

Using Qualitative Simulation Models for Monitoring and Diagnosis

Ankita Das¹ 

Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Roxane Koitz-Hristov 

Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Franz Wotawa² 

Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Abstract

Many systems in our daily lives control physical processes, which are parametrized and adapted, such as heating systems in buildings. Faults and non-optimized settings lead to a high energy demand and, therefore, need to be detected as early as possible. Unfortunately, due to specific adaptations, only the basic principles remain the same, but not the concrete implementations, making the use of techniques like machine learning difficult. Therefore, we suggest using abstract models that cover the basic behavior in a way that allows us to reuse the models in different installations. In particular, we discuss the application of qualitative simulation for fault detection and introduce a formal definition of conformance between the results of qualitative simulation and the monitored behavior. We discuss arising difficulties and provide a basis for further research and applications.

2012 ACM Subject Classification Computing methodologies → Causal reasoning and diagnostics; Computing methodologies → Spatial and physical reasoning; Computing methodologies → Modeling methodologies

Keywords and phrases Qualitative Simulation, Fault Detection, Model-based Diagnosis, Monitoring, Application

Digital Object Identifier 10.4230/OASICS.DX.2025.4

Funding The work presented in this paper has been supported by the FFG project Artificial Intelligence for Smart Diagnosis in Building Automation (ALFA) under grant FO999914932.

Ankita Das: Supported by the Christian-Doppler Forschungsgesellschaft (CDG) project AI-based Diagnosis for Energy Transition and the Circular Economy (AID4ETCE).

1 Introduction

In 2022, the International Energy Agency estimated the buildings' energy-related CO₂ emissions at around 27% of the total worldwide CO₂ emissions. To bring the emissions down, several actions have to be taken, including monitoring and diagnosis of heating and cooling systems that often operate in non-optimal operational spaces using far more energy than necessary. This is well visible when considering previous work (i) in the context of radiant ceiling cooling systems, where model predictive control can reduce energy consumption by up to 27% (see, e.g., [10]), or (ii) heat pumps where fault diagnosis can reduce energy loss when detected, diagnosed, and repaired early by about 40% [3]. As a consequence, there is a strong need for automated diagnosis of heating and cooling systems in buildings to reduce the overall CO₂ emissions.

¹ Authors are listed in alphabetical order.

² Corresponding author.



© Ankita Das, Roxane Koitz-Hristov, and Franz Wotawa;
licensed under Creative Commons License CC-BY 4.0

36th International Conference on Principles of Diagnosis and Resilient Systems (DX 2025).

Editors: Marcos Quinones-Grueiro, Gautam Biswas, and Ingo Pill; Article No. 4; pp. 4:1–4:14

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Unfortunately, monitoring and fault localization of such systems is complicated because each building is unique, comprising tailored heating and cooling facilities. When using machine learning, we need either to retrain every model for every building, which is expensive and time-consuming or to find a way to adapt trained models. This similarly holds for physical simulation models that would need to be parametrized for every building. However, the basic principles and their corresponding abstract representation of the behavior of each of the facilities are the same, which follow physical principles. Hence, when being able to represent the behavior of systems that follow physical principles in an abstract way, we can use this for at least detecting faults when monitoring the current behavior of a system.

In this paper, we tackle this challenge and suggest utilizing qualitative reasoning to provide an abstract model of a system that can be used for fault detection and, finally, localization. In particular, we discuss how to use and couple qualitative simulation [20] to ordinary monitoring systems. Rinner and Kuipers [24] already suggested the use of qualitative simulation in monitoring. In contrast, we formally define conformance between the outcome of qualitative simulation and the observations obtained from a system. This formal conformance relationship serves as the basis for detecting faults and may also be of use for fault localization, providing that the qualitative simulation models also capture faulty behavior. The idea is to compare the different qualitative states over time with the abstracted observations. This comparison requires not only abstraction but also specific enhancements. For example, we do not continuously observe the behavior over time but at certain time points. Hence, we might not observe reaching landmarks, which would raise false alarms. Therefore, we need to add reasonable qualitative states to the observations for comparison.

We organize this paper as follows: In Section 2, we discuss related work. Afterward, in Section 3, we introduce a simple tank example, which we use in the rest of our paper for illustration purposes. Section 4 introduces qualitative simulation to be self-contained and discusses conformance in detail. Finally, we conclude the paper.

2 Related work

Qualitative reasoning provides a powerful method for modeling dynamic systems under uncertainty by abstracting continuous variables into symbolic associations. Three formalisms form the foundation [26]: Qualitative Process Theory (QPT) [12], which models physical processes using causality and process activation; Qualitative Physics [15], which uses component-level modeling and constraint propagation to envision possible system behaviors; and Qualitative Simulation (QSIM) [20], which simulates all consistent qualitative trajectories from an initial state using a constraint-based model. QSIM represents system dynamics using Qualitative Differential Equations (QDEs), which abstract sets of Ordinary Differential Equations (ODEs). Starting from an initial qualitative state, QSIM generates a behavior tree by enumerating all possible successor states using a transition table of permissible qualitative changes, and then pruning inconsistent states based on qualitative constraints [21].

QSIM has been extended and applied in various diagnostic frameworks. Subramanian and Mooney [25] extended QSIM to model systems with multiple concurrent faults by associating fault-specific constraints and using constraint-based reasoning to isolate them. Similarly, the SEXTANT system [23] combines QSIM with the HS-DAG algorithm [14] to compute minimal diagnoses based on behavioral conflicts. Another example is Mimic [11], a semi-quantitative system that integrates QSIM-style simulation with model-based reasoning for process monitoring. Mimic identifies faults by detecting discrepancies between simulated

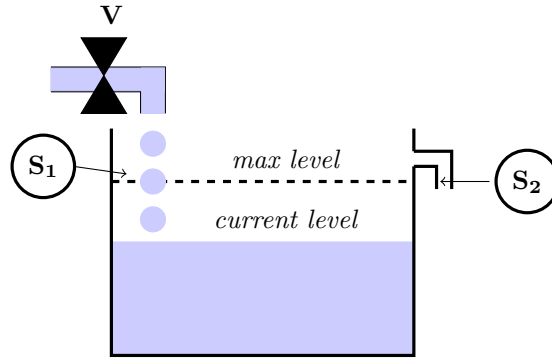


Figure 1 A simple tank example, comprising an input pipe with a valve, a sink, a sensor S_1 measuring the maximum water level and a sensor S_2 measuring whether water is passing the sink. The ordinary behavior is that the valve is open enabling an inflow until the maximum water level is reached and the valve is closed. The outward flow is only for cases where the inflow cannot be controlled anymore to prevent from flooding.

and observed behavior, and then tests fault hypotheses by adapting the model. Similarly, DIAMON [22] combines QSIM with consistency-based diagnosis in a layered monitoring framework. It uses qualitative simulation to detect abnormal behavior and incrementally refines the model abstraction level to isolate the root cause.

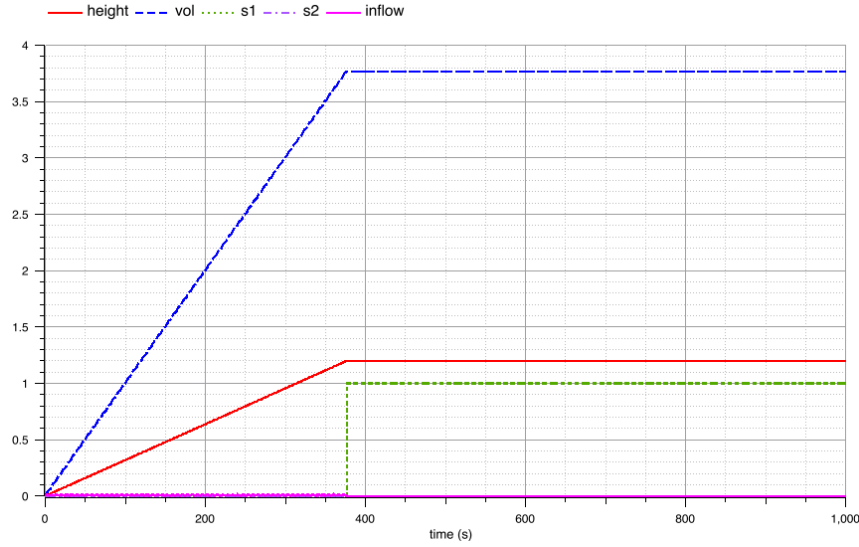
Beyond classical QSIM applications, efforts have been made to scale qualitative simulation to more complex hybrid systems. Klenk et al. [16, 17] propose translating a subset of Modelica models into qualitative constraints to enable QSIM-style simulation. The approach reduces spurious trajectories by incorporating continuity, higher-order derivatives, and landmark ordering.

QSIM has been successfully combined with frameworks based on constraints, including Constraint Logic Programming over Finite Domains (CLP(FD)) [2] and Answer Set Programming (ASP) [29]. Wiley et al. [29, 28] demonstrate that robots may autonomously learn qualitative action sequences, such as scaling obstacles or navigating uneven terrain, without requiring accurate quantitative representations of the robot or its environment when using their ASP approach to QSIM called ASP-QSIM.

Our approach builds directly on these QSIM-based diagnosis principles but introduces a conformance-based interpretation of system behavior for system monitoring and fault detection. By exploiting ASP-QSIM, our method identifies deviations between observed system trajectories and simulation-derived qualitative behaviors.

3 Illustrative example

In this section, we illustrate the suggested diagnosis approach considering a tank comprising a pipe where water flows in that has a valve for enabling water flow or stopping it, and a sink for preventing overflowing of the tank. We assume that the sink is designed such that even the amount of water coming from a fully opened input valve can be handled. The overall system has also two sensors. Sensor S_1 is for indicating that the maximum water level is reached. In this case the valve will be closed. Otherwise, the valve is set to open. Another sensor S_2 measures whether there is an outflow, which indicates a trouble. The second sensor can be seen as a form of safeguard for the outpipe. Figure 1 graphically depicts the tank example.

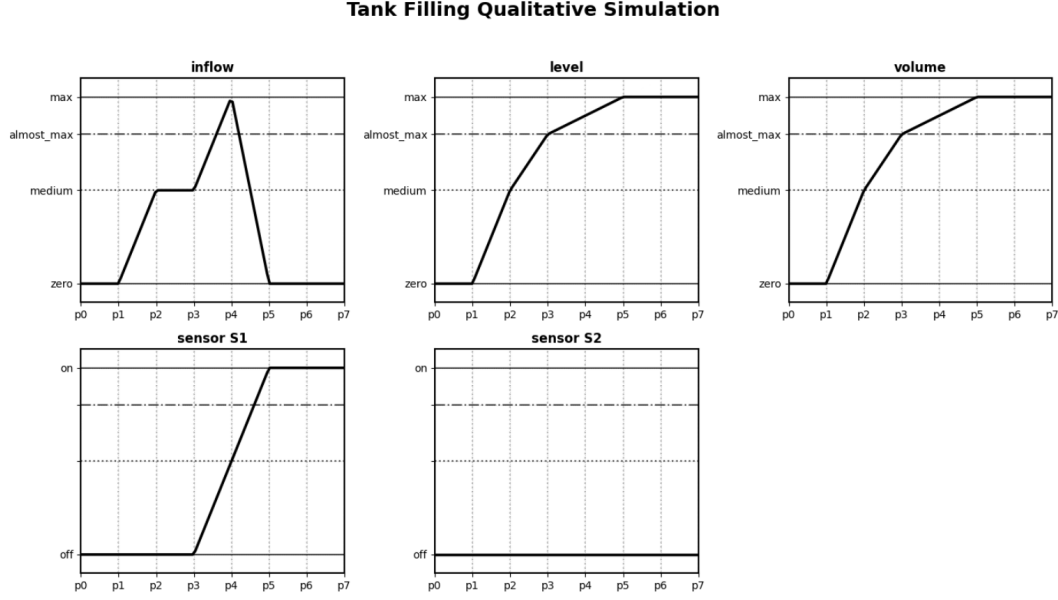


■ **Figure 2** The correct behavior of the water tank example considering the height of the water level, the volume of water in the tank, the sensors and the inflow value. Before time 400s, the water level reaches the maximum causing sensor S_1 to fire which leads to closing the valve preventing from additional inflow.

A mathematical model of this tank systems comprises one input, i.e., the volumetric flow rate, which fills the tank until reaching the maximum level. If there is still an inflow, then the water will flow through the output of the tank (of course unless it is open and not blocked). The ordinary behavior of this tank example can be modeled as follows: The tank has an area A (e.g., in form of a circle of a given radius r where $A = r^2 \cdot \pi$), a maximum height h_{max} where the water should stop, and the height of the tank h_{top} , which is higher than h_{max} . The inflow in is given in the amount of water per time, i.e., in m^3/s . The current volume of water is a function of time $V(t) = h(t) \cdot A$ where $h(t)$ is the current height at time t . This volume depends on the sum of the inflows over time, i.e., $V(t) = \int_0^t in(t)dt$, and, hence, $\frac{dV(t)}{dt} = in(t)$. If the water reaches the maximum level, the valve should close. If this is not the case the water level reaches the outpipe causing an outflow. This outflow should be the same than the inflow, i.e., $out = in$, and no further water is added to the tank. When implementing this model in a simulation language like Modelica [13], we obtain a behavior like the one depicted in Figure 2. For this behavior we assumed a tank with a radius of $1m$, a maximum level of $1.2m$, a top level of $1.5m$, and a flow rate of $0.001m^3/s$. Note that we varied the inflow over time using a sinus function. This allows us also to see that the inflow into the tank stops when the valve closes. There are fluctuations of the inflow until reaching the maximum level in Figure 2. Afterward, the inflow is zero without fluctuations.

Note that the behavior is similar in case of changed parameters but the steepness of the increase would be different. However, when using concrete values for diagnosis, i.e., the detection and explanation of misbehavior, we need to adapt any comparison function allowing us to show differences between the real and the expected behavior. If we use simulation, we further would need to adapt parameters to fit the simulation outcome with the one of the real systems due to variations between real components and ideal ones. Hence, a qualitative representation of the behavior to be used to check conformance of behavior would be a good idea. To confirm that a water tank system will operate as expected under typical operating

conditions, a qualitative simulation using ASP-QSIM was carried out. Figure 3 depicts the qualitative behavior our simulation produces for the tank example. Analyzing the plots, we see that as the tank fills, the inflow progressively decreases and eventually stabilizes at zero, while both volume and water level increase until they reach a maximum state, representing the threshold of the tank. Once this maximum is reached, the sensor S_1 switched off as expected. Hence, our simulation accurately captures the tank systems nominal behavior.



■ **Figure 3** Qualitative simulation behavior for tank system, showing qualitative transitions of inflow, level, volume and sensors across key landmarks.

In the following, we discuss the applicability of qualitative simulation for diagnosis focusing on fault detection. In particular, we want to clarify the following questions:

- What means conformance of a concrete system run and the qualitative simulation result?
- What are the limitations of qualitative simulation when applied for diagnosis? Here, we want to outline limitations considering different fault scenarios. Which type of faults can hardly or not being detected?

For the discussions on limitations, we consider the following fault cases of the tank example, assuming that the fault happens at 200s and remains permanent until the end of simulation:

Fault case 1: Sensor S_1 gets stuck at false. In this case, the controller would not get any information and not initiate closing the valve. Hence, the water will reach the outpipe causing sensor S_2 to go from false to true.

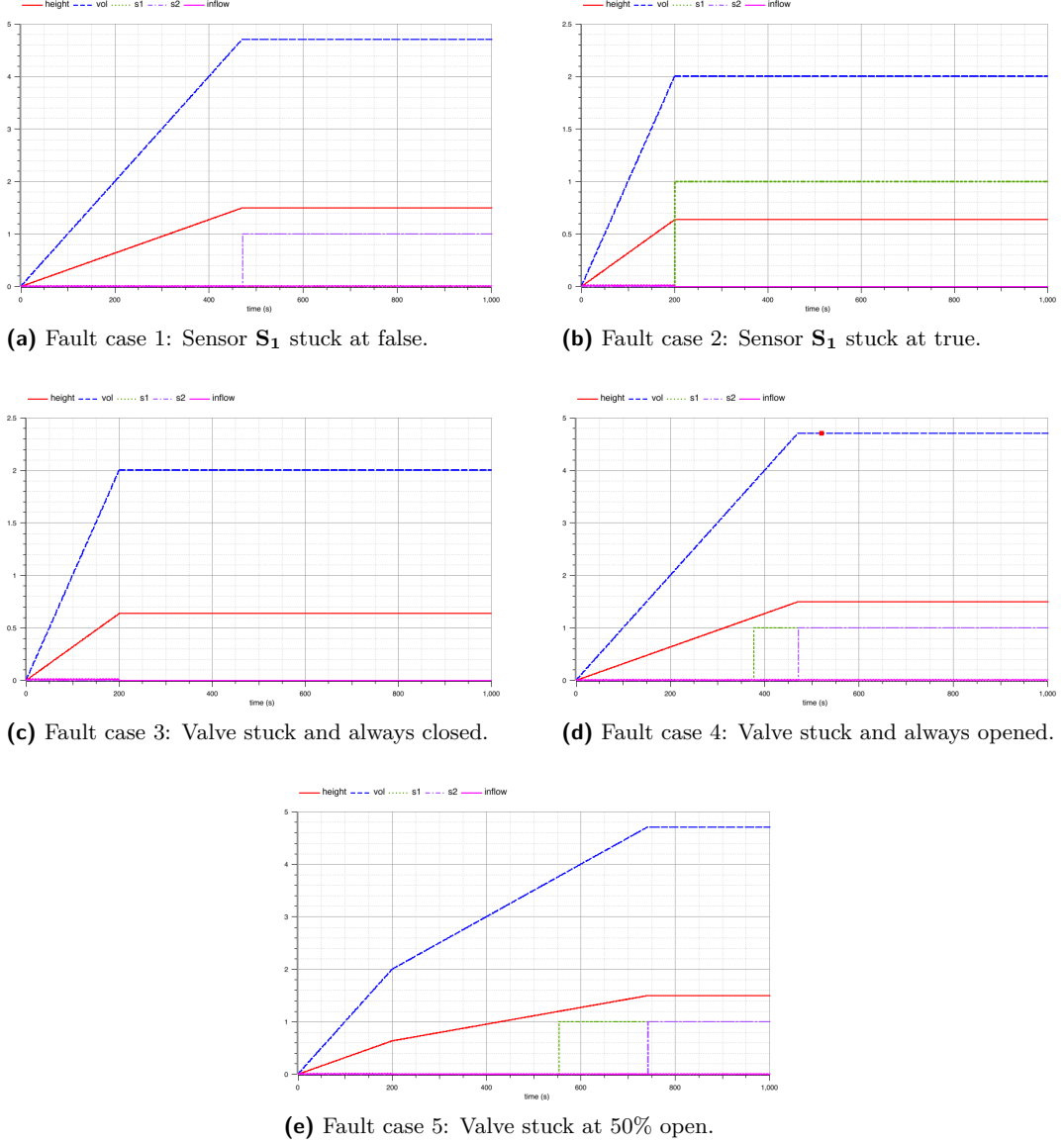
Fault case 2: Sensor S_1 is stuck at true, leading to closing the valve immediately. Hence, the tank will not be filled completely.

Fault case 3: The valve is stuck and always closed leading to zero inflow after 200s. Hence, none of the sensor will go to true and the tank cannot be filled.

Fault case 4: The valve is stuck and remains open. In this case sensor S_1 indicates reaching the maximum level but having no effect. Hence, the top level is reached and sensor S_2 will go from false to true.

Fault case 5: The valve gets stuck at 50% leading to continuous and not stopping inflow. Both sensor indicate reaching a certain water level but the water inflow does not stop.

Figure 4 shows the faulty behaviors resulting from the different fault cases. Before discussing the limitations and challenges, we introduce the basic definitions of qualitative simulation in the next section of this paper.



■ **Figure 4** Faulty behavior cases of the tank example.

4 Qualitative Simulation (QSIM)

QSIM provides a formal basis for reasoning about physical systems in situations where precise numerical data is unavailable or not needed. Unlike numerical simulation techniques, which require precise parameter values, in QSIM continuous variables are abstracted into symbolic categories such as increasing, decreasing, or landmark values. QSIM simulates all possible behaviors of a system given a qualitative model. Its inputs are: (1) QDEs represented by variables and constraints, and (2) an initial state. QSIM completes the initial state by solving

a constraint satisfaction problem (CSP) over the QDEs. For each consistent state, it then generates all valid successors and filters them using constraint consistency. Rather than predicting a single numeric outcome or trace of a simulation, QSIM captures all plausible system behaviors over time [20].

► **Example (cont.).** To illustrate how QSIM is applied in practice, consider out tank system and its system variables V . We model a single *control variable* – the **inflow** – whose value can be externally manipulated. The remaining variables are *state variables*, which represent internal or observable properties of the system. The state variables include the current water level in the tank (**level**), the volume (**volume**), and **sensor S1** and **sensor S2**, which represent sensor readings related to the tank state.

Each variable in a qualitative model is described over a symbolic structure called a *quantity space*. This space consists of key reference points in the domain of the variable, denoted as *landmarks*, and the intervals between them. Landmarks represent semantically meaningful thresholds without requiring precise numeric values. We formalize the *quantity space* as follows.

► **Definition 1** (Quantity Space (adapted from [30])). *For each variable $v \in V$, the quantity space $Q(v)$ is defined as an ordered set of landmark values l_i , i.e., $Q(v) = \{l_0, l_1, \dots, l_m\}$ with $l_0 < l_1 < \dots < l_m$.*

► **Example (cont.).** In the tank system, the landmarks for **level** may include **zero** (representing an empty tank), **medium**, **almost_max**, and **max** (representing the threshold for the tank level). The sensors **S1** and **S2** may use the landmarks **on** and **off** as shown in Figure 3.

Based on the quantity space of a variable, we can now define its *qualitative value* at a particular time step t :

► **Definition 2** (Qualitative Value (adapted from [19])). *A qualitative value $QV(V, t)$ of a variable $v \in V$ at time step t is represented as a tuple $\langle qmag, qdir \rangle$, where:*

- *$qmag$ is either a landmark or an interval between two landmarks,*
- *$qdir \in \{\text{increasing (inc)}, \text{steady (std)}, \text{decreasing (dec)}\}$ is the direction of change of v .*

► **Example (cont.).** Suppose that in our tank system model, landmarks define the quantity space for the variable **level**, which represents the water level in the tank. The ordered landmarks for this variable are: **zero** < **medium** < **almost_max** < **max**. Suppose that at time t , the water level lies within the interval (**medium**, **almost_max**) and the level is increasing. Then the qualitative value of **level** at time t is given as $QV(\text{level}, t) = \langle (\text{medium}, \text{almost_max}), \text{inc} \rangle$.

These qualitative values form the building blocks of a system's *qualitative state*, which captures the complete system snapshot at a given time:

► **Definition 3** (Qualitative State (adapted from [30])). *A qualitative state $S(t)$ at time t is defined as $S(t) = \{QV(v_0, t), QV(v_1, t), \dots, QV(v_n, t)\}$ over all $v \in V$.*

The collection of the qualitative values of all the variables in a system at a given point in time is its *qualitative state* [30].

► **Example (cont.).** For instance, in our tank example in Figure 3, at time point p_4 we obtain the following qualitative state:

$$S(p_4) = \left\{ \begin{array}{l} \text{inflow} \mapsto \langle \text{max}, \text{dec} \rangle, \text{ level} \mapsto \langle (\text{almost_max}, \text{max}), \text{inc} \rangle, \\ \text{volume} \mapsto \langle (\text{almost_max}, \text{max}), \text{inc} \rangle, \text{ sensor S1} \mapsto \langle \text{off}, \text{inc} \rangle, \\ \text{sensor S2} \mapsto \langle \text{off}, \text{std} \rangle \end{array} \right\}$$

4.1 Qualitative dynamics

A qualitative model defines valid qualitative states of the system and how changes in variables lead to transitions between them. When describing a model, qualitative constraints which restrict the magnitude and direction of variable change are represented by Qualitative Differential Equations (QDEs), which include basic constraints such as derivatives, sums, and monotonic dependencies [29].

To better reflect actual system causality in different system operational scenarios, Wiley et al. [29] introduced *qualitative rules*, which allow constraints to be applied conditionally only within specific regions of the system's state space, i.e., when the associated preconditions are met. The rule-based approach models system behavior in a more context-sensitive and dynamic manner.

► **Example (cont.).** In our tank system model, the following domain specific qualitative constraints (C1–C4) and rules (R1 and R2) are applied: ³

- C1: `deriv(level, inflow)`: The rate of change in water level is determined by inflow.
- C2: `mplus(level, volume)`: Volume increases proportionally with tank level.
- C3: `const(sensor_S1, land(off))`: Under condition `sensor_conditions_1` (see R1), the sensor `sensor_S1` is constrained to remain `off`.
- C4: `const(sensor_S1, land(on))`: Under condition `sensor_conditions_2` (see R2), the sensor `sensor_S1` is constrained to remain `on`.
- R1: `precond(sensor_conditions_1, bound, level, i, zero, i, almost_max) ⇒ const(sensor_S1, land(off))`⁴: When the tank level is between `zero` and `almost_max`, sensor `S1` remains `off`.
- R2: `precond(sensor_conditions_2, equal, level, land(max)) ⇒ const(sensor_S1, land(on))`: When the tank level reaches `max`, sensor `S1` is switched `on`.

These qualitative constraints reflect physical principles such as flow regulation (C1) and conservation of volume (C2), while rules reflect the physical behavior of the systems. For instance, the sensor stays inactive during normal filling (R1) and activates only at the threshold (R2).

Qualitative transitions define how the system evolves over time by moving from one qualitative state to another. For example, if `inflow` exceeds `zero`, the `level` of the tank increases. While *qualitative constraints* describe relationships between variables (e.g., how `level` depends on `inflow`), *transitions* capture the system's dynamics over time. The QSIM algorithm [8] systematically computes these transitions. At each time step t_i and interval (t_i, t_{i+1}) QSIM uses a transition table, which encodes all valid value combinations permitted by the active constraints [20], to generate a set of candidate successor states S_{i+1} for the current qualitative state S_i . For simplicity, we do not distinguish whether we go from one time point to an interval or vice versa. We only distinguish sequences of states. Further note that this transition process has been extended in ASP-QSIM [29] to support qualitative rules. As a result, only contextually valid transitions – as defined by the qualitative rules – are considered, ensuring that the qualitative simulation reflects both the physical structure and operational logic of the system.

³ We express constraints and rules using the ASP-QSIM formalism [29], where `const` indicates a constraint, `land` a landmark, and `precond` a precondition.

⁴ `bound` indicates that the variable `level` is between two values. The `i` indicates that the range of the variable can be between `zero` and `almost_max` including those two values.

The output of a qualitative simulation is a set of *qualitative behaviors* – each a possible sequence of qualitative states that the system might exhibit given an initial and goal condition. We refer to a single qualitative behavior also as a qualitative trajectory and one qualitative behavior reflects one consistent qualitative path the system may follow.

► **Definition 4** (Qualitative Behavior [20]). *A qualitative behavior QB is an ordered sequence of qualitative states $QB = \langle S_0, S_1, \dots, S_n \rangle$, where each state S_i maps each variable $v \in V$ to its qualitative value $\langle qmag, qdir \rangle$ at a time point or interval corresponding to i .*

► **Example** (cont.). Consider this first portion of qualitative behavior $QB = \langle S_0, S_1, S_2, \dots \rangle$ from the tank example with the following qualitative states:

$$\begin{aligned} S_0 &= \{\text{inflow} \mapsto \langle \text{max}, \text{dec} \rangle, \text{level} \mapsto \langle \text{zero}, \text{inc} \rangle, \text{volume} \mapsto \langle \text{zero}, \text{inc} \rangle, \\ &\quad \text{sensor S1} \mapsto \langle \text{off}, \text{std} \rangle, \text{sensor S2} \mapsto \langle \text{off}, \text{std} \rangle\}, \\ S_1 &= \{\text{inflow} \mapsto \langle (\text{zero}, \text{max}), \text{dec} \rangle, \text{level} \mapsto \langle \text{medium}, \text{inc} \rangle, \\ &\quad \text{volume} \mapsto \langle (\text{zero}, \text{medium}), \text{inc} \rangle, \text{sensor S1} \mapsto \langle \text{off}, \text{std} \rangle, \\ &\quad \text{sensor S2} \mapsto \langle \text{off}, \text{std} \rangle\}, \\ S_2 &= \{\text{inflow} \mapsto \langle (\text{zero}, \text{max}), \text{dec} \rangle, \text{level} \mapsto \langle \text{almost_max}, \text{inc} \rangle, \\ &\quad \text{volume} \mapsto \langle (\text{zero}, \text{medium}), \text{inc} \rangle, \text{sensor S1} \mapsto \langle \text{off}, \text{std} \rangle, \\ &\quad \text{sensor S2} \mapsto \langle \text{off}, \text{std} \rangle\} \end{aligned}$$

4.2 Conformance

Several approaches to define conformance between a system and its qualitative model have been proposed over the years. In the context of testing, Aichernig et al. [1] introduced the *griofconf* relation to formally define when an implementation under test (IUT) conforms to a qualitative model, based on whether the observed outputs under a sequence of inputs are a subset of the outputs predicted by the model. Similarly, the *ioco*-based [27] approach for qualitative action systems checks if outputs after a given input trace remain within the allowed qualitative behaviors [7]. In both cases, the qualitative models were constructed using *Garp3* [9], which is founded in Qualitative Process Theory [12]. Both approaches treat qualitative traces as symbolic abstractions of continuous behaviors. Our work adopts a similar idea by comparing traces from system observations against qualitative simulations to check the conformance of the real-world system with the simulation.

First, let us define an *Abstract Qualitative Behavior* (AQB). An AQB is a qualitative trace produced by the simulation or the result of abstracting a quantitative trace from a system run using a system-appropriate discretization. Defining AQB allows us to formally compare expected behaviors from the model with observed behaviors from the physical system using a conformance relation.

► **Definition 5** (Abstract Qualitative Behavior). *An abstract qualitative behavior AQB is an ordered sequence of qualitative states $QB = \langle S'_0, S'_1, \dots, S'_n \rangle$, where each state S'_i maps each variable $v \in V$ to its reduced qualitative state only consisting of $qmag$.*

In our AQB, we focus solely on the qualitative magnitude $qmag$ of each variable and omit the direction of change $qdir$. First, directionality can be more susceptible to noise and measurement imprecision in real-world systems, making it unreliable for behavior alignment. Second, the key behavioral distinctions we want to detect are typically captured by changes in magnitude (e.g., crossing a landmark), which directly reflect system state transitions.

In addition, as we are mainly interested in capturing changes within the system and the qualitative simulation, we abstract away redundant repetitions in a trajectory by minimizing the ABQs, i.e., $\min(AQB)$. That is, we only consider transitions from a state to a successor in case the qualitative value changes; otherwise, we consider two states to be qualitatively equivalent [6]. Consecutive repetitions of qualitatively equivalent states are removed.

► **Definition 6** (Minimal Abstract Qualitative Behavior). *Given an AQB, we define the minimal abstract qualitative behavior, $\min(AQB)$, as the subsequence of AQB in which all consecutive states are qualitatively distinct, i.e., $\min(AQB) = \langle S'_0, S'_1, \dots, S'_n \rangle$ where $\forall 0 \leq i < n : S'_i \neq S'_{i+1}$.*

► **Example** (cont.). For our tank example, we created a Modelica simulation to generate system behavior data in place of real-world measurements. However, in practice, the same abstraction process we apply to this simulated data is used to transform time-series data from real physical sensor measurements.

In order to create an abstract qualitative behavior (AQB) from the concrete quantitative data, we first create a quantitative-to-qualitative mapping, i.e., an abstraction. First, we down sample the original time series at a predetermined rate (i.e., per timestep in our case). Then, each sampled value is compared to a set of predefined landmarks $\{\theta_{\text{zero}}, \theta_{\text{medium}}, \theta_{\text{almost_max}}, \theta_{\text{max}}\}$. If it is within a small tolerance of one of these thresholds, it is considered a landmark (e.g., $\text{land}(\theta)$); if not, it is assigned to the open interval between two adjacent landmarks (e.g., $\text{interval}(\theta_i, \theta_{i+1})$).

However, direct changes from one sample to another may also affect intermediate qualitative areas. To mitigate this loss of information, we introduce n equally spaced “intermediate steps” between each pair of sampled timepoints by linearly interpolating the quantitative values and reclassifying each intermediate value. By altering the number of intermediate stages, trace length and qualitative fidelity can be compromised. We found that $n = 1$ is sufficient to capture all landmark and interval transitions in the underlying numeric trace, while $n = 0$ may overlook significant qualitative changes in our tank example. After building the complete qualitative trace, including intermediate steps, a minimization step is employed. This process, denoted as $\min(AQB)$, removes successive states that are qualitatively similar or in which all system variables retain the same qualitative values. This decrease significantly lowers the computing complexity for subsequent reasoning tasks and improves interpretability by focusing exclusively on significant behavioral adjustments. The result satisfies the formal criterion of minimal abstract qualitative behavior by ensuring that no two successive states are identical and encouraging precise and effective qualitative reasoning.

Based on the definition of AQB, we can define a conformance relation. The goal of the conformance relation is to determine whether the real system behaves in accordance with the simulation, i.e., whether at least one of the AQB generated by the qualitative simulation matches the AQB derived from an observed system run.

► **Definition 7** (Conformance). *Let $\min(AQB_{\text{real}})$ be the minimal abstract qualitative behavior derived from a system run, and let $\mathcal{A}$ be the set of AQB generated by the qualitative simulation and afterwards minimized. We define conformance as:*

$$\text{conformance}(AQB_{\text{real}}, \mathcal{A}) \iff \exists AQB_{\text{model}} \in \mathcal{A} : \min(AQB_{\text{real}}) = \min(AQB_{\text{model}})$$

This conformance relation requires an exact match over the entire trace. In principle, both the simulation and the real system could produce infinite traces, especially when viewed as executions of transition systems. However, in practical applications we only ever observe

finite prefixes of such behaviors. To address this, we adopt a bounded trajectory conformance approach, inspired by similar bounding techniques in model checking (e.g., bounded model checking [5]). Instead of requiring a full match across infinite behaviors, we check whether the real system trace matches a simulated behavior up to a fixed prefix length m .

► **Definition 8** (Bounded Conformance). *We define bounded conformance w.r.t. a prefix length m as:*

$$\text{conformance}_m(AQB_{\text{real}}, \mathcal{A}) \iff \exists AQB_{\text{model}} \in \mathcal{A} : [\min(AQB_{\text{real}})]_m = [\min(AQB_{\text{model}})]_m$$

Here, $[\min(AQB)]_m$ denotes the prefix of a minimal AQB of length m . Thus, bounded conformance reflects cases where the trajectories are equivalent for the first m qualitatively distinct states. While this bounded conformance approach is practical and scalable, it inherently limits fault detection to the prefix of length m , i.e., fault occurring after the prefix cannot be captured posing a challenge in scenarios where faults manifest beyond the checked horizon.

► **Example** (cont.). Once we have determined all qualitative observations by abstraction, including the interpolated states from the real-world data, we encode them as ASP constraints:

```
:- not holds(p(t), var, land(), _), time(p(t)).
```

`holds(p(t), var, land(), _), time(p(t))` is a fact used in ASP to define a qualitative state, where $p(t)$ is a time point, `var` refers to the qualitative variable (e.g., `level`), `land()` is the qualitative value, and `_` refers to the qualitative direction that is absent in our abstraction as mentioned earlier. `holds(...)` stipulates that a particular qualitative state must exist at a particular point in time, while `:- not holds(...)` ensures that the model is rejected if the fact `holds(...)` is absent. In essence it guarantees that the observed behavior is present in all valid answer sets. Under the guidance of its internal rules, landmarks, and preconditions, ASP-QSIM attempts to generate valid pathways that logically “fill in the gaps” around an initial state and a goal state taken from the quantitative trace. The result of this, as stated earlier, is a set of minimal AQBs.

If one of the simulation model’s produced AQBs exactly matches the abstract qualitative behavior AQB_{real} of the real system, then we have shown bounded conformance. In our case, we compared the full observed trace with the complete simulated trajectories. Our experimental results confirmed that the qualitative model accurately captures the system’s dynamics, i.e., every qualitative trace derived from the quantitative data of the nominal system behavior matched at least one valid trajectory generated by the simulation model.

4.3 Fault detection

While conformance checking determines whether an observed system behavior matches one of the expected qualitative behaviors generated by the model, it also offers a principled basis for revealing faults. In practice, deviations from nominal qualitative trajectories – especially in the ordering or presence of sensor events – may indicate abnormal or unexpected conditions. By systematically comparing abstracted observations to simulated behaviors, we can detect faults as violations of the conformance relation.

► **Example** (cont.). Qualitative simulation is especially effective in detecting any anomaly that adds, removes, or reorganizes sensor “on/off” transitions relative to the expected nominal model behavior. We were able to detect the following faults, as those faults alter the logical structure of the qualitative state transitions, which QSIM explicitly models:

Fault case 1 (Sensor S1 stuck false): Sensor S2 turns `true` without a prior Sensor S1 activation, breaking the expected causal sequence.

Fault case 2 (Sensor S1 stuck true): Sensor S1 changes to `true` prematurely, i.e., before the tank reaches the defined threshold.

Fault case 4 (Valve stuck and always opened): Sensor S1 turns `true` but fails to trigger valve closure; Sensor S2 becomes `true` as the valve is not closed.

Fault case 5 (Valve stuck at 50% open): An unanticipated pattern of sensor events appears that does not match any valid qualitative trajectory.

However, we cannot detect **Fault case 3 (Valve stuck and always closed)**. The system enters a stagnant condition as both sensors remain off and the water level remains constant due to the absence of inflow. Since QSIM observes only changes in landmarks or directional trends, such “silent” failures, where there are no new transitions, are indistinguishable from nominal behaviors, e.g., the tank has partially stabilized [20]. Slight quantitative differences that do not cross these symbolic boundaries (i.e., `increasing`, `decreasing`, `steady`) and landmarks are ignored [18].

5 Conclusions

By abstracting quantitative system behavior into qualitative representations, we demonstrated how qualitative simulation can effectively detect anomalies across various scenarios by formally defining conformance between the qualitative simulation and concrete quantitative data. At the same time, our evaluation uncovered important limitations. In particular, the exclusion of directional information hindered the detection of one of the faults, which might have been caught had directionality been preserved during abstraction. This highlights a central problem of exclusively qualitative models, even though they can capture high-level patterns and behavior, they may not be able to detect defects that depend on small or deliberate aberrations.

However, the power of qualitative simulation is in its ability to provide a methodical and understandable approach to diagnosis, especially in situations where complete quantitative data is unavailable or unreliable. When considering fault scenarios, it offers a model-based foundation and excels at identifying anomalies and discrepancies across different system behaviors.

Future improvements could include the integration of directional cues to enhance sensitivity to small or gradual deviations, which are currently lost in the reduced abstraction. While the exclusion of the direction of changes simplifies the qualitative behavior space and reduces computational overhead, selectively reintroducing it either for specific variables or during suspected fault windows could strike a balance between fidelity and tractability.

Additionally, the development of hybrid approaches that combine qualitative and quantitative reasoning [4] could further improve fault detection robustness. Such methods may allow the system to benefit from the interpretability and flexibility of qualitative models, while leveraging quantitative models for precision in ambiguous or borderline cases.

Another area of enhancement lies in the application of bounded conformance not just as a binary check but as a tool for localizing faults temporally within the trace. By identifying the prefix length at which divergence from expected behavior occurs, diagnosers could potentially infer the onset time and probable subsystem associated with the deviation. To address the challenge of “silent” faults failures that do not result in a change of qualitative state and

thus evade detection future work could explore the use of context-aware thresholds, state persistence checks, or anomaly scoring models that monitor for suspicious invariance in critical variables.

Moreover, QSIM may generate spurious or physically implausible trajectories, particularly in more complex systems. Incorporating domain knowledge, constraint filtering, or ranking heuristics will therefore be crucial for scaling the approach to real-world applications.

References

- 1 Bernhard K. Aichernig, Harald Brandl, and Franz Wotawa. Conformance testing of hybrid systems with qualitative reasoning models. *Electronic Notes in Theoretical Computer Science*, 253(2):53–69, 2009. Proceedings of Fifth Workshop on Model Based Testing (MBT 2009). doi:10.1016/j.entcs.2009.09.051.
- 2 Aleksander Bandelj, Ivan Bratko, and Dorian Suc. Qualitative simulation with clp. In *Proceedings of the 16th International Workshop on Qualitative Reasoning (QR02)*, 2002.
- 3 I. Bellanco, E. Fuentes, M. VallÅss, and J. Salom. A review of the fault behavior of heat pumps and measurements, detection and diagnosis methods including virtual sensors. *Journal of Building Engineering*, 39:102254, 2021. doi:10.1016/j.jobe.2021.102254.
- 4 Daniel Berleant and Benjamin J. Kuipers. Qualitative and quantitative simulation: bridging the gap. *Artificial Intelligence*, 95(2):215–255, 1997. doi:10.1016/S0004-3702(97)00050-7.
- 5 Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman, and Yunshan Zhu. Bounded model checking. *Adv. Comput.*, 58:117–148, 2003. doi:10.1016/S0065-2458(03)58003-2.
- 6 Harald Brandl, Gordon Fraser, and Franz Wotawa. Qr-model based testing. In *Proceedings of the 3rd international workshop on Automation of software test*, pages 20–17, 2008.
- 7 Harald Brandl, Martin Weiglhofer, and Bernhard K. Aichernig. Automated conformance verification of hybrid systems. In Ji Wang, W. K. Chan, and Fei-Ching Kuo, editors, *Proceedings of the 10th International Conference on Quality Software, QSIC 2010, Zhangjiajie, China, 14-15 July 2010*, pages 3–12. IEEE Computer Society, 2010. doi:10.1109/QSIC.2010.53.
- 8 Ivan Bratko. *Learning Qualitative Models*. Pearson Education, 4 edition, 2012.
- 9 Bert Bredeweg, Floris Linnebank, Anders Bouwer, and Jochem Liem. Garp3 — workbench for qualitative modelling and simulation. *Ecological Informatics*, 4(5):263–281, 2009. Special Issue: Qualitative models of ecological systems. doi:10.1016/j.ecoinf.2009.09.009.
- 10 Qiong Chen and Nan Li. Model predictive control for energy-efficient optimization of radiant ceiling cooling systems. *Building and Environment*, 205:108272, 2021.
- 11 Daniel Dvorak and Benjamin Kuipers. Process monitoring and diagnosis: a model-based approach. *IEEE expert*, 6(3):67–74, 1991. doi:10.1109/64.87688.
- 12 Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24(1-3):85–168, December 1984. doi:10.1016/0004-3702(84)90038-9.
- 13 Peter Fritzson. Modelica - a language for equation-based physical modeling and high performance simulation. In *Proceedings of the 4th International Workshop on Applied Parallel Computing, Large Scale Scientific and Industrial Problems, PARA '98*, pages 149–160, London, UK, UK, 1998. Springer-Verlag. doi:10.1007/BFB0095332.
- 14 Russell Greiner, Barbara A. Smith, and Ralph W. Wilkerson. A correction to the algorithm in Reiter's theory of diagnosis. *AI*, 41(1):79–88, 1989. doi:10.1016/0004-3702(89)90079-9.
- 15 Johan De Kleer and John Seely Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24(1-3):7–83, December 1984. doi:10.1016/0004-3702(84)90037-7.
- 16 Matthew Klenk, Johan de Kleer, Daniel G. Bobrow, and Bill Janssen. Using modelica models for qualitative reasoning. In *Proceedings of the 27th International Workshop on Qualitative Reasoning (QR 2013)*, pages 1–8, 2013.

- 17 Matthew Klenk, Johan de Kleer, Daniel G. Bobrow, and Bill Janssen. Qualitative reasoning with modelica models. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1084–1090. AAAI Press, 2014. doi:10.1609/AAAI.V28I1.8876.
- 18 Benjamin Kuipers. The limits of qualitative simulation. In *IJCAI*, 1985.
- 19 Benjamin Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. *Automatica*, 25(4):571–585, July 1989. doi:10.1016/0005-1098(89)90099-X.
- 20 Benjamin J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–338, 1986. doi:10.1016/0004-3702(86)90073-1.
- 21 Benjamin J. Kuipers. Qualitative simulation, 2001. In *Encyclopedia of Physical Science and Technology*, pages. 287–300.
- 22 Franz Lackinger and Wolfgang Nejdl. Integrating model-based monitoring and diagnosis of complex dynamic systems. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pages 1123–1128. Morgan Kaufmann, 1991. URL: <http://ijcai.org/Proceedings/91-2/Papers/074.pdf>.
- 23 Bruno Lucas, Jean Michel Evrard, and Jean Pierre Lorre. A qualitative diagnosis method for a continuous process monitor system. In *Proceedings of the IMACS International Workshop on Qualitative Reasoning and Decision Technologies -QUARDET'93*. CIMNE, 1993.
- 24 Bernhard Rinner and Benjamin Kuipers. Monitoring piecewise continuous behaviors by refining semi-quantative trackers. In Thomas Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 1080–1086. Morgan Kaufmann, 1999. URL: <http://ijcai.org/Proceedings/99-2/Papers/059.pdf>.
- 25 Siddharth Subhramanian and Raymond J. Mooney. Multiple fault-diagnosis using general qualitative models with fault modes. In *Working Papers of the 5th International Workshop on Principles of Diagnosis*, pages 321–325, New Paltz, NY, October 1994.
- 26 Louise Travé-Massuyès, Liliana Ironi, and Philippe Dague. Mathematical foundations of qualitative reasoning. *AI Magazine*, 24(4):91–106, 2003. doi:10.1609/AIMAG.V24I4.1733.
- 27 Jan Tretmans. Conformance testing with labelled transition systems: Implementation relations and test generation. *Comput. Networks ISDN Syst.*, 29(1):49–79, 1996. doi:10.1016/S0169-7552(96)00017-7.
- 28 Timothy Wiley, Claude Sammut, and Ivan Bratko. Using planning with qualitative simulation for multistrategy learning of robotic behaviors. In *Proceedings of the 27th International Workshop on Qualitative Reasoning*, pages 24–30, 2013.
- 29 Timothy Wiley, Claude Sammut, and Ivan Bratko. Qualitative simulation with answer set programming. In *European Conference on Artificial Intelligence (ECAI)*, May 2014.
- 30 Timothy Colin Wiley. A planning and learning hierarchy for the online acquisition of robot behaviors, 2017. PhD Thesis Defence Paper.