PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 34, NO. 3, PP. 179–193 (1990)

QUALITATIVE SIMULATION OF CONTINUOUS DYNAMIC SYSTEMS

M. E. WIEGAND

Intelligent Automation Laboratory
Department of Electrical and Electronic Engineering
Heriot-Watt University, Edinburgh, Scotland

Received July 2, 1990.

Abstract

This paper describes a *constructive* algorithm for performing the qualitative simulation of continuous dynamic systems. The algorithm may be thought of as the qualitative analogue of conventional numerical simulation. The equations of the system model are causally ordered to improve the efficiency of the algorithm; this also facilitates the production of causal explanations of system behaviour. We introduce the notion of 'differential planes' to allow the system model to be repeatedly differentiated. An additive input function is described by piecewise singularity functions and the algorithm includes the appropriate level of differential plane to remove the qualitative ambiguity. In this way the algorithm is able to cope with inputs as functions of time. We illustrate the application of the algorithm to a number of examples, and show that it overcomes some problems encountered with the *non-constructive* approach.

Keywords: causal ordering, differential planes, qualitative ambiguity, qualitative reasoning.

Introduction

This paper describes a novel algorithm for performing the qualitative simulation of continuous dynamic systems. This 'Predictive Algorithm' forms the outer module of the system architecture described in (WIEGAND, 1989), and replaces the Predictive Rules module in the earlier design of the Predictive Engine. The algorithm is coded in Prolog.

The Predictive Algorithm uses a well-defined notion of system causality, familiar (and intuitive) to systems engineers. This notion of causality forms the basis of Iwasaki's work on causal ordering (IWASAKI, 1988). The algorithm is 'event-based' and uses the notion of 'persistence': an event is assumed to persist until another event causes it to change. For this reason we call the algorithm *constructive*.

We contrast our algorithm with the *non-constructive* approach used in the QSIM algorithm (KUIPERS, 1986). At each new state generation in QSIM, every possible (in a local sense) qualitative transition of each system variable is considered. QSIM then refutes impossible (in a global sense) transitions of the system considered as a whole by applying a succession of filters. A phenomenon known as 'chattering' (KUIPERS, 1987) has been described and investigated in the QSIM algorithm. We claim that chattering can be split into (at least) two quite distinct phenomena and that in one case it is the non-constructive nature of QSIM that causes chattering. Local transitions are considered that cannot be filtered by global information. We contend that the consideration of all local transitions is unnecessary if the correct transition can be inferred constructively.

The Predictive Algorithm separates constructive inferences about the system behaviour into three temporal categories. Those events that will occur *instantaneously* are discovered by a phase termed 'Causal Propagation'. Events that will occur after *infinitesimal* time are found by a phase termed 'Qualitative Integration'. Lastly, those events that *may* occur after some *finite* time are also found by Qualitative Integration, but since the results are hypothetical further investigation is required by a 'Transition Analysis' phase. The algorithm terminates when it deduces that equilibrium has been reached. This process may be thought of as the qualitative analogue of conventional numerical simulation.

In some respects the Predictive Algorithm is similar to TQA (WILLIAMS, 1984). However, it differs from TQA in important ways. The Causal Propagation phase uses a fixed notion of system causality with a consequent improvement in efficiency. The Transition Analysis phase calls the Predictive Algorithm recursively, avoiding the need for separate rules of Transition Ordering. By comparison with the work of DE KLEER and BOBROW (1984) the Predictive Algorithm treats asymptotic transitions to equilibrium in the same way as finite time transitions (although technically these would constitute discontinuous behaviour). We also introduce an explicit notion of 'differential planes' to avoid qualitative ambiguity around feedback loops with additive inputs.

Differential planes allow the system model to be differentiated repeatedly so that input functions described by piecewise singularity functions (D'AZZO and HOUPIS, 1981; OPPENHEIM et al, 1983) can be considered. The qualitative model of the system includes the order of differential plane that removes the qualitative ambiguity introduced by the additive input function. The Predictive Algorithm is called whenever the description of the input function changes. We assume that all changes to the input description are left-continuous (Heaviside). Differential planes also allow higher-order information about the functional relationships between system variables to be used.

The next section describes the various phases of the Predictive Algorithm. The results sections that follow are used to illustrate some of the features that have been discussed in this paper. An appendix shows how the phases of the algorithm construct the qualitative behaviour for one of the example systems considered.

The Predictive Algorithm

The Predictive Algorithm consists of three separate phases. A flowchart for the algorithm is presented in *Fig 1.* An 'event' in the algorithm is a tuple consisting of the qualitative value (for simplicity, either +, 0, or −) of a system variable and the time duration for which the variable held that value. An event 'persists' if the end point of its duration is not yet known.

The Causal Propagation phase of the algorithm is responsible for inferring new events that will begin at the *same* time as the antecedent events (viewing qualitative arithmetic as an inference process). Events created during this phase begin *instantaneously* in response to the persisting events used to make the inference. Causal Propagation differs from the more usual 'constraint propagation' in that it assumes that the equations describing the qualitative model of the system have been manipulated into an ordered sequence that permits an intuitive interpretation of causality. Each equation of the model is rearranged so that the variables appearing in the right hand side of the equation *cause* the single variable that appears on the left hand side. Also, the equations themselves are ordered into a sequence

$$\{e_1, e_2, \ldots e_n\}$$

so that for any equation e_i, the left hand variable of e_i does not appear in the right hand side of any $e_j, 1 \leq j < i$. Causal Propagation treats each equation as an inference rule to determine the instantaneous value of the left hand variable, and proceeds linearly through the equation sequence. This process is directly analogous to the way that system equations are rearranged for the operation of computational algorithms in conventional numerical simulation. In the Predictive Algorithm, 'causality' is the normal system theoretic notion as understood by users of block diagram notation for describing models of physical systems. The method used for determining this causality is based on the dynamic causal ordering algorithm of IWASAKI (1988). This provides the information needed to rearrange the equations into the above 'qualitative structural equations'.

TQA (WILLIAMS, 1984) also uses a phase termed 'Causal Propagation'. However, Causal Propagation in TQA appears to search for an appropriate equation at each inference step. Our Causal Propagation is able to exploit system causality to a greater extent since the ordering of the equations means that the phase can be completed in a single linear pass

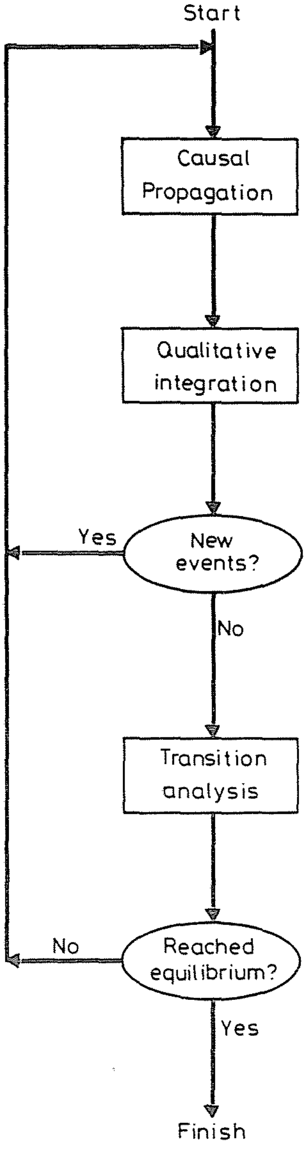

Fig. 1. Flow chart of the predictive algorithm

(without searching). This appears to be much more efficient than in TQA. In addition, the process of Feedback Analysis in TQA occurs automatically in our Causal Propagation by virtue of the ordering of the equations and the use of higher-order differential planes.

Continuous dynamic systems store energy, i.e. they have *memory*. This energy is released to produce the behaviour of the system. In forming dynamic analytic models of such systems, we often model this ability to store energy using the time-integral primitive. The effects of integration then model the storage and release of energy in the system. In our qualitative structural equations, we say that any differentiated variable in the model, such as V', will effect its base variable, V, via the process of Qualitative Integration. The events inferred by the phase of Qualitative Integration are considered to begin some time *after* the start of the persisting events used to infer those new events. For example, if V is 0 and V' is $+$, then V *will become* $+$.

Qualitative Integration characterises the algorithm as one able to predict continuous dynamic behaviour by providing a means of using the information contained in a dynamic modelling primitive (here differentiation or integration). The Qualitative Integration phase is genuinely *causal*. In physical systems, causality is always considered to be directed from the integrand to the integral, reflecting the dissipation of energy in the system. It is the nature of this phase in particular (though also our implementation of Causal Propagation) that characterises the Predictive Algorithm as *constructive*, as opposed to a *non-constructive* algorithm such as QSIM (KUIPERS, 1986). Qualitative Integration is linear in the number of differentiated variables in the model (including all differential planes).

A new event inferred by Qualitative Integration could occur after some *infinitesimal* time has passed, e.g. $V=0$ and $V'=+$ implies $V=+$ after infinitesimal time. However, the new event might occur after some *non-infinitesimal* (but finite) time has passed, e.g. $V=+$ and $V'=-$ implies $V=0$ after finite time. (Note that we treat asymptotic changes, i.e. new events after infinite time, as though they were finite time changes). In the latter case, there is the possibility of another event starting before the new event, and this might actually prevent it occurring at all! Therefore, Qualitative Integration is separated into the creation of new events after *infinitesimal* time, and the *possibility* of new events after *finite* time. The latter case is called Transition Recognition. If there are more than one new events possibly starting after finite time, then we must deduce which will start first. The task of determining which of the hypothesized events starts first is performed by Transition Ordering. Transition Recognition and Ordering together form the Transition Analysis phase. The only method we have of determining the subset of hypothesized events that will occur first is 'generate-and-test'. Transition Recognition generates the possibilities, and Transition Ordering tests to see if a valid dynamic behaviour is produced. We take our terminology from Williams who first used this two-phase notion of Transition Analysis (WILLIAMS, 1984).

Transition Recognition takes a list of possible transitions (hypothesized events) and produces the power set of these transitions. It orders the subsets in the power set in accordance with two heuristics. The first is that the subsets are arranged in increasing size (least commitment). The second principle employed in ordering the subsets is that the lowest-order state variable transitions should be considered first (DE KLEER and BOBROW, (1984) *only* consider these). The computation of subsets for consideration by Transition Ordering indicates that this phase is of exponential complexity. The only way of alleviating the problem (apart from the heuristic methods above) is to cut down the exponential term, i.e. the number of possibly transitioning variables returned from Qualitative Integration. The way this is done in the Predictive Algorithm is by 'dynamic plane switching'. This is the technique of ignoring higher-order differential planes when they will contribute no more information to the simulation. By utilising dynamic plane switching and the two heuristics mentioned above it is always possible to reduce the exponential term to the order of the system model.

Transition Ordering calls the Predictive Algorithm itself recursively to perform the analysis, trying to discharge the assumption made by Transition Recognition. TQA (WILLIAMS, 1984) has a separate analysis phase to do this, but DE KLEER and BOBROW (1984) use a similar recursive scheme. The testing of whether the results of the Predictive Algorithm are consistent with the behaviour of a dynamic system is left to the following rules. The first rule is that non-causal inferences are not allowed. This means that inferences in Causal Propagation cannot effect a variable's history. The second rule disallows discontinuous behaviour, and in particular is concerned with finding *prior* events. A prior event is found if the rules of continuity demand that the transition being considered must be preceded by another transition. Finally, the third rule concerns one way in which QSIM may start 'chattering' (KUIPERS, 1987). In certain coupled systems we find that the transition of one state variable derivative without the others causes an oscillation. This sort of oscillation is disallowed as being an impossible behaviour; it occurs when there is a 'qualitative symmetry' between state variables. Note that we have identified two quite separate phenomena that cause chattering in QSIM: firstly, the non-constructive nature of the algorithm, and secondly, the genuine problem of qualitatively symmetric systems. We contend that QSIM's method of avoiding these phenomena is not sufficiently general (this was also a conclusion of DALLE-MOLLE (1989)). At present, we are only able to formally justify the third rule for *linear* models of physical systems. One way of extending the justification is to use a qualitative version of the Euler-Cauchy method (HENRICI, 1964); in fact, this is the principle behind QSIM's use of the 'Smoothness Assumption' to avoid the problem of chattering.

Simple Water Tank with Laminar Flow

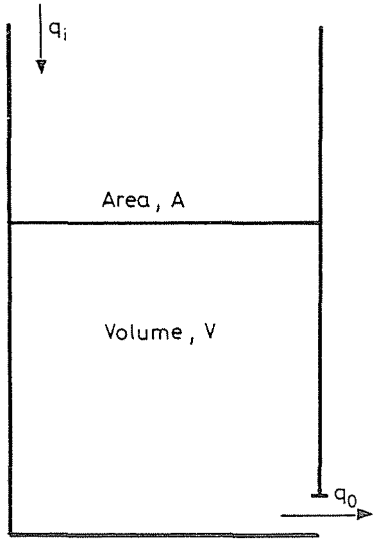

Fig. 2. Simple water tank

The physical system consists of the simple water tank shown in *Fig. 2.* We will assume that there is laminar flow at the output q_o . This is modelled by a linear relationship between the height of fluid in the tank $\frac{V}{A}$ and the output flow q_o. We therefore have the following system model:

$$q_o = \left(\frac{\rho g}{R}\right)\left(\frac{V}{A}\right) ,$$

$$V' = q_i - q_o ,$$

where ρ is the density of the fluid, g is the acceleration due to gravity, and R is the laminar fluid resistance. We wish to supply an input flow q_i

described by a linear ramp over time. We model this as a step change to q_i' (with q_i'' held constant at 0).

The process of developing a qualitative model of this system is difficult to formalise (and therefore difficult to automate). Firstly, we differentiate the system model to derive the appropriate differential planes for the order of the singularity function that describes the additive input q_i . Each plane is then subjected to the causal ordering algorithm and the equations rearranged into a list of qualitative structural equations. For clarity of exposition, we will ignore constant parameters in the model. Finally, we obtain the following qualitative model:

Plane 0:

$$[q_o] = [V] ,$$

$$[V'] = [q_i] - [q_o] .$$

Plane 1:

$$[q_o'] = [V'] ,$$

$$[V''] = [q_i'] - [q_o'] .$$

Plane 2:

$$[q_o''] = [V''] ,$$

$$[V'''] = [q_i''] - [q_o''] .$$

The linear ramp, modelled as a step change to q_i' , is applied at (symbolic) time t_1. *Fig. 3* shows the resulting qualitative behaviour predicted by the algorithm as the response of the laminar tank model to the ramp. Note that after the initial transient between times t_1 and t_2, the system reaches an equilibrium condition where the responses of the state variable V and the output q_o are linear. The appendix shows the constructive effect of the three phases of the algorithm: Causal Propagation, Qualitative Integration, and Transition Recognition.

Simple Water Tank with Turbulent Flow

The physical system is the same as before (see *Fig. 2*), but now we will assume that there is turbulent flow at the output q_o. This is modelled by a non-linear relationship between $\frac{V}{A}$ and q_o. We have the following system model:

$$q_o = C_d a \sqrt{2g\left(\frac{V}{A}\right)} ,$$

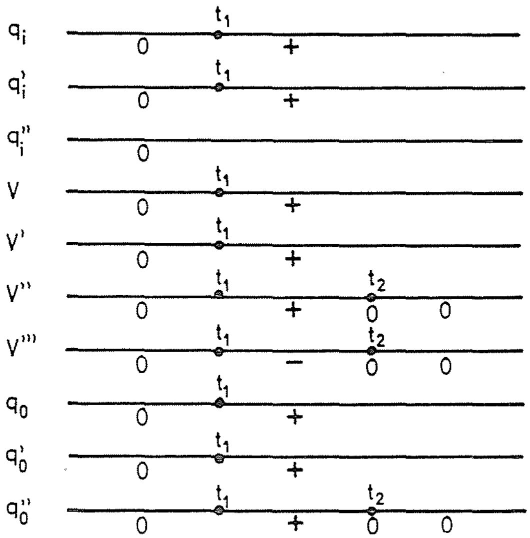

Fig. 3. Response of laminar tank model to ramp

$$V^{'} = q_{\rm i} - q_{\rm o} \ ,$$

where C_d is the flow coefficient at the output aperture, and a is the area of the aperture. We differentiate and simplify to obtain the following qualitative model:

Plane 0:

$$[q_{\rm o}] = [V] \ ,$$

$$[V^{'}] = [q_{\rm i}] - [q_{\rm o}] \ .$$

Plane 1:

$$[q_{\rm o}^{'}] = [V^{'}] \ ,$$

$$[V^{''}] = [q_{\rm i}^{'}] - [q_{\rm o}^{'}] \ .$$

Plane 2:

$$[q_{\rm o}^{''}] = [V^{''}] - [V^{'}] \ ,$$

$$[V^{'''}] = [q_{\rm i}^{''}] - [q_{\rm o}^{''}] \ .$$

Note that the non-linear square root relationship has introduced a structure into plane 2 (we have applied a heuristic to handle the zero denominator). This higher-order information would not be available to an algorithm such as QSIM that only supports basic 'weak' functional primitives. However, since we have an explicit notion of differential planes, we can use these planes to hold the additional information that is available about the relationship between $\frac{V}{A}$ and q_o.

The linear ramp, modelled as a step change to q_i', is applied at (symbolic) time t_1. *Fig. 4* shows the resulting qualitative behaviour predicted by the algorithm as the response of the turbulent tank model to the ramp. Note that after the initial transient between times t_1 and t_3, the system reaches an equilibrium condition where the response of the output q_o is linear, but the response of the state variable V is non-linear.

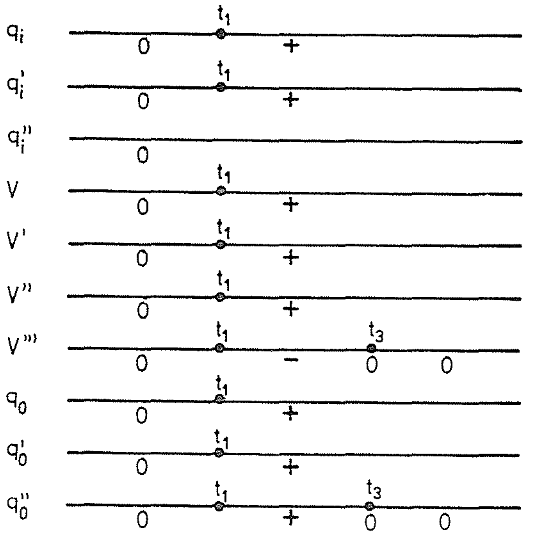

Fig. 4. Response of turbulent tank model to ramp

Two Coupled Water Tanks

The physical system consists of an arrangement of two coupled water tanks shown in *Fig. 5.* We use the following system model:

$$q_1 = C_{d1} a_1 \sqrt{2g(H_1 - H_2)}\ ,$$

$$q_o = C_{do} a_o \sqrt{2gH_2}\ ,$$

$$H_1' = \frac{q_i - q_1}{A_1}\ ,$$

$$H_2' = \frac{q_1 - q_o}{A_2}\ .$$

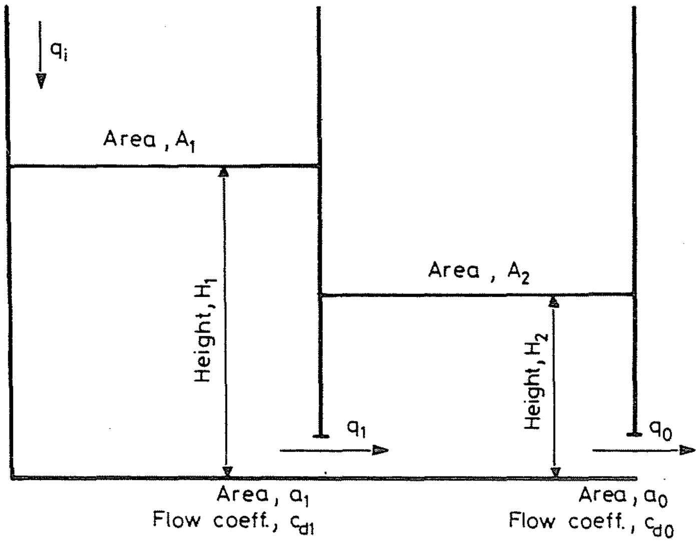

Fig. 5. Two coupled water tanks

We wish to supply an input flow q_i described by a step change over time. We model this as a step change to q_i (with q_i' held constant at 0). We need only derive differential planes to the appropriate level for dealing with the order of the input singularity function. We therefore obtain the following qualitative model:

Plane 0:

$$[q_1] = [H_1] - [H_2] ,$$
$$[q_o] = [H_2] ,$$
$$[H_1'] = [q_i] - [q_1] ,$$
$$[H_2'] = [q_1] - [q_o] .$$

Plane 1:

$$[q_1'] = [H_1'] - [H_2'] ,$$
$$[q_o'] = [H_2'] ,$$
$$[H_1''] = [q_i'] - [q_1'] ,$$
$$[H_2''] = [q_1'] - [q_o'] .$$

The step change to q_i is applied at (symbolic) time t_1. *Fig. 6* shows the resulting qualitative behaviour predicted by the algorithm as the response of the coupled tanks model to the step. Note that at time t_4 a *non-constructive* algorithm such as QSIM would have investigated three possibilities for the value of H_2'' after t_4, i.e. +, 0, or −. Without appropriate filtering, the phenomenon of 'chattering' would have been encountered. However, in our *constructive* algorithm the change to H_2'' has been inferred as a result of the changing H_1' and our algorithm goes on to infer that H_2'' becomes − after t_4.

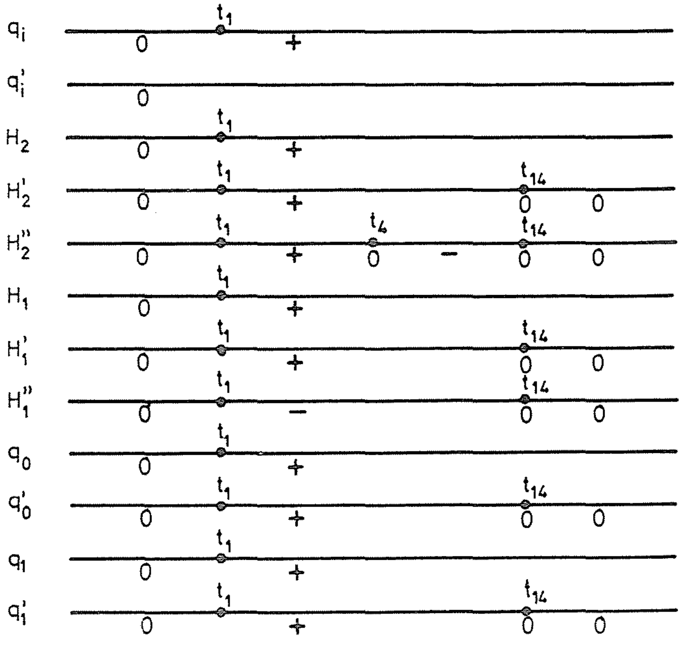

Fig. 6. Response of coupled tanks model to step

The determination of what happens during t_4 to t_{14} involves finding out which of H_1' and H_2' will transition to 0 first. This aspect of Transition Analysis is unavoidably exponential; we must investigate the power set of possible transitions. However, notice that by our mechanism of using differential planes we can *always* limit the exponential term to the order of the system. Here the system is second order, and we must investigate $2^2 - 1 = 3$ possibilities: H_1' first, H_2' first, or H_1' and H_2' together.

Conclusion

We have presented a novel algorithm for performing the qualitative simulation of continuous dynamic systems. The algorithm is *constructive* and is founded on system theoretic principles. We stress the important role played by 'causality' in our algorithm. The notion of 'differential planes' has been introduced and we have illustrated how these can be used both to limit the exponential nature of the algorithm and to use higher-order information about functional primitives. The algorithm has been compared with the results of other research in qualitative reasoning. We have shown that it exhibits a number of advantages, not least its efficiency.

All the results in this paper have been generated by the Prolog implementation of the Predictive Engine. The particular systems presented were chosen for their simplicity. The algorithm has been applied successfully to much larger scale systems, including a laboratory system rig (with heat and fluid flow loops) and a full-size industrial-scale gas recirculation fan (part of a boiler system). The adherence to system theoretic principles in our approach has made the algorithm suitable for use with large-scale systems that can be structurally decomposed.

The algorithm forms a module of a general architecture for a Predictive Engine (WIEGAND, 1989) capable of combining reasoning with qualitative and quantitative information.

References

D'AZZO, J. J. – HOUPIS, C. H. (1981): *Linear Control System Analysis and Design: Conventional and Modern,* Electrical Engineering, McGraw-Hill, Tokyo, Japan.

DALLE-MOLLE, D. T. (1989): Qualitative Simulation of Dynamic Chemical Processes. AI89–107, The University of Texas at Austin, Austin, U.S.A.

HENRICI, P. (1964): *Elements of Numerical Analysis,* John Wiley & Sons, New York.

IWASAKI, Y. (1988): Causal Ordering in a Mixed Structure. *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI–88),* Vol. 1, pp. 313–318, Saint Paul, Minnesota, U.S.A.

DE KLEER, J. – BOBROW, D. G. (1984): Qualitative Reasoning with Higher-Order Derivatives. *Proceedings of the National Conference on Artificial Intelligence (AAAI–84),* pp. 86–91, William Kaufmann, Los Altos, Ca., U.S.A.

KUIPERS, B. (1986): Qualitative Simulation. *Artificial Intelligence,* Vol. 29, pp. 289–338, North-Holland/Elsevier, Amsterdam.

KUIPERS, B. – CHIU, C. (1987): Taming Intractible Branching in Qualitative Simulation. *Proceedings of Tenth International Joint Conference on Artificial Intelligence (IJCAI 10),* Vol. 2, pp. 1079–1085, Milan, Italy.

OPPENHEIM, A. V. – WILLSKY, A. S. – YOUNG, I. T. (1983): *Signals and Systems.* Signal Processing Series, Prentice-Hall Inc., Englewood Cliffs, N.J.

WIEGAND, M. E. – LEITCH, R. R. (1989): A 'Predictive Engine' for the Qualitative Simulation of Dynamic Systems. *Artificial Intelligence in Manufacturing: Proceedings of the fourth International Conference on the Applications of Artificial Intelligence in Engineering, Cambridge, U.K., July 1989,* ed. G. Rzevski, pp. 141–150. Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin.
WILLIAMS, B. C. (1984): Qualitative Analysis of MOS Circuits. *Artificial Intelligence,* Vol. 24, No. 1, pp. 281–346, North-Holland/Elsevier, Amsterdam.

Appendix

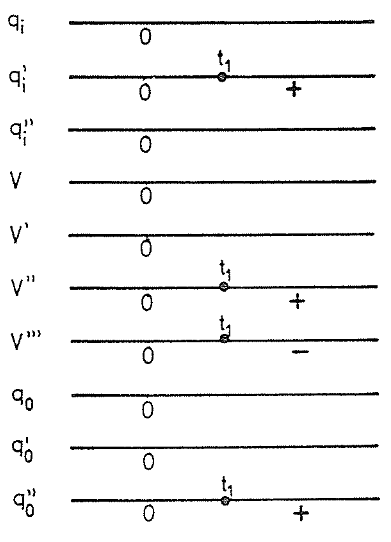

Fig. 1A. Causal propagation on laminar tank model

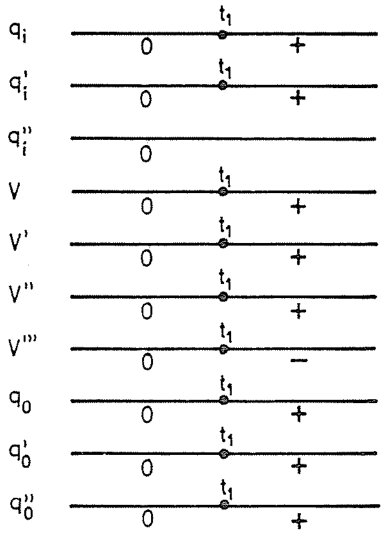

Fig. 2A. Qualitative integration on laminar tank model

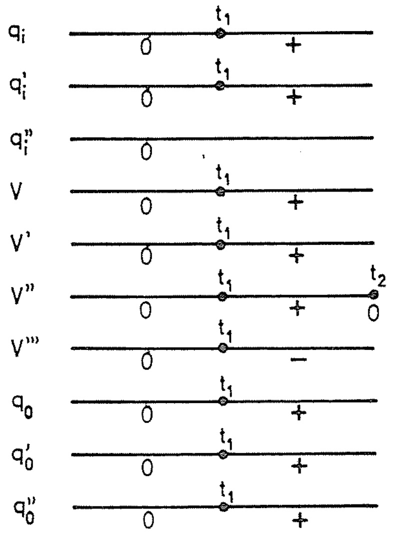

Fig. 3A. Transition recognition on laminar tank model

Address:

Mark Eric WIEGAND
Department of Electrical and Electronic Engineering
Heriot-Watt University
31-35 Grassmarket
Edinburgh, EH1 2HT
Scotland