195

An Assessment of Current Qualitative Simulation Techniques†

Pierre Fouché & Benjamin Kuipers

Department of Computer Sciences
The University of Texas at Austin, Austin TX 78712, USA
Email: fouche@cs.utexas.edu, kuipers@cs.utexas.edu

Abstract: QSIM is a powerful Qualitative Simulation algorithm, which now includes many features that have proven to be necessary in Qualitative Simulation. These features are: reasoning with Higher-Order Derivatives, having Multiple Levels of Abstraction, reasoning in the Phase Space representation, and reasoning about Energy. The aim of this paper is to provide a comprehensive view of all these techniques, by explaining their rationale, showing the problems they address and how they interact. Remaining problems in Qualitative Simulation are also discussed.

Main Topic: Qualitative Simulation

† This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-8602665, IRI-8905494, and IRI-8904454, by NASA grants NAG 2-507 and NAG 9-200, and by the Texas Advanced Research Program under grant no. 003658175. Pierre Fouché holds a grant from Rhône-Poulenc.

Introduction

Qualitative Physics has experienced a rapid growth since its birth, generally dated to the special issue of the *Artificial Intelligence Journal* of December 1984. Among all the formalisms that have been developed, QSIM[1]. originally designed by Kuipers [86] has been greatly improved by many researchers since. It now includes several features that have proven to be necessary in Qualitative Simulation, like reasoning with higher-order derivatives [de Kleer & Bobrow, 84; Kuipers & Chiu , 87], having multiple levels of description [Hobbs, 85; Kuipers & Chiu , 87], reasoning in the phase space representation [Sacks, 87; Struss, 88; Lee & Kuipers, 88; Doyle & Sacks, 89] and reasoning about energy [Fouché & Kuipers, 90a, 90b]. So far these techniques have been described separately and have not been compared to each other. Consequently it was not easy to decide, even for someone very familiar with Qualitative Simulation, which one to apply to solve a practical problem. This paper is an attempt to provide a comprehensive view of these techniques: for each of them we describe their rationale and intuitive appeal, and we compare their relative efficiency on two simple examples, widely used in the Qualitative Physics literature[2]: a block-spring system with or without friction.

The first part of the paper presents the models and the result of their simulation with the QSIM kernel. It is shown that simulation is intractable, mainly because of a phenomenon known as *Chatter*. The second part is devoted to local reasoning techniques. Section 2.1 shows that reasoning with higher-order derivatives and introducing curvature constraints allows to eliminate this phenomenon, but that the real problem for the damped spring is that the level of description is not appropriate: the system is not constrained enough for QSIM to predict a total ordering of the relative occurrence of events, and behaviors keep proliferating. This phenomenon is referred to as *Occurrence Branching*. An alternate way to get rid of chattering, which turned out to solve in part the occurrence branching problem is to shift to a higher level of description by ignoring irrelevant distinctions. This is described in section 2.2.

All the preceding techniques still do not provide a good behavioral description of the damped-spring: QSIM still derives behaviors which are genuinely incompatible with any actual system that abstracts to the model. This problem of incompleteness mainly stems from a combination of the loss of quantitative precision with the local character of qualitative inferences. A way to get a global view of a system behavior is to use the Phase Space representation, as described in section 3.1. This allows QSIM to derive that some properties of a system cannot change through time. The final step to get a correct behavioral description of the spring system is to introduce energy considerations, as shown in section 3.2. A summary is provided in section 4.

With the help of all these techniques, QSIM is now able to derive important properties of industrially significant systems [Fouché & Kuipers, 90a, 90b]. However some problems still remain: The way QSIM handle correspondences between qualitative values and creates new landmarks is not satisfactory; QSIM provides no result about asymptotic behaviors; Finally QSIM cannot always determine whether a behavioral property is a system property. This is described in section 4.

1. Basic Qualitative Simulation

The Spring-Block system (figure 1.1) consists of a block connected to a spring laying on a horizontal table. The block position is referenced by a variable X, the origin being the rest position. The frictionless system will often be referred to as the *simple spring*, and the other as the *damped spring*. Though extremely simple from a structural point of view, deriving their behaviors qualitatively has turned out to be challenging. We know that the

---

[1] For a detailed description of QSIM, see [Kuipers, 86]; see [Kuipers, 89] for a tutorial view.

[2] For instance: [de Kleer & Brown, 84; Forbus, 84; Kuipers, 86; Weld, 87; Travé & Dormoy, 88; Struss, 88; Lee & Kuipers, 88; Ishida, 89]

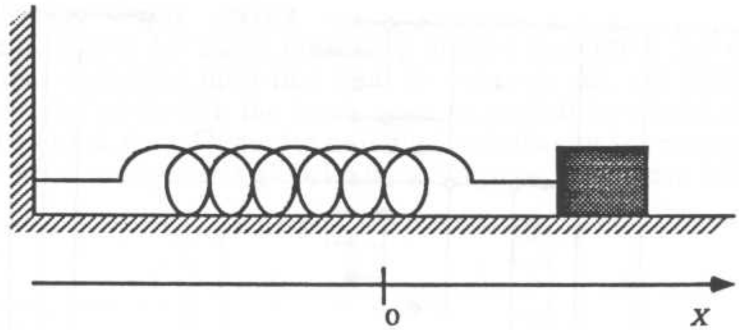


Figure 1.1: The Spring-Block System

force F_S exerted by the spring on the block is inversely proportional to its elongation X. While the relation between F_S and X happens to be linear (Hook's law), we shall not make any linearity assumption, to demonstrate that Qualitative Simulation applies to non-linear systems. For the simple spring, we can directly model that the acceleration is inversely related to the position of the block. For the damped spring, the friction force F_F is inversely proportional to the speed of the block and again we shall not assume that this relation is linear. Figure 1.2 shows the models as they are given to QSIM[1].

```
(define-QDE simple-spring
  (quantity-spaces
    (X (minf 0 inf))
    (V (minf 0 inf))
    (A (minf 0 inf)))
  (constraints
    ((d/dt X V))
    ((d/dt V A))
    ((M- X A) (minf inf)
              (0 0)
              (inf minf))))
```

```
(define-QDE damped-spring
  (quantity-spaces
    (X  (minf 0 inf))
    (V  (minf 0 inf))
    (A  (minf 0 inf))
    (FF (minf 0 inf))
    (FS (minf 0 inf)))
  (constraints
    ((d/dt X V))
    ((d/dt V A))
    ((M- FS X) (minf inf) (0 0) (inf minf))
    ((M- FF V) (minf inf) (0 0) (inf minf))
    ((add FS FF A))))
```

Figure 1.2: QSIM models of the simple and damped springs

Simulation with the QSIM kernel: what we expect...

We start the simulation with the spring stretched and the block immobile. The beginning of the expected behavior is the block moving towards its rest position. What it will do next depends on the friction force. If the motion is frictionless then the block will move across its rest position, reach another extreme and move back to its original position. One can describe this behavior as a stable oscillatory behavior. If friction occurs then it can move towards the rest position without crossing it, if the friction force is strong enough. This is an over-damped behavior. Otherwise, the system will exhibit decreasing oscillations.

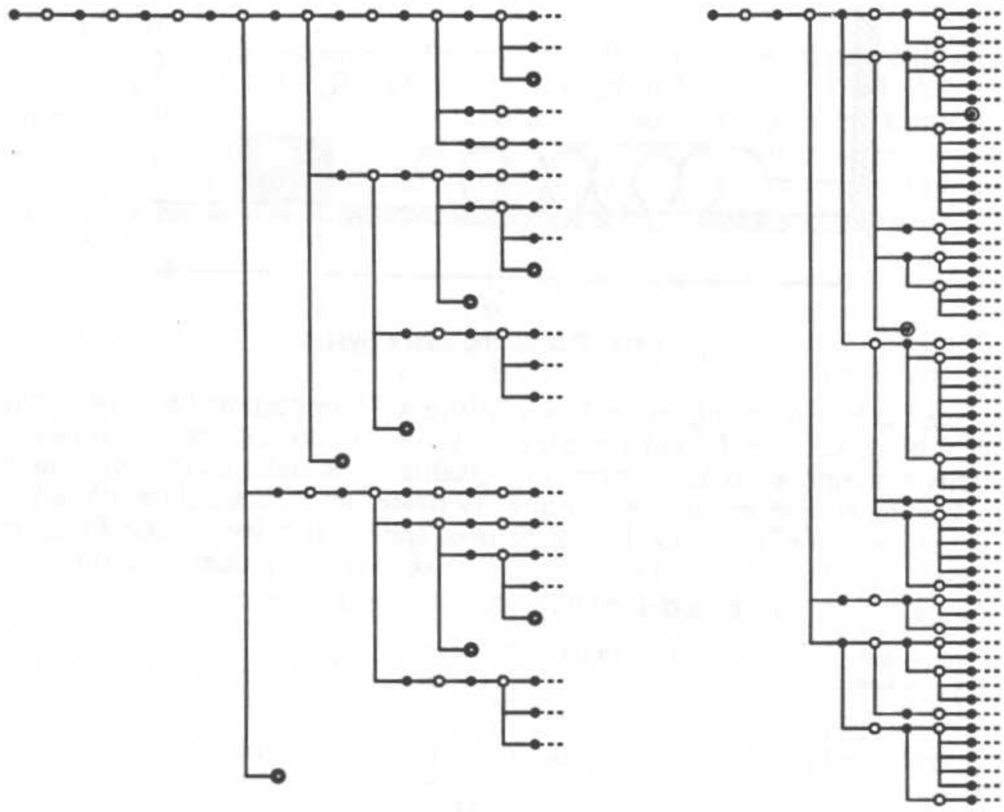
... and what we really get

Figure 1.3 shows the behavior tree[2] of the simple and damped springs when behaviors are allowed to reach time point t_8 and t_4. Clearly this is not as simple as expected.

---

[1] [Farquhar & Kuipers, 90; Throop *et al.*, 90] describe in detail how to use the QSIM is implementation.

[2] A filled circle represents a state at some time point, an empty circle a state at some time interval, a filled circle surrounded by a larger circle a quiescent state and an empty circle surrounded by a larger one a cyclic state, identical to a prior state in the same behavior. States followed by dashed lines are states whose successors have not been computed yet, due to a resource cut-off. Time increases from left to right.



(a) (b)

Figure 1.3: Behavior trees produced by the QSIM kernel
(a) simple spring, time limit t_8, 26 behaviors
(b) damped spring, time limit t_4, 56 behaviors

Problems with the damped spring

Figure 1.4 shows a particular behavior of the damped spring. Analyzing the distinctions among all the behaviors reveals that only the acceleration has distinct behaviors. The explanation is that the acceleration is constrained only by continuity: it is the sum of a continuously increasing function (the friction force) and a continuously decreasing function (the spring force). Since we have no more information about these two functions, their sum can be a decreasing, steady or increasing function of time. While all these behaviors are real behaviors, they only occur if the functions relating speed and friction force and position and spring force have uncommon shapes. If one can make additional assumptions about their second derivatives, which are valid for actual spring systems, these behaviors are no longer possible. This is developed in section 2.1.

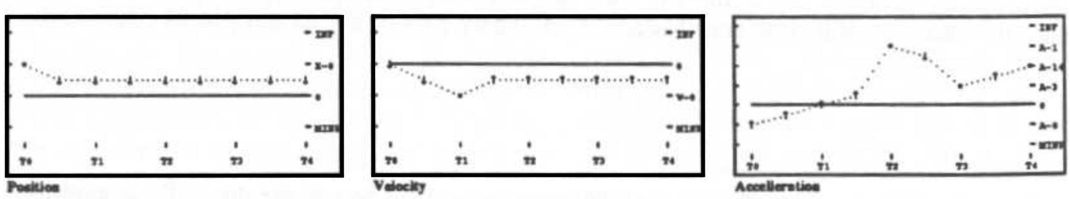
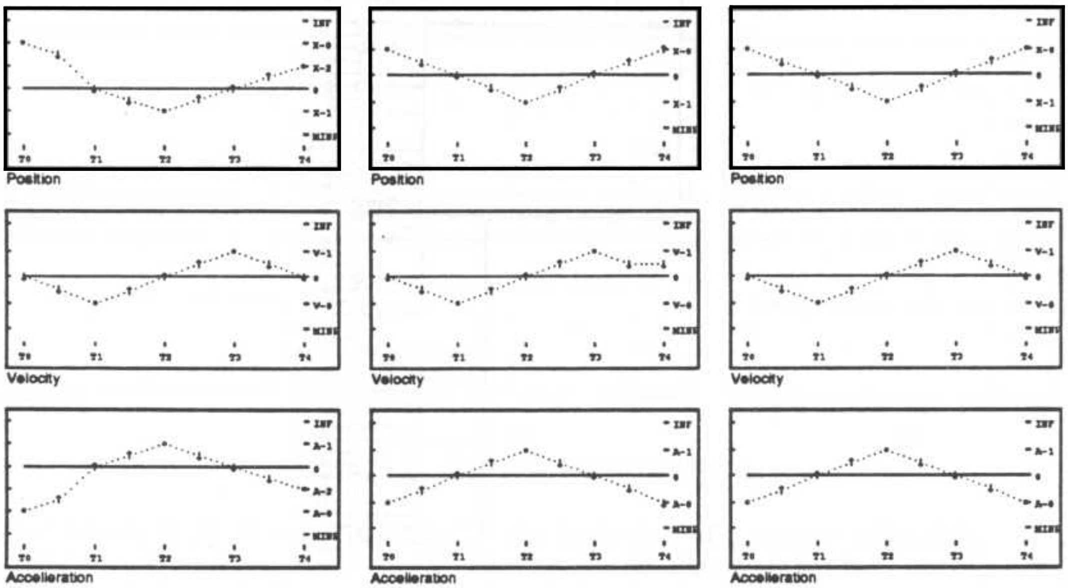


Figure 1.4: One chattering behavior of the damped spring

Problems with the simple spring

The first tree of figure 1.3 starts branching at time t_4. Figure 1.5 displays the three possible behaviors computed until that time. In behavior (a), the block stops before its initial position. In behavior (b), the block goes beyond and in behavior (c) the system comes back to its initial state. Of course only the last behavior is genuine [Kuipers, 86]. In fact solving the problem requires a global point of view on the system behavior.



(a) (b) (c)

Figure 1.5: Three way branching of the simple spring

2. Local Reasoning Techniques

2.1. Reasoning with Higher-Order Derivatives

The phenomenon that occurs for the damped spring is a particular instance of a more general problem, called *Chatter*. de Kleer & Bobrow [84] and Kuipers & Chiu [87] proposed solutions. The state of the art is presented in [Kuipers *et al.*, 89].

Principle

Roughly speaking, a variable may exhibit chatter if its derivative is unconstrained. Chatter doesn't occur for the simple spring because the acceleration has a behavior similar to the position whose derivative is explicitly represented. Basically, if at some time point a variable transitions to a critical point (that is, its derivative becomes zero) then its qualitative value in the next open interval of time is determined by its second derivative. If no information is provided about this second derivative then QSIM will branch on each possible future. Multiple occurrence of this phenomenon leads to intractable branching. The method proposed in [Kuipers *et al.*, 89] is performed in three steps: Identify equivalence classes of variables in the QDE likely to chatter; Derive an expression (called the curvature constraint) for the second derivative of one variable among each class[1]; Use the sign of the second derivative to constrain simulation[2].

---

[1]Derivation can be performed manually or automatically by an integrated algebraic manipulator.

[2]Curvature constraints provide useful information only when they are not ambiguous, that is when the sign of the second derivative is positive or negative. When it is zero, then the same problem occurs and the next qualitative value of a chattering variable is given by the sign of its third derivative. For the damped spring

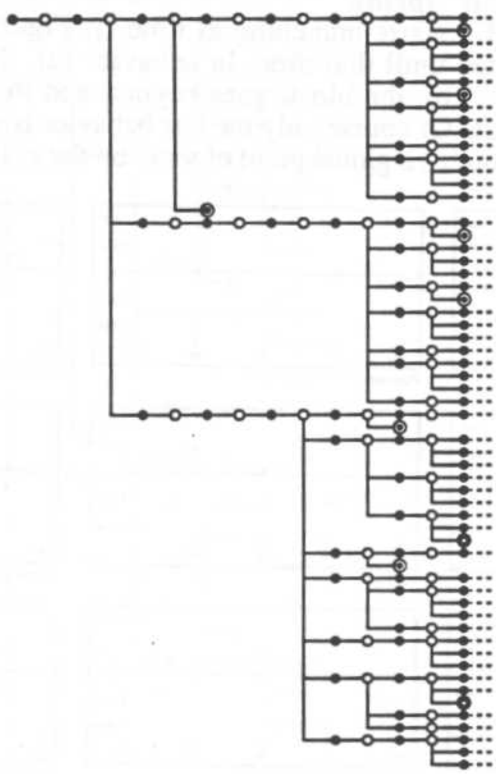


Figure 2.1: Tree of behaviors of the damped spring using curvature constraints

Deriving curvature constraints in the presence of monotonic (M^+ or M^-) constraints requires additional assumptions about them: the semantics of $M^+(Y, X)$ is that there exists a function f such that $\forall\ t$, $Y(t) = f(X(t))$ with $f'(x) > 0$. This implies that $\partial Y = \partial X$ [1] but $\partial^2 X$ and $\partial^2 Y$ are not related. The assumption that $\partial^2 X = \partial^2 Y$ is called the *sign-equality assumption* and is applied by the algebraic manipulator. This assumption is obviously satisfied when f is linear but more generally when f's shape is relatively smooth[2].

Results

Figure 2.1 shows the tree of behaviors for the spring with friction when simulation is allowed to reach time point t_7. Instead of 56 behaviors at time t_4, only 4 are produced. But the tree is still growing exponentially.

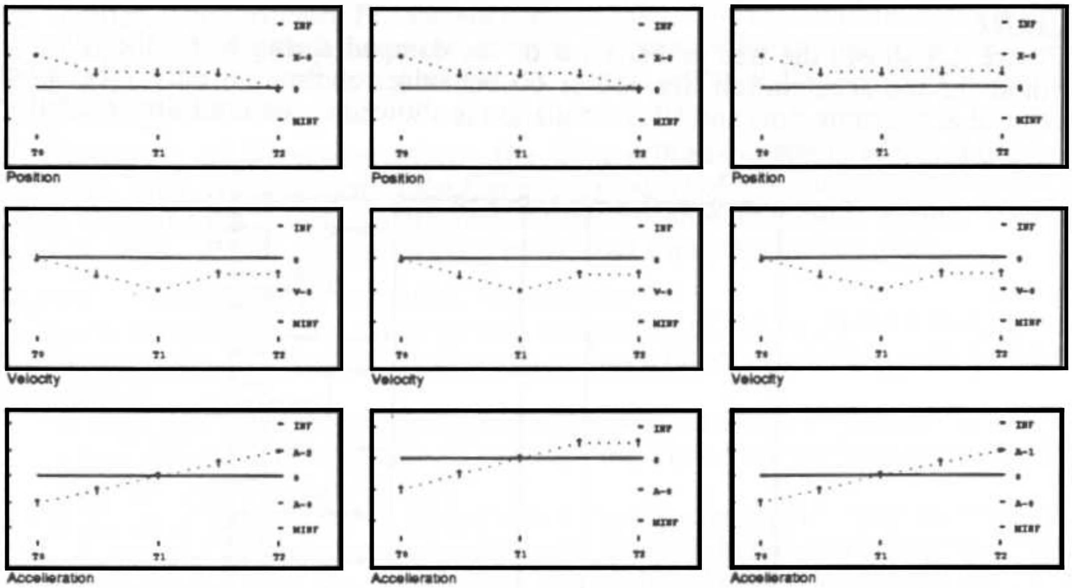
Figure 2.2 shows the three possible behaviors at time t_2. Time t_2 is defined either by X crossing zero, A reaching a critical value, or both at the same time. In the linear case, the relative occurrence of these events is a *system property* [Lee *et al.*, 87], that is, a property that depends on the parameters of the system, but that that does not vary during simulation. In the general case, it depends on the functions F_F and F_S and thus any order on the occurrence of events is possible. The point is that even if these distinctions are real we do

---

curvature constraints are always not ambiguous. See [Kuipers *et al.*, 89] for a complete discussion of the use of Higher-Order Derivatives in Qualitative Simulation.

[1] ∂X is a notation used by de Kleer & Bobrow [84] and represents the sign of X's derivative. By extension, $\partial^n X$ represents the sign of X's n[th] derivative and $\partial^0 X$ the sign of X.

[2] See [Kuipers *et al.*, 89] for a discussion of physical situations in which these assumptions may be violated.



(a) (b) (c)

Figure 2.2: Three way branch at time t_2

not care about them. When we describe the behavior of a spring informally, we are interested in the position of the block, its speed, and also whether it is accelerating or decelerating. In other words we are just interested in the sign of the acceleration but not its derivative: the level of description is not appropriate and we are getting lost in too detailed a behavioral description.

2.2. Switching to a Higher Level of Description

Ignoring irrelevant distinctions

In the preceding section, simulation was branching according to the ordering of the following events: *A reaches a critical value* and *X crosses zero*. The former event is not of particular interest and Kuipers & Chiu [87] developed a method to ignore this irrelevant distinction: If the directions of change *inc*, *dec* and *std* are collapsed into a single value *ign* then branching will no longer occur since it is impossible to detect critical values. Ignoring directions of change has three main advantages over using information about higher order derivatives :

- No assumption is made about the nature of M^+ or M^- instances and thus soundness is preserved.
- No algebraic manipulation has to be made to derive higher-order derivative properties.
- It eliminates occurrence branching caused by a variable reaching a critical value.

However, in some cases:

- Information about higher-order derivatives is required to filter out genuinely spurious behaviors [Dalle Molle, 89; Kuipers *et al.*, 89], that the ignore-qdirs[1] method does not rule out. Higher-order derivative constraints usually perform better than ignore-qdirs on non-oscillatory systems.
- It may be very interesting to know explicitly the direction of change of some variable.

[1]Short for *Ignoring the qualitative direction of change*

Results

Figure 2.3 shows the tree of behavior of the damped spring using the *ignore-qdirs* method for the acceleration. Instead of 60 behaviors at time t_7, only 7 behaviors are produced at the same time and 50 if simulation is allowed to run until time t_{10}. but we still have too many behaviors...

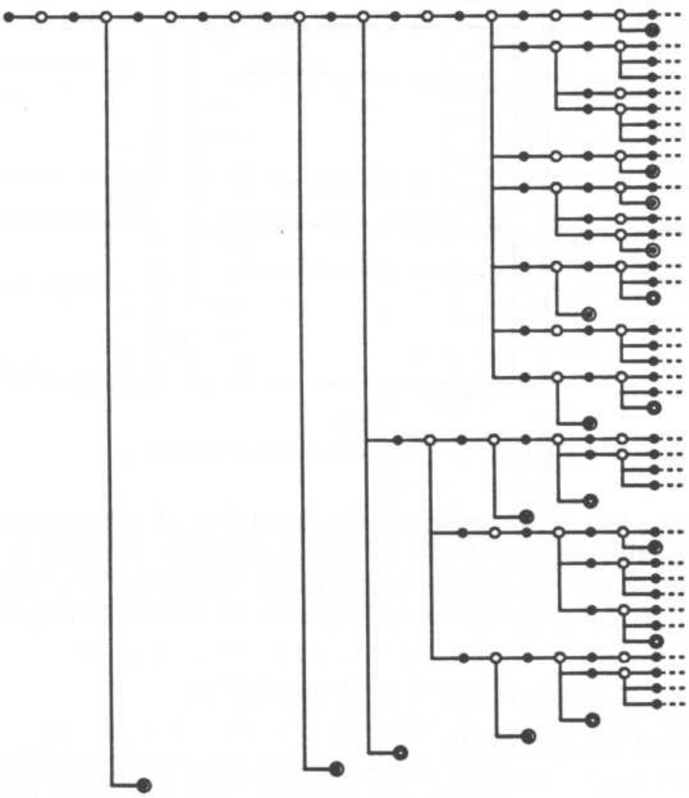


Figure 2.3: Tree of behaviors of the damped spring, ignoring the direction of change of the acceleration

Browsing the tree reveals that many of them are intuitively spurious and figure 2.4 shows one of them: the block seems to oscillate with increasing amplitude when X is positive and constant amplitudes when X is negative. But reasoning about oscillations requires a global point of view.

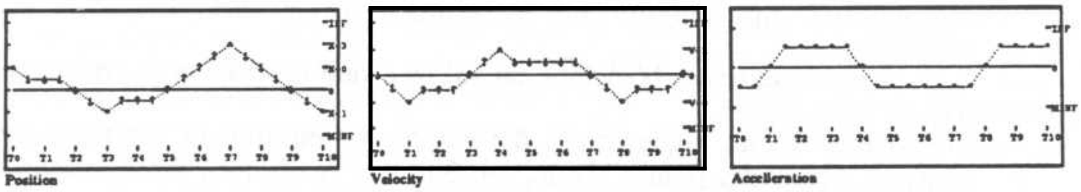


Figure 2.4: An intuitively spurious behavior

3. From a Local to a Global Point of View

The filtering techniques we have discussed so far have been purely local: Determining the validity of state transitions has been made by considering only two successive states, and the basic validity criterion has been continuity of variables. But the first thing that we are thinking of when we are looking at a spring is oscillation. And dealing with oscillation requires a global view of a system behavior : it involves comparing the state of a system at a given time with its state at some time before, and these two time points are not successive in our qualitative description. For instance, expressing that "the amplitude of oscillations starts increasing" in figure 2.4 requires comparing the qualitative magnitude of the block position at time points t_7 and t_0. One way to reason globally about a system is to use a phase space representation.

3.1. Changing of Representation: the Phase-Plane View

The *Phase Space* for a system is the Cartesian product of a set of independent variables that fully describe the system. In practice, it provides another view of system behavior: a point in the phase space represents a system state and a trajectory a behavior.

Principle of the Non-Intersection Constraint

A major theorem about the existence and uniqueness of the solution of an autonomous first-order ordinary differential equation system has a direct equivalent in the phase space representation:

> *A trajectory which passes through at least one point that is not a critical point cannot cross itself unless it is a closed curve. In this case the trajectory corresponds to a periodic solution of the system.*

Lee & Kuipers [88] and Struss [88] discovered that this property can be conveniently translated into qualitative terms for second order systems. In this case, the phase space is a plane and a general intersection criterion can be established even if trajectories are described qualitatively. This criterion can then be used to rule out trajectories that intersect themselves.

Figure 3.1 shows the Non-Intersection Constraint at work[1]: the behavior corresponds to figure 2.4. In the phase plane (V A) the trajectory is a closed curve and thus the behavior should be cyclic and trajectories in other phase planes be closed curves too. But this is not the case and the behavior is labeled as spurious (**x** at the end of each trajectory).

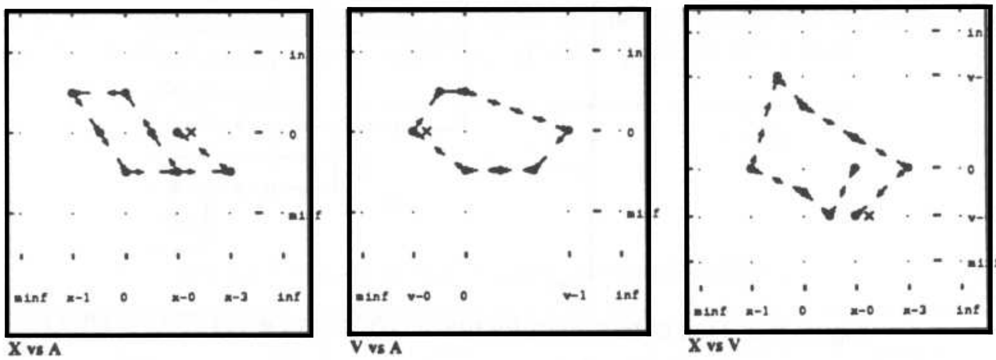


Figure 3.1: The Non-Intersection Constraint at work

Results

Figure 3.2 shows the relative efficiency of the Non-Intersection Constraint on the damped spring example. Instead of 50 behaviors only 15 are created at time t_{10}. Note that only one cyclic behavior is now detected. One can see that the tree is divided into three main branches at time t_6. The upper branch corresponds to increasing oscillations, the middle to damped oscillations and the lower to stable oscillations. It is interesting to compare this tree with the one obtained for the simple spring using the Non-Intersection Constraint (figure 3.3). The two trees are very similar. For the simple spring branching

---

[1] One must notice that in the quantitative case, only one among all possible couples of independent variables is necessary to check the non-intersection property. However, since reasoning qualitatively implies losing a certain amount of information, all possible phase planes must be taken into account to fully capture the Non-Intersection Constraint: one phase plane may contain information that is not present in another one.

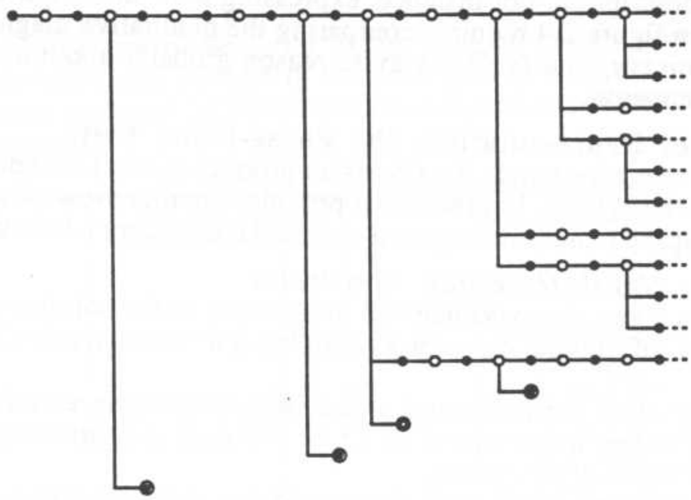


Figure 3.2: Tree of behaviors for the damped spring with NIC until time t_{10}

occurs at time t_4 instead of time t_6 and the branch corresponding to decreasing oscillations is placed at the top of the tree instead of at the middle.

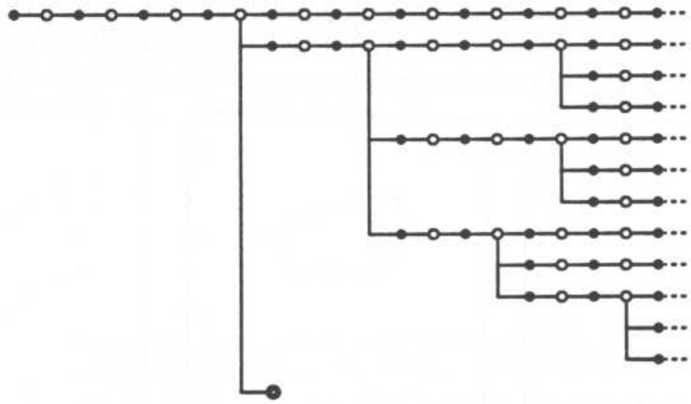


Figure 3.3: Tree of behaviors for the simple spring with NIC until time t_{10}

Let us analyze more deeply the differences:

- Qsim does not allow the simple spring to reach quiescence.
- The reason for the time difference between the first branching occurrences (t_4 for the simple spring and t_6 for the damped spring) is that the variables X and A are in phase for the simple spring but there is a phase difference for the damped spring. This phase difference implies the creation of distinct time points for the events *A crosses 0* and *X crosses 0*.
- Both trees exhibit more branching on the branch corresponding to increasing oscillations. The explanation is given in section 5.

But of course only damped oscillations are valid for the damped spring and only stable oscillations for the simple spring. Determining unambiguously the character of oscillations (damped or stable) requires energy considerations.

3.2. Reasoning about Energy

Intuitively increasing and decreasing oscillations for the simple spring (figure 1.5.a and b) are spurious because the system is conservative (the mechanical energy is conserved) and the potential energy of the block depends only on its position. Fouché & Kuipers [90a, 90b] developed a method, implemented as a global filter called the *Energy Constraint*, that automates this reasoning. It is based on a *qualitative interpretation of the Law of Conservation of Energy*, and on the *decomposition of processes into conservative and non-conservative ones*[1]. It consists of computing the sign of the work of conservative and non-conservative forces and checking that this is compatible with the sign of the variation of kinetic energy. The Energy Constraint, like the Non Intersection Constraint, takes advantage of two main features of QSIM: it creates new landmarks during simulation, thus providing enough information to compute the sign of the quantity defined above, and it describes behaviors of a system directly in terms of its variable histories, which makes non-local reasoning very natural.

Results

Currently, QSIM has to be provided with the name of variables representing conservative and non-conservative terms, C and N_c. For both the simple and the damped springs, the decomposition of the acceleration is trivial:

- For the simple spring: $C(X(t)) = A(t)$ and $N_c(t) = 0$.
- For the damped spring: $C(X(t)) = F_S(t)$ and $N_c(t) = F_F(t)$.

With the Energy Constraint, QSIM is able to determine that the two first behaviors of figure 1.5 are spurious with the following justifications:

- a: `Inconsistent: between t0 and t4, Ke-var = 0, C-work = +, NC-work = 0`
- b: `Inconsistent: between t0 and t4, Ke-var = +, C-work = 0, NC-work = 0`

and thus *we come up with only one, genuine behavior for the simple spring.*

Applying the Energy Constraint to the damped spring also allows to get the correct behavioral description as shown in figures 3.4 and 3.5. We come up with one infinite, pseudo-cyclic behavior, exhibiting decreasing oscillations, and an infinite set of quiescent behaviors, for the system can become over-damped each time the block is moving toward its rest position. If the system was linear, it could not become over-damped after the first oscillation, but without a linearity assumption, these are genuine behaviors.

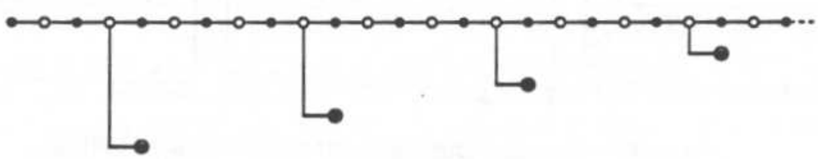


Figure 3.4: Tree of behaviors of the damped spring

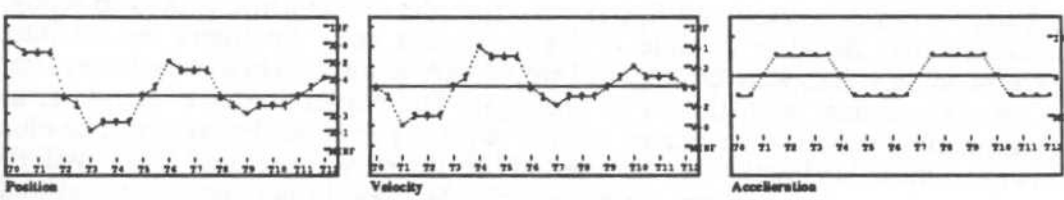


Figure 3.5: Decreasing oscillations of the damped spring

4. Summary

Here are two tables that summarize the results of applying these different techniques to the simple and damped springs. Figures in bold type-face indicate that the number of behavior is correct.

---

[1]This method is actually not restricted to mechanical systems and is applicable to any second- or higher-order system.

Simple Spring	t_4	t_8
QSIM Kernel	3	26
Non-Intersection Constraint	3	7
Energy Constraint	1	-

Damped Spring	t_4	t_7	t_{10}
QSIM Kernel	56	Too many	Too many
Higher-Order Derivatives	4	60	Too many
Ignore Qdir	2	7	50
Ignore Qdir *and* Non-Intersection Constraint	2	5	15
Ignore Qdir *and* Energy Constraint	2	3	4

5. Open Problems

Apparently we are now able to correctly simulate any reasonably complex system without any problem. Of course this is not true and some problems still remain.

Constraint Checking VS Limit Analysis

Currently QSIM uses landmarks to both check constraints (by the means of corresponding values) and perform limit analysis. Here are two examples to illustrate that this can cause problems sometimes.

Suppose for some reason we have to know the direction of change of the acceleration for the damped spring. Simulating it taking into account higher-order derivatives and applying energy filtering produces 103 behaviors at time t_{10}, many of them being spurious. Figure 5.1 shows one of them where the variables F_F, F_S and A only are plotted.

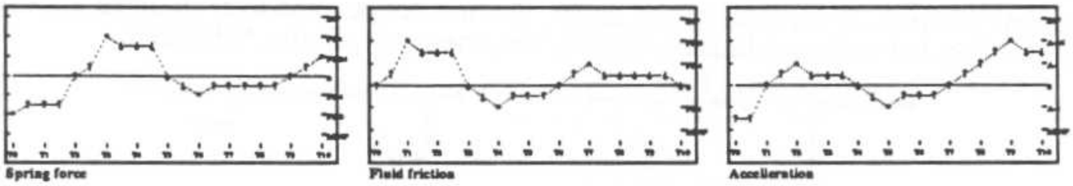


Figure 5.1: One spurious behavior of the damped spring

In this behavior, the acceleration A is greater at time t_{10} than at time t_3. But between t_3 and t_{10}, F_S has decreased and $F_F(t_3) = F_F(t_{10})$: the *add* constraint fails to recognize an inconsistency. Because at time t_3 A's qmag is a not a landmark but an interval no corresponding values were created, that would have allowed to detect the inconsistency.

One way to deal with this is to propagate landmarks across constraints, allowing corresponding values to be created and consequently spurious behaviors to be eliminated. However, these landmarks will be inserted into quantity spaces and QSIM will then take them into account to perform limit analysis. As they do not represent real qualitative distinctions, this will lead to intractable occurrence branching. One could also allow correspondences between intervals. This would solve the above problem, but may fail in other cases if an interval is refined later in the simulation[1]. The appropriate solution is to *distinguish the two tasks constraint checking and limit analysis*: landmarks necessary to check constraints do not always have to appear in a quantity space.

As a second example, consider a spring for which energy is provided instead of being dissipated. Simulation of such a system produces a tree similar to figure 4.2, except that

[1]For instance, consider the following episode:
qval(A, t_0)= <(A_1 A_2), *inc*>, qval(A, (t_0,t_1)) = <(A_1 A_2), *inc*>, qval(A, t_1)= <A_3 , *std*> with $A_3 \in (A_1\ A_2)$
At t_1, we know that qmag(A, t_0)= (A_1 A_3)

branches corresponding to steady or decreasing oscillations are pruned. The tree exhibits a three way pattern of branching: at each oscillation, once QSIM has determined that the block goes further away than the previous maximal position X_n, then, when the block comes back, the ordering of events *X crosses* X_n and *A crosses zero* is undetermined, leading to occurrence branching. But after X_n is crossed a first time, X_n is not of particular interest any longer and it should be withdrawn from the quantity space. We are currently experimenting with new, more flexible methods to manage creation and withdrawal of landmarks.

Asymptotic behaviors: what are they ?

Another issue is to determine the asymptotic behavior of oscillatory systems. Let us see on the damped spring example how this can be formulated. Let Xl_n be the maximal value of the position when X is positive in the n^{th} cycle. Intuitively we can say that the oscillations tend to a limit if the sequence (Xl_n) has a limit when n tends to infinity. From the simulation we know that Xl_n is a decreasing sequence and that zero is a lower bound. Thus Xl_n has a limit. Let Xl be this limit. Suppose we choose Xl_0 equal to Xl. From the definition of a limit we must have $Xl_n = Xl$ for all n. The only value Xl_0 for which Xl_n is constant is zero and thus $Xl = 0$. We plan to generalize and incorporate this kind of reasoning into QSIM.

Asymptotic behaviors: can they be reached in finite time ?

Determining whether this limit can be reached in finite time can be done using the phase-plane representation. So far the non-intersection constraint has been used to check that a trajectory does not intersect itself, but it also prohibits intersection of any two trajectories in the same phase portrait, if they correspond to different initial conditions. Since we know that the asymptotic behavior of the damped spring corresponds to immobility at the rest position, which is itself a possible trajectory (actually a point), we can conclude that the asymptot can never be reached in finite time. Doyle and Sacks [88, 89] developed a general methodology to interpret trajectories in the phase space representation and certainly some of their techniques could be used in the QSIM framework.

Is a behavioral property a system property ?

Currently QSIM is not able to determine that some behavioral properties of a system do not change through time. For instance if the damped spring is linear then it cannot become over-damped after the first oscillation. We have to extend QSIM so that it can automatically determine whether a behavioral property is a system property. But at the moment these words are written, we do not know how...

Conclusion

We have seen that with the help of several filtering techniques (using Curvature Constraints, Ignoring the Direction of Change of a variable, using the Non-Intersection or the Energy Constraint), QSIM is now able to simulate systems that were previously intractable. We hope this paper will help QSIM users choose the appropriate methods to simulate their models. However, another possible improvement and direction of research is to automate the process of choosing these methods.

Acknowledgments

Pierre Fouché is particularly indebted to his advisor Jean-Paul Barthès, from the University of Compiègne, France, for his constant support and useful advice. This paper benefited from discussions with many members of the Qualitative Reasoning Group at the University of Texas at Austin, especially Wood W. Lee. Thanks to Adam Farquhar for a careful reading of a previous draft of this paper.

References

[de Kleer & Brown, 84]
de Kleer , J. & Brown, J.S.
A Qualitative Physics Based on Confluences
Artificial Intelligence 24, 7-83, 1984

[de Kleer & Bobrow, 84]
de Kleer , J. & Bobrow, D.G.
Qualitative Reasoning with Higher-Order Derivatives
Proceedings of AAAI 84, 86-91, 1984

[Dalle Molle, 90]
Dalle Molle, D.
Qualitative Simulation of Dynamic Chemical Processes
Technical Report AI-TR-89-107, The University of Texas at Austin, 1989

[Doyle & Sacks, 89]
Doyle, J. & Sacks, E.
Stochastic Analysis of Qualitative Dynamics
Proceedings of IJCAI 89, 1187-1192, 1989

[Farquhar & Kuipers, 90]
Farquhar, A. & Kuipers, B. J.
QSIM User's Manual
Technical Report AI-TR-90-123, The University of Texas at Austin, 1990

[Forbus, 84]
Forbus, K. D.
Qualitative Process Theory
Artificial Intelligence 24, 85-168, 1984

[Fouché & Kuipers, 90a]
Fouché, P. & Kuipers, B.J.
Reasoning about Energy in Qualitative Simulation
Technical Report AI-TR-90-xx (forthcoming), The University of Texas at Austin, 1990

[Fouché & Kuipers, 90b]
Fouché, P. & Kuipers, B.J.
Introducing Energy into Qualitative Simulation
Submitted to AAAI-90

[Hobbs, 85]
Hobbs, J.R.
Granularity
Proceedings of IJCAI 85, 432-436, 1985

[Ishida, 89]
Ishida, Y.
Using Global Properties for Qualitative Reasoning: A Qualitative System Theory
Proceedings of IJCAI 89, 1174-1179, 89

[Kuipers, 86]
Kuipers, B. J.
Qualitative Simulation
Artificial Intelligence 29, 289-338, 1986

[Kuipers & Chiu, 87]
Kuipers, B. J. & Chiu, C.
Taming Intractable Branching in Qualitative Simulation
Proceedings of IJCAI 87 1078-1085, 1987

[Kuipers, 89]
Kuipers, B. J.
Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge
Automatica, Vol. 25, No. 4, pp. 571-585, 1989

[Kuipers *et al.*, 89]
Kuipers, B. J., Chiu, C., Dalle Molle, D., Throop, D.
Higher-Order Derivative Constraints in Qualitative Simulation
Technical Report AI-TR-90-116, The University of Texas at Austin, 1989

[Kuipers *et al.*, 87]
Lee, W. W., Chiu, C., Kuipers, B. J.
Developments Towards Constraining Qualitative Simulation
Technical Report AI-TR-87-44, The University of Texas at Austin, 1987

[Lee & Kuipers, 88]
Lee, W. W. & Kuipers, B. J.
Non Intersection of Trajectories in Qualitative Phase Space : A Global Constraint for Qualitative Simulation
Proceedings of AAAI 88, 286-291, 1988

[Sacks, 87]
Sacks, E.
Piecewise Linear Reasoning
Proceedings of AAAI 87, 655-659, 1987

[Struss, 88]
Struss , P.
Global Filters for Qualitative Behaviors
Proceedings of AAAI 88, 275-27, 1988

[Throop *et al.*, 90]
Throop, D. & the Qualitative Reasoning Group
QSIM Maintainer's Guide
Technical Report AI-TR-90-124, The University of Texas at Austin, 1990

[Travé & Dormoy, 88]
Travé, L. & Dormoy, J. L.
Qualitative Calculus and Applications
Proceedings of the 12th World Congress on Scientific Computation, Paris, 1988

[Weld, 87]
Weld, D. S.
Comparative Analysis
Proceedings of IJCAI 87, 959-965, 1987

[Weld & de Kleer, 89]
Weld, D.S. & de Kleer, J., ed.
Readings in Qualitative Reasoning about Physical Systems
Morgan Kaufmann Publishers, 1989