



HAL
open science

Qualitative Reasoning and Cyber-Physical Systems: Abstraction, Modeling, and Optimized Simulation

Baptiste Gueuziec, Jean-Pierre Gallois, Frédéric Boulanger

► **To cite this version:**

Baptiste Gueuziec, Jean-Pierre Gallois, Frédéric Boulanger. Qualitative Reasoning and Cyber-Physical Systems: Abstraction, Modeling, and Optimized Simulation. Innovations in Systems and Software Engineering, 2024, 10.1007/s11334-024-00567-0 . hal-04661330

HAL Id: hal-04661330

<https://hal.science/hal-04661330v1>

Submitted on 24 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Qualitative Reasoning and Cyber-Physical Systems: Abstraction, Modeling, and Optimized Simulation

Baptiste GUEUZIEC^{1*}, Jean-Pierre GALLOIS¹ and
Frédéric BOULANGER²

¹Université Paris-Saclay, CEA LIST, Palaiseau, F-91120, France.

²Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire Méthodes Formelles, Gif-sur-Yvette, F-91190, France.

*Corresponding author(s). E-mail(s): baptiste.gueuziec@cea.fr;
Contributing authors: jean-pierre.gallois@cea.fr;
frederic.boulanger@centralesupelec.fr;

Abstract

Complex systems modeling and simulation are critical in many industrial and research fields, specifically to predict, prove, verify, and understand the behavior of cyber-physical systems. The diversity of variables in a system creates complexity and a need for more efficient modeling and simulation methods. In the case of hybrid systems, the heterogeneity of the discrete and continuous parts makes these tasks more challenging, adding the necessity to manage different types of variables and trajectories separately. Qualitative reasoning offers a paradigm to study the behavior of such systems with a high level of abstraction, trading precision and specificity against generality and formalism. In some situations, this paradigm can be preferred to numerical analysis, especially in the upstream study of a system in its design phases when information and knowledge about its components are incomplete. However, the different representations and contexts of such systems create a significant obstacle to defining a general methodology for applying qualitative reasoning and modeling to every case. This article proposes a method and presents a tool prototype to unify different qualitative reasoning techniques on complex cyber-physical systems described by polynomial ordinary differential equations. Both are illustrated in case studies to highlight their benefits. It also presents some possibilities of generalization to systems that do not satisfy the criteria currently required by our approach.

Keywords: qualitative reasoning, modeling, abstraction, cyber-physical systems

1 Introduction

Cyber-Physical Systems (CPS) [1] are central to many disciplines. The ability to study, decompose, and understand them and their behavior is crucial to improving our technical abilities in many industrial and scientific areas. Many methodologies exist to study them, either at a component level or from a more systematic point of view. However, all these methods lack generality: they were mainly thought for specific applications, be it numerical simulation, diagnosis, or monitoring. Their high degree of specialization makes them poorly adapted to reason on systems or tasks requiring more general and abstract properties. Moreover, the hybrid nature of many CPSs, including feedback or human interactions, adds a new form of complexity. As hybrid systems [2] combine discrete and continuous variables (see Definition 1 and Definition 2), their study is a more challenging task and must involve more efficient and general methods.

Definition 1 (Hybrid System). *We consider a system to be hybrid if it exhibits both continuous and discrete behaviors. These systems can be described with:*

- a discrete variable Q representing the control mode of the system, defined on the finite domain \mathbf{Q} , and whose value is noted m and called mode.
- a set X of continuous variables defined on the continuous set \mathbf{X} , whose valuation is noted x . The components of X are noted X_i , and their valuations x_i . We often consider $\mathbf{X} = \mathbb{K}^n$ with $n = |X|$ and \mathbb{K} a field with the usual relations ($<$, $>$, $=$). In most cases, $\mathbb{K} = \mathbb{R}$.
- a function I mapping each mode m to a set of predicates $Inv_m \subset 2^{\mathbf{X}}$ representing the invariant conditions of the system for the mode m .
- a set $T \subset \mathbf{Q} * 2^{\mathbf{X}} * \mathbf{Q} * \mathbf{X}^{\mathbf{X}}$ of modal transitions represented by the departure mode m_1 , a guard condition associated to the transition, the target mode m_2 , and a reset function giving the new value of X when entering m_2 .
- the flow F mapping every mode m of Q to the dynamics vector containing ordinary differential equations (ODE) on X and its derivatives.

Definition 2 (Variable). *We define a variable as a measurable quantity with a physical unit, a value, and a variation set.*

Qualitative reasoning, as defined by Forbus [3], refers to the area of computer sciences aiming at representing and reasoning on models with very little and imprecise knowledge. In a complementary way, Sugeno defines qualitative modeling as the design of models adapted to support qualitative reasoning [4]. Originally, qualitative modeling was introduced by Brown [5] and De Kleer [6], who developed the concept of qualitative knowledge about systems and processes. They designed this paradigm mainly for electronics and computer-assisted physics computation. They did not see this representation of knowledge as a substitute for numerical computation but as a complementary strategy. Indeed, they presented their approach as a tool to solve general problems by reasoning at a high level of abstraction on the system and to refine

the more specific sub-problems that could only be solved with a more classic numerical computation. Therefore, the initial idea was to add intelligence in problem-solving and optimize the use of computational resources that should be kept for sub-problems that require them. If the notions implied by qualitative reasoning have evolved since then, the ambition remains unchanged: proposing a new reasoning tool to supplement the exact but too specific numerical approaches available to study the different kinds of systems. Major elements have been added to the theory of qualitative modeling, such as naive physics [7, 8] by Hayes and Smith, the theory of dynamic processes [9], and the concept of conceptual closure of such a theory [10] by Forbus. Qualitative modeling has made significant advances with the works of Kuipers [11] on the qualitative representation of the state space and the value of system variables. He introduced a form of reasoning based on sign algebra, using the values $\{-, 0, +\}$ as abstractions of the numerical values of the variables. The authorized operators are $\{+, *\}$, and they illustrate the advantage of the sign algebra as they preserve all their properties of transitivity, associativity, and commutativity [12]. This method allows the qualitative study of the behavior of a system based on qualitative differential equations, which are abstractions of numerical differential equations. The development of this analysis led to the development of the **QSIM** tool. However, the non-determinism of such operations and the lack of precision showed the limits of the approach in the case of systems with feedback or multiple successors for a given state. Kuipers and Berleant partially solved these problems with their work on semi-qualitative reasoning [13]. In this approach, the sign algebra is completed with interval propagation to integrate a part of numerical analysis and resolve the uncertainties that cannot be studied with only sign knowledge. This complementarity allowed the study of more complex systems and the development of more advanced versions of his tool, such as **SQSIM** and **Q3**. Some work has been undertaken to combine it with orders of magnitude, like Medimegh in [14], but the results did not meet the expectations. However, interval propagation has since progressed, allowing advances such as flow-pipe computation presented by Bouissou in [15] or more precise uncertainty measures and correction. Tiwari completed this approach [16] and generalized it to ordinary differential equations (ODE) under the condition that the terms of the equations must be polynomial according to the system's variables. This methodology requires an additional step of numerical analysis upstream of the system study. However, it can give more interesting results as it considers the links between the dynamics and the values of a system, while previous methods had the drawback of separating the two aspects.

In terms of applications, qualitative reasoning mainly allows formal computation and proof using dynamic logic [17], diagnosis [18, 19], verification [20, 21] and approximation [22]. Qualitative modeling implies reasoning on multiple levels of abstraction. The most important ones are the design space (also named 'domain space' by Forbus [10]), centered on constructing an adapted representation of the system and its context. Working on the design is the most advanced form of qualitative reasoning. It involves non-instantiated and even partially defined systems and requires the ability to compute and solve equations with purely symbolic and non-valuated constants. The dynamic equations of such a system can, for example, be defined as $\dot{x} = ax^3 - 3xy^2$ with a a symbolic constant with an unknown value. The stakes of such a challenge are

to help design complex systems very upstream in their development and to work on models defined with incomplete knowledge.

The state space of a system is a more tangible field, as we can expect all the constants of the system to be at least partially known. In this publication, we will focus on exploring the state space. Still, we consider these results preliminary to future work exploring both the design and state spaces. This article presents the current state of our methodology and tool to create a qualitative model from a representation of the system, explore this model and build its behavior tree, and exploit it for behavior analysis, simulation piloting, or system monitoring.

This article first presents the technical concepts on which we intend to apply qualitative reasoning in [section 2](#). Then, in [section 3](#), it develops a classic abstraction method to create a qualitative model from a numerical system described by polynomial equations. [Section 4](#) presents our main contribution to qualitative reasoning by developing the concept of qualitative zones. Then, [section 5](#) describes and explains our work to develop the prototype of a tool aiming at creating qualitative models from numerical CPSs. [Section 6](#) briefly develops applications allowed by qualitative reasoning upgraded by our contribution. Finally, [section 7](#) overviews our works to generalize qualitative reasoning to more general complex systems with less convenient structures.

2 Concepts

2.1 Cyber-Physical Systems

The main application of our work on qualitative modeling is the analysis of CPSs. In this term, we include every system consisting of physical elements controlled by a computational process. Such a system can be represented using the classic form of hybrid systems [23] $S = \langle Q, X, I, F, T \rangle$ with Q the discrete variable defined on \mathbf{Q} , X the vector of continuous variables defined on \mathbb{K}^n , F the flow of the system (i.e., its dynamics), which maps the value m of Q to the set of ODE that define the dynamics of X in mode m , m_{init} the initial mode and x_{init} the initial value of X . I is defined as explained in [1](#) and represented as a mapping from \mathbf{Q} to predicates on \mathbf{X} . T is the set of all modal transitions, represented as quadruplets (initial mode, guard condition, arrival mode, reset function), where the reset function is defined in $\mathbf{X}^{\mathbf{X}}$ and gives the value of the continuous variables in m_2 after the transition from the value they had in m_1 before the transition. The equations of F are generally limited to $\mathbb{K}[X_i]_{X_i \in X}$ the set of polynomials on \mathbb{K} with variables in X (we will use the notation $\mathbb{K}[X]$ for the rest of this article). As F defines the continuous behavior of the system for each mode and T the discrete switches between modes, we can now model the system's operation. For example, let us define a model of a thermostat as:

$$\begin{aligned} Q &= \{mode\}, & \mathbf{Q} &= \{\text{off}, \text{on}\}, \\ X &= \{x\} \text{ with } x \text{ the temperature of the system in Celsius,} \\ \mathbf{X} &= \mathbb{R}, \\ I &= \{mode = \text{off} : x > 60, mode = \text{on} : x < 100\}, \\ F &= \{mode = \text{off} : \dot{x} = -x, mode = \text{on} : \dot{x} = 100 - x\}, \\ T &= \{(\text{off} \xrightarrow{x \leq 70} \text{on}, x \mapsto x), (\text{on} \xrightarrow{x \geq 80} \text{off}, x \mapsto x)\} \end{aligned}$$

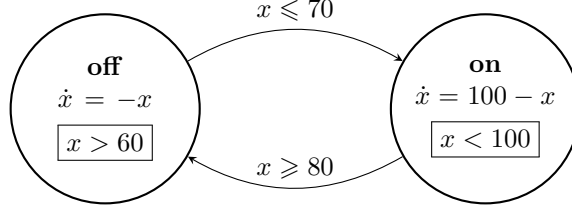


Fig. 1 Hybrid model of a thermostat

2.2 Hybrid Automaton

Hybrid automata (see [Definition 3](#)), as described by Henzinger [2], are a standard and efficient tool to represent and study CPSs.

Definition 3 (Hybrid Automaton). *An hybrid automaton can be defined as $H = \langle Q, X, V, E, Init, I, F, J, L \rangle$ with*

- Q , X , I , and F corresponding respectively to the discrete variable, the set of continuous variables, the invariant constraints, and the flow equations as defined in definition 1 for the associated CPS.
- V the vertices of H representing all its control modes and E the edges between the vertices, corresponding to modal transitions.
- $Init$ the set of initial conditions, with predicates constraining the initial values of Q and X .
- J a jump condition function that associates to each transition $e \in E$ a predicate. It corresponds to the guard condition of T in definition 1.
- L a pair composed of a set of labels and a mapping from E to these labels. The label associated with a transition can be the nature of the transition, its cause, or its specificity.

The structure of hybrid automata is particularly adapted to represent the behavior of a CPS. For example, we represented the thermostat system described above as a hybrid automaton in [Figure 1](#). This classic definition isolates the different modes of the system as different states of the automaton. Each one is characterized by its specific flow conditions (i.e., the continuous dynamics associated with the mode) and by transitions and initial conditions, defining the values of the elements of X that allow a transition to another mode and the reset function imposed after an incoming transition. However, the trajectory of X is not taken into account in this automaton model. In simple cases like this one, it can be deduced from the expression of the flow, whereas it can be impossible for more complex systems. This structure is oriented toward studying discrete behaviors and possible mode jumps rather than continuous behavior and effective transitions. However, studying CPSs also requires observing the continuous behavior of the system. To this extent, we must not only deal with the dynamics of the continuous variables but also with their trajectories and properties, as these elements can give us helpful information about the behavior of the global system. Moreover, knowing the behavior of the continuous variables is necessary to

know which transitions (constrained by the value of X) can occur and which ones are impossible in the current mode. To this extent, we need to be able to compute the trajectory of these variables.

One can compute these trajectories at two levels of abstraction, with two main families of techniques. The first one, and the most commonly used, is the family of numerical methods. Here, the trajectories are obtained with numerical simulation techniques, including, for example, Euler integration or Runge Kutta, that can give exact and reliable results. The main problem of numerical methods in the context of partially known systems is that they do not allow any generality in the simulation and may require very high computation time for complex systems. Moreover, any uncertainty can propagate and give very unsatisfactory results. Uncertainty management must use more complex techniques, such as interval propagation methods, which require more time and computational resources. Therefore, as we especially seek an important generality more than precision at this step of the system study, we prefer the other option, which is qualitative reasoning. This choice implies discretizing the state space in a finite number of qualitative states, which are abstractions of the numerical values they include. This discretization will allow the computation and study of an abstract trajectory among a set of qualitative states. These trajectories will benefit from being very general and represent a whole family of numerical traces as a type of behavior. We will represent such trajectories in another type of hybrid automata featuring both modal and intra-modal transitions. From now on, we will present how to obtain these trajectories.

3 System Abstraction

3.1 State Space Discretization

Definition 4 (Discretization). *If \mathbb{K} is a continuous set, we call discretization of \mathbb{K} a finite partition of \mathbb{K} in $\mathbb{K}_i \subset \mathbb{K}$ with $i \in [1, k]$ and $k \in \mathbb{N}$, meaning that $\bigcup_{i \in [1, k]} \mathbb{K}_i = \mathbb{K}$ and $\forall (i, j) \in [1, k]^2, i \neq j \implies \mathbb{K}_i \cap \mathbb{K}_j = \emptyset$.*

The first step in computing the qualitative trace of a system is to transform our representation of the system to a qualitative model to serve as computation support.

Definition 5 (Qualitative Model). *Let S be either a CPS or a numerical representation of a CPS. A qualitative model associated with S is a representation of S trading precision and knowledge for a convenient highlight of the relations between the elements of a discretization of the system. These relations can be a causal inference or a temporal ordering.*

Let us consider a system $S = \langle Q, X, I, F, S_0, T \rangle$ with the notations already defined. We consider that the flow of the system is represented by a mapping of each mode to an ordinary differential equation $\dot{X} = F(X(t), t)$ with t the time parameter of the system.

To illustrate our explanations, we will take as an example the hybrid Brusselator system, expressed as:

$$\begin{aligned}
S = \langle & \\
& Q = \{mode\}, X = \{x, y\}, \\
& \mathbf{Q} = \{1, 2\}, \mathbf{X} = \mathbb{R}^2, \\
& I = \{mode = 1 : x, y > 0; mode = 2 : x, y > 0\}, \\
& F = \{mode = 1 : \\
& \quad (\dot{x}, \dot{y}) = (1 - (b_1 + 1)x + a_1x^2y, b_1x - a_1x^2y), \\
& \quad mode = 2 : \\
& \quad (\dot{x}, \dot{y}) = (1 - (b_2 + 1)x + a_2x^2y, b_2x - a_2x^2y), \\
& S_0 = (1, (5.3, 2.6)), \\
& T = \{(1, x < y, 2, Id), (2, x > y, 1, Id)\} \\
& \rangle
\end{aligned}$$

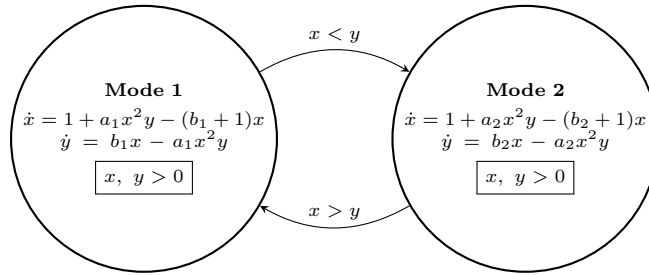


Fig. 2 Hybrid model of a Brusselator system

With this representation, it is possible to visualize the discrete part of the model as a hybrid automaton (see [Figure 2](#)). Different abstraction methods have been introduced and used for continuous evolution. The choice of the abstraction method is critical because it will strongly influence state space exploration. Among the methods developed and studied, the most noticeable for CPS study are the methods of Kuipers and Tiwari. The use of more advanced reasoning techniques implies converting the continuous trajectory into a discrete evolution with the help of an abstraction function α that associates each position of the system to a qualitative state, which belongs to a finite set.

The nature of this function is what differentiates the various modeling approaches. The first method, brought by Kuipers, only reasons in terms of landmarks (i.e., hyperplanes defined by $X_i = c$ with c a constant and $i \in \llbracket 0, |X| - 1 \rrbracket$). The abstraction of the state space is made separately for the different variables of X and for their derivatives. The abstraction of the state space is processed using landmarks on the zeros of the variables and abstractions of the system's differential equations named qualitative differential equations (QDE). These abstractions allow any change of sign of a component X_i to influence the other components. For example, if $X = (x, y)$ and if x has a positive influence on y , then the QDE characterising it will be $y = M^+x$.

In the case of our case study system, $\dot{x} = 1 - (b + 1)x + ax^2y$ will be replaced by $\dot{x} = M^+y + M^+x * M^+y$. As variables and their derivatives are not completely linked anymore because of the high level of abstraction, using these landmarks on the components of \dot{X} is more complex and gives little information. Actually, $\dot{x} = ay$ and $\dot{x} = ay^2$ are not different in this abstraction space applied to \mathbb{R}^+ .

It is also possible to use other landmarks (i.e., with $X_i = c \neq 0$) deduced from prior knowledge about the system. For example, suppose we know that 100 km.h^{-1} and 1 km.h^{-1} are important milestones around which the qualitative reasoning about an autonomous car should process. In that case, they will be considered as reference landmarks to compare and abstract the current value of the speed. This integrates elements of order of magnitude reasoning as presented by Travé-Massuyès and Dague in [12]. However, as these values are not from the sign algebra, they cannot be propagated in the equations as their properties do not match all the properties of the most convenient algebra. This method allows simple studies of systems based on explicit values chosen according to the objectives and the context of the CPS. It means it is required to have a predefined instance and context of the system and to know where and why it will be used. This constraint contradicts the main objective of qualitative modeling: we must create models with as little information as possible, so we should avoid contextual frontiers.

Aichernig [24] further formalized the abstraction process but did not improve the generality of the approach, as its contribution is still based on the comparison of variables to landmarks that cannot always be anticipated. This abstraction method fits an already defined system with sufficiently precise knowledge about the intended use and behaviors but is not suitable for partially known or designed systems.

The approach by Tiwari [16] has resolved this drawback: for each mode $m \in \mathbf{Q}$, using the equations defining the system (both the dynamic equations, the invariant expressions, and the transition conditions), the algorithm defines new variables derived from X . Tiwari assumes that all these equations have a polynomial form according to the components of X . Using all the elements p from F_m , I_m , and T_m (with F_m , I_m , and T_m being the subset of F , I and T associated with the mode m) such that $p \in \mathbb{K}[X]$, we define a set P_m initiated with $P_m = \{X_0, \dots, X_{n-1}\}$ with $n = |X|$. $\forall p \in \mathbb{K}[X] \cap (F_m \cup I_m \cup T_m)$, we set $x_p = p$ and we add each x_p to P_m . One can note that previously mentioned landmarks can be integrated into these polynomial equations once represented as $X_i - c_i$ for a landmark $X_i = c_i$. Then, $\forall p \in P_m$, if $\dot{p} \neq 0 \wedge \dot{p} \notin P_m \wedge \nexists (b, d) \in \mathbb{K}[X] * P_m$ such that $\dot{p} = b * d$, then \dot{p} is added to P_m . This condition quickly suppresses the nilpotent and idempotent polynomials. As the considered polynomials include terms only defined by their differential equation, it is likely that many of them are neither nilpotent nor idempotent. Therefore, the more they are derived, the more complex their expression will become. For example, in the case of the Brusselator, the expression of $\dot{x} = 1 + ax^2y - (b + 1)x \in \mathbb{K}[x, y]$ will be added to P . Once derived, the obtained expression will be $\ddot{x} = a(-ax^2y + bx)x^2 + 2a(ax^2y - (b + 1)x + 1)xy - (b + 1)(ax^2y - (b + 1)x + 1)$. Deriving polynomials from x and y many times according to the time parameter creates a refinement of the qualitative model. However, it may increase computational complexity up to a certain point without major precision improvement. Therefore, choosing a criterion to

stop the filling of P_m is necessary. The more elements P_m will contain, the more the qualitative model will be refined, so this choice corresponds to the search for a trade between precision and complexity. This criterion must be chosen before the execution, so there is still a problem with non-instantiated systems. If no information is available about the needed precision of the abstraction or the usefulness of new reference values, the criteria will have to be chosen arbitrarily. In [16], the authors did not consider this a problem, as whenever the discretization is stopped, the result is still an abstraction of the system. They do not see over-refinement to be disturbing. However, depending on the use of the model, the searched precision will be completely different.

The search for stopping criteria for the abstraction process is a challenge on its own and would require further investigations. We do not see the expression of a universal criterion as possible from what emerged from the current works. However, something quite general can be proposed. The first criterion we could add to the modeling process is a constraint on the maximal complexity of the resulting qualitative model. This complexity can be expressed with the number of elements of P_m , giving a maximum estimation of the possible number of existing qualitative states and, by a simple computation, the maximum estimation for the number of transitions. This computation of the complexity of the model is an analogy of state space complexity (see Gao in [25]) to the discretization of the system state space. For a discretization achieved with a set P_m of size l_m , we can bound the maximum number of qualitative states in the mode m to 3^{l_m} , and therefore the maximum number of transitions between these states to 3^{2l_m} . The maximum complexity of the qualitative model is, in consequence, directly dependent on $|P_m|$. An idea would be to stop the abstraction process when each P_m reaches a limit size. Still, a new challenge is raised by the hybrid nature of treated systems. Should the different modes be treated separately with a maximum complexity associated with each, or should the model be treated as a whole with a global maximum complexity? We still have to study this question more profoundly. Anyway, the intuitive answer is that the consideration of the complexity of the complete model could provoke a disturbing imbalance between the granularity of the state spaces of the different modes. Therefore, the choice of a maximal authorized complexity per mode, equivalent to a maximal number of qualitative states for each mode, seems to be the most adapted solution. It would also be possible to evaluate the complexity of the abstraction and to refine in consequence not only considering the number of polynomials in P_m and their supposed inherited states and transitions but rather on the actual qualitative states and the existing transitions between them. This would require complete execution of the further process before coming back to possibly refine or reduce the qualitative abstraction of the state space. The computation of the transitions would then have to be achieved for each execution, as every refinement of the qualitative state may affect every formerly computed qualitative transition. Yet, as the creation of the qualitative model is supposed to be computed offline without severe time constraints, this possibility is not to be rejected.

In contrast, another option to define a stopping criterion is to evaluate the quality of the obtained discretization. This is equivalent to an optimization exercise. It requires the definition of a utility function f_u to evaluate the quality of a given abstraction. Precision cannot be sufficient to constitute a criterion, as qualitative reasoning implies

inherently a loss of precision. Again, defining a utility function pertinent to every qualitative abstraction seems impossible, but some specific points could be considered to develop a generic optimization process. The main quality intended for a qualitative model is the ability to correctly reason on qualitative values to get satisfying results without needing numeric computation. The criterion to determine a sufficiently precise abstraction could be the ability to discriminate a given set of numerical values in the same number of different qualitative states. If two of the given values were to be abstracted in the same qualitative state, the abstraction would be considered insufficiently refined, and the computation of P_m would restart to generate more qualitative frontiers and, therefore, more qualitative states discretizing the state space.

The last challenge regarding the management of P_m and the discretization of \mathbf{X} for each mode once the stopping criteria are chosen is the priority order to attribute to each element of P_m . Should the same element be derived many times in a row due to its important role in the description of the system, or should there be no priority between the considered polynomials? As the flow equations and their derivative contain information about the system's dynamics, giving them higher priority than the derivatives of the transition condition would make sense. This decision is crucial to drawing the abstraction policy and must be made according to the desired qualitative information to integrate into the qualitative model.

For the rest of this article, the stopping criterion was chosen as simple as possible to allow the algorithm to terminate without risking influencing the results in an undesired way. We fixed for each of the polynomial equations of the system corresponding to the initial elements of P_m a maximum number of times to be derived during the process before stopping the P_m computation.

Finally, in order not to integrate the same polynomial p more than once in the same P_m , it is necessary to add a unicity test during this discretization phase.

In the hybrid Brusselator system, with a simple criterion of a maximum of 2 derivations, P_1 and P_2 are both initialized to $\{x, y\}$. Then $F[mode_1][x]$ and $F[mode_1][y]$ are added to P_1 . $I[mode_1]$ is also added as is the transition condition from $mode_1$ to $mode_2$. At that time, we have $P_1 = \{x, y, \dot{x}, \dot{y}, x - y\}$. Then, each element is derived: as \dot{x} and \dot{y} are already in P_1 , they are not added. However, $\frac{d(x-y)}{dt}$, \ddot{x} and \ddot{y} are added.

Once P_m is computed, the next step is to take every $p \in P_m$ and use a polynomial solver to solve the equation $p = 0$. The obtained solutions give the expression of the nullclines (i.e., a curve supported by an equation $\dot{v} = 0$ with v a function of time) of the dynamics and allow a discretization of the state space of the variables. Each value of X will now be abstracted by comparison to these nullclines.

Proposition 1. *Using the so-obtained discretization of the state space of each mode of a CPS, each valuation x of X can be abstracted in exactly one qualitative state.*

This proposition comes from the nature of discretization, which is a partitioning of the state space.

Definition 6 (Qualitative State). *We call **qualitative state** of a system the pair (m, qs) with $m \in \mathbf{Q}$ the current mode and $qs \in \{+, 0, -\}^{|P_m|}$ a vector such that*

$\forall i \in \llbracket 0, |P_m| - 1 \rrbracket$, $qs[i] = -$ if $P_m[i](X) < 0$, 0 if $P_m[i](X) = 0$, and $+$ otherwise. A qualitative state corresponds to a set of abstracted values of X expressed as a vector of elements from $\{+, -, 0\}$ representing respectively for every $p \in P_m$ the fact that $p(X)$ is negative, zero, or positive.

Definition 7 (Abstraction Functions). *Given a CPS S defined with the previous notations, we define the system abstraction function*

$$A : \begin{cases} \mathbf{Q} \rightarrow \mathcal{B}_{\mathbf{X}} \\ m \mapsto \mathbb{S}_{m, \mathbf{X}} \end{cases} \quad (1)$$

with $\mathbb{S}_{m, \mathbf{X}}$ the set of qualitative states defined on \mathbf{X} for the mode m and $\mathcal{B}_{\mathbf{X}}$ the set of all partitions of the space \mathbf{X} inspired from the notation of the Bell number $\mathcal{B}_{|\mathbf{X}|}$.

From this definition and the definition 6 of the qualitative state, we deduce the state abstraction function in the mode m noted

$$\alpha_m : \begin{cases} \mathbf{X} \rightarrow \mathbb{S}_{m, \mathbf{X}} \\ x \mapsto qs \in \{-, 0, +\}^{|P_m|} \end{cases} \quad (2)$$

which abstracts each value x of X as a vector of length $|P_m|$ representing the sign of each polynomial p of P_m for $X = x$.

A qualitative state represents an abstraction of a whole set of numerical values of X , therefore implying a loss of knowledge in the representation. For a qualitative state qs corresponding to the abstraction of two different numerical values x_1 and x_2 , coming back from the qualitative knowledge to numerical representation implies the implementation of a concretization function as defined by Tiwari in [16], which cannot give with complete certainty one value rather than the other. Instead, it will return a continuous set of values allowed for the variables. As the qualitative states are computed by partitioning the state space of S for each mode m , the mutual exclusion principle automatically applies to them by definition of a partition.

The advantage of this method is that it can be applied even on non-instantiated systems: it does not require prior knowledge about the context or the system's objective, meaning it is convenient to generalize.

Moreover, to apply this method, it is necessary to know the explicit formula of the ODE, which means that it does not automatically apply to systems defined by more abstract structures such as bond graphs developed by Rosenberg in [26], causality graphs, or even proportionality relations. Therefore, the method is limited to a subset of CPS where the relations and dynamics are perfectly known with symbolic formulas. We are currently working on generalizing this abstraction to systems defined by causality or proportionality. In the case of explicit equations with non-valuated constants, α can be defined but will not be reliable before a complete definition of all the symbolic constants as its return value depends on it.

3.2 Evolution Abstraction

Once the system's state space is discretized in qualitative states, the qualitative model consists of a finite partition of its state space in qualitative states for each

mode. Upgrading it and making it fit for complex applications such as prediction or diagnosis requires computing the possible evolution for each qualitative state, given the dynamics of the system.

Definition 8 (Continuous Evolution). *Let t be the time parameter of the system S . A continuous evolution of S is a function $\sigma : t \mapsto \mathbf{X}$ that maps every defined instant with a value of X .*

The stake of qualitatively studying a system relies not only on the ability to create and manipulate an abstraction of its state space. A major challenge of qualitative reasoning consists in applying this qualitative state abstraction to the system's dynamics to observe a concise view of its evolution. The continuous evolution of a system is often observed to verify specific properties and to visualize the behavior of a concrete or a well-designed system. However, when the design process is not over, numerical resolution and computation do not fit the available knowledge nor the qualitative model previously created.

To create a behavior representation fitting qualitative reasoning, the study of the dynamics must follow the mindset of qualitative modeling and create an abstraction of the continuous evolution σ . Therefore, we will consider a decomposition of the system evolution inspired by the works of Forbus [10] using events and tendencies.

Definition 9 (Events and Tendencies). *Let us consider a continuous evolution σ of the system S on the time interval T . We note m_t the operating mode of S at each instant t . We consider as events the values of t corresponding to valuations x of X associated to a change of qualitative state of S , i.e. a time instant t such that $S(t) = (m_t, x)$, $\exists p \in P_{m_t}$ such that $p(x) = 0$, and such that $\exists \epsilon > 0, \forall \theta \in]0, \epsilon[$, $S(t \pm \theta) = (m_2, x_2) \neq (m_t, x)$. On the opposite, we consider as tendency an interval $T_i \in T$ such that $\forall t \in T_i$, if $S(t) = (m_t, x_t)$, then $\forall p \in P_{m_t}, p(x) \neq 0$.*

For example, in a sine function $f : x \in \mathbb{R} \mapsto \sin(x)$, where the function and its first order derivative are studied, the events will be the points $\{k\frac{\pi}{2}\}_{k \in \mathbb{Q}}$ because the even values of k will imply a change of sign of f while the odd values of k correspond to the critical points and to a change of sign of $\frac{df}{dx}$. Consequently, the tendencies of f will correspond to the intervals $]k\frac{\pi}{2}, (k+1)\frac{\pi}{2}[$ where neither f nor $\frac{df}{dx}$ change their sign.

Events and tendencies offer support to the abstraction of continuous trajectories, as any continuous behavior can be represented using a variation table, whose extrema and thresholds correspond to events while the tendencies represent variation directions between the events.

Definition 10 (Qualitative Evolution). *Let t be the time parameter of the system S . A qualitative evolution of S is a function $\kappa : t \mapsto \mathbf{Q} * \mathbb{S}_{q,m}$ that maps every defined instant with a couple (m, s_q) with m the mode and s_q the qualitative state corresponding to the abstraction of the continuous value X of the state of the system at instant t . Said otherwise, a qualitative evolution of a system corresponds to its continuous evolution abstracted with tendencies and events.*

To abstract all the possible behaviors of a system, we must first fix and abstract its initial state to figure out which qualitative state to begin with. To this extent, it is necessary to define which mode m of Q is the initial one. We then apply the abstraction function α_m on the initial value X_0 of X . The result s_0 corresponds to the initial qualitative state of the system S . Without an initial numerical state, the initial qualitative state may be chosen arbitrarily in the authorized set of qualitative states.

From s_0 , the objective is now to apply the dynamics F of S on the state space abstraction from s_0 to propagate the qualitative states and explore all the possible qualitative behaviors κ of the system.

Then, while we are not in an absorbent state or an already explored state, we explore all the neighbors of the current state and add them to the list of qualitative states to be treated. Two structures are used to memorize the qualitative states: a *frontier* list and an *explored* list. *frontier* contains the qualitative states from which the analysis should progress, and *explored* memorizes the ones already studied.

3.2.1 Qualitative Transitions

Definition 11 (Qualitative Transition). *A transition $tr = (m_i, s_1) \rightarrow (m_j, s_2)$ is called qualitative or intra-modal iff $m_i = m_j$ and $s_1 \neq s_2$. Moreover, tr must be allowed by the theorems of continuity, such as the intermediate value theorem.*

While *frontier* is not empty, we consider s_i the next state in *frontier*. From this state s_i , the objective is to compute all the successor states according to the dynamics of the current mode m of S . The first step is to compute all the states sharing a border with s_i . This is possible by using a polynomial constraint solver such as $Z3$ and by translating each digit of the state-vector $v_s \in \{-, 0, +\}^{|P_m|}$ to a constraint corresponding to the associated sign and imposed to the related polynomial. Changing one digit of the current qualitative state and respecting the intermediate value theorem allows the creation of all the theoretical neighbors of s_i . However, many of the found neighbors do not exist or do not offer a transition from s_i .

Using constraint solving, we then find the possible neighbors s_j . Considering the hypothesis that every equation defining the qualitative states is polynomial, this resolution only consists of a conjunction of polynomial inequalities that can be solved using the chosen solver. Then, if s_j exists, it is placed in the structure *RealNeighbors*.

For s_k in *RealNeighbors*, we must verify whether or not a transition from s_i to s_k is possible in the behavior of the system. This is achievable by using the Lie derivative formula (see Tiwari in [16] and Yano in [27]):

$$L_X(p) = \sum_{X_i \in X} \frac{\partial p}{\partial X_i} \frac{\partial X_i}{\partial t}. \quad (3)$$

with $p \in P_m$ the polynomial function supporting the border between s_i and s_k and the X_i the components of X .

Proposition 2. *The transition from s_i to s_k is possible iff $L_X(p) * (p(s_i) - p(s_k)) > 0$ where p is the polynomial function supporting the border between s_i and s_k .*

Proof. see [16]. □

Using the Lie derivative, it is, therefore, possible to compute every qualitative transition in a mode by applying this process to every state $s_i \in \mathbb{S}_{m, \mathbf{X}}$, and to visualize the behavior tree of any execution in a continuous model or a single mode of a hybrid model. The different tests allow the suppression of the non-existing states and the computation of only the real qualitative transitions. Using the *existing* structure allows us to avoid the loops and suppress the quiescent, the already explored, and the terminal states. However, dealing with modal transitions of a hybrid system is more complex and requires more computation.

3.2.2 Discrete Transitions

Definition 12 (Modal Transition). *A transition $tr = (m_i, s_1) \rightarrow (m_j, s_2)$ is called discrete or modal if $m_i \neq m_j$.*

Computing the possible discrete transition in a qualitative hybrid model requires incorporating the guard conditions of each transition from the current mode in the abstraction process. To simplify the computation, we consider the guard condition necessary and sufficient to provoke the transition, i.e., a discrete transition happens as soon as its associated guard condition is verified. As the guard conditions are considered polynomials and are incorporated in $P_m \forall m \in \mathbf{Q}$, the verification of the guard predicates can be integrated into the qualitative transitions detection process. When a polynomial p associated with a guard condition changes its sign, the guard condition is verified, and the transition is triggered. For the involved transitions, the jump is not from a qualitative s_i to another s_j in the same mode, but between the qualitative states (m_1, s_i) and (m_2, s_k) with $i \neq j$. The intended transition is not qualitative anymore but becomes modal. Therefore, the change does not only concern the continuous variables of X but also Q . Moreover, a shift in mode implies activating an associated reset function *res* that applies on X and may also cause a discontinuity in its valuations.

Proposition 3. *Abstracting the discrete transition $tr = (m_i, s_1) \rightarrow (m_j, s_2)$ requires to compute $\alpha_2(s_1)$, i.e. $\alpha_2(x)$ for all $x \in s_1$.*

The computation of $\alpha_2(s_1)$ can be achieved by creating another iteration of a polynomial solver: by using branch-and-bound solving on all the constraints defining the qualitative states on m_2 , computing all the possible destination states for a reset function applied on (m_1, s_i) is possible. The list of obtained authorized states $[(m_2, s_j)]_{s_j \in \mathbb{S}_{m_2, \mathbf{X}}}$ will constitute the successors of (m_1, s_i) .

Doing this for all the triggering transitions in a mode m_1 highlights all m_1 outgoing transitions. Once computed for every mode $m \in \mathbf{Q}$, all the system's outgoing and incoming modal transitions will be represented.

3.3 Qualitative Behavior Computation

By neighborhood propagation, we can compute all the paths among the defined qualitative states of the qualitative model considering its dynamics. Adjacent state propagation allows us to determine all possible variation directions from an initial

state. The obtained set of qualitative traces contains all the theoretically possible behaviors of the system S .

Computing complete qualitative behaviors makes it possible to represent it as a qualitative automaton.

Definition 13 (Qualitative automata). *We consider the structure of a qualitative automaton $A_q = \langle Q, X, V, E, Init, I, F, J, L, A, T_Q \rangle$, where $Q, X, V, E, Init, I, F, J, L$ are the elements defined in [Definition 3](#) and*

- A is the system abstraction function mapping each mode m of Q to a partition of X where each set corresponds to a qualitative state of the system.
- T_Q is a function mapping each mode to the set of qualitative transitions computed in [subsection 3.2](#).

This representation, introduced in [\[16\]](#) as abstract transition systems, proposes an alternative visualization of the behavior of a hybrid system, giving more information about the different qualitative trajectories in the modes. It can provide, for example, more knowledge about possible attractive states, intra-modal cycles, or even two qualitatively different trajectories that could not be separated using a classic hybrid automaton but which could imply various constraints for the system. This new layer of information completes our qualitative model with a more precise mapping of the qualitative states and gives more possibilities of anticipation, diagnosis, and piloting [\[18, 28\]](#).

3.4 Problem of Dimensions

The presented process is simple to execute when $|X| = 2$ because polynomial solvers easily handle polynomial problems with two variables. One of the components of X can trivially be expressed depending on the other for each polynomial function with at least one reference variable.

However, the presence of more dimensions raises problems that do not appear in two dimensions. The more $|X|$ increases, the more difficult it will become for the solver to return a usable solution. Especially symbolic solvers in high dimensions tend to find a unique solution that satisfies the constraints and will, therefore, avoid general solutions of the expected form. To ensure that the returned solution has the desired form, we must specify the anticipated solution format and the priority order of the variables for the resolution to the solver. The presence of solutions of dimension less than $|X| - 1$ will create frontiers that may be circumvented without being crossed, which is a major problem in computing qualitative transitions. Therefore, we added a filter to the solver that suppressed the solutions of low dimensions. Secondly, the choice of the component of X that should be considered as a reference to express the others is a question. Should X_0 always be regarded as the reference variable, or should there be a smarter decision criterion? Theoretically, the best solution would be to avoid the presence of fractions in the expressions of the solutions as much as possible. As fractions may cause singularity in the presence of a fraction pole in the state space, the best expression of a solution would be the one that minimizes the number of fractions. However, we made a concession due to the complexity of

submitting such criteria to already implemented polynomial solvers. We just created a reference priority order favoring the lower factors of X to the upper. Finally, an inherent problem in this situation is that the used solver may not find any solution in high dimensions associated with high polynomial degrees. Even if polynomials of very high degrees are not commonly used in CPS, the situation may happen for specific systems and possibly in critical situations. This shows that there exists a possibility of generalizing this contribution even more, using elements from other works, such as qualitative tendencies.

4 Introduction of Qualitative Zones

The model obtained by the process presented in [section 3](#) allows us to reason about the qualitative states and behaviors. This representation constitutes the heart of what is currently qualitative modeling. The major qualitative abstraction methods in the literature are based on discretizations of this kind. However, this representation of systems lacks knowledge about orders of magnitude and distance/time to an event that could allow the management of critical situations of the system. This means that qualitative models sacrifice all their knowledge about the magnitude of the variables apart from their comparison to landmarks as in [\[12, 24\]](#). As landmarks values depend on the use and the design of the system, this choice is not an option when the abstraction takes place at the design stage of CPSs. This section introduces the concept of qualitative zone, a contribution that we present to improve the reasoning capabilities on qualitative models by introducing systematic order of magnitude reasoning on system abstraction.

The exposed qualitative models represent the equivalent for the system state space of a topographic chart of the summits of a mountain chain with the altitude gradient between them. Yet, exploring such an important chain cannot be achievable without information about the gradient magnitude and the possible presence of cliffs. In a CPS, this is illustrated by the difficulty of making predictions or applying simulation on a model with no further information about the intended trajectories than the sign or the gradient. For an autonomous car, an acceleration of 1 m.s^{-2} and another directed in the same direction of 100 m.s^{-2} cannot be considered the same way in simulation or worse, in real-time execution. Moreover, the same car will not behave the same way on an empty country road or a crowded freeway with other vehicles at less than three meters. Therefore, the two corresponding challenges are introducing elements from order of magnitude reasoning (see Mavrovouniotis in [\[29\]](#) and Raiman in [\[30\]](#)) in our qualitative models and adding thresholds surrounding and preventing the important events. As qualitative reasoning imposes constraints on the considered models, these upgrades to the representation must imply a process of refinement of the state space partition that matches the described ambitions. This means adding new borders to create more qualitative states, therefore adding new elements in the sets P_m . The inclusion of new polynomials in the P_m must be made with specific consideration to the interest and the significance of the added elements: adding more polynomials that correspond to no physical phenomenon does not have great interest. For example, creating qualitative states defined by the sign of the 5^{th} order derivatives will rarely

bring useful knowledge to the model. Yet, when all the equations describing the system and the most significant variable valuations have already been used to construct the previous model, the newly introduced equations have a very small chance of being of any significance to describe the system's behavior by themselves. Therefore, an option is to define new border equations from the previously defined ones to have reasoning values associated with them. Modifying the elements p_i of P_m makes it possible to create two thresholds of borders surrounding the frontier defined by $p_i = 0$. These thresholds will then determine a neighborhood around the event frontier and describe a form of proximity before the possible occurrence of the related event. The consideration of these neighborhoods, or proximity areas, will introduce information about the distance to the potential future events of the system behavior without clearly implying numerical evaluation. They can also have the role of precaution frontier, defining either a particularly far or critically close proximity to the event. Now, let us see how to define such neighborhood limit (that we also call secondary frontiers, by opposition to the primary/main ones, which are supported by the elements of P_m). We mainly evoked and studied three main ideas. The first was to translate the main equations of P_m from a chosen distance d to obtain the neighborhood limit. The advantage of this idea is its simplicity of execution: each border from any dimension can be translated from a specific value in a given direction without requiring heavy computation. However, the choice of the translation direction is entirely arbitrary, and such a frontier may not have any physical meaning, especially in the case of borders defined by nullclines (here, the zeros of the derivatives of the system's variables and equations). This is, however, the best solution for the events defined by $X_i = k$, with X_i a component of X and k_i a constant in \mathbb{K} defining a value for X_i : the translation direction is then naturally following the axis defined by X_i . The obtained hyperplane defined by the equation $X_i = k \pm d$ is parallel to the main border and represents a proximity area around it at a constant distance d . This solution is not possible in the case of nullclines supported by $\dot{X}_j = 0$.

Another idea was to compute surfaces completely parallel to the border: this would have assured a constant distance for every direction and would have been visually understandable. However, the computation of such surfaces is much trickier than expected: even in two dimensions, computing the equations defining a curve parallel to another is far from trivial. In two dimensions, in the case of a parametric curve defined at any instant t by $x = f(t)$ and $y = g(t)$ with f and g two functions, the parallel curves to the parametric curve (x, y) are the parametric curves which equations are (x', y') with $x' = f(t) \pm \frac{c\dot{g}(t)}{\sqrt{\dot{f}(t)^2 + \dot{g}(t)^2}}$, and $y' = g(t) \pm \frac{c\dot{f}(t)}{\sqrt{\dot{f}(t)^2 + \dot{g}(t)^2}}$, with c a constant. This choice would make little sense as simplification and limited computation are the main objectives of qualitative modeling. Moreover, such a limit would not have any physical meaning. Consequently, this idea of parallel surfaces does not fit the ambitions of qualitative reasoning to compute a neighborhood limit for polynomial frontiers. Finally, the best of the raised propositions was to use isoclines (i.e., surfaces defined with a function f by $\dot{f} = c \neq 0$). For each element p of P_m , we must define a value $c_p \in \mathbb{K}$, then use the already defined solver to solve $p = c_p$ and $p = -c_p$. The symbolic results will define the surfaces that will delimit the neighborhood of each main border. Solving these equations does not require specifically heavy computation,

and it corresponds to something concrete: a very low proximity value will be associated with a critical proximity to the event and a very little time interval remaining before its occurrence. On the contrary, a very high value defining the neighborhood will inform that any state out of this area will be unstable and maybe untrustworthy as the associated variable or quantity will vary very fast.

We chose to separate these borders from the previous ones. As we already have our discretization of the state space in qualitative states, we now have another discretization in areas defined depending on these latter, and that we called **qualitative zones** (see [Definition 14](#)).

As the qualitative zones correspond to a subset of the state space defined by the proximity to a frontier, it is also possible to expand this notion to represent the neighborhood of a point. When representing the proximity of a chosen point x^p valuation of X , the proximity frontier of distance d is represented by a n -dimensional sphere of radius d of equation $\sum_{i=1}^n (X_i - x_i^p)^2 - d^2 = 0$. The corresponding qualitative zone is, therefore, the associated ball of the same dimension and of equation $\sum_{i=1}^n (X_i - x_i^p)^2 - d^2 < 0$.

Definition 14 (Qualitative Zone). *A qualitative zone is a set of abstracted values in a neighborhood of a qualitative frontier or of a point. A qualitative zone is defined by a set of 2-uplets p_i, d_i with $p_i \in P_m$ corresponding to a qualitative frontier or the coordinates of a point and $d_i \in \mathbb{K}$ to the chosen neighborhood distance.*

It is to note that some subsets of \mathbf{X} may be included in many qualitative zones at a time: the creation of a function able to compute the intersection between qualitative zones is helpful to keep as much knowledge as possible about the proximity to the different events.

Suppose the qualitative states give abstracted knowledge about the distance from the current numerical state to each frontier. In that case, the qualitative zones offer an abstraction of the distance separating it from the nearest borders. This allows us to define a notion of distance in the qualitative models that may be used to evaluate the system's stability, the likelihood of a forthcoming event, or even the criticality of a situation. It gives a new tool to anticipate and reason about a system state, its risks, and possible futures.

The superposition of the two knowledge layers given respectively by qualitative states and qualitative zones creates a more complete abstract representation of the state space of a system. We call a position of the system in this representation a **qualitative position** (see [Definition 15](#)).

Definition 15 (Qualitative Position). *We call qualitative position the triplet $(mode, qualitative_state, qualitative_zone)$ of a system.*

This complementary information creates a complete qualitative state space map designed to locate a numerical state and reason about its successors.

Finally, considering qualitative zones is a good criterion for taking apart two qualitative behaviors that could not have been distinguished by reasoning only on qualitative states. For example, in the Brusselator system defined in (4)

$$\begin{cases} \dot{x} = 1 - (b + 1)x + ax^2y \\ \dot{y} = bx - ax^2y \end{cases} \quad (4)$$

with a and b two positive constants, the analysis of qualitative states and transitions shows that the system is cyclic around the convergence point. However, this knowledge is insufficient to deduce whether the trajectory will be convergent: the two qualitatively different behaviors (convergence and cycling around the convergence point) follow exactly the same trajectory among the qualitative states described earlier.

The consideration of a qualitative zone around the nullclines $\dot{x} = 0$ and $\dot{y} = 0$ and around the stable point allows us to determine, using the directional differentiation of the Lie derivative, if the system will converge towards $\dot{x} = \dot{y} = 0$, or stay in a critical cycle around it. With a symbolic expression of the distance d , the study of the surface defined by $\|a - X\| = d$ or by $|\dot{x}| < d \wedge |\dot{y}| < d$ will show if the convergence is possible or not. If for any value of d , the inward transition is possible, and if from a threshold $d_s > 0$, the outward transition is impossible for $d < d_s$, the system will be considered as convergent. Otherwise, it can be proved that it will follow a limit cycle.

However, just like qualitative states, qualitative zones require the instantiation of parameters such as the chosen proximity d to be computed. As the automation of the choice of the number and the size of the qualitative zones has not been achieved, we did not integrate the creation of the qualitative zones in the abstraction process: we instead created a functionality to add qualitative zones on a qualitative model around the desired qualitative frontiers with a given size. Functions to test the intersection between a qualitative state and a qualitative zone and to compute the trajectory direction on the border of a qualitative zone are also available.

5 Computation and Tool Creation

To build our tool, we considered two categories of systems corresponding to different situations regarding the knowledge we have access to. These two categories will be represented using the possibilities of object-oriented programming with two different classes. They will correspond respectively to instantiated and non-instantiated systems. As a primary hypothesis, we suppose that in the two categories of systems, the precise dynamic equations will be available, as will the invariant constraints and transition guard conditions. For a system with more available knowledge (especially about essential values of the variables, the objectives or context of the system, or the effects of the environment), we favor the second category, supporting the instantiated systems. Here, a new data structure is added to the model, representing all the landmarks that will be considered to study the system and its behavior.

These two categories are distinguished using two classes of objects, the first named *System*, and the second one, inheriting from the first, named *Instantiated-System*. We use object-oriented programming to impose a standard structure for the treated systems. We want to unify methods from polynomial solving, symbolic computation, constraint solving, and graph logic, so we developed our program in *Python* to benefit

from its numerous libraries. To be able to solve all the equations symbolically, we use the *Sympy* library. For each mode taken separately, the discretization of the state space is conducted as presented in [section 3](#).

The discretization of the state space is done by placing all the polynomial equations that define the system in the sets P_m and using the functions of symbolic solving provided by *Sympy* to resolve the equalities $p_i = 0$. *Sympy* also comes with tools to differentiate the polynomials according to time, to test if the newly obtained formula is a factor of existing polynomials using its polynomial module, and to create the qualitative state based on the comparison of a value to every polynomial of the set. The results of the resolution of the borders are given as n -uplets with $n = |X|$, in the form $(f_0(X_i), \dots, X_i, \dots, f_{n-1}(X_i))$, that express the equations corresponding to the borders.

Then comes the abstraction function: given the entered numerical value x of X and the current mode m , for each $p \in P_m$, we compute the sign of $p(x)$ to determine on which side of each of the qualitative border the current state is. The qualitative state is then represented by an array of -1 , 0 , and 1 , corresponding respectively to $-$, 0 , and $+$ but making operations easier to compute.

In our example of the Brusselator, let us suppose that the current mode is $mode_1$, that the numerical value of the system is $(1.5, 4.3)$, and that the parameters have value $(a_1, b_1) = (1.2, 1.4)$. If $P_1 = [x, y, x - y, 1 - (b_1 + 1)x + a_1x^2y, b_1x - a_1x^2y]$ as we computed before, the qualitative state of the system is $[+, +, -, +, -]$. Then, the algorithm of qualitative state propagation explores the system's behavior and all the system's possible trajectories from an initial state, using mathematical properties such as the intermediate value theorem, inequality solving, or Lie derivatives.

To propagate the current state and to compute the behavior tree of the system, the algorithm first looks at all the qualitative states in contact with each found state. This is done by taking the array of a state as a reference and by creating copies of it where one digit is changed to a possible adjacent state (in the direction of the intermediate value theorem), meaning that a 1 or a -1 can change to 0 and that a 0 can be converted either to a 1 or a -1 . In the case of a state $s_1 = [1, 1, -1, 0, -1]$, its computed possible successors will be $[0, 1, -1, 0, -1]$, $[1, 0, -1, 0, -1]$, $[1, 1, 0, 0, -1]$, $[1, 1, -1, 1, -1]$, $[1, 1, -1, -1, -1]$ and $[1, 1, -1, 0, 0]$.

Then, the tool must check whether each of these possible successors exists or not. It uses a constraint solver and checks if the conjunction of all the inequalities defining the states can be *True* at a point of the space. If it can, the state exists and is kept for the next steps. Otherwise, the state does not exist, and we forget it. Moreover, if the new state corresponds to an invariant violation, we keep it in a special category: we will still study the feasibility of the transition but keep it as an invalid one. The difficulty arising in this step is that *Sympy* does not allow constraint solving for inequalities: only equality solving is allowed when it comes to a conjunction of equations. Therefore, we used a second *Python* library allowing symbolic computation specifically for this task, which is *Z3*. To this extent, we added a translation program to the algorithm, converting *Sympy* expressions and variables to *Z3* format. The *Z3* Python library uses the *Z3* SMT solver to solve a conjunction of symbolic inequalities on a specified set of variables and returns *sat* if the created problem admits a solution. Therefore, to check if a state exists, we convert all the equations and inequalities that characterize it to

the $Z3$ format, create a problem based on the conjunction of all these inequalities, and call the $Z3$ SMT solver to know whether a solution exists.

Once all the neighbors of a qualitative state s_1 have been isolated, the program must determine which are successors and which are predecessors. As explained, it achieves this by computing the Lie derivative of the equation corresponding to the border. It executes it with a scalar product between $[\frac{\partial p}{\partial X_i}$ for $X_i \in X$] and $[\frac{\partial X_i}{\partial t}$ for $X_i \in X$], with $p = 0$ defining the border. Ideally, this should be evaluated on a numerical value x of X located in the current state to determine this scalar's sign. However, regarding the difficulty of finding a satisfying concretization function expression (which is still one of the biggest challenges of qualitative reasoning), there is not yet a possibility of getting numerical values corresponding to a qualitative state. $Z3$ offers a way to bypass this difficulty. It can create a virtual state of size $|P_m| + 1$, with the $|P_m|$ first elements being the elements of the initial state, to which we associate a digit whose value depends on the sign variation implies by the considered transition. This new element is linked to the equation obtained with the Lie derivative formula. Then, we call the constraint solver again to determine if this virtual qualitative state exists. If the associated problem is satisfiable, then the Lie derivative on the border allows the transition, and the state s_2 considered as a possible s_1 successor is an actual successor of s_1 . We add s_2 to the list of successors of s_1 and s_1 to the list of predecessors of s_2 . Otherwise, the transition from s_1 to s_2 is impossible, so we do not add s_2 as a real successor of s_1 . If s_2 happens to violate invariant conditions, it is not added at all in the behavior tree, but a state named *Invariant Violation* is added as a successor of s_1 .

With all the intra-modal transitions computed for each mode, the algorithm now browses the states corresponding to the validation of a jump condition. As the natures of the equations defining the qualitative states are kept, one can refer to them and observe which actual successors of s_1 change the value of the associated digit. When one transition to such a state is found, a modal transition to the corresponding target mode replaces the previously computed qualitative transition. The reset function calculates the qualitative state of the new mode in which the discrete transition arrives.

In the end, the algorithm returns an automaton expressed as a dictionary, where the identifier of each mode and qualitative state are used as keys, giving each qualitative state its predecessors and successors. The qualitative state corresponding to the initial value of the system contains the string *start* in its predecessors' list.

Applied to the hybrid Brusselator and translated into a graphic version, the algorithm gives the result shown in [Figure 3](#) as a qualitative automaton drawn from the automaton obtained as an output of the algorithm.

In this figure, the blue vertical arrows correspond to intra-modal (or qualitative) transitions, while the red (oblique) ones are the modal (or discrete) transitions. The thin red arrows represent the one-way transitions, and the thick ones represent transitions that can happen in both directions. Qualitative transitions are all unidirectional, so this distinction does not apply to them. No label was added to qualitative transitions, while discrete transitions are labeled with their respective direction. The chosen parameters values in this model were $(a_1, a_2, b_1, b_2) = (1.2, 3.6, 1.4, 2.5)$. The labels of the modes describe the sign of the elements of P_m and are ordered in the same order.

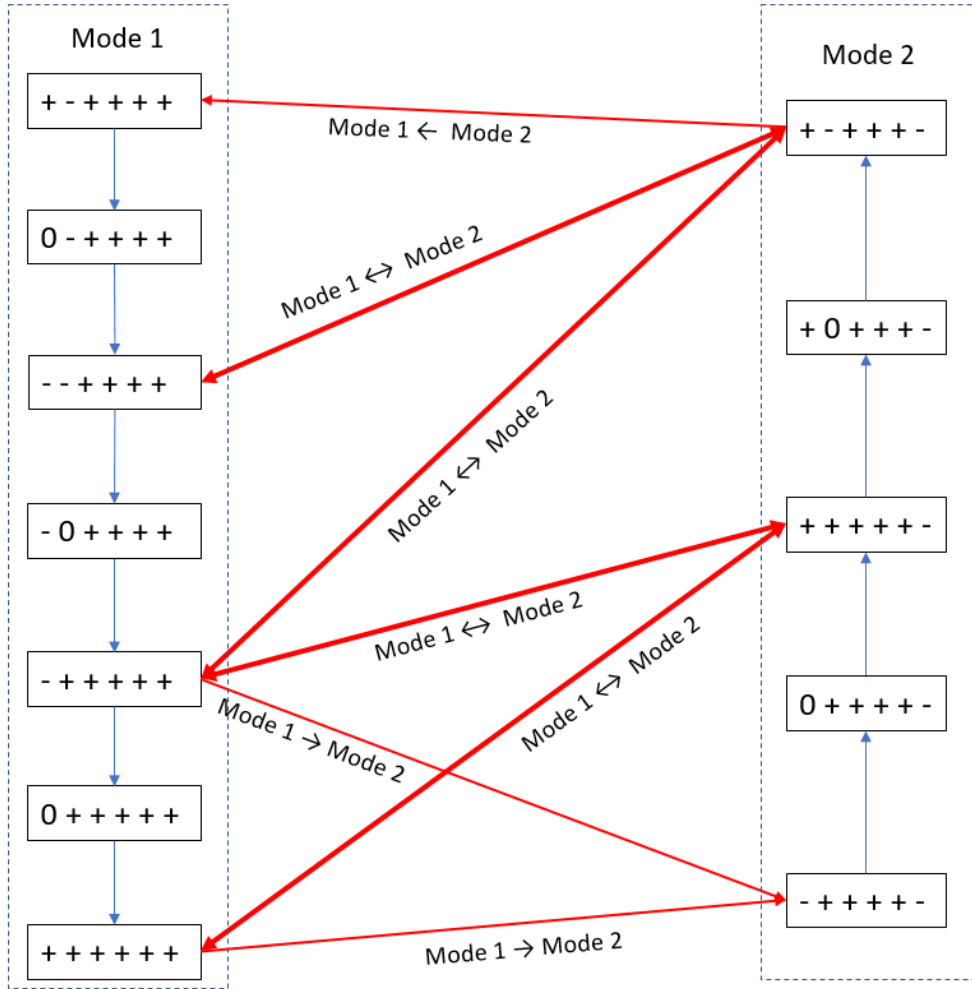


Fig. 3 Complete qualitative automaton of the hybrid Brusselator system

Having chosen a simple stopping criterion for the discretization of the state space of at most one derivation for any polynomial of P_m , the computed sets P_m are respectively equal to $\{1 - (b_m + 1)x + a_m x^2 y, b_m x - a_m x^2 y, x, y, x - y\}$, containing equations corresponding respectively to the two flow equations, the two invariant constraints and the transition condition. In each state, the set of elements from $\{-, 0, +\}$ characterizes the sign of the corresponding elements from P_m . Many qualitative cycles coexist, all of them transiting by the two modes. There is no behavior staying in only one of them. The existence of many different cycles shows the interest of the state space discretization, as it separates trajectories that would have been considered identical in a classic hybrid automaton. We now have the complete qualitative automaton of the

CPS. Both the discrete transitions and intra-modal behavior are included, and the trajectories are more visible than in traditional hybrid automata.

Once the qualitative automaton is obtained, the following step is to compute the qualitative zones around the qualitative frontiers and their different interactions with qualitative states. The distances d_i between every qualitative border and its associated secondary frontier must be entered as an argument: we still need to automatize the choice of the value characterizing the optimal distance. Once we have the equations of the different hyperplanes and isoclines, we aim to determine how they fit into the qualitative model. This involves a new call of the constraint solver: the intersection between qualitative states and zones can be achieved by testing the satisfiability of the conjunction of the constraints associated with the intersection of a state with a zone. If the problem has a solution, then the considered qualitative zone exists in the qualitative state. Knowing which zone exists in each qualitative state and computing the transition direction once more creates a qualitative map that gives more knowledge about the trajectories of the system, with a good balance between qualitative and numerical information. Here, we gave an example of the result we can obtain on a Van der Pol oscillator: this system is a continuous system defined by [Equation 5](#), where c is a constant.

$$\begin{cases} \dot{x} = 10(y + x - \frac{x^3}{3}) \\ \dot{y} = c - x - \frac{3y}{4} \end{cases} \quad (5)$$

This system's few equations and borders make it an excellent example, as the qualitative map is still understandable and not overloaded. In [Figure 4](#), we drew the main borders with plain lines and the associated secondary borders with dotted lines. This representation corresponds to an instance of the system where c is arbitrarily set to 0.465. The color code associates each border with its secondary limits. For both the straight lines (hyperplanes in two dimensions), we arbitrarily chose a distance of 0.2 to place their proximity limits. We computed the isoclines with $\dot{x} = \pm d_1$ and $\dot{y} = \pm d_2$ where $d_1 = 10$ and $d_2 = 1$. Finally, the black arrows show the transition direction allowed according to the computation of the Lie derivatives. Double arrows highlight that both transition directions are possible at a point of the border.

6 Application to Piloting and Supervision

The process of qualitative model creation can now be automatized to generate a hybrid automaton representing the system's behavior. Qualitative models are mainly used for theoretical and upstream tasks such as verification, proof, and diagnosis. The contribution of qualitative zones to the qualitative model gives a glimpse of possible concrete applications of qualitative models on concrete CPS. Using the abstraction algorithm and the Lie derivative formula, the qualitative model already shows all the possible trajectories and, therefore, perfectly delimits the normal operation of the system. Real-time verification is already possible with such modeling abilities. In addition, the knowledge of the distance to events inherent to the notion of qualitative zone creates the possibility of anticipating and monitoring the execution of a system during its execution or adapting the simulation policy of a simulator depending on the observed

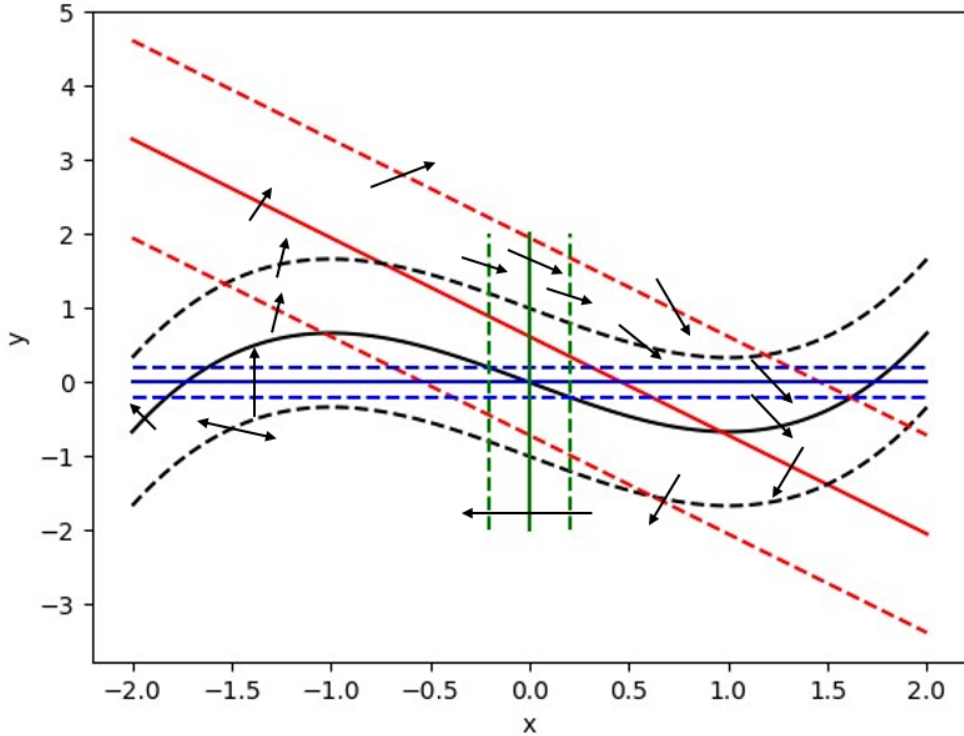


Fig. 4 Qualitative map of a Van der Pol oscillator system

results. Such possibilities offer the ability to drive and pilot a simulation of the system by modifying both the integration step dt and the quantization precision when entering a zone defining a critical neighborhood of an event or, on the contrary, when one of the derivatives of the variables is found very far from its nullcline, meaning that the corresponding variable may vary very fast and cause unexpected events. The controller is then more careful about the appearance of mistakes or unplanned behavior in these situations. The adaptation to the proximity of an event can be compared to the adaptation of someone walking near a cliff on a mountain path who will walk more slowly and more carefully when close to the cliff in order to avoid any bad move. The reduction of the integration step in a simulation can prevent the appearance of false positives (i.e., computed transitions that do not exist in the system) and detect with precision the transition timing without requiring rollback as evoked by Bouissou in [15]. The modification of the integration step depending on the qualitative position is inspired by quantized states simulation, developed by Kofman [31], Cellier [32] or Floros [33]. It creates a compromise between qualitative and numerical reasoning, joining what Kuipers considered as semi-qualitative reasoning. The adaptation to the variation amplitude is processed by making the formula $\max_i(\lceil \log_k(|\dot{X}_i dt|) \rceil)$ to be bounded, with dt the integration step and k a chosen numeration base. A higher value of k will cause a more important reduction of dt to the same acceleration and assure a reduced risk but will increase the simulation time. The limit value imposed to $\max_i(\lceil \log_k(|\dot{X}_i dt|) \rceil)$ will depend on the researched precision of the model, which

is not something automatically quantifiable. Therefore, the intervention of a human agent is required. As a result, we can upgrade the precision of a numerical simulation compared to a constant-step execution and get results with initial time steps that would not have allowed a classic simulation to work. The detection of the precise instant of the transitions is also more accurate because adapting the simulation step near the neighborhood limits implies that any post-transition point will be in immediate proximity to the transition border. Consequently, the deviations caused by late detection of the transition threshold decrease. We also noted a reduction of false transitions caused by significant inertia on the variation of the variables when some derivatives have high values. As the simulation controller automatically reduces the time step in these situations, the inertia is canceled due to the high number of computation points. It does not cause any false transition that could completely deviate the simulated trajectory from the actual behavior. Moreover, the execution time of an adaptive simulation is far lower than the one of a precise simulation with constant and small time steps. The simulator saves time when the system is not in any proximity zone and has no high derivative among its variables. It keeps it for areas of the state space where we seek precision and reliability. Therefore, simulation piloting using qualitative reasoning offers an interesting trade between time complexity and result precision.

In the case of CPS monitoring, it is also possible to change the frequency and the quality of the sensor sampling to optimize the use of resources and keep results of good quality. Modifying the quantization precision during a simulation allows a pilot to adapt its precision and use better-quantized values in more critical zones. However, adding precision back in the values of the system after having decided to discard it to improve the computation time is still a problem: earning knowledge to get a finer quantification is not possible without sensors or at the risk of choosing false values. Now that we can nearly automatize the creation of such a qualitative pilot, it is almost possible to generalize it to any kind of CPS expressible with ODE dynamics. We still need to define some parameters, such as a reference time step that strongly depends on the working context of the system and that cannot be determined by a general rule for any CPS. We will also have to fully automatize the definition of such a pilot using the qualitative analysis presented above. More details on both these applications are presented in [28, 34].

7 Generalization

As this article follows and completes works focusing on polynomial systems, the proposed abstraction methods were designed mainly for systems with polynomial constraints and dynamic equations. This is due to the simplicity of solving polynomial equations to a certain degree, while more general functions do not offer such convenience for resolution. As polynomials are almost the simplest mathematical functions to manipulate, the framework of the polynomial system is very advantageous but still quite limited. It is, however, possible to generalize this work to other categories of functions, such as rational functions.

7.1 Rational Functions

Considering that every rational function can be written in the form $\frac{p_1}{p_2}$ with p_1 and p_2 polynomials, the set of all rational functions on the variable set X and written $\mathbb{K}(X)$ can, by definition, be expressed as a product of set $\mathbb{K}[X] * (\mathbb{K}[X]^*)$. Therefore, computing the zeros of a rational function is equivalent to computing the zeros of its numerator polynomial p_1 and verifying that they do not match with the zeros of its denominator p_2 . This implies that *Sympy* and *Z3* solvers, primarily thought for polynomial functions, can perfectly deal with rational equations.

Moreover, as it is easy to prove that $\mathbb{K}(X)$ is closed under the derivation just like $\mathbb{K}[X]$, manipulating rational functions for state space abstraction does not change the situation for the computation. It makes the resolution more complex and the execution longer. However, as computing the qualitative map is an offline operation, the computation time does not constitute an important criterion. The qualitative model is supposed to be generated before the execution of the processes (be it a simulation or the monitoring of the system) that use it.

The main problem will be the apparition of a singularity area near the poles of the fractions, which will cause the qualitative borders to have unpredictable shapes.

In spite of this inherent complexity, the generalization of our process to rational fractions does not pose major problems.

7.2 Qualitative Tendencies and Hybridization

As exposed in [subsection 3.4](#), complex polynomials can become a problem if the capacity of the solver is exceeded. This can happen when X is composed of many variables and if the treated polynomials are of high degree. To keep these case studies in the range of qualitative reasoning, we worked on a function abstraction process [\[35\]](#) based on order of magnitude [\[29\]](#) and order of growth reasoning. After having defined a reference value $v_{i,m}$ for every component $X_i \in X$ and every mode $m \in \mathbf{Q}$ expressing the anticipated order of magnitude of the variable X_i in the mode m and set of negligibility criteria $k_{i,m}$, the process applies a hybridization method as explained by Maler or Asarin in [\[20, 36\]](#) to abstract each function considered as too complex in a piece-wise continuous simplified function, keeping for each sub-division of a mode state space the terms considered as dominants and suppressing the terms whose impact on the dynamics are considered as locally negligible. Hybridization is often used on continuous complex systems to simplify them by transforming them into a hybrid system that reproduces their properties and behavior. The idea of hybridization consists in approximating a too-complex continuous vector field by a hybrid system with a collection of simpler (constant, affine, or even polynomial) vector fields [\[36\]](#). In the case of already hybrid systems, it would lead to splitting each mode of the system into several modes with simpler equations. The comparison between the different terms of an equation is possible using order of magnitude reasoning (OMR). More precisely, as OMR can be either absolute or relative, as explained by Raiman in [\[30\]](#), we especially use relative OMR to perform a comparison between the relative magnitude of an equation. Absolute OMR compares different variables to a set of fixed reference values, while relative OMR compares variables with each other as long as they are comparable. Different

comparison systems exist, such as *FOG* [30], *O(M)* [29], or *Rom* [37]. Each of them uses its comparison operators, but the operators *neg* (Negligible) and *Vo* (Neighbor) seem to make a consensus. We also chose to use *Co* (Comparable) from *Rom*.

Previous works also proposed a generalization of OMR to functions and dynamics by introducing the notion of order of growth (see Borel [38], Hardy [39] or Dague [37]). Orders of growth (OG) initially aimed to characterize the behavior of functions to ∞ and work with equivalent classes of applications. The order of growth of an application f is computed as: $c(f) = \lim_{t \rightarrow \infty} \frac{\ln|f(t)|}{\ln|t|}$ and $c(f = 0) = -\infty$. This formula allows the polynomial functions to have an order of growth equal to their degree. Abstracting functions to simplify qualitative model computation requires both reasoning on the magnitude of functions at a precise instant t and their behavior to infinity at the limits of the state space. Relative OMR and OG reasoning give a good basis for abstracting the functions defining the CPS. In the case of high-degree polynomials, equivalent classes impose the monomials of different OG to have different behaviors to infinity. Consequently, monomial terms of different degrees applying on the same variable of X or parameter can be simplified if the considered variable/parameter reaches sufficiently high values. The chosen negligibility criterion associated with the variable gives the threshold authorizing this abstraction.

The protocol for applying relative OMR and OG for the polynomial abstraction of system equations is as follows.

After separating each of the target equations into monomial terms, the algorithm will discretize the state space based on the relative prevalence of each of the monomial terms on the others to create a piecewise continuous function constituted of sub-functions simpler than the original one. Each sub-function will correspond to a new mode of the qualitative model, meaning that hybridization is applied to each mode separately. This increases the number of operating modes of the qualitative model and the computation capacity for each.

Definition 16 (Qualitative tendencies). *If $f : \mathbb{K}^n \rightarrow \mathbb{K}$ is a function on \mathbb{K} , let f_r be the piecewise continuous function abstracting f_C obtained with the previously presented instructions for negligibility criteria C_{X_i} for each variable X_i . We call the qualitative tendencies of f each continuous function f_i defined on a rectangle of \mathbb{K}^n composing a continuous section of f .*

Qualitative tendencies are thus defined as the section of the abstracted formulation of the equations of the system. It corresponds to a subspace of the state space where the respective orders of magnitude of the variables and where the local magnitude or growth of the equation implies a specific influence relationship between its different terms. In the case of equations involving different variables, the homogeneous terms are compared to each other as the presence of an addition supposes a physical homogeneity in the added terms. As example, consider a function $f : x \mapsto x^3 - 2x^2 - 2$ defined on \mathbb{R} . If the negligibility criterion is chosen to be of 0.1 (meaning that x *Ne* y if $\frac{|x|}{|y|} < 0.1$),

then the obtained abstracted piecewise continuous function can be expressed as

$$f_p = \begin{cases} -2 & \text{if } |x| \leq \frac{1}{\sqrt{10}} & x^3 \text{ and } x^2 Ne 2 \\ -2x^2 - 2 & \text{if } \frac{1}{\sqrt{10}} \leq |x| \leq \frac{1}{\sqrt[3]{10}} & x^3 Ne 2 \text{ and } x^2 \\ x^3 - 2x^2 - 2 & \text{if } \frac{1}{\sqrt[3]{10}} \leq |x| \leq \sqrt{10} & \\ x^3 - 2x^2 & \text{if } \sqrt{10} \leq |x| \leq 10 & 2 Ne x^3 \text{ and } x^2 \\ x^3 & \text{if } |x| > 10 & x^2 \text{ and } 2 Ne x^3 \end{cases}$$

It is also possible to slightly change this expression to make the abstracted function continuous on \mathbb{R} by evaluating the suppressed terms on the discontinuity points when the continuity of the abstraction is a constraint. This modification requires a few more computations but does not radically change the process or the result.

This way, it becomes possible to create another qualitative model of the same system, which will contain less precise and more artificial frontiers, but it will guarantee the possibility of processing and refining the model better than with high-degree polynomial equations.

7.3 Transcendental Functions

In the case of non-rational transcendental functions such as exponential, logarithms, or trigonometric functions, the ability of polynomial solvers to symbolically find the zeros of the equations does not hold anymore. The previous propositions and processes do not apply to systems defined at least partially by such equations. Therefore, abstracting the expressions before constructing a qualitative model is a necessity. Depending on the nature of these functions, the abstraction possibilities are different and offer different benefits and precision.

Periodic and stochastic functions have the advantages of showing some regularity and a certain predictability in the case of long-term executions. By definition, periodic functions repeat a behavior after a period t_f and will not diverge to $\pm\infty$ when $t \rightarrow \infty$. Moreover, all the values of a periodic function f can be bounded with absolute certainty between the maximum and the minimum of f on a single period. Therefore, knowing the mean value and the amplitude of a periodic function f part of the expression of an equation of the system, the bounded nature and predictability makes it possible to neglect the variations of f around its mean value m_f if they are negligible compared to other terms of the equations according to the chosen negligibility criterion, and if the period t_f is sufficiently small compared to the considered time interval T . The second condition is necessary to observe the function's periodic nature. If $t_f > T$ or $t_f \approx T$, the function will not be observed as locally periodic, and the deviations imposed by its variations to the mean may not compensate for each other. If the conditions are satisfied, f can be replaced in the concerned areas by m_f without causing a major deviation from the exact behavior. Consequently, periodic functions may be abstracted by their mean values in the appropriate conditions.

In the presence of a stochastic function, the potential simplification may come from the law of large numbers. If the considered time interval is sufficiently long, the mean of the experienced behavior will converge almost surely to the expected value of the

associated random variable. If the variance of the stochastic function is negligible compared to the defined negligibility criterion, its variations may compensate themselves, and the result should not diverge too much from the expected value. This approximation is less safe than the previous ones, but it still gives a tool to help the system abstraction by transforming non-derivable terms into constant monomials, allowing resolution and derivation, both impossible with purely stochastic terms.

Finally, once the degrees of the most complex polynomials have been reduced and that the periodic and stochastic functions have been simplified to their mean value when possible, the problems come from the terms that fit none of the previously mentioned categories, such as exponential and logarithmic functions. Considering the formula of orders of growth presented in [subsection 7.2](#), it appears that $c(f : t \mapsto e^{kt}) = +\infty$ for any constant $k > 0$, meaning that orders of growth are not sufficient to distinguish diverging exponential terms by their relative behavior to ∞ . This implies that simplifications cannot be achieved in the same way as with polynomials. The direction we would like to follow now is to apply polynomial approximations to these embarrassing functions. By combining different interpolation methods, such as Taylor or Chebyshev polynomials, we hope to be able to generalize the state space abstraction approach to any system defined with transcendental ODEs and equations.

7.4 Causality Reasoning

Finally, one last developed possibility is to unify the different generations of qualitative reasoning on systems to consolidate their respective influence and combine their application area. Originally, qualitative reasoning was mainly supported by causal ordering as developed by Iwasaki [40] or De Kleer [41]. This representation of a system highlights the causality dependence between its variables and components and may integrate a notion of temporality. Mythical causality, for example, uses the temporal ordering between cause and consequence to represent the behavior. Causality applied to qualitative reasoning, as developed by de Kleer and Brown, uses a specific form of differential equations named qualitative differential equations (QDE), which can be seen as abstractions of ODE applied to define the confluences of the system. These qualitative equations associate a sign to a formula composed of simple operations between variables and parameters of the system. This idea of QDE has been used in more recent productions such as [12, 42] to represent as simply as possible some system's properties and to reason on incomplete equations. This can be related to our approach, as the choice to abstract the dynamics of the system is at the core of the contributions. One of the drawbacks of this choice was that when models do not integrate sufficiently usable knowledge, reasoning on it becomes irrelevant because all the achievable conclusions are too simple to present significant interest. A possibility to unify this area of qualitative reasoning to the state space abstraction is to change the formulation of the causality bounds (currently expressed with operators such as *PROP*, *CPROP* or *IPROP*) by more standard equations with non valued parameters. For example, the relation between two variables x and y , represented in the absence of further knowledge as $x \text{ PROP } y$ could be expressed with $x = cy$ with c a non valued positive constant. Also, by authorizing more categories of relations with, for example, an operator representing a quadratic proportionality (meaning that $x = cy^2$ with

$c > 0$) or even the existence of a polynomial relation between x and y . That would ensure that the literal expression of every relation could be included in a model using polynomial functions to abstract the state space. By enriching the set of allowed operators with, for example, QPROP, PⁿPROP, PPROP, REL (meaning respectively the existence of a quadratic, polynomial of the n^{th} order, polynomial of unknown order and unknown but existent relations between variables or components) and by relating each of these operators to a symbolic equation linking the variables to be used in an abstraction process, it is possible to apply the presented system abstraction. The absence of instantiated values for the parameters of the equation stops the possibility of computing transition directions and, therefore, the dynamics of the system but still allows a discretization of the state space and reasoning about the qualitative position on the qualitative states of the system.

8 Conclusion and Future Works

In this article, we presented a state of the art and the current state of our research to develop a qualitative abstraction and reasoning approach for the study of CPS and for improvements in unifying the various forms of qualitative reasoning to generalize it to as many situations as possible. In the study of dynamic systems described with polynomial ordinary differential equations, the discretization of the state space builds the qualities to reason about; the study of qualitative transitions using the Lie derivative allows the visualization of a qualitative trajectory of the system, while the introduction of the qualitative zones adds a notion of proximity management, allowing for more concrete applications such as verification, real-time monitoring or simulation piloting. Finally, our previous and current works to generalize this method to more complex, stochastic, or less defined systems give us hope that our contribution can be generalized to more complex and significant use cases.

The current limitations of applying qualitative reasoning to CPS mainly concern the systems represented using less convenient behavioral equations, such as algebraic differential equations, or systems that include stochastic components that cannot be ignored, given the relative magnitude of the different terms. These two cases require the ability to solve equations that are not handled by traditional solving methods and are entirely out of the field considered in this article. The use of remarkable results might give hints to help in the qualitative abstraction of algebraic differential equations. However, most of the possible situations still pose a significant problem for qualitative reasoning. Stochastic systems, which can exhibit stochastic properties under various forms (dynamics, transition condition, uncertainty, ...), have been partially treated in other works such as [43] from Kroger. In this example, a concept of Bayesian hybrid automaton allows the author to deal with major uncertainty in various aspects of the system by representing the current state of the system using different and related Bayesian networks, with weighted values for the state variables integrating the probabilistic aspects of the dynamics and relations between the networks that take into account the stochastic nature of the thresholds of the system. Integrating this contribution to ours will be arduous, but it may solve one of the main remaining weaknesses of our approach.

References

- [1] Lee, E.A., Seshia, S.A.: Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition. MIT Press, Cambridge (2017)
- [2] Henzinger, T.A.: The theory of hybrid automata. In: Verification of Digital and Hybrid Systems, pp. 265–292. Springer, Heidelberg (2000)
- [3] Forbus, K.D.: Qualitative Reasoning. (2014)
- [4] Sugeno, M., Yasukawa, T.: A fuzzy-logic-based approach to qualitative modeling. IEEE Transactions on fuzzy systems **1**(1), 7 (1993)
- [5] Brown, A.L.: Qualitative Knowledge, Causal Reasoning, and the Localization of Failures. MIT Artificial Intelligence Laboratory, Cambridge (1974)
- [6] De Kleer, J.: Qualitative and Quantitative Knowledge in Classical Mechanics. AI-TR-. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge (1975)
- [7] Hayes, P.J.: The naive physics manifesto. Expert systems in the microelectronic age (1979)
- [8] Smith, B., Casati, R.: La physique naïve: un essai d’ontologie. Intellectica **17**(2) (1993)
- [9] Hayes, P.J.: The second naive physics manifesto. Readings in qualitative reasoning about physical systems, pp. 46–63. Morgan Kaufmann, New York (1985)
- [10] Forbus, K.D.: Qualitative process theory. Artificial intelligence **24**(1-3), 85–168 (1984)
- [11] Kuipers, B.: Qualitative simulation. Artificial intelligence **29**(3), 289–338 (1986)
- [12] Louise Travé-Massuyès, P.D.: Modèles et Raisonnements Qualitatifs. Hermes, Paris (2003)
- [13] Berleant, D., Kuipers, B.J.: Qualitative and quantitative simulation: bridging the gap. Artificial Intelligence **95**(2), 215–255 (1997)
- [14] Medimegh, S., Pierron, J.-Y., Boulanger, F.: Qualitative simulation of hybrid systems with an application to sysml models. In: MODELSWARD, pp. 279–286 (2018)
- [15] Bouissou, O., Chapoutot, A., Mimram, S.: Computing flowpipe of nonlinear hybrid systems with numerical methods. arXiv preprint arXiv:1306.2305 (2013)

- [16] Tiwari, A., Khanna, G.: Series of abstractions for hybrid automata. In: International Workshop on Hybrid Systems: Computation and Control, pp. 465–478 (2002). Springer
- [17] Selvaraj, Y., Ahrendt, W., Fabian, M.: Formal development of safe automated driving using differential dynamic logic. *IEEE Transactions on Intelligent Vehicles* **8**(1), 988–1000 (2022)
- [18] Zaatiti, H., Ye, L., Dague, P., Gallois, J.-P., Travé-Massuyès, L.: Abstractions refinement for hybrid systems diagnosability analysis. In: *Diagnosability, Security and Safety of Hybrid Dynamic and Cyber-Physical Systems*, pp. 279–318. Springer, Cham (2018)
- [19] Mosterman, P.J., Biswas, G.: Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **29**(6), 554–565 (1999)
- [20] Maler, O.: Algorithmic verification of continuous and hybrid systems. arXiv preprint arXiv:1403.0952 (2014)
- [21] Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.-H.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: *International Hybrid Systems Workshop, Heidelberg*, pp. 209–229 (1991). Springer
- [22] Maler, O., Batt, G.: Approximating continuous systems by timed automata. In: *International Workshop on Formal Methods in Systems Biology*, pp. 77–89 (2008). Springer
- [23] Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. *Proceedings of the IEEE* **88**(7), 971–984 (2000)
- [24] Aichernig, B.K., Brandl, H., Krenn, W.: Qualitative action systems. In: *International Conference on Formal Engineering Methods*, pp. 206–225 (2009). Springer
- [25] Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. arXiv preprint arXiv:1509.03254 (2015)
- [26] Rosenberg, R.R., Karnopp, D.C.: *Introduction to Physical System Dynamics*. McGraw-Hill, Inc., New York (1983)
- [27] Yano, K.: *The Theory of Lie Derivatives and Its Applications*. Courier Dover Publications, New York (2020)
- [28] Gueuziec, B., Gallois, J.-P., Boulanger, F.: Qualitative models for the supervision of cps simulations. In: *Proceedings of the 25th International Conference on Model*

- Driven Engineering Languages and Systems: Companion Proceedings, pp. 612–616 (2022)
- [29] Mavrouniotis, M.L., Stephanopoulos, G.: Formal order-of-magnitude reasoning in process engineering. *Computers & Chemical Engineering* **12**(9-10), 867–880 (1988)
- [30] Raiman, O.: Order of magnitude reasoning. In: *Readings in Qualitative Reasoning About Physical Systems*, pp. 318–322. Elsevier, San Francisco (1990)
- [31] Kofman, E., Junco, S.: Quantized-state systems: a devs approach for continuous system simulation. *Transactions of The Society for Modeling and Simulation International* **18**(3), 123–132 (2001)
- [32] Cellier, F.E., Kofman, E., Migoni, G., Bortolotto, M.: Quantized state system simulation. *Proc. GCMS'08, Grand Challenges in Modeling and Simulation*, 504–510 (2008)
- [33] Floros, X., Bergero, F., Cellier, F.E., Kofman, E.: Automated simulation of modelica models with qss methods-the discontinuous case. In: *8th International Modelica Conference*, pp. 657–667 (2011)
- [34] Gueuziec, B., Gallois, J.-P., Boulanger, F.: Abstraction qualitative et surveillance de systèmes cyber-physiques. In: *Modélisation des Systèmes Réactifs* (2023)
- [35] Gueuziec, B., Boulanger, F., Gallois, J.-P.: Qualitative tendencies for hybrid system simulation. In: *MPM4CPS 2023-Multi-Paradigm Modelling for Cyber-Physical Systems-26th International Conference on Model Driven Engineering Languages and Systems* (2023)
- [36] Asarin, E., Dang, T., Girard, A.: Hybridization methods for the analysis of nonlinear systems. *Acta Informatica* **43**, 451–476 (2007)
- [37] Dague, P.: Numeric reasoning with relative orders of magnitude. In: *AAAI*, pp. 541–547 (1993)
- [38] Borel, É., Denjoy, A.: *Leçons sur la Théorie de La croissance* vol. 12. Gauthier-Villars, Paris (1910)
- [39] Hardy, G.H.: Properties of logarithmico-exponential functions. *Proceedings of the London Mathematical Society* **2**(1), 54–90 (1912)
- [40] Iwasaki, Y., Simon, H.A.: Causality in device behavior. *Artificial intelligence* **29**(1), 3–32 (1986)
- [41] De Kleer, J., Brown, J.S.: Theories of causal ordering. *Artificial intelligence* **29**(1), 33–61 (1986)

- [42] Medimegh, S.: Analyse formelle de spécifications hybrides à partir de modèles sysml pour la validation fonctionnelle des systèmes embarqués. PhD thesis, Université Paris Saclay (COMUE) (2018)
- [43] Kröger, P., Fränzle, M.: Bayesian hybrid automata: A formal model of justified belief in interacting hybrid systems subject to imprecise observation. *Leibniz Transactions on Embedded Systems* **8**(2), 05–1 (2022)