as a single entity, the first support gets more prominent. In the other cases, it is shown here that each component is modeled

as a separate model, and the first support gets weaker but the second support is still available.

5.1. Performance Measurement. The example is executed for five (5) minutes. For time slicing simulation execution all state variables are updated at each second. That means, for Case 1, simulation takes 300 ($5 * 60$) steps, if there is no close event received during execution. In qualitative calculation case, it takes just one step. In Case 2, time slicing simulation approach takes 300 steps plus number of flow change events. In qualitative calculation case, number of simulation steps is equal to two times the number of flow change events.

The most complex situation is seen in Case 3. Qualitative landmark points are discretized by the values of inflow and outflow function roots and of their derivative roots. The roots are $[0, \pi/2, \pi, 3 * \pi/2]$ for inflow and $[\pi/4, 3 * \pi/4, 5 * \pi/4, 7 * \pi/4]$ for outflow. Each root value is taken into consideration as a discrete time point in a continuous universe. During simulation execution, in five minutes, we have eight discrete time points in total (4 from inflow and 4 from outflow) because of the discretized movements of the

valves. At each step, the system reaches one landmark point and this makes eight steps. In qualitative calculation phase, a qualitative inference overhead is added to processing effort; it takes approximately 0.001 seconds. Processing effort for time slicing is directly related to simulation duration. In the second case, simulation time effort depends on both simulation time and number of events processed. In the third case, the effort is calculated in a similar way. But none of the cases in qualitative calculation and simulation execution effort does not depend on simulation execution duration.

6. Discussions

Combining qualitative and quantitative information for simulation based analysis gives several advantages. In addition to the advantages to cope with the problem in qualitative analysis such as data size to be analyzed, variety of data type, and hardness in generalizing qualitative analysis to particular cases, [15], in this study, how qualitative reasoning provides a fast and accurate simulation execution providing time step mechanism via qualitative meaning of the mathematical properties and qualitative information known prior to discrete event simulation is shown. The simulation is advanced, instead of iterating simulation time for a predefined time step and checking whether or not there is any activity in the interval, directly to the time points that are qualitatively different.

As seen from the literature, qualitative discrete event simulation studies are mostly focused on alleviating vagueness on qualitative model and variables by enhancing them using quantitative information such as quantitative intervals. Unlike other approaches, the approach presented here is based on qualitative reasoning and it is used to guide quantitative discrete event simulation determining qualitatively (common sense) meaningful time points.

In the approach given here, qualitative simulation is achieved as a parallel processes, it is executed and its filtering rules are applied, and whole process is followed. By this way, total envision (all possible behaviors), in other words, qualitative behavior trajectories, is got. All these behaviors serve quantitative simulation to find which qualitatively distinct point is to be reached for each qualitative variable and using this information quantitative simulation calculates how long it takes. This ensures simulation execution is done by stepping on qualitatively meaningful time points, instead of advancing a continuous simulation step by step and checking the rule set (e.g., if tub is burst or not and checking clearance of valves). The study aims to provide a higher order rule set depending on mathematical properties of the system and qualitative model, not depending on domain specific information.

The proposed approach maps continuous event simulation into discrete event simulation by discovering qualitatively distinct time points both in sign algebra (sign of derivatives) and in qualitative value set consisting of landmark points of state variables. Discovered time points are interpreted as event points created by continuous state variables. For a hybrid system, the approach poses events arising from discrete part of the system onto trajectories of continuous

state variables. In the approach, looking at discrete event occurrences on continuous state variable trajectories, this is taken into consideration as rare event simulation [16, 17]. The threshold values defined in rare event simulation are seen as landmark point intervals and time interval placed on qualitative simulation trajectories.

The approach provides not only faster simulation executions but also allows more accurate computations. Since simulation advances directly to qualitatively meaningful discrete time points computed on a continuous state variable trajectory, the approach alleviates errors that come from truncation and round-off. In other words, each simulation step is as wide as simulation model needed rather than centrally granted time step. Because time step intervals are computed by both taking algebraic characteristics of function sign and qualitative value and function sets with change directions into consideration, this gives an opportunity to advance time points mathematically and qualitatively, in this sense, linguistically reasonable function points during simulation execution without missing any important point and unnecessarily computing qualitatively the same adjacent time points. This makes simulation execution associated with model mathematical properties by making it independent from model parameters. For example, let us take a moving entity from point A to point B with a constant velocity V into consideration. The execution duration of this example case using time slicing execution depends on velocity of the entity and size of simulation step size. It is also known that the bigger the step size (than necessary), the bigger the Fourier error and the smaller the step size, the bigger the double precision error [14]. But the approach proposed is independent from simulation parameters, since it uses model mathematical properties; the execution takes a known number of steps and calculates the exact time points to advance. The approach assumes to use integration methods to solve model equations instead of numerical methods. If it is necessary to use numerical method each simulation entity consumes a wide time step that is calculated using landmark points and/or derivation time point presented in the approach by dividing into the smaller steps. The time saving opportunity allows modelers to change differential equation solution technique from numerical methods to integration methods that give more accurate result but need more computational effort.

Since simulation time advances directly to the qualitatively distinguished time points, the concept of time is taken into consideration as 4th dimension in the space. This is explained by space time (relativity). In contrast with this, slicing time advance mechanism is basically Euclidian time definition. There is a time axis that is continuously advancing, the system variables are calculated for each time step, and for each time step there is an activity scanning phase to discover whether or not there are any events to be processed.

The approach also provides some important advantages for tracing simulation records. Jumping directly to qualitatively distinct time points on a continuous contour instead of transverse function point (as seen in Table 5) and stepwise qualitative computation of the same functional values (not in quantitative manner) at each time step gives less record points for each variable to be recorded. The property makes

TABLE 6: Filtering behaviors using $M+$ (amount, outflow) and Add (netflow, outflow, inflow).

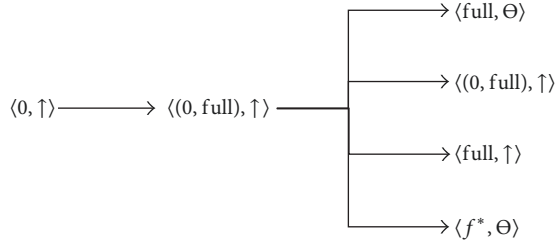
Outflow	Amount				Netflow			
	$\langle \text{full}, \uparrow \rangle$	$\langle \text{full}, \Theta \rangle$	$\langle (0, \text{full}), \uparrow \rangle$	$\langle f^*, \Theta \rangle$	$\langle 0, \downarrow \rangle$	$\langle 0, \Theta \rangle$	$\langle (0, \infty), \downarrow \rangle$	$\langle \text{of}, \Theta \rangle$
$\langle \infty, \uparrow \rangle$	R2	R1	R2	R1	R4	R3	R4	R3
$\langle \infty, \Theta \rangle$	R1	R2	R1	R2	R3	R4	R3	R4
$\langle (0, \infty), \uparrow \rangle$	B1	R1	B4	R1		R3		R3
$\langle \text{of}, \Theta \rangle$	R1	B2	R1	B3	R3		R3	

R1: $[d(\text{amount})/dt] = [d(\text{outflow})/dt]$; R2: if *outflow* has *inf* value then *amount* must have *inf* value; R3: $[d(\text{netflow})/dt] + [d(\text{outflow})/dt] = [d(\text{inflow})/dt]$; R4: $\infty + \text{finite} = \infty$; R5: $[d(\text{amount})/dt] = [\text{netflow}]$.

TABLE 7: Filtering behaviors using Deriv (amount, netflow).

Netflow	Amount			
	$\langle \text{full}, \uparrow \rangle$	$\langle \text{full}, \Theta \rangle$	$\langle (0, \text{full}), \uparrow \rangle$	$\langle f^*, \Theta \rangle$
$\langle 0, \downarrow \rangle$	R5			
$\langle 0, \Theta \rangle$	R5	B2	R5	B3
$\langle (0, \infty), \downarrow \rangle$	B1	R5	B4	R5
$\langle \text{of}, \Theta \rangle$		R5		R5

R1: $[d(\text{amount})/dt] = [d(\text{outflow})/dt]$; R2: if *outflow* has *inf* value then *amount* must have *inf* value; R3: $[d(\text{netflow})/dt] + [d(\text{outflow})/dt] = [d(\text{inflow})/dt]$; R4: $\infty + \text{finite} = \infty$; R5: $[d(\text{amount})/dt] = [\text{netflow}]$.

FIGURE 5: Behavior trajectory for the variable *amount*.

it unable to record the necessary time points and obtain qualitative records of the same values. In a constant time step simulation, it is possible to miss qualitative meaningful time points. In quantitative function sign algebra based analysis, for the variable, defining the function time advance request is computed depending on the first function and the second derivation solution points. Bringing simulation steps together with common-sense reasoning is another important property provided by the approach. Because at each simulation step a qualitatively meaningful system state is reached, the execution trajectory gives a concise, close to human reasoning, and meaningful result.

Appendix

Filtering Spurious Behaviors

From t_0 to t_1 , the behavior trajectory for amount is given in Figure 5. The same representation can be envisioned for the rest of the variables. Filtering spurious behaviors that violate model constraints is achieved by using mathematical properties of the model constraints for the time point t_1 and this is shown in Tables 6 and 7. The mathematical properties

used are given as R1, R2, R3, R4, and R5. The main aim is to filter out the invalid alternative qualitative values following the value at $[t_0, t_1]$. After qualitative filter, we just have three valid behaviors. These are given as B1, B2, B3, and B4. Because B4 is the same behavior at $[t_0, t_1]$, it is eliminated.

Conflicts of Interest

The author declares that they have no conflicts of interest.

References

- [1] B. Kuipers, "Qualitative simulation," *Artificial Intelligence*, vol. 29, no. 3, pp. 289–338, 1986.
- [2] B. Kuipers, *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, MIT Press, New York, NY, USA, 1994.
- [3] J. De Kleer and J. S. Brown, "A qualitative physics based on confluences," *Artificial Intelligence*, vol. 24, no. 1-3, pp. 7–83, 1984.
- [4] K. D. Forbus, "Qualitative process theory," *Artificial Intelligence*, vol. 24, no. 1-3, pp. 85–168, 1984.
- [5] A. Muzy, R. Jammalamadaka, B. P. Zeigler, and J. J. Nutaro, "The Activity-tracking paradigm in discrete-event modeling and simulation: The case of spatially continuous distributed systems," *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 5, no. 87, pp. 449–464, 2010.
- [6] F. M. Hocaoglu, *Agent Based Simulation: Lecturer Notes*, Istanbul University, Istanbul, Turkey, 2014.
- [7] R. Ingalls, D. Morrice, and A. Whinston, "Interval time clock implementation for qualitative event graphs," in *Proceedings of the Winter Simulation Conference*, pp. 574–580, Lake Buena Vista, FL, USA, 1999.
- [8] Y. Leow-Sehwail and R. Ingalls, "Qualitative Discrete Event Simulation," in *Proceedings of the Winter Simulation Conference*, pp. 648–653, Orlando, FL, USA, 2005.
- [9] Y. Leow-Sehwail and R. G. Ingalls, "Estimating the probability of an event execution in Qualitative Discrete Event Simulation," in *Proceedings of the 2010 Winter Simulation Conference - (WSC 2010)*, pp. 1133–1144, Baltimore, MD, USA, December 2010.
- [10] B. Zeigler and S. Chi, "Symbolic discrete event system specification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 22, pp. 1428–1443, 1992.
- [11] R. G. Ingalls, D. J. Morrice, and A. B. Whinston, "The implementation of temporal intervals in qualitative simulation graphs," *ACM Transactions on Modeling and Computer Simulation*, vol. 10, no. 3, pp. 215–240, 2000.
- [12] G. Garani and G. K. Adam, "Qualitative Modelling of Manufacturing Machinery," in *Proceedings of the IECON 2006 - 32nd*

Annual Conference on IEEE Industrial Electronics, pp. 3591–3596, Paris, France, November 2006.

- [13] F. M. Hocaoglu, “A Comparison between Qualitative Simulation and Traditional Simulation?: Bridging the Conceptual Gap,” in *AI, Simulation Planning in High Autonomy Systems-AIS2002*, Lisbon, Portugal, 2002.
- [14] D. J. Murray-Smith, *Continuous System Simulation*, University of Glasgow, Glasgow, UK, 1995.
- [15] T. Eldabi, Z. Irani, R. J. Paul, and P. E. Love, “Quantitative and qualitative decision-making methods in simulation modelling,” *Management Decision*, vol. 40, no. 1, pp. 64–73, 2002.
- [16] J. Bucklew, *Introduction to Rare Event Simulation*, Springer, S. University of Wisconsin, Madison, Wis, USA, 2004.
- [17] J. Nutaro, P. T. Kuruganti, V. Protopopescu, and M. Shankar, “The split system approach to managing time in simulations of hybrid systems having continuous and discrete event components,” *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 3, no. 88, pp. 281–298, 2011.

