

When Oscillators Stop Oscillating

Philippe Dague, Olivier Jehl
Centre Scientifique IBM
3-5 Place Vendôme
75001 Paris - France

Philippe Devès, Pierre Luciani, Patrick Taillibert
Dassault Electronique
55 quai Marcel Dassault
92214 St Cloud - France

Abstract

Diagnosing analog systems, i.e. systems for which physical quantities vary over time in a continuous range is, in itself, a difficult problem. Analog electronic circuits, especially those with feedback loops, raise new difficulties. Oscillators, a special case of such circuits, are even more difficult to diagnose. This is because most faults lead to the same type of symptom: the oscillator just stops oscillating, the different electric signals remaining hopelessly flat!

This paper shows how a program implementing a model-based diagnosis theory can be used to diagnose oscillators. We first present CATS, an ATMS-like domain-independent conflict recognition engine, adapted for diagnosing continuous time-varying physical systems. Its original feature is that it works on data that are arrays of numeric intervals so as to deal with values that change with time and are inaccurate.

Then we show how to use this program for diagnosing oscillators, pointing out the advantage of using external stimulation. Experimental results are given through the example of an astable multivibrator.

1 Introduction

In addition to pure analog circuits, the increasing usage of analog/digital hybrid circuits and the recognition of the importance of "analog" faults in digital circuits makes analog fault diagnosis ever more important. Unfortunately this raises many specific problems:

- There is a continuum of possible failures because of the continuous character of the parameters of such circuits, whereas for digital circuits a large percentage of faults can be modeled by stuck-at-one faults and stuck-at-zero faults.

- Both measurements and component parameters are inaccurate.

- Not even 50% of the measurements can be taken (voltages but not currents).

- A faulty component can provoke other components to shift well away from their nominal behavior and yet be correct.

- Complex feedback structures are encountered. The effects of a fault propagate around loops, making it impossible to use conventional diagnostic techniques which try to track fault effect backward to a single component. Breaking the loop (as is often proposed in repair manuals [Loveday, 1989]) is impossible in a large number of cases since, in addition to the practical problems that would arise, it would require the elaboration of a replacement signal so as to avoid pushing the circuit towards an unacceptable state of behavior.

- Multiple faults are not a rare occurrence, since the failure of one component can have a knock-on effect on others.

The consequence of these difficulties is that the kind of "brute force" fault simulation classically used for diagnosing digital circuits is unrealistic for analog circuits. Only partial solutions have been obtained in this way [Liu, 1987].

Some of these difficulties are naturally overcome by using the general model-based diagnostic framework [de Kleer and Williams, 1987, Reiter, 1987], which allows diagnosis based exclusively on the description of the structure of the circuit, the models of correct behavior of its components (possibly extended to include faulty behavior) along with the assumptions governing the validity of these models and, lastly, the observation (measurements). The central task of diagnosis is thus to detect discrepancies between predicted behavior (from models) and observation and to build the corresponding minimal conflicts, i.e. minimal sets of assumptions which support a discrepancy. This task is classically carried out by coupling a predictive engine with an ATMS [de Kleer, 1986]. The diagnoses, i.e.

all sets of component behavioral modes that are consistent with current measurements, are then computed from the conflicts.

However, this approach does not tackle certain difficulties specific to analog circuits, i.e. modeling, and inaccuracy and time management. This explains why most of the model-based diagnostic systems (GDE [de Kleer and Williams, 1987], SHERLOCK [de Kleer and Williams, 1989], GDE+ [Struss and Dressler, 1989], XDE [Hamscher, 1990]) are in fact applied only to digital circuits. Only a few systems have been designed for analog circuits (SOPHIE [Brown et al., 1982], DEDALE [Dague et al., 1987], ELECTRA [Sierzega and Rastogi, 1990]), but they propose only partial solutions.

In [Dague et al., 1990a], we have recently presented a system for applying model-based diagnosis to analog circuits and begun to experiment them. Continuing with this investigation we show in section 3 of this paper that our solutions also apply to the special case of oscillators (circuits without input and containing a positive feedback loop) in spite of specific difficulties. Through experiments we show that the use of external stimulation is a good way to reduce the ambiguity of the diagnoses obtained by our program. But beforehand, in section 2, we describe the domain-independent conflict recognition engine (CATS) used by our program. CATS has been specially designed to reason on arrays of intervals representing continuous time-varying signals. This algorithm allows the coupling of an extended ATMS incorporating the interval semantics with any given interval computation primitive, in such a way as to focus on the detection of nogoods. It also reduces the cost of label propagation by recording only one label for an array of intervals.

2 Conflict Recognition

CATS is a domain-independent conflict recognition engine for diagnosing continuous dynamic physical systems. It receives as input both constraints under assumptions coming from the structure and the models, and values coming from measurements; it produces as output all the minimal inconsistent sets of assumptions (bearing in mind that this completeness is defined w.r.t. a given modeling and numeric propagation, both incomplete in essence).

2.1 Representation of Physical Quantities

We have to deal both with physical quantities, the values of which change over time in a continuous range and are inaccurate, and with the unavoidable imprecision of modeling. In DEDALE [Dague et al., 1987] we used qualitative modeling in terms of relative orders of magnitude, since only direct voltages were

measured and thus all the quantities were constant over time. Consequently, only failures observable in direct mode could be diagnosed. We are now interested in circuits where the important region of behavior of the components is when they behave dynamically. A qualitative scale cannot be used to detect fast variations and thus to detect subtle discrepancies (it often happens that discrepancies occur over a very short period of time). It is the reason why we have decided to represent inaccuracy upon values (of either a quantity at a given instant or a parameter of a model) by an interval, which is computed by taking into account the precision of the measuring device or of the modeling.

We have rejected the idea of using an interval representation of time and of expressing quantities in terms of their concise histories made up of episodes, i.e. maximal time intervals during which the (interval) value of the quantity remains constant. Although this is used by temporal propagators such as TCP [Williams, 1986] and TINT [Hamscher, 1989] for discretely modeled systems (e.g. XDE uses TINT for troubleshooting digital circuits), it seems difficult to transpose to continuous systems. Firstly, since it is rare to obtain the same interval value at two different instants, defining episodes is not simple. Secondly, even if such a representation could be obtained, it would rapidly become useless since computations on values are performed to predict others, using models. Indeed, during a given time interval of observation, the regions where different quantities vary very quickly (corresponding to very small episodes) or vary very slowly (corresponding to large episodes, which are interesting from the point of view of conciseness of information) would be different from one quantity to another. Simply adding two quantities might lead to a new partition in very small episodes. What we would end up with is a kind of fine sampling for a lot of quantities, except that the samples would not coincide!

We have thus decided, in contrast with [Pan, 1984, Ng, 1990], to rely upon a proper sampling procedure. Every value of a quantity is represented by an array of numeric intervals, where each interval expresses the value at the corresponding sample. It is accompanied by a triplet defining the start instant, the sampling increment and the number of samples. During a sequential measurement process the sampling increment and the number of samples are generally kept constant and it is assumed that measurements can be repeated in such a way that start instants can be synchronized: comparison of values or computation on values is thus possible.

The most general form for an input of CATS is: $x_1, \dots, x_k \Rightarrow x$ where each antecedent x_i of the left-hand side conjunction (possibly empty) and the

consequent x are either propositional assumptions or constraints (equalities or inequalities) on given quantities. For example, the model of a capacitor c with $10 \mu F \pm 10\%$ capacitance and accesses a and b is:

correct(c) $\Rightarrow$

$i_{c_a} = [9.E-6, 11.E-6] \times \text{derivative}(v_{c_a_b})$

Part of the model of a transistor t with a gain ranging from 50 to 200 is:

correct(t) $\Rightarrow i_{t_c} \geq -1.E-4$

correct(t), $v_{t_b_e} \geq 0.6$, $v_{t_c_e} \geq 0.3$

$\Rightarrow i_{t_c} = [50, 200] \times i_{t_b}$

The structural constraints which declare that access a of component c is linked to access k of node n are:

$v_{c_a} = v_{n_k}$, $i_{c_a} = -i_{n_k}$

A value resulting from a measurement is:

$v_{r1_a_b} = ([2, 2.2], [2.5, 2.8], [2.9, 3.4])$

All this input may be given incrementally.

2.2 Interval Propagation and Conflict Recognition

For reasons of clarity we consider here the problems that arise with continuous quiescent quantities whose values are intervals; temporal aspects will be examined later.

2.2.1 Semantics of the Data

A datum is a pair (q, i) where q is the name of a quantity and i a numeric interval, meaning that the value of q belongs to i . The set of all computed values of a given quantity q is thus partially ordered by the inclusion of intervals.

The semantics of intervals is expressed by the three following first-order interval rules, which are needed to fire propagation and detect discrepancies. For example, given 3 data $(q_1, [0, 4])$, $(q_1, [3, 6])$, $(q_2, [4, 8])$ and one input rule provided by a constraint $(q_1, [2, 5]) \Rightarrow (q_2, [1, 3])$, we have to use successively:

- the **intersection** rule $(Q, I_1), (Q, I_2) \mapsto (Q, I_1 \cap I_2)$ which builds the new datum $(q_1, [3, 4])$.

- the **inclusion** rule

$(Q, I_1), (Q, I_2), I_1 \subset I_2 \mapsto ((Q, I_1) \Rightarrow (Q, I_2))$ which

records the justification $(q_1, [3, 4]) \Rightarrow (q_1, [2, 5])$, needed to fire the input rule in order to deduce the datum $(q_2, [1, 3])$.

- the **falsity** rule $(Q, \emptyset) \mapsto \perp$ which, after reapplying the intersection rule to $(q_2, [1, 3])$ and $(q_2, [4, 8])$ to build $(q_2, \emptyset)$, detects a discrepancy.

2.2.2 Interval Propagation

CATS uses the classical interval propagation techniques already found in [Brown et al., 1982] and described by [Davis, 1987]. In practice CATS may use any interval refinement computation primitive which is sound and also monotonic w.r.t. interval inclusion.

Soundness means that the computed interval of a given quantity always contains all possible values of

this quantity. Although this theoretical domain of values is usually an interval (because the constraints are usually expressed as continuous piecewise linear functions), it is generally impossible to compute it exactly (see [Moore, 1966] for mathematical aspects of interval calculus), i.e. the propagation procedure is not complete. For example, a naive propagation of interval i through the function $y = x + 1/x$ according to the formula $i + 1/i$ gives for $i = [0.5, 2]$ the interval $[1, 4]$ which is larger than the exact result $[2, 2.5]$. When the domain is not an interval, but can yet be computed, we nevertheless enlarge it to an interval. For example, dividing by zero is not continuous and would lead to a disjunction of intervals that is too complex to manage. $1/[-1, 2] =]-\infty, -1] \cup [0.5, +\infty[$ is thus enlarged to $]-\infty, +\infty[$.

Monotonicity (w.r.t. interval inclusion), which is obviously theoretically true for continuous functions, is also true in practice when classical propagation procedures are used.

Any input of CATS such as $f(q_1', \dots, q_m') \geq 0 \Rightarrow c(q_1, \dots, q_n)$ is expressed as predictive first-order rules on data:

$(q_1', I_1'), \dots, (q_m', I_m'), f(I_1', \dots, I_m') \subseteq \mathbb{R}_+,$

$(q_1, I_1), \dots, (q_n, I_n) \mapsto (q_k, \subseteq_k(I_1, \dots, I_n)),$

where f and each $\subseteq_k$ are the computation primitives associated with function f and constraint c .

A problem of interval refinement is that it may go into an infinite loop [Davis, 1987]. To prevent this, a halt condition is required. It is defined by the following qualitative equality between intervals using two thresholds Λ and ϵ :

$[a, b] \approx [\max(a, -A), \min(b, A)]$

$[a, b] \approx [c, d]$ if $|a - c| + |b - d| < \epsilon$

Considering the way floating numbers are represented in a computer, this halt condition is satisfied in practice. Moreover Λ and ϵ have physical counterparts: the upper bound for a value of a physical quantity can be known as a property of the artifact and the lower bound results from the imprecision of both measurement and modeling. In this sense, this halt condition ensures a "computationally and physically complete" saturation.

2.2.3 Interval Semantics and Assumption Management

In a classical refinement process, without assumptions, only the smallest interval for each quantity q is kept at any moment. But this is not true for propagation under assumptions. In the coupling of an interval propagator with a traditional ATMS, the semantics of the data is taken care of by the propagator itself which has to fire all possible instantiated forms not only of the input rules but also of the three interval rules, recording each of them as a justification. This saturation process leads to a combinatorial explosion.

Consider for example a single input rule $r: q_2 = [q_0 - q_1]$ and the eight following data: $\{h_1, h_2\} \rightarrow (q_1, [0, 6])$, $\{h_1\} \rightarrow (q_1, [2, 8])$, $\{h_2\} \rightarrow (q_1, [3, 16])$, $\{h_2\} \rightarrow (q_1, [4, 18])$, $\{h_3\} \rightarrow (q_1, [6, 20])$, $\{h_4\} \rightarrow (q_0, [8, 9])$, $\{h_5\} \rightarrow (q_0, [7, 13])$, $\{h_5\} \rightarrow (q_2, [5, 15])$. Total saturation leads to the computation of 46 new data and 310 justifications (see Figure 1 where each interval $[a, b]$ is represented by the point (a, b) in a half plane).

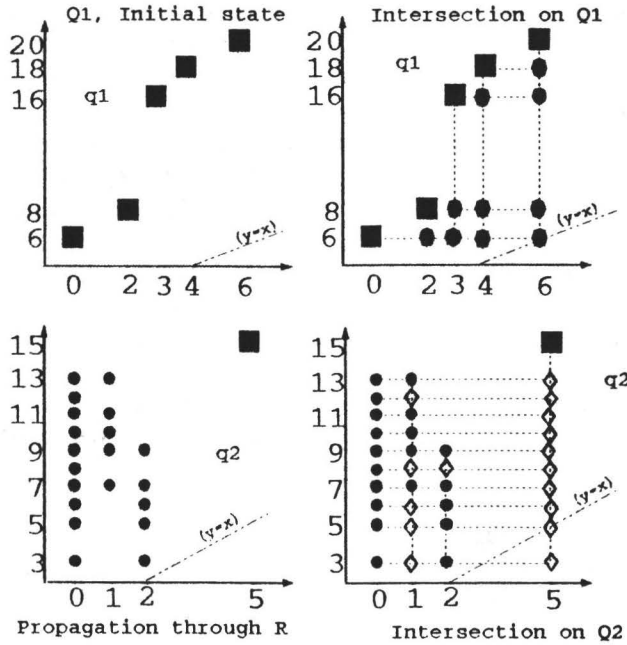


Figure 1

When extending ATMS so as to merge interval semantics (intersection and inclusion rules) and assumption management, some optimization is obtained.

- **Intersection with assumptions:** any intersection of n intervals is in fact the intersection of two of them. Computing the intersection of an intersection is unnecessary as it would only give bigger environments. Superfluous justifications are thus removed.

- **Inclusion with assumptions:** by logical subsumption, bigger intervals that hold under bigger environments can be suppressed, since this redundancy remains valid through propagation thanks to monotonicity. Thus the only pairs (datum, environment) which are kept are those that are minimal for the following order

$$((q, i_1), e_1) \leq ((q, i_2), e_2) \Leftrightarrow (i_1 \subseteq i_2) \wedge (e_1 \subseteq e_2).$$

Labels are thus simplified, nodes may be removed and inclusion justifications are not recorded.

Total optimization now leads to the computation of only 18 new data, 28 new justifications and the suppression of 3 initial data.

2.2.4 Focusing on Nogoods

Previous optimizations are still insufficient to prevent a

combinatorial explosion. We now have to define a propagation procedure which exploits the fact that, for diagnosis purposes, only minimal nogoods are needed. The idea is to propagate only minimal values for each quantity, so as to detect the contradictions (thanks to monotonicity none is missed). In our example, CATS just computed $(q_1, [0, 6]) \cap (q_1, [6, 20]) = (q_1, [6, 6])$ and fired rule $r: ((q_1, [6, 6]), (q_0, [8, 9])) \rightarrow (q_2, [2, 3])$, which detected a contradiction by intersection with $(q_2, [5, 15])$ and thus produced the nogood $\{h_1, h_2, h_3, h_4, h_5\}$ (see Figure 2).

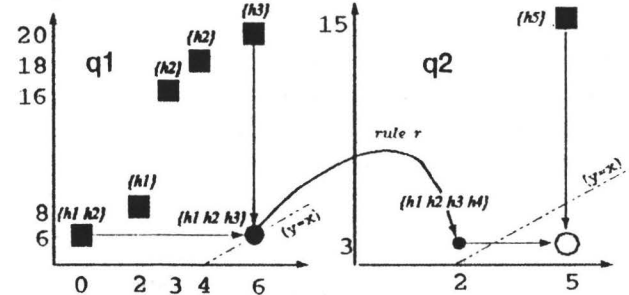


Figure 2

The minimality of the nogoods thus obtained is not guaranteed. CATS therefore uses the recorded path leading to the contradiction to compute minimal values larger than those leading to the contradiction, then activates them. These values may have been computed previously and in this case were hibernating. In our example, CATS computes $(q_1, [4, 6])$ and $(q_1, [6, 8])$ using the intersection rule and reactivates $(q_0, [7, 13])$. Rule r is then used with $(q_0, [8, 9])$ and the new computed values of q_1 to create $(q_2, [2, 5])$ and $(q_2, [0, 3])$, and with $(q_1, [6, 6])$ and the reactivated value of q_0 to create $(q_2, [1, 7])$. Intersection is then used first to create $(q_2, [5, 5])$ and then $(q_2, \emptyset)$ under $\{h_1, h_3, h_4, h_5\}$ (see Figure 3).

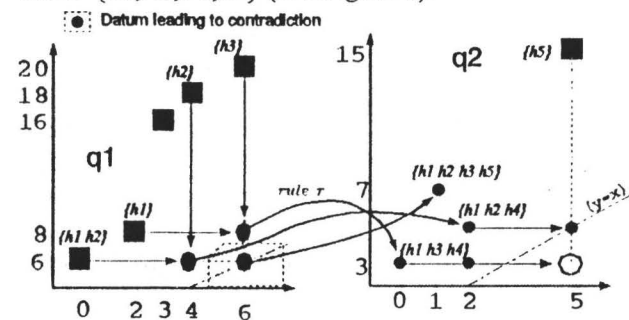
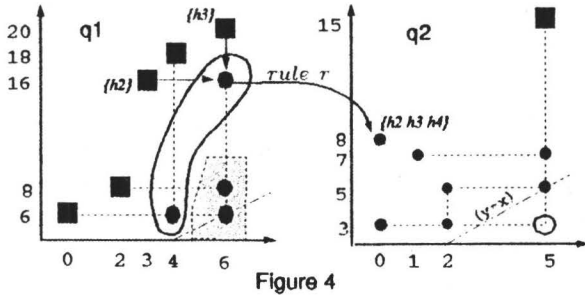


Figure 3

A contradiction thus remains and a smaller nogood $\{h_1, h_3, h_4, h_5\}$ is obtained. The process is repeated: $(q_1, [6, 16])$ is computed by intersection and $(q_2, [0, 8])$ through rule r . As no contradiction is now detected, propagation stops. The final situation is shown in Figure 4 where $[4, 6]$ and $[6, 16]$ are the minimal values for q_1 which do not lead to a contradiction. Only 10 new data and 12 new justifications

have been created. See [Dague et al., 1990b] for a detailed description of the algorithm.



2.3 Extension to Arrays of Intervals

Up to now, only (quantity,interval) data have been considered. Taking variations of quantities over time into account means here manipulating data that are arrays of numeric intervals, because of the sampling of quantities. Propagation of arrays raises no particular problems because most computations are local to a sample, with a few exceptions such as the derivative of a given datum (q,i) that we approximate by (q',i') where $i'_t = (i_{t+\delta} \ominus i_{t-\delta}) / 2\delta$ (δ is the sampling increment and $\ominus$ is the difference between intervals).

Applying the previous techniques of propagation by interval rules, label computation and reactivation of hibernating values to each sample of an array is impracticable (several hundred samples per array). For this reason, we take each array as a whole. The array semantics can now be described by three array rules. The **array intersection** rule creates the same array as would have been obtained by the intersection rule per element, but only one justification is created. The **array falsity** rule holds as soon as the falsity rule holds for one element of the array, i.e. if one element is the empty interval. We are thus sure to detect all discrepancies. However, the **array inclusion** rule between (q,i) and (q,j) applies only if every element i_t is included in j_t . Logical subsumption is not therefore applied when some, but not all, of i_t are included in j_t . Environments computed for the whole array are in general larger than those that would have been computed for each interval. Consequently, not all minimal conflicts are obtained.

One way to recover the minimal nogoods for a contradiction is to apply the previous interval algorithm to each sample (that has an empty interval) involved in the contradiction, using the justifications path that leads to it. The correct local label for these samples can thus be computed and hibernating intervals reactivated. As in general these samples are few in number compared to the size of the arrays, this method becomes tractable.

2.4 Implementation

The present implementation (with IBM Prolog and the object-oriented language IPW) of our diagnostic program does not incorporate all the features of CATS described here. To simplify programming, we have not yet implemented the complete reactivation procedure or the technique to recover (possibly smaller) conflicts between intervals once a conflict between arrays has been found. Not all minimal conflicts are therefore necessarily obtained. Moreover, although the algorithm can work on labels made up of more than one environment, our implementation has been limited to just one because two equal intervals are virtually never obtained.

However, our program offers substantial improvements over the previous one, DIANA [Dague et al., 1990a]. Firstly, intersections of intervals are processed completely; secondly, there is no limit to the number of times a rule can be fired in a given propagation path; lastly, at the end of propagation, it is now possible to reactivate each hibernating value in turn (and not only when the active value becomes inconsistent as was the case in DIANA).

3 Diagnosing Oscillators

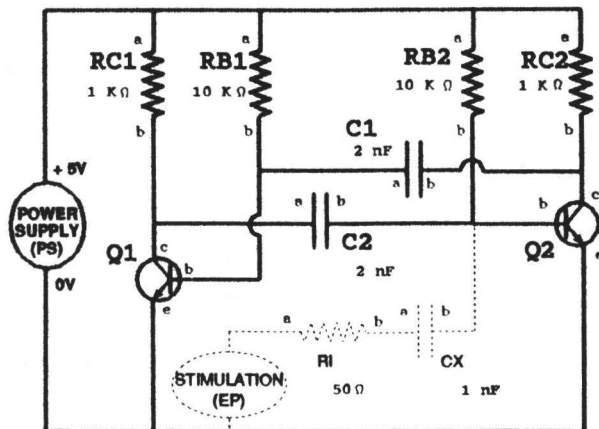
3.1 The Case of Oscillators

We had already experimented our algorithm on various kinds of circuits but when we tried to diagnose oscillators the results were poor. It so happens that most of the faults which may occur bring the oscillator to a state where all voltages are steady. The reason is that the feedback loop which is at the heart of the circuit cannot accommodate deviations from nominal behavior due to a faulty component, even less so as the feedback for oscillators is positive: correct components are drawn to their limit states, sometimes break down themselves, and all electrical quantities remain constant. As a consequence, many components do not appear to be malfunctioning while the diagnostic program is applied; it is the case, for instance, for a capacitor if the voltage at the terminals does not vary, or for a transistor which remains saturated (here the collector current is badly defined). More generally, it is the case when an equation describing a model is not determined for the parameters of the component.

To improve the situation it is necessary to add an external stimulation so as to disturb the circuit and then to perform diagnosis on the disturbance. The fact that a global model of the behavior of the circuit is not required means that there are few constraints in the choice of stimulation: almost any kind of stimulus can be chosen provided this causes variations in the voltage at the different nodes.

3.2 Advantages of Using External Stimulation

We carried out our experiment on the astable multivibrator shown below.



We have studied 12 cases of failure. Taking into account the symmetry of the circuit, they correspond to all the short circuits and all the breaks in the circuit that could occur in each component. Although the fail-

ures considered are the most frequent ones, they only represent a small fraction of all possible faults; the results should therefore not be analyzed in statistical terms. We started by performing diagnosis without stimulation and in this case all observed voltages were constant. We then applied a pulse which lasted $1 \mu s$ to the base of the transistor $q2$ via a capacitor cx of $1 nF$ (dotted lines on the diagram). The results obtained are summarized in the table below. There was a marked improvement in the diagnosis when external stimulation was introduced. Without stimulation, no diagnoses were proposed in 5 cases of failure, because no contradictions appeared; with stimulation, diagnoses were proposed for all of these cases. In 2 other cases of failure the diagnosis is more precise (short circuit of $rc1$ and $c1$) and, in the case of $c1$ under single fault hypothesis, the faulty component can be identified.

The residual ambiguity of the diagnoses should not be blamed on our algorithm or its implementation but, on the contrary, is a consequence of the impossibility of measuring all quantities.

DEFECT	DIAGNOSES WITHOUT STIMULATION	DIAGNOSES WITH STIMULATION
short-circuit on $rc1$	$rc1, q1, rb1, c1, \{c2\} \times \{ps, ep, q2, rb2, rc2, ri\}$	$rc1, q1, \{c2\} \times \{ps, cx, c1, ep, q2, rb1, rb2, rc2, ri\}$
short-circuit on $rb1$ which induces destruction of $q1$	$q1^1$	$q1^1$
short-circuit on $c1$	$c1, q1, rb1$	$c1$, and 10 double candidates
short-circuit on $q1$ (collector-emitter)	nil	$q1, c2, \{rc1\} \times \{cx, c1, ep, q2, rb2, rc2, ri\}$
short-circuit on $q1$ (base-emitter)	$q1, c1, rb1$	$q1, c1, rb1$
short-circuit on $q1$ (base-collector)	$q1, c1, rb1, rc1, \{c2\} \times \{ps, ep, q2, rb2, rc2, ri\}$	$q1, c1, rb1, rc1, \{c2\} \times \{ps, ep, q2, rb2, rc2, ri\}$
open circuit on $rc1$	nil	$rc1, q1, c2$
open circuit on $rb1$	$rb1, q1, c1$	$rb1, q1, c1$
open circuit on $c1$	nil	$c1, q2, rc2$
open circuit on $q1$ (emitter)	$q1$	$q1$
open circuit on $q1$ (base)	$q1$	$q1$
open circuit on $q1$ (collector)	nil	$q1, c1, rb1, c2, rc1$

¹ Any superset of a diagnosis is in our case a diagnosis (analogous to a candidate in [de Kleer and Williams, 1987]), since we don't use fault models or alibis [Raiman, 1990]. Only minimal diagnoses are thus given. The diagnosis $\{q1, rb1\}$ is not proposed directly by the program since any contradiction containing $rb1$ also includes $q1$ and a contradiction has been found containing $q1$ alone (high base-emitter voltage).

We know that the use of fault models could help to reduce the ambiguity in our example where the faults tested for came from an existing list of possible faults. Unfortunately such a list can in no way cover all the cases in analog electronics.

4 Conclusion

We have presented an ATMS-like constraint propagation engine for processing arrays of numeric intervals, focusing on the recognition of all minimal nogoods. This engine, CATS, constitutes a powerful tool for model-based diagnosis of continuous dynamic physical systems.

Its application to analog electronics allows the handling of failures observable only in transient modes and does not require a description of the nominal behavior.

However, as has been shown in this paper, there are cases where, if a system has failed, taking measurements under the conditions where the failure was detected is not sufficient for model-based diagnosis to be performed. It is necessary to introduce external stimulation and to observe the resulting disturbances in order to obtain a diagnosis. Fortunately, since a global model of the behavior of the circuit is not required for diagnosis, a great scope in the choice of stimulation is possible, as can be seen through the example of oscillators. All we have to do is to kick the ant-hill and see what happens.

Acknowledgments

We would like to thank the referees for their pertinent remarks which enabled us to considerably improve the first draft of this paper. We also thank Rosalind Greenstein for helping us to squeeze a gallon of ideas into a pint pot of acceptable English.

References

- J.S. Brown, R.R. Burton, and J. de Kleer, "Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III," *Intelligent Tutoring Systems*, Academic Press, 1982.
- P. Dague, P. Devès, P. Luciani, and P. Taillibert, "Analog Systems Diagnosis," *Proceedings of the Ninth ECAI*, Stockholm, August 1990a.
- P. Dague, P. Devès, and O. Raiman, "Troubleshooting: when Modeling is the Trouble," *Proceedings of AAAI Conference*, Seattle, July 1987.
- P. Dague, O. Jehl, and P. Taillibert, "An Interval Propagation and Conflict Recognition Engine for Diagnosing Continuous Dynamic Systems," *Lecture Notes in Artificial Intelligence*, vol. 462, Springer-Verlag, Vienna, September 1990b.
- E. Davis, "Constraint Propagation with Interval Labels," *Artificial Intelligence* 32, 1987.
- J. de Kleer, "An Assumption-based TMS," *Artificial Intelligence* 28, 1986.
- J. de Kleer and B.C. Williams, "Diagnosing Multiple Faults," *Artificial Intelligence* 32, 1987.
- J. de Kleer and B.C. Williams, "Diagnosis with Behavioral Modes," *Proceedings of the Eleventh IJCAI*, Detroit, August 1989.
- W. Hamscher, "Temporally Coarse Representation of Behavior for Model-based Troubleshooting of Digital Circuits," *Proceedings of the Eleventh IJCAI*, Detroit, August 1989.
- W. Hamscher, "XDE: Diagnosing Devices with Hierarchic Structure and Known Component Failure Modes," *Proceedings of the Sixth IEEE Conference on AI Applications*, Santa Barbara, March 1990.
- R.-W. Liu, *selected papers on Analog Fault Diagnosis* Advances in Circuits and Systems, IEEE Press, 1987.
- G. Loveday, *Electronic Testing and Fault Diagnosis*, Second edition Longman Scientific and Technical, 1989.
- R. Moore, *Interval Analysis* Series in Automatic Computation, Prentice-Hall, 1966.
- H.T. Ng, "Model-based Multiple Fault Diagnosis of Time Varying, Continuous Physical Devices," *Proceedings of the Sixth IEEE Conference on AI Applications*, Santa Barbara, March 1990.
- Y. Pan, "Qualitative reasoning with deep-level mechanism models for diagnoses of mechanism failures," *Proceedings of the First IEEE Conference on AI Applications*, Denver, 1984.
- O. Raiman, "A Circumscribed Diagnosis Engine," *Lecture Notes in Artificial Intelligence*, vol. 462, Springer-Verlag, Vienna, September 1990.
- R. Reiter, "A Theory of Diagnosis from first Principles," *Artificial Intelligence* 32, 1987.
- K. Sierzega and R. Rastogi, "Model-based reasoning finds Analog PCB faults," *Electronics Test*, pp. 26-30, March 1990.
- P. Struss and O. Dressler, "Physical Negation - Integrating Fault Models into the General Diagnostic Engine," *Proceedings of the Eleventh IJCAI*, Detroit, August 1989.
- B. C. Williams, "Doing Time: Putting Qualitative Reasoning on Firmer Ground," *Proceedings of AAAI Conference*, Philadelphia, August 1986.