Exaggeration

Daniel S. Weld
Computer Science Department, FR-35
University of Washington
Seattle, WA 98195

Abstract

Exaggeration is a technique for solving comparative analysis problems by considering extreme perturbations to a system. For example, exaggeration answers the question "What happens to the output temperature of a heat exchanger if fluid flow rate increases?" by simulating the behavior of an exchanger with infinite flow rate. This paper explains the three phases of the exaggeration algorithm: transform, simulate, and scale. The transform phase takes a comparative analysis problem and generates the description of an exaggerated system. The simulate phase predicts the behavior of the transformed system. Finally, the scale phase compares the original and exaggerated behaviors to answer the original comparative analysis question.

1 Introduction

The symbolic analysis of real-world systems is central to many problems in artificial intelligence. In order to cope with a changing world one must be able to understand its behavior. Many types of analytic activities have been investigated, for example qualitative simulation [de Kleer and Brown, 1984; Forbus, 1984; Williams, 1984; Kuipers, 1986], measurement interpretation [Forbus, 1983], and diagnosis [Davis, 1984; de Kleer and Williams, 1987]. Recently, a new qualitative reasoning task has been isolated: comparative analysis [Forbus, 1984; Weld, 1988b]. Whereas qualitative simulation takes a description of a system and predicts its behavior, comparative analysis takes as input this behavior, the original description, and a perturbation, then describes how and why the perturbation changes the behavior.

For example, given a description of heat exchanger in which hot oil passes through a pipe surrounded by cold water, a qualitative simulator would say that the oil will exit from the pipe cooler than when it entered. Comparative analysis, on the other hand, takes this description of cooling and evaluates the effects of perturbations. For example, it might be asked to deduce what would happen to the oil output temperature if the oil moved more quickly through the heat exchanger.

Previous discussions of comparative analysis have dealt with a solution method called differential qualitative (DQ) analysis [Forbus, 1984; Weld, 1988b]. This paper introduces exaggeration, a technique which solves a larger class of comparative analysis problems than DQ analysis [Weld, 1988a]. For example, DQ analysis generates the following answer to the heat exchanger question:

> Since the rate of cooling is dependent only on the initial temperature and thermal conductivity and these are unchanged, the rate of cooling is unchanged as a function of time. Since the oil will spend less time in the pipe, it will exit with a higher temperature.

Exaggeration's approach to comparative analysis is very different from that of DQ analysis. Instead of tracing the effect of a perturbation through the causal structure of the system, exaggeration considers the behavior of a system in which the perturbation is taken to a limiting value. If this new system has a *qualitatively different* behavior from the original, then exaggeration postulates a general trend caused by the perturbation. Exaggeration produces the following explanation:

> If the fluid flow rate was infinite, the oil would spend negligible time in the exchanger. Since the rate of cooling is finite, the oil would lose negligible heat and exit hotter than oil moving at finite speed. Thus any increase in oil flow rate will cause a corresponding increase in output temperature.

Exaggeration changes a comparative analysis question into a simulation problem about a system with infinite or infinitesimal valued parameters. Figure 1 provides an overview of the program, EXAG, that implements the theory of exaggeration in three parts. Given a perturbation and a description of the system including initial values, the TRANSFORM PHASE produces a new model in which the perturbation has been taken to an extreme. The SIMULATE PHASE (denoted HR-QSIM in the figure) simulates this exaggerated model to produce an exaggerated behavior that is qualitatively different from the behavior QSIM [Kuipers, 1986] produces using the original model: in one case the heat has dropped a finite amount, in the other it has fallen negligibly. Finally, the SCALE PHASE compares the two behaviors and predicts the answer to the original comparative analysis question.

Although exaggeration handles a larger class of comparative analysis questions than DQ analysis, it does not

*This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's Artificial Intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124. Portions of this research were done at Xerox Palo Alto Research Center and at Schlumberger CAS Palo Alto Research.

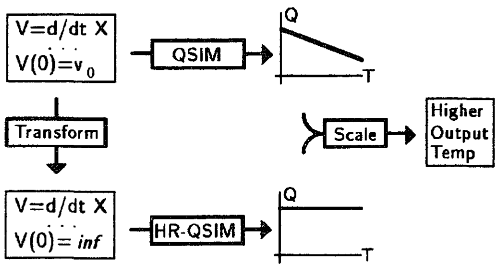

Figure 1: Overview of the Exaggeration Algorithm

always answer them correctly. If the system does not respond monotonically to the perturbation, then exaggeration may generate false predictions. For a comparison of the two techniques, including an explanation of exaggeration's limitations, see [Weld, 1988a]. This paper explains the details of the exaggeration algorithm; the transform, simulate, and scale phases are discussed in turn with an emphasis on the HR-QSIM implementation of the simulate phase. In particular, HR-QSIM is critically dependent on two temporal reasoning innovations: predecessor-persistence and successor-arrival filtering.

2 Transform Phase

The transform phase converts a comparative analysis problem into a simulation problem by creating a model of the system that has an exaggerated initial value for some parameter. The trick is to produce a description which, when simulated, generates a behavior qualitatively different than the original's. To do this, the parametric perturbation in the comparative analysis question is amplified: an increasing perturbation is transformed into an infinite initial value while a decreasing change results in an infinitesimal value.

The critical requirement is a qualitative representation that can express infinitesimal and infinite values. The QUALITATIVE HYPERREAL REPRESENTATION [Weld, 1988c] meets the requirement by extending Kuipers' QSIM quantity space using the hyperreal numbers of nonstandard analysis [Robinson, 1966; Keisler, 1976]. As in QSIM, parameters are continuous functions from time into a value space, but both time and the value spaces are abstractions of the hyperreal numbers. In this extended representation, the qualitative value of a parameter has two parts. The HR-QVAL encodes magnitude information, and the HR-QDIR abstracts the parameter's derivative. Suppose a parameter P has landmark values $p_0 < \ldots < p_k$. For any time t, the following HR-QVAL's are possible:

inf,	if $P(t)$ is infinite, and > 0; *minf* if < 0
p_i	if $P(t) =$ landmark p_i
(HALO p_i +)	if $P(t) - p_i$ is infinitesimal and > 0
(HALO p_i −)	if $P(t) - p_i$ is infinitesimal and < 0
$\prec p_i, p_{i+1} \succ$	if $P(t) - p_i$ and $p_{i+1} - P(t)$ are both non-infinitesimal > 0
$\prec p_k, inf \succ$	if $P(t)$ is finite and $P(t) - p_k$ is non-infinitesimal > 0

Every finite landmark, p, has a halo of numbers that are infinitesimally close; the two halves of these halos are denoted (HALO p +) and (HALO p −) respectively. The positive infinitesimals, for example, are represented (HALO 0 +). The QSIM expression for an open interval, (p_1, p_2), is not used since it overlaps with (HALO p_1 +) and (HALO p_2 −). This explains the definition of $\prec p_1, p_2 \succ$.

It also proves useful to extend the representation of qualitative derivatives. QSIM uses a simple description of the sign of the parameter's derivative: *inc*, *dec*, or *std*. The qualitative hyperreal representation supplements this representation with information on the order of magnitude of growth. A hyperreal number, x, has four possible orders of magnitude:

inf	if $\lvert x \rvert >$ every finite number
fin	if $\lvert x \rvert =$ a positive standard real number
negl	if $\lvert x \rvert =$ negligible, i.e. a positive infinitesimal
0	if $x = 0$

Qualitative derivatives are represented as a pair of the direction and order of magnitude of change. Thus (*dec inf*) denotes the HR-QDIR of a parameter that is decreasing infinitely fast. If a parameter's HR-QDIR is (*std* 0), then it may be abbreviated *std* since 0 is the only possible order of magnitude of *std*.

Thus a parameter P may be qualitatively described at a point of time, t, by its HR-QVAL and HR-QDIR; square brackets denote this abstraction:

$$[P(t)] \equiv (\text{HR-QVAL}(P(t)), \text{HR-QDIR}(P(t)))$$

If the same qualitative description is valid for an interval, $\mathcal{A}$, of time, then it can be written $[P(\mathcal{A})]$.

As described in [Weld, 1988c], the transform phase uses this representation to describe an exaggerated system. Suppose that the original heat exchanger is described in terms of two independent parameters, thermal conductivity K, and fluid velocity through the pipe V (both assumed constant), and three dependent parameters: heat Q, heat flow F, and position of a unit volume of oil[1] X. The parameters obey the following constraints: $V = \frac{d}{dt}X$, $F = \frac{d}{dt}Q$, and $F = KQ$. The transform phase modifies the initial conditions to produce a description of a heat exchanger with infinite flow rate (x_0, k_0, and f_0 are standard, finite negative landmark values, but q_0 is positive.):

$$\begin{aligned}
[V(0)] &= (\mathit{inf},\ \mathit{std}) \\
[X(0)] &= (x_0,\ (\mathit{inc\ inf})) \\
[K(0)] &= (k_0,\ \mathit{std}) \\
[Q(0)] &= (q_0,\ (\mathit{dec\ fin})) \\
[F(0)] &= (f_0,\ (\mathit{inc\ fin}))
\end{aligned}$$

3 Simulate Phase

Since the advantage of exaggeration is that it reduces a comparative analysis problem to a problem of qualitatively simulating a transformed system, it should be no surprise that the simulate phase is the most difficult of the three. The trick is to demonstrate a qualitative simulation technique which can handle parameters with infinite and infinitesimal values. Because Kuipers' QSIM [Kuipers, 1986]

[1] For simplicity, the 'liquid-individual' model of fluids is used here. In addition this model does not distinguish between temperature and heat.

is simple, precisely defined and widely available, I chose it as basis for the simulate phase.

The addition of infinite and infinitesimal values requires a number of modifications. The fundamental problem is the strong reliance that all qualitative simulation algorithms place on the order topology of the standard real numbers [Williams, 1984]; QSIM, for example, assumes that the value spaces of time and the various parameters alternate between open intervals and closed points. The presence of infinitesimals in the hyperreals results in a more complex topology where this is no longer the case.

I call my implementation of the simulate phase HR-QSIM, to acknowledge its ancestry. The next section explains its overall control. Then I present two of HR-QSIM's most interesting technical innovations: the predecessor-persistence filter and the successor-arrival filter.

3.1 HR-QSIM Control

HR-QSIM has essentially the same control structure as QSIM. They take as input a set of parameters, a set of constraints, and a set of initial qualitative values. As output, they produce a tree of states; each path through the tree represents a possible behavior of the system. To generate a state's successors, they use continuity information to predict the possible next values of each parameter independently. Conceptually, the space of possible successor state values is the cross product of the parameter values. Waltz filtering efficiently prunes this space of states without explicitly representing it. After Waltz filtering, the states are constructed to represent the remaining tuples of parameter values. Global filters may prune some of these states; the rest are marked as successors to the original state and pushed on the control queue. Space considerations preclude treatment of the many difference between QSIM and HR-QSIM; see [Weld, 1988c] for a discussion of additional next-value tables used to generate parameter values, and of extended constraint filters used in Waltz filtering. Instead the next sections focus on two global filters based on predecessor-persistence and successor-arrival times.

3.2 Persistence and Arrival Times

QSIM's temporal representation is simple; states persist for either an instant (a closed point of time) or a finite open interval. Furthermore, QSIM can easily tell how long any state will last; if the predecessor state lasted for an instant, the successor will persist for an interval and vice versa. For HR-QSIM the qualitative hyperreal representation allows derivatives to have a negligible order of magnitude so a state might last for an infinite time before a parameter transitions to a new landmark value. If some parameter has an *inf* derivative, then the state might persist for only a negligible time. Since the original QSIM cases are also still possible, I distinguish between the following four qualitative lengths of time: 0, *negl*, *fin*, and *inf*. HR-QSIM uses two techniques, predecessor-persistence filtering and successor-arrival filtering (section 3.4), to deduce the temporal extent of qualitative states and to prune inconsistent successors.

The difference between the two techniques results from the following observation about transitions in the qualitative hyperreal representation:

> It may take longer for a parameter to transition to a new qualitative value than it spends in its old value.

Lest this sound confusing, consider the following concrete example. Let I be a parameter, in other words a function from the hyperreals to the hyperreals, defined as the identity function $I(t) = t$. Consider the length of the interval, $\mathcal{A}$, in which $[I(\mathcal{A})] = ((halo\ 0\ +), (inc\ fin))$, termed the PERSISTENCE of the qualitative value [Weld, 1988c]. I claim that I persists in $(halo\ 0\ +)$ for *negl* time. For example, if I persisted in the halo for a standard finite time, t_0, then that would imply that $t_0 \in (\text{HALO}\ \ 0+)$, in other words that t_0 is an infinitesimal. Since 0 and *inf* persistences also lead to contradictions, I persists in $(\text{HALO}\ \ 0\ +)$ for *negl* time.

Now consider the time it takes for I to reach the qualitative value $\prec 0, inf \succ$ (formalized as SUCCESSOR-ARRIVAL TIME [Weld, 1988c]). I argue that I's successor-arrival time is *fin*. By definition of $\prec 0, inf \succ$, when I reaches this qualitative value it must be greater than some standard real value, r_0. Thus r_0 time must have elapsed since I left 0. Since only *negl* time passed reaching $(\text{HALO}\ \ 0\ +)$ from 0 [Weld, 1988c], I takes $fin - negl = fin$ time to arrive at its new qualitative value. In other words, even though there is no intervening hyperreal value sandwiched between $(\text{HALO}\ \ 0\ +)$ and $\prec 0, inf \succ$, I takes longer to reach its new qualitative value than it spends in its original value.

Several benefits result from considering persistence and arrival measures separately. The unintuitive topology of the hyperreals is made clear, exposing the relationship between the time when one value ends and another starts. The result is a powerful algorithm for temporal reasoning in qualitative hyperreal simulation. Section 3.3 discusses the filtering of successor states based on persistence times while section 3.4 deals with the successor-arrival filter.

Both techniques use a common mechanism, the DISTANCE-RATE-TIME TABLE (figure 2) to compute temporal values. This table is indexed by rate and distance values and returns the time required to traverse the distance. In both cases, the rate values come directly from the parameter's qualitative derivative. The difference between persistence and arrival times comes from the distance used to index into the table. To calculate the time a parameter can persist in a qualitative value, the 'width' of the value is used as a table index.

Formally, the width of a qualitative value is the order of magnitude of the maximum distance between any two members of the set of hyperreal points that underlie the qualitative value [Weld, 1988c]. From this definition, the following characteristics can be derived. The width of a landmark point is 0, the width of a landmark's halo is *negl*, the width of a finite interval (e.g., $\prec p_i, p_{i+1} \succ$ or $\prec p_j, inf \succ$) is *fin*, and the width of *inf* or *minf* is *inf*. By using these width values as an index to the distance-rate-time table, HR-QSIM calculates how long each parameter can persist in its current qualitative value. An entry of '?' in the table indicates that *inf*, *fin*, *negl*, or 0 time may elapse

3.3 Predecessor-Persistence Filtering

HR-QSIM calculates persistence values for two reasons. From the persistences of each parameter, one can determine how long a qualitative state is a valid description of

		Distance			
		inf	*fin*	*negl*	0
Rate	*inf*	?	*negl*	*negl*	0
	fin	*inf*	*fin*	*negl*	0
	negl	*inf*	*inf*	?	0

Figure 2: The Distance-Rate-Time Table

a system. Secondly, by comparing the persistences of all the parameters in a system, one can often filter inconsistent transitions that were not eliminated by HR-QSIM's other techniques.

For example, suppose the two parameters, A and B, are both increasing at the same *fin* rate, and this rate is held constant. In state S_i, $A = (0, (inc\ fin))$ and $B = ((\text{HALO}\ \ 0\ -), (inc\ fin))$. After Waltz filtering, three sets of possible next values remain. Either A leaves 0 before B reaches 0, or B reaches 0 before A leaves 0, or they both transition at the same time. The question is, which of these successor states is possible? The answer comes from analyzing the persistence of the predecessor state, S_i. The width of A's qualitative value is 0 and A is moving with *fin* speed, thus the distance-rate-time table lists A's persistence as 0. B has the same rate and has *negl* width, so B's persistence is *negl*. This means that B must persist in its qualitative value for longer than A. In other words, A must transition before B.

3.4 Successor-Arrival Filtering

Like persistence values, arrival times are useful as a means for eliminating inconsistent transitions. Calculating the time that a parameter takes to arrive at a new qualitative value from an old one requires a notion of the distance between the two different values.

The distance between two qualitative values is defined as the order of magnitude of the minimum distance between any two points in the hyperreal sets underlying the two qualitative values [Weld, 1988c]. For example, the distance between a landmark and its halo is *negl*, the distance between a halo and a neighboring finite interval is *fin*, and the distance between *inf* and any different value is *inf*.

The predecessor-persistence and successor-arrival filters are implemented together. The inputs are parameter values for the predecessor and proposed successor states. Two variables, SP and SA, store successive approximations to the state's persistence and arrival values respectively. SP is initialized to the set $\{0, negl, fin, inf\}$, and SA to $\{negl, fin, inf\}$. For each parameter, X, let P be the set of possible persistence values, and let A be the set of possible arrival values. If X transitions to a new qualitative value in the successor state, set SP to SP intersect P and let SA be SA intersect A. Otherwise, if X has the same qualitative value in the predecessor and successor states, remove any time values from SP that are greater than the largest value in P, and remove any time values from SA that are greater than the largest value in P, not A! Since this parameter is not changing, the next state must arrive while this parameter is still persisting. If SP or SA is empty, the successor state is inconsistent, otherwise SP and SA are the sets of possible persistences and arrivals respectively.

3.5 Heat Exchanger Example

Successor-arrival filtering is nicely illustrated by the heat exchanger. The initial state generated by the transform phase persists for 0 time because several parameters are moving from landmarks. Waltz filtering generates a single successor state which arrives in *negl* time and has new values for X, Q, and F:

$$\begin{aligned}
[X(\mathcal{A}_1)] &= ((\text{HALO}\ x_0+), (inc\ inf)) \\
[Q(\mathcal{A}_1)] &= ((\text{HALO}\ q_0-), (dec\ fin)) \\
[F(\mathcal{A}_1)] &= ((\text{HALO}\ f_0+), (inc\ fin))
\end{aligned}$$

Unfortunately, Waltz filtering does not predict a unique successor to this state. The question is whether X will transition from its halo before, after or at the same time as Q and F transition from their halo. Since each parameter is in a halo, each has a qualitative width of *negl*, and since each is moving towards a finite interval, each parameter must travel a *fin* distance before transitioning. Plugging these values into the distance-rate-time table leads to the conclusion that every parameter persists for *negl* time, so $\mathcal{A}_1$ represents a time interval of *negl* length. In addition, X takes *negl* to arrive, but Q and F take *fin* to arrive. Successor-arrival filtering uses these values to eliminate the two successor states that don't have X transitioning before Q and F. The only set of next values which pass the test are the following; they arrive in *negl* time.

$$\begin{aligned}
[X(\mathcal{A}_2)] &= (\prec x_0, 0 \succ, (inc\ inf)) \\
[Q(\mathcal{A}_2)] &= ((\text{HALO}\ q_0-), (dec\ fin)) \\
[F(\mathcal{A}_2)] &= ((\text{HALO}\ f_0+), (inc\ fin))
\end{aligned}$$

Since the distance to X's next value is still *fin*, similar reasoning holds again. $\mathcal{A}_2$ has *negl* length; next X transitions to $(\text{HALO}\ \ 0\ -)$ and then to 0 (always arriving in *negl* time) while Q and F remain in the halo of their original values. Without successor-arrival filtering, HR-QSIM could not be sure that negligible heat is lost when oil moves infinitely fast.

4 Scale Phase

The scale phase answers comparative analysis questions by comparing a standard QSIM behavior of the original system with the hyperreal behavior generated by HR-QSIM from the transformed initial conditions. For example, QSIM generates three possible behaviors for the heat exchanger: in one, thermal equilibrium ($Q = 0$) occurs before the oil leaves the pipe ($X = 0$), in one X transitions to 0 before Q reaches 0, and in the third they transition at the same time. Since Q drops a finite amount in all these standard behaviors but stays at $(\text{HALO}\ \ q_0\ -)$ in the hyperreal simulation, the scale phase concludes that in general, output heat rises as oil velocity increases.

Although this is a correct answer for this problem, the scale phase can draw false conclusions. Since exaggeration approximates the sign of a partial derivative (e.g., $\frac{\partial Q}{\partial V}$ has positive sign) by evaluating at an infinite or infinitesimal asymptote and scaling, it may answer incorrectly if the system does not respond monotonically [Weld, 1988a].

5 Related Work

Like Raiman's FOG system [Raiman, 1986], HR-QSIM's qualitative hyperreal representation is grounded in the theory of nonstandard analysis [Robinson, 1966]. Unlike FOG, which only handles algebraic equations, HR-QSIM can simulate the time behavior of differential equations.

Davis' CHEPACHET program [Davis, 1987] is very similar to HR-QSIM. In fact, HR-QSIM's four next-value tables [Weld, 1988c] are derived from CHEPACHET's temporal topology rule. However, CHEPACHET's qualitative representation is less expressive than the qualitative hyperreal representation. For example, CHEPACHET cannot distinguish between q_0, (HALO q_0 $-$), and $\prec 0, q_0 \succ$ — each value is MEDIUM. Of course, the values could be distinguished by introducing another parameter called HEAT-LOST, but how would the transform phase know when to do this? Until this question is addressed, exaggeration can solve more comparative analysis problems using HR-QSIM as the simulation phase.

DQ analysis [Weld, 1988b] also solves comparative analysis problems. Unlike exaggeration, DQ analysis only predicts correct answers [Weld, 1988b] to comparative analysis questions. However, exaggeration appears to solve more problems than DQ analysis [Weld, 1988a] and often generates simpler explanations [Weld, 1988c].

ACKNOWLEDGMENTS

Tomas Lozano-Perez and Johan de Kleer advised and made this research possible. Ernie Davis contributed technically in many ways. Mark Shirley straightened me out when confused and rescued me when imperiled. Ben Kuipers provided QSIM. Brian Williams, Jerry Roylance, David Jacobs, and Walter Hamscher read early drafts. Jeff Shrager, Paul Horwitz, Pat Hayes, Ken Forbus, Randy Davis and Steve Bagley provided interesting ideas.

References

[Davis, 1987] E. Davis. *Order of Magnitude Reasoning in Qualitative Differential Equations.* Technical Report 312, NYU Computer Science Department, August 1987.

[Davis, 1984] R. Davis. Diagnostic Reasoning Based on Structure and Behavior. *Aritificial Intelligence*, 24, 1984.

[de Kleer and Brown, 1984] J. de Kleer and J. Brown. A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24, December 1984.

[de Kleer and Williams, 1987] J. de Kleer and B. Williams. Diagnosing Multiple Faults. *Artificial Intelligence*, 32, April 1987.

[Forbus, 1983] K. Forbus. Measurement interpretation in qualitative process theory. In *Proceedings of the Eighth IJCAI*, pages 315–320, 1983.

[Forbus, 1984] K. Forbus. Qualitative Process Theory. *Artificial Intelligence*, 24, December 1984.

[Keisler, 1976] J. Keisler. *Foundations of Infinitessimal Calculus.* Prindle, Webber and Schmidt, Inc., Boston, 1976.

[Kuipers, 1986] B. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29, September 1986.

[Raiman, 1986] O. Raiman. Order of Magnitude Reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, August 1986.

[Robinson, 1966] A. Robinson. *Non-Standard Analysis.* North-Holland Publishing Company, Amsterdam, 1966.

[Weld, 1988a] D. Weld. Choices of Comparative Analysis: DQ Analysis or Exaggeration? *International Journal for Artificial Intelligence in Engineering*, To Appear 1988.

[Weld, 1988b] D. Weld. Comparative Analysis. *Artificial Intelligence*, To Appear 1988.

[Weld, 1988c] D. Weld. *Theories of Comparative Analysis.* AI-TR-1035, MIT AI Lab, May 1988.

[Williams, 1984] B. Williams. Qualitative Analysis of MOS Circuits. *Artificial Intelligence*, December 1984.