

Prioritising Behaviours in Qualitative Simulation

Roy Leitch and Qiang Shen

Intelligent Automation Laboratory

Department of Computing and Electrical Engineering

Heriot-Watt University

Edinburgh EH14 4AS, Scotland

Abstract

We present a simple but potentially extremely powerful technique for associating degrees of commitment to the possible behaviours produced by qualitative simulation algorithms. The 'distance' between the predicted successor states and those determined from local propagation through the constraints is used to determine the most likely state and hence prioritise the behaviours. The method forms a basis for a technique for performing progressive reasoning to cope with real-time constraints. In this manner, the most likely solution is generated first and the subsequent behaviours are only generated if higher priority behaviours fail. Ultimately all possible behaviours are generated, as in current algorithms, and the soundness of qualitative predictions is guaranteed. The approach is applicable to any non-constructive simulation algorithm (e.g. QSIM and its derivatives) when the landmarks are represented by known real numbers or semi-quantitative information is utilised.

1 Introduction

The major goal of qualitative simulation research is to develop techniques for generating a description of the behaviour from a structural model that captures only those distinctions that are necessary for the purpose for which the qualitative model has been constructed. Such distinctions determine the necessary *precision* of the description of a system's behaviour. Models that generate behaviours (solutions) with less precision than that of a given reference model have been called *abstraction* models [Leitch, 1987; Struss, 1991]. In which case, qualitative models are abstractions of real-valued models and form a major contribution that AI is currently making to the modelling of physical systems. However, other properties of behaviours can also be defined. Another 'dimension' upon which behaviours can be classified concerns the degree of *commitment* or uncertainty associated with a given state, behaviour or, indeed, (constraint-based) model. This degree of commitment is important in the representation of complex and/or ill-defined systems [Shen and Leitch, 1992a] as it allows inherent uncertainty to be represented explicitly. It is fundamentally different from modifying the precision through abstraction. In abstraction complete certainty is still assumed albeit in a lower

resolution quantity space. Whereas, in commitment uncertainties in real-world modelling are explicitly represented. In spite of uncertainty being a major attribute of AI approaches to knowledge representation, it has yet to be fully explored within approaches for Qualitative Reasoning (QR); until now, only a few proposals [D'Ambrosio, 1987; Dubois and Prade, 1989; Shen and Leitch, 1992b, 1993] have been made incorporating uncertainty within QR researches. In our view the reason that the QR community almost entirely focuses on abstraction operations [Weld and de Kleer, 1990], is in the proper preoccupation with maintaining the formal properties of the simulation algorithm. However, we argue, in this paper, that by incorporating uncertainty within QR we can associate degrees of commitment to behaviours which offer very significant advantages for practical applications. Loosely speaking, we can use commitment to prioritise behaviours, thereby making qualitative simulation more effective and time-constrained simulation or model-based diagnosis [Dvorak and Kuipers, 1991; Leitch and Shen, 1991] much more practical.

In this paper, we present a basically simple technique for assigning commitment to states and hence priorities to behaviours. The approach is not restricted to techniques that utilise uncertainty in the definition of the quantity space such as FuSim [Shen and Leitch, 1993], but can easily be incorporated within non-constructive 'crisp' (zero uncertainty) simulation algorithms, e.g. Q3 [Berleant and Kuipers, 1992] and HR-QSIM [Weld, 1990], when semi-quantitative information is utilised. The following sections give formal definitions of our approach to assigning commitment to states and to describing methods for propagating priority measures for a particular behaviour. We then outline the potential uses of the technique which appear to us to be very significant indeed. Finally, we conclude and suggest some of the many possible extensions to the method.

2 Priorities of Qualitative Behaviours

In present approaches to qualitative simulation, except for FuSim, the worst case solution is always assumed. That is, all *theoretically* possible successor states of a present state are maintained and, in fact, propagated with equal status. This, of course, leads to the generation of spurious behaviours that discredits qualitative simulation in the eyes of application engineers, in that, such multiple behaviour predictions are at variance with the uniqueness of the behaviour of the physical

world. However, attaching an uncertainty measurement to those potential successor states, and hence a commitment to an associated behaviour, allows prioritised generation of behaviours or an efficient mechanism for their use within applications.

2.1 State Priority

Over the last few years a family of qualitative simulation algorithms [Berleant and Kuipers, 1992; Shen and Leitch, 1993; Weld, 1990] have been developed from the seminal work of Kuipers [1986]. This generated a highly innovative 'non-constructive' approach to the propagation of system states. In this method, the system model, usually a set of (qualitative) constraints, is used within a 'generate-and-test' procedure that first determines all possible successor states based on assuming continuity of system variables and then removes those states that are inconsistent with the system model and other known system properties. This has the very great advantage that the quantity space remains finite but suffers from the inevitable problem that, due to the inherent ambiguity of the underlying qualitative calculi [Struss, 1988a], not all invalid behaviours are removed. This results in a set of behaviours that contain the qualitatively correct behaviour but also other spurious behaviours that do not reflect any real situations. Spurious behaviours tend to obscure the 'real' behaviour hence making qualitative simulation less attractive to application engineers, though much progress has been made in developing so-called global filters [Fouche and Kuipers, 1992; Lee and Kuipers, 1988; Struss, 1988b] that can reduce the number of such behaviours.

To formalise the presentation, without losing generality, it is assumed that a finite set of possible states for each variable survive all possible filterings and that this set is denoted by $S = \{s_1, s_2, \dots, s_n\}$. Apparently, even in the worst case where no possible next states produced by the use of continuity are filtered, the cardinality of this set, n , is a small number. For instance, in QSIM, HR-QSIM, and FuSim n cannot be larger than 3, 4, and 6 respectively. Within this set, each element s_i , $i = 1, 2, \dots, n$, represents a pair of qualitative magnitude and qualitative rate-of-change and can be denoted by (A_i, B_i) . With the exception of FuSim, all other qualitative simulation algorithms treat the successor states that remain after constraint filtering as equally likely and, hence, propagate each state with equal priority. If any possible successor state intersects with the constraints it is retained; no information about the degree of matching is utilised or indeed generated. However, intuitively, those states that maximally intersect with the constraints can be considered as more likely and therefore can be given a higher degree of commitment and hence priority in further processing. We propose that the commitment of a particular state within such a set of possible states can be determined by a metric function that can faithfully represent certain preference criteria. Notationally, for each possible new state s_i the priority is represented by $p(s_i)$, where $p(s_i) = j$, $j \leq n$, indicates that s_i is the j th preferable state, namely the i th state is of the j th priority.

For convenience, the states of a system variable generated by the state transition tables and/or rules (based on continuity) [Kuipers, 1986; Shen and Leitch, 1993; Weld, 1990] are called *predicted* states of the variable and those obtained by performing local propagation of the predicted states of other system variables throughout a set of constraints or the system model are termed *propagated* states. The basic criterion used herein is to assign the highest priority to a predicted state that maximises the possibility degree to which the state is considered to be the actual one. A predicted state of a particular variable is said to be the most possible actual state of the variable if it results in the shortest distance between all the predicted states of the variable and their corresponding propagated states. This can be better understood by considering the following simple example based on the 4-tuple parametric representation $[a, b, \alpha, \beta]$ of the fuzzy number used in FuSim. Due to lack of space the semantics of the 4-tuples are omitted here but can be found in [Shen and Leitch, 1993]. Importantly, it is worth emphasising that this development is not restricted to the fuzzy quantity space but is applicable to any interval based non-constructive algorithm.

Suppose that a system has three variables $\{a, b, c\}$ related by the addition constraint $c = a + b$ and that these variables take values from a fuzzy quantity space. Consider that, based on continuity, two possible successor states are *predicted* as follows:

$$\begin{aligned} s_1 &= \{([10, 12, 0.1, 0.1], [4, 5, 0.1, 0.1]), \\ &\quad ([3, 4, 0.2, 0.2], [6, 7, 0.2, 0.2]), \\ &\quad ([13.5, 16.5, 0.3, 0.3], [10, 12, 0.3, 0.3])\}, \\ s_2 &= \{([10, 12, 0.1, 0.1], [4, 5, 0.1, 0.1]), \\ &\quad ([3, 4, 0.2, 0.2], [6, 7, 0.2, 0.2]), \\ &\quad ([4, 13, 0.3, 0.3], [10, 12, 0.3, 0.3])\}, \end{aligned}$$

where, for instance, $([3, 4, 0.2, 0.2], [6, 7, 0.2, 0.2])$ denotes a fuzzy qualitative state of the variable b with $[3, 4, 0.2, 0.2]$ and $[6, 7, 0.2, 0.2]$ representing the qualitative magnitude and the qualitative rate-of-change of b at this state, respectively. Evaluating the constraint a *propagated* state for the variable c , $([13, 16, 0.3, 0.3], [10, 12, 0.3, 0.3])$, can be obtained since

$$\begin{aligned} [13, 16, 0.3, 0.3] &= [10, 12, 0.1, 0.1] + [3, 4, 0.2, 0.2], \\ [10, 12, 0.3, 0.3] &= [4, 5, 0.1, 0.1] + [6, 7, 0.2, 0.2]. \end{aligned}$$

Clearly, both predicted states will survive the constraint filtering as shown in figure 1 (where two dotted fuzzy numbers denote the predicted qualitative magnitude values of the variable c and the highlighted one expresses the propagated qualitative magnitude) and also reflected by the following:

$$\begin{aligned} [13.5, 16.5, 0.3, 0.3] \cap [13, 16, 0.3, 0.3] &\neq \Phi, \\ [4, 13, 0.3, 0.3] \cap [13, 16, 0.3, 0.3] &\neq \Phi, \end{aligned}$$

with both predicted rates of change being equal to the propagated one.

Although both $([13.5, 16.5, 0.3, 0.3], [10, 12, 0.3, 0.3])$ and $([4, 13, 0.3, 0.3], [10, 12, 0.3, 0.3])$ are the possible states of the variable c and maintaining them guarantees the soundness of the algorithm, the predicted state $([13.5, 16.5, 0.3, 0.3], [10, 12, 0.3, 0.3])$ of c is much closer to the propagated state and hence, intuitively,

tively, more likely to correspond to the underlying real state than $([4, 13, 0.3, 0.3], [10, 12, 0.3, 0.3])$. If this distance information is not evaluated and recorded it is lost and cannot subsequently be used to compare the possible behaviours. In this example, the predicted and propagated values of the rates of change of the variables match. This simplifies the above discussion but is not a necessary condition and will be removed in later developments.

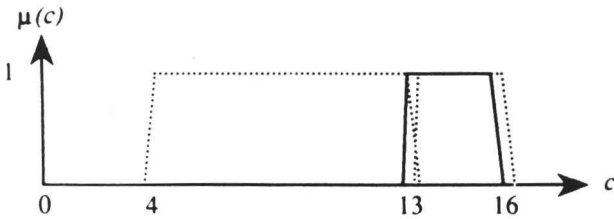


Figure 1. Intersections between Predicted and Propagated Values

2.2 Distance Metric

A discussion on the properties of an appropriate metric is given in [Leitch and Shen, 1991]. In this work, we utilise a distance metric already proposed by the authors to realise the approximation principle within the FuSim algorithm. In fact, this metric is general enough to be used for 'crisp' algorithms, however, other metric functions may also be developed and utilised in such cases. Within FuSim a distance function $d(.,.)$, used to find the difference between any two fuzzy qualitative values A and $\hat{A}$, is given as the following:

$$d(A, \hat{A}) = \frac{1}{2} [(Power(A) - Power(\hat{A}))^2 + (Centre(A) - Centre(\hat{A}))^2]^{\frac{1}{2}},$$

where, for 4-tuple parametric fuzzy numbers,

$$Power([a, b, \alpha, \beta]) = \frac{1}{2} [2(b - a) + \alpha + \beta],$$

$$Centre([a, b, \alpha, \beta]) = \frac{1}{2} [a + b].$$

For simplicity, the common coefficient, $1/2$, on the right hand side of the above two expressions can be omitted when substituting these expressions into the distance expression [Shen and Leitch, 1993].

It is important to notice that this distance function produces a distance measure between two qualitative values. However, each qualitative state within a qualitative simulation algorithm is denoted by a pair of qualitative values representing the magnitude and the rate-of-change. For two qualitative states (A, B) and $(\hat{A}, \hat{B})$, distance between the values of their qualitative magnitudes $d(A, \hat{A})$ and that between their rates of change $d(B, \hat{B})$ are usually different. To ensure a correct commitment to the possible states, the following synthesised distance $D(.,.)$, based on the function $d(.,.)$ is used to measure the difference between two states (A, B) and $(\hat{A}, \hat{B})$:

$$D((A, B), (\hat{A}, \hat{B})) = \max \{d(A, \hat{A}), d(B, \hat{B})\}.$$

It is straightforward to prove that $D(.,.)$ satisfies the three basic properties of a distance metric [Leitch and Shen, 1991].

Using such a distance function, the distances between the predicted states,

$$s_{c1} = ([13.5, 16.5, 0.3, 0.3], [10, 12, 0.3, 0.3])$$

$$s_{c2} = ([4, 13, 0.3, 0.3], [10, 12, 0.3, 0.3]),$$

and the propagated state,

$$\hat{s}_{c1} = \hat{s}_{c2} = ([13, 16, 0.3, 0.3], [10, 12, 0.3, 0.3]),$$

of the variable c in the example given in the preceding sub-section can be computed such that:

$$D(s_{c1}, \hat{s}_{c1}) = 1, \quad D(s_{c2}, \hat{s}_{c2}) = 17.$$

Based on this, it is considered that s_{c1} is much more likely to include the actual 'real' state and, therefore, a higher commitment can be assigned to it.

Importantly, the above-shown distance metric is also suitable for determining distances between (crisp) interval-valued qualitative states. This is due to the fact that if the parameters α and β within the 4-tuple representation of a fuzzy qualitative value (i.e., a fuzzy number) $[a, b, \alpha, \beta]$ are both assigned to be 0 this fuzzy number degenerates to an ordinary real interval $[a, b]$ [Shen and Leitch, 1992b].

2.3 Computational Method for State Priority

Having defined a distance metric $D(.,.)$, the priority of a state can then be assigned. A general computational method for this purpose will be presented later in this sub-section. To better understand the technique for determining state priority, let us investigate a simplest case first. Suppose that only one constraint, $z = x + y$, within a system model involves the variable z and that, after local filtering, the following N three-tuples of qualitative states remain as the possible new state combinations of the variables x, y, z :

$$((A_{xi}, B_{xi}), (A_{yi}, B_{yi})), \quad i = 1, 2, \dots, N.$$

From this, a set of propagated states $(\hat{A}_{zi}, \hat{B}_{zi})$ for the variable z can be obtained, namely,

$$\hat{A}_{zi} = A_{xi} + A_{yi}, \quad \hat{B}_{zi} = B_{xi} + B_{yi}.$$

Using the given distance metric, for each $i, i = 1, 2, \dots, N$, the distance between (A_{zi}, B_{zi}) and $(\hat{A}_{zi}, \hat{B}_{zi})$ can be calculated.

With respect to a particular quantity space, both states (A_{zi}, B_{zi}) and $(\hat{A}_{zi}, \hat{B}_{zi})$ should ideally be identical to each other, or, equivalently, $D((A_{zi}, B_{zi}), (\hat{A}_{zi}, \hat{B}_{zi}))$ should be equal to 0 (if they both are supposed to reflect the underlying unique 'real' state of the variable z). However, due to the nature of (fuzzy) interval algebra or theoretically unavoidable qualitative ambiguity, this cannot be guaranteed. This leads to a difference between the two that is directly represented by the measured distance $D((A_{zi}, B_{zi}), (\hat{A}_{zi}, \hat{B}_{zi}))$. Intuitively, the larger this distance is the less likely the state (A_{zi}, B_{zi}) will be the actual one and, hence, the lower the degree of the com-

mitment that should be assigned to this state. From this point of view, any monotonically decreasing function of the distance metric can be used to measure the degree of commitment that a particular state is associated with provided that the monotonical function employed maps 0 to full commitment. Notice that, however, generating degrees of commitment to different possible qualitative states is aimed at prioritising such states in order to perform progressive simulation from the most likely state. Thus, information embedded in the distance metric function itself is sufficient for the purpose of determining state priorities. We shall, therefore, omit the discussion about the concrete forms of the monotonical decreasing functions (which may be instantiated in an infinite number of ways) but exploit the distance metric to meet the same end.

Different tuples of predicted states associated with the variables x , y , and z usually result in different distances between the predicted states of z and their corresponding propagated ones. This forms the very basis to prioritise the predicted possible states. In fact, when distances between $s_i = (A_{zi}, B_{zi})$ and $\hat{s}_i = (\hat{A}_{zi}, \hat{B}_{zi})$, $i = 1, 2, \dots, N$, are obtained, the state priorities can then be determined such that $\rho(s_i) = \rho((A_{zi}, B_{zi})) = j$ if

$$D(s_i, \hat{s}_i) = \min\{D(s_k, \hat{s}_k) \mid k = 1, 2, \dots, N\} - \{D(s_l, \hat{s}_l) \mid l < j\}.$$

Following this rule, for example, the state with the highest priority ($\rho((A_z, B_z)) = 1$) will be the one that satisfies:

$$D((A_z, B_z), (\hat{A}_z, \hat{B}_z)) = \min\{D((A_{zk}, B_{zk}), (\hat{A}_{zk}, \hat{B}_{zk})) \mid k = 1, 2, \dots, N\}.$$

The above discussion is based on the simplified case where only one constraint within the system model involves the variable z . When there are several constraints sharing this variable the corresponding computational method will, of course, become more complex. To extend this to the general situation, it requires a clarification of two basic notions, *constrained variables* and *constraining variables*. Informally, a single variable on one side of a given constraint is termed the constrained variable while those appearing on the other side are called constraining variables. However, it should be emphasised that causal implications are not used in the generation of the behaviours but rather they are only used in the calculation of the distance metric and hence for the associated priority. Due to space limitations this is not further discussed here, but a full treatment is given in [Leitch and Shen, 1991].

Once the distance measures, of a variable, associated with each constraint have been determined the overall priority of a particular state of the variable can then be easily adjudged by ordering all the distance measures from the smallest to the largest and assigning the state attached with the smallest distance to the highest priority while that with the largest distance to the lowest priority. Summarising the discussion presented above results in a basic algorithm for computing the state priorities of system variables. In principle, for computational efficiency, the determination of state priority can be carried out while performing the constraint filtering through

the use of the Waltz filter [Kuipers, 1986; Shen and Leitch, 1993; Weld, 1990]. However, for the sake of clearer presentation, the algorithm outlined below is executed after accomplishing local filtering.

- 1) For each variable x find all constraints relating to it.
- 2) When x is the constrained variable within a constraint, find the N propagated qualitative states of the x $\{(\hat{A}_i, \hat{B}_i) \mid i = 1, 2, \dots, N\}$ by operating on the values of constraining variables; where N is the number of the tuples of the predicted possible states associated with this constraint; find the distances between (A_i, B_i) and $(\hat{A}_i, \hat{B}_i)$ respectively:

$$D((A_i, B_i), (\hat{A}_i, \hat{B}_i)) \mid i = 1, 2, \dots, N,$$

or

$$\{\max\{d(A_i, \hat{A}_i), d(B_i, \hat{B}_i)\} \mid i = 1, 2, \dots, N\};$$

then, attach each resulting distance measure to its corresponding predicted possible state of x and go to 4).

- 3) When x is a constraining variable in a constraint, for each of its predicted states (within the tuples of the possible states associated with that constraint), find the propagated state of the constrained variable (but not the x) within the constraint and then the distance between this propagated state and its respective predicted state; attach so resulting distance to the predicted possible state of the constraining variable x .
- 4) Redo 2) or 3) until all the predicted states associated with all the constraints that are related to x have been attached with distance measures.
- 5) Prioritise the states of x such that $\rho((A_i, B_i)) = j$, $i = 1, 2, \dots, M$, if

$$D_i = \min\{D_k \mid k = 1, 2, \dots, M\} - \{D_k \mid k < j\},$$

where D_i , $i = 1, 2, \dots, M$, is the distance label attached with state (A_i, B_i) and M is the total number of possible states of the variable x associated with all the constraints.

2.4 Behavioural Priority

It is natural, with each possible new state having been prioritised, to generate a system's behaviour from the state assigned with the highest priority. A useful strategy to perform effective qualitative simulation can then be described as follows. Start with a given initial state and predict the next possible states. Then, find the highest prioritised state and perform further simulation from this state. Repeat this at each simulation stage (of prediction and local filtering) unless at some stage the highest prioritised state has an attached distance label less than that attached with the second highest prioritised state at the previous simulation stages. In which case, the simulation will carry on from the predicted state labeled with the second shortest distance measure within one of the behavioural branches that were temporarily terminated earlier.

Such a strategy can be graphically illustrated in figure 2 through a simple example. Without losing generality, suppose that the given initial state is represented by s_{01} with $\rho(s_{01}) = 1$ and that three immediate successor states are predicted: s_{11} , s_{12} , and s_{13} that survive constraint filtering. If the distance measures associated with these three states are 0.6, 0.8, and 0.9 respectively, their corresponding state priorities can then be determined such that $\rho(s_{11}) = 1$, $\rho(s_{12}) = 2$, and $\rho(s_{13}) = 3$. From this, the next possible states are first generated from state s_{11} and, as shown in the figure, we have $\rho(s_{21}) = 2$, $\rho(s_{22}) = 1$, and $\rho(s_{23}) = 3$. From state s_{22} , currently the highest prioritised one, two future states, s_{31} and s_{32} are obtained. However, the distances (0.8 and 1.0) labeled with these two new states are both larger than that (0.7) attached to the second highest prioritised state s_{21} and, thus, further prediction from s_{22} is halted, although states s_{31} and s_{32} can still be prioritised between themselves. The simulation process will then carry on from state s_{21} that was temporarily stopped as the figure shows, producing the next highest prioritised state s_{32} .

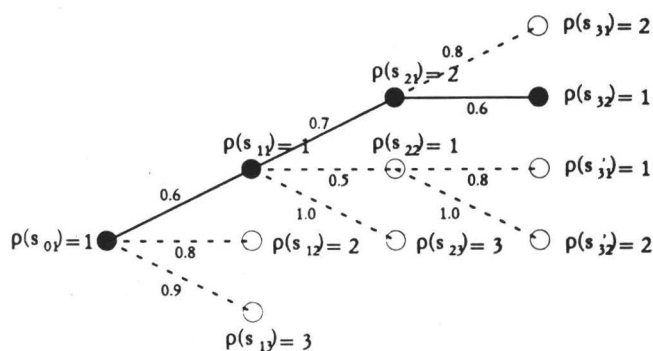


Figure 2. Progressive Simulation Strategy

Once the state priorities are determined the priority assigned to a particular behavioural branch (such as the partial behaviour consisting of states s_{01} , s_{11} , and s_{21} in the above example) can easily be adjudged. In fact, with respect to the above given simulation strategy, at each simulation stage the ordering in which the partial behaviour from the initial state to one of the present states is generated reflects the priority of this partial behaviour. Using the distance metric we can determine the behavioural priorities by the following method:

For the partial behavioural branches generated so far, B_i ($i = 1, 2, \dots, n$), each of which consists of a sequence of (predicted) states $\{s_{i0}, s_{i1}, \dots, s_{iN}\}$ and is consequently labeled with a sequence of distance measures $\{D_{i1}, D_{i2}, \dots, D_{iN}\}$, the behaviour B_i is said to have the j th priority if

$$c(B_i) = \min\{ \{c(B_k) \mid k = 1, 2, \dots, N\} - \{c(B_k) \mid k < j\} \},$$

$$\text{where } c(B_i) = \min_{k=1}^N \{D_{ik}\}.$$

This method effectively constitutes a technique for realising progressive reasoning within qualitative simulation. The simulation system will explore the behaviour with the current highest priority, thereby maximising the

chance of producing a solution within the minimum computation and hence time. If unsuccessful the next highest priority behaviour will be explored and so on until, in the limit, all possible behaviours are generated, ensuring the eventual soundness of the qualitative simulation algorithm used.

3 Application of Prioritised Behaviours

We have presented a basically simple but yet potentially very useful technique for making qualitative simulation more efficient and hence more attractive for practical applications. Indeed, in the majority of application cases, the number of simulation steps will be related to the number of behavioural distinctions supported in the quantity space. For stable systems, a qualitative behaviour represents the transient from one equilibrium to another and, hence, will normally be restricted to less than 2 times the number of the distinctions. This can, therefore, be used as a heuristic bound on the depth of search in the progressive reasoning. The potential uses of the present technique include any application tasks for which efficiency is important such as real-time critical-event simulation and real-time model-based diagnosis of dynamic systems [Dvorak and Kuipers, 1991; Shen and Leitch, 1991]. Here, due to the limit of space, we only outline a current application of this technique in model-based diagnostic systems.

In general, within real-time model-based diagnosis of dynamic systems, the use of prioritised behaviours offers several distinct advantages. For such applications qualitative simulation is used to produce predictions of the evolution of a physical plant that are synchronously compared to the evolution of the observations from the real plant to detect discrepancies [Shen and Leitch, 1991]. If no discrepancy exists then the (normal) model currently being simulated reflects the real operation condition of the plant and the plant is assumed to be working normally. However, even in this case, the computation time of the qualitative simulation is required to be fast enough so that the generation of the next simulated system state occurs faster than the evolution of the real process. For processes with relatively fast dynamics, say less than 1 second transient response, this presents a problem even for current affordable computers. Prioritising the behaviour generation allows progressive reasoning and thereby easing the time-criticality of the simulation.

When fault conditions are present, i.e. the predicted behaviour does not match the observed state, the plant model is modified until a new predicted state is generated that does match the observed state and, hence, the possible fault associated with the modified model (constraint) is identified. The key is, of course, to find a suitable method for making model adjustments. The authors have proposed a method which they call *iterative search* [Leitch and Shen, 1991] that provides a systematic way of exploring model adjustment that might result in a reduction of discrepancy between the observed and predicted behaviours. An important advantage of this is that, in principle, it does not require explicit fault models, although they may be used if available. However, this potentially results in exploring a number of models, at

different stages of prediction, as an attempt to get the best match (that minimises discrepancy metric). Even if only one model is currently being used to track against observations for detecting and/or evaluating discrepancies, the behaviour predicted by a qualitative simulation algorithm often branches because of the inherent ambiguity of qualitative representation. The behavioural comparison has therefore to be performed for all such branches when they are considered equally likely to cover the actual underlying real behaviour [Dvorak and Kuipers, 1991]. Fortunately, the use of prioritised behaviours offers the prospect of a significant reduction of the computation required to realise the iterative search method in general and the synchronous tracking [Shen and Leitch, 1991] in particular. Indeed, with prioritised behaviours, predicted states with higher priority can be compared with the observations first. Whenever a predicted behaviour is explored that results in a value of the discrepancy metric less than a prescribed threshold, the search procedure could be halted and the associated fault reported as a possible cause. Further, the degree of commitment implied by the priority can be combined with the value of the resultant discrepancy metric to produce an 'integrated' measure reflecting both the likelihood of that behaviour being the correct one (of the assumed underlying fault happening in the physical plant) and the degree of correspondence between observed and predicted values. Such possibilities seem to us to be a very exciting prospect for real-time applications of qualitative simulation.

4 Conclusion

We have proposed a simple but potentially very powerful technique for prioritising the behaviours generated by qualitative simulation algorithms. Although the presentation and associated examples utilise the fuzzy quantity space employed in FuSim [Shen and Leitch, 1993], this is not essential and, in fact, as discussed in the paper, the technique is equally applicable to other non-constructive simulation algorithms that employ semi-quantitative information. The method forms a basis for a technique for performing progressive reasoning to cope with real-time constraints. In this manner, the most likely solution is generated first and the subsequent behaviours are only generated if higher priority behaviours fail. Ultimately all possible behaviours are generated, as in current algorithms, and the soundness of qualitative predictions is guaranteed. The technique has relevance to any application where computational resources are important.

References

- [Berleant and Kuipers, 1992] D. Berleant and B. Kuipers. Qualitative-numerical simulation with Q3. In B. Faltings and P. Struss (Eds.), *Recent Advances in Qualitative Physics*, pp. 3-16, 1992.
- [D'Ambrosio, 1987] B. D. D'Ambrosio. Extending the mathematics in qualitative process. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pp. 595-599, 1987.
- [Dubois and Prade, 1989] D. Dubois and H. Prade. Order-of-magnitude reasoning with fuzzy relations. In *Proceedings of IFAC Symposium on Advanced Information Processing in Automatic Control*, Nancy, 1989.
- [Dvorak and Kuipers, 1991] D. Dvorak and B. Kuipers. Process monitoring and diagnosis: a model-based approach. *IEEE EXPERT*, 6(3):67-74, 1991.
- [Fouche and Kuipers, 1992] P. Fouche and B. Kuipers. Reasoning about energy in qualitative simulation. *IEEE Transactions on Systems, Man and Cybernetics*, 22(1):47-63, 1992.
- [Kuipers, 1986] B. J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289-338, 1986.
- [Lee and Kuipers, 1988] W. W. Lee and B. Kuipers. Non-intersection of trajectories in qualitative phase space: a global constraint for qualitative simulation. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pp. 286-290, 1988.
- [Leitch, 1987] R. R. Leitch. The modelling of complex dynamic systems. *IEE Proceedings, Part D*, 134:245-250, 1987.
- [Leitch and Shen, 1991] R. R. Leitch and Q. Shen. Finding faults with model based diagnosis. In *Proceedings of the 2nd International Workshop on Principles of Diagnosis*, pp. 121-134, 1991.
- [Leitch and Shen, 1992] R. R. Leitch and Q. Shen. Being committed to qualitative simulation. In *Proceedings of the 6th International Workshop on Qualitative Reasoning about Physical Systems*, pp. 281-293, 1992.
- [Shen and Leitch, 1991] Q. Shen and R. R. Leitch. Synchronised qualitative simulation in diagnosis. In *Proceedings of the 5th International Workshop on Qualitative Reasoning about Physical Systems*, pp. 171-185, 1991.
- [Shen and Leitch, 1992a] Q. Shen and R. R. Leitch. Application studies of fuzzy qualitative simulation. In P. Borne and S. G. Tzafestas (Eds.), *Mathematics of the Analysis and Design of Process Control*, 1992.
- [Shen and Leitch, 1992b] Q. Shen and R. R. Leitch. On extending the quantity space in qualitative reasoning. *Artificial Intelligence in Engineering*, 7(3):167-173, 1992.
- [Shen and Leitch, 1993] Q. Shen and R. R. Leitch. Fuzzy qualitative simulation. *IEEE Transactions on Systems, Man and Cybernetics*, 23(4), 1993.
- [Struss, 1988a] P. Struss. Mathematical aspects of qualitative reasoning. *Artificial Intelligence in Engineering*, 3(3):156-169, 1988.
- [Struss, 1988b] P. Struss. Global filters for qualitative behaviours. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pp. 275-279, 1988.
- [Struss, 1991] P. Struss. A theory of model simplification and abstraction for diagnosis. In *Proceedings of the 5th International Workshop on Qualitative Reasoning about Physical Systems*, pp. 25-57, 1991.
- [Weld, 1990] D. Weld. Exaggeration. *Artificial Intelligence*, 43:311-368, 1990.
- [Weld and de Kleer, 1990] D. Weld and J. de Kleer, editors. *Readings in Qualitative Reasoning about Physical Systems*, 1990.