# Causal ordering for multiple mode systems

**Louise Travé-Massuyès and Renaud Pons**

*Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS*
*7 avenue du Colonel-Roche*
*31077 Toulouse cedex*
*Tel : (33) 5 61 33 63 02*
*email : louise, rpons@laas.fr*

## Abstract

After a brief overview of the work on causality in the area of qualitative reasoning, this paper proposes an algorithm for causally ordering the variables appearing in a set of equations. The main originality of the algorithm compared to existing work is that it copes with systems that have several operating modes and performs the **causal ordering** in an **incremental** way. The algorithm is implemented in the software module CAUSALITO written in C.

## 1. Introduction

The work presented in this paper results from a problem encountered in the framework of the Esprit project TIGER. This project aimed at integrating several artificial intelligence technologies, including qualitative model-based reasoning to perform condition monitoring of gas turbines. It resulted in the TIGER system which has been installed on several gas turbines in the UK to date (1996) [13]. The TIGER diagnostic mechanism uses three independent systems, among which the qualitative model-based diagnosis system Ca~En [2]. Ca~En includes a general diagnosis engine plus a knowledge representation language associated with a simulation engine [13, 1].

A Ca~En model is defined as a set of causal relations (influences) among the variables of the physical system and a set of equations (analytical expressions) relating the variables. These constitute the so called *causal level* and *global constraint level*. Both levels cope with imprecise knowledge by allowing parameter interval values. The causal level is supported by a directed graph in which the nodes stand for the variables and the edges stand for the influences. Influences describe dynamic relationships ; they are implemented by a predicate *I+(X,Y,c,K,Td,Tr)* (or *I-*) in which $Y$ is the variable influenced by $X$, $c$ is an activation condition, $K$ is the gain of the influence (ratio between the variation amplitude of $Y$ with respect to that of $X$), $Td$ is the delay (time taken by $Y$ to react to $X$) and $Tr$ is the response time (time needed by $Y$ to reach a new equilibrium value after being perturbed by $X$). The Ca~En prediction (simulation) and diagnosis procedures are both driven by the causal level, the global level being only used in the prediction procedure [1]. In consequence, whereas the global level may only represent part of the knowledge available about a physical system, all the knowledge must be represented at the causal level.

The Ca~En model of a physical device can be built from deep knowledge (analytical expressions) and/or empirical knowledge (influence relationships known by the experts).

Implementing deep knowledge at the causal level requires to explicitly set the causal structure underlying the set of equations. We are hence faced with the problem of **causal ordering**, and we want to automate this step. Our objective differs however from other works in that the causal order is not only to be used for explanatory purposes (when used by the diagnosis mechanism) but also for prediction purposes (when used by the simulation mechanism).

## 2. Causal ordering

The problem of causal ordering has been approached by several authors in the qualitative reasoning context, generally for providing an explanation of why a device produces the behaviour it does. Since the behaviour is generally obtained from an equational model, the problem can be set as the one of deriving a causal pattern from a set of equations which may be algebraic or differential ones. The causal pattern obviously depends on the context in which the device operates, which is determined by the set of *exogenous* variables, i.e.

variables which are controlled by factors external to the system currently modelled. We can refer in particular to the following approaches:

- the mythical causality approach of de Kleer and Brown [3]
- the causal ordering of Iwasaki and Simon [5, 6, 7]
- the QUAF[1] causal graph generation algorithm of Rose and Kramer [10]
- the bond graphs approach[8, 11]

The QUAF and the bond graphs approaches only provide partial solutions. The QUAF method requires the user to change algebraic equations into differential equations from the knowledge of the temporal scales of the different mechanisms, meaning that the problem is really solved by hand. The bond graphs approach defines a preferred causality for every component from which it builds up the whole causal pattern. Unfortunately, inconsistencies may occur at some points and the algorithm must backtrack.

The solution proposed by Iwasaki and Simon derive the causal ordering from a structural analysis of the equations. Their approach does not require equation solving. It differs from the de Kleer and Brown (1986) process for finding *mythical causality* which performs a runtime computation for determining the propagation paths followed by disturbances given as input signals. We agree with Iwasaki and Simon that finding a causal structure can be viewed as a more general problem than determining the effect of a disturbance for which standard qualitative techniques can be used once the causal structure is obtained. This being so, both approaches are consistent and they provide the same causal order when no feedback loops are involved. When a feedback loop is present, then mythical causality determines all the interpretations, each one specifying a possible causal order around the loop. Unlikely, Iwasaki and Simon do not specify any causal ordering, such an order being pointless for them.

## 3. The requirements of our problem

We do agree with the main ideas of Iwasaki and Simon causal ordering. However, their approach is still limited for our problem.

First, it is not convenient for prediction problems. Indeed when the causal structure is to be used for predicting the values of unmeasured variables by propagating through the values of measured ones, it is impossible to abstract existing feedback loops. Any variable must be reachable from exogenous variables. The prediction engine needs a full causal structure and requires to determine at least *one* possible interpretation around the loops. Notice that all possible interpretations around the loops are equivalent for prediction purposes.

Iwasaki and Simon's approach just provides no causal order in these cases. In the TIGER project, the APU fuel system (cf. section 6.) was a good example of a strongly connected system in which all the 18 unknown variables of the assembled model are in a loop.

Second, we want to address an important problem. It is the very common problem of systems which have **several operating modes**. In hydraulic circuits for example, switch valves are very common. These valves are either fully open or fully closed, adding or retracting new branches to the circuit. The equational models of such systems have conditions associated to some of the equations. A brute force approach would consist in generating a new causal structure for every different mode. This can be significantly optimised by performing an *incremental* generation of the causal structure.

Before presenting our algorithm, we recall the main lines of Iwasaki and Simon approach on which we have built our own contribution.

## 4. The causal ordering of Iwasaki and Simon

### *4.1 Static systems*

Static systems are composed of algebraic equations relating the values of the variables at anytime.

*Self-containment.* A (qualitative) static system of $n$ algebraic equations with $n$ variables is self-contained if every subset of $k$ ($k \leq n$) equations contains at least $k$ variables.

*Minimal Complete Subsystem (MCS).* Given a self-contained system $S$, a proper subset $s$ of $S$ that is also self-contained and does not contain a proper self-contained subset is called a *MCS*.

*Causal Ordering.* Given a self-contained system $S$, let $S_0$ be the union of all its *MCS*, called of zero order. Since $S_0$ is self-contained, the variables in $S_0$, can be determined by solving the equations in $S_0$. Substitute these values for all the occurrences of these variables in the equations of $(S - S_0)$. A new self-contained system is obtained, called the *derived structure of first order*. Let $S_1$ be the union of all its *MCS*, called of first order.

[1] QUalitative Analysis Feedback

The above procedure is repeated until the lastly derived self-contained system contains no proper subsystem that is self-contained.

For each equation $e_i$ of $S$, let $V_i$ denote the set of variables appearing in $e_i$ and $W_i$ the subset of $V_i$ containing the variables belonging to the *MCS* of highest order in $V_i$. Then, the variables in $W_i$ are defined as causally dependent on the variables in $(V_i - W_i)$.

Determining causal ordering is therefore equivalent to computing the *MCSs* of the successive derived structures.

*Remark.* A loop corresponds to the existence of an *MCS* with more than one variable, all the variables of such *MCS* being mutually dependent. One can then notice that, as mentioned before, this causal ordering does not provide a causal order for the variables in a loop.

*Implementation within a graph theoretic framework.* The problem of computing the causal ordering is closely related to the one of finding a perfect matching in a bipartite graph as shown by [9]. This result is presented below.

Given a self-contained system $S = (E, X)$ formed by a set of $n$ equations $E$ in $n$ variables $X$ and the context of the system given by the set of exogenous variables[2]. The problem of causal ordering is the one of determining the dependency paths among variables which would indicate in which order every equation should be used to solve successively for the $n$ unknown variables. The system being self-contained, this comes back to the problem of associating one variable to one equation.

If we define $G = (E \cup X, A)$ as the labelled bipartite graph associated to $S$ in which every equation of $S$ is represented by a node labelled by $e_i$ (equation-node), each variable of $S$ by a node labelled by $x_j$ (variable-node) and there exists a non-oriented edge $a(j,i)$ between the node $x_j$ and the node $e_i$ if $x_j$ appears in $e_i$, then the problem of associating one variable to one equation is the well-known problem of finding a perfect matching in the bipartite graph $G = (E \cup X, A)$.

*Proposition 1.* [9] The bipartite graph $G = (E \cup X, A)$ associated with a self-contained system has a perfect matching.

Once a perfect matching $C$ has been found, every equation $e_i$ can be interpreted as a mechanism which determines the value of its matched variable $x_i$ as a function of the other variables appearing in the equation. $x_i$ is hence viewed as *causally dependent* from the other variables in the equation $e_i$. A causal graph, $G_c = (X, A_c)$, can be derived from $G$ in two steps :

1. Derive the oriented graph $G' = (E \cup X, A')$ by orienting the edges of $A$ from $x_i$ towards $e_j$ if $a(i,j) \in C$ and from $e_j$ towards $x_i$ if $a(i,j) \notin C$ ;

2. Deduce $G_C = (X, A_C)$ from $G' = (E \cup X, A')$ by fusioning in a local manner the matched variable and equation nodes. $G_c$ hence provides a full causal ordering among the variables, exhibiting one possible interpretation around the loops.

Let us now relate $G_c$ to Iwasaki and Simon's causal ordering. $G_c$ by itself does not show the *MCSs* explicitly.

*Proposition 2.* [9] Every *MCS* corresponds to a maximal bipartite elementary subgraph in $G = (E \cup X, A)$.

*Proposition 3.* [9] Every maximal bipartite elementary subgraph of order more than 2 in $G = (E \cup X, A)$, corresponds to a strongly connected component (*SCC*) in the directed bipartite graph $G' = (E \cup X, A')$.

*Proposition 4.* There is a one to one correspondence between the SCCs in $G'$ and the SCCs in $G_C$.

The proof is trivial from the construction of $G'$ and $G_c$. The above propositions show that the *SCCs* in $G_c$ correspond to the *MCSs* of $S$. Consequently, if a graph $G_{c0}$ is built from $G_c$ by aggregating all the nodes within the *SCCs* of $G_c$ into a single node, then $G_{c0}$ provides the causal ordering proposed by Iwasaki and Simon [5]. The variables belonging to the same *SCC* in $G_c$, i.e. the same *MCS* of $S$, are all mutually dependent. Contrary to the causal ordering directly obtained from $G_c$, this causal ordering does not assign any ordering among them.

*Example.* Consider the system composed by a voltage generator $U_0$ feeding two resistors branches (this example will also be used in section 5.2) presented in figure 1. The equations are:

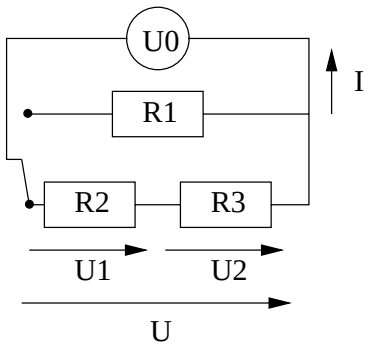



Figure 1: Example: three resistors with a switch

[2]Every exogenous variable is accounted for by an *exogenous equation* which artificially sets the value of the exogenous variable to a constant value parameter

Assuming that the switch is in position 1, the corresponding equations are the following:

$$\begin{array}{rcll} U & = & U_0 & (e_1) \\ U & = & U_1 + U_2 & (e_2) \\ U_1 & = & R_2 I & (e_3) \\ U_2 & = & R_3 I & (e_4) \end{array}$$

The different graphs corresponding to the system are in figure 2.

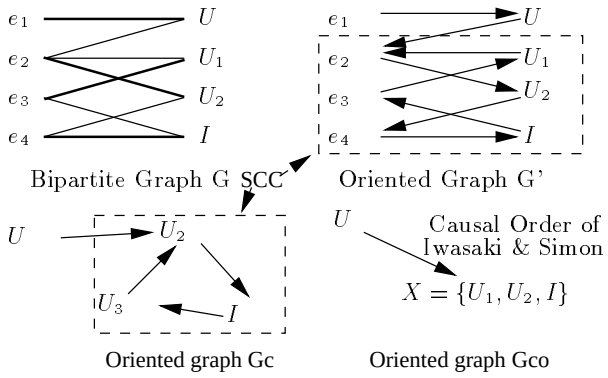



Figure 2: Example: graphs

### *4.2 Dynamic systems*

Dynamic systems are modelled by differential equations. It is commonly accepted that they have a natural causal interpretation. Any differential equation can be brought back to a set of differential equations in canonical form, i.e. there is only one derivative in every equation and the derivative is the only term appearing on the left hand side: $X_i/dt = f(X_1, \ldots, X_i, \ldots, X_n, t)$. Every differential equation in canonical form can then be interpreted as a mechanism which determines the value of a derivative as a function of the variables which appearing in the right-hand side of the equation.

*Self-containment.* A (qualitative) dynamic system of n first order differential equations (in canonical form) is said to be self-contained if every subset of $k$ ($k \leq n$) equations contains at least the first derivatives of $k$ variables.

*Causal ordering.* Iwasaki and Simon [5, 6, 7] distinguish two kinds of causal relations in dynamic systems: differential causality and integral causality. Integral causality means that each variable depends on its derivative ($\forall t, X(t) = X(t - dt) + dX/dt$), and differential causality notifies the dependence of each derivative in relation to all the variables occurring in its expression.

The causal ordering of a self-contained dynamic system in canonical form is given by, for each equation:

1. the integral link between the derivative and its primitive
2. the differential links between the derivative and the other variables of the equation.

### *4.3 Mixed systems*

Mixed structures are composed of differential and algebraic equations.

*Mixed system instanciation.* Since $\forall t, X(t) = X(t - dt) + dX/dt$, Iwasaki and Simon [5, 6, 7] consider $X(t - dt)$ as an exogenous variable for the system instanciation at time $t$. Hence, the causal ordering of a mixed system $M$ at time $t$ is obtained from a new system $Inst(M)$ including all the equations of $M$ and, for each derivative $dX_i/dt$ occurring in $M$, one constant equation $X_i = c$ to represent the fact that $X_i$ is exogenous in $Inst(M)$.

*Self-containment.* A mixed system is self-contained iff:

1. $M$ contains 0 or more first-order differential equations, the rest being algebraic equations.
2. $Inst(M)$ verifies the self-containment criteria of static self-contained systems when the variables and their derivative are treated as distinct variables.

*Causal ordering.* Causal ordering of a mixed system is determined by:

1. The application of static systems ordering rules to $Inst(M)$.
2. The addition of the integral links between each derivative and its primitive.

## 5. Causal ordering for Ca~En

Our problem is to generate automatically the causal structure of the Ca~En models when the original knowledge is in the form of an equational model. This section presents the algorithm that we have devised and implemented in the software module Causalito. Let us recall the requirements of our problem:

1. The causal structure must account for all the causal links (influences) acting within the system, in particular feedback phenomena must be explicitly accounted for, i.e. the causal structure must provide one possible interpretation around the loop(s). This is because Ca~En needs a full causal structure in its prediction mechanism which is based on propagating the variables values through the causal

graph (any internal variable must be reachable from the exogenous variables).

2. The causal structure must be generated for systems which have several operating modes as well.

### *5.1 One single operating mode systems*

#### *5.1.1 Static systems*

Let us consider the self-contained system $S = (E, X)$ and the bipartite graph $G = (E \cup X, A)$ as defined in 4.1. Then we have the following interesting results which help understand the causal ordering problem:

*Proposition 5.* The bipartite graph $G = (E \cup X, A)$ contains no bipartite elementary subgraph ($S$ contains no $MCS$ with multiple variables) iff $G = (E \cup X, A)$ has a unique perfect matching.

**PROOF.** This result is easily proved by reasoning about the occurrence matrix $M$ of $S$. The rows of $M$ correspond to the equations $E$ and the columns to the variables $X$ and the entry $m_{ij}$ is non null iff the variable $x_j$ occurs in the equation $e_i$. This matrix is also the non null submatrix of the adjacency matrix of $G$. Hence every $m_{ij_0}$ corresponds to an edge of $A$. A perfect matching associates one variable to one equation and it can therefore be represented on the occurrence matrix by the selection of $n$ entries with the property that there is one and only one selected entry per row and one and only one selected entry per column. It is a well-known result that the adjacency matrix of a graph with no cycles can be put in diagonal form after a permutation of its rows and its columns. Therefore there is a unique perfect matching which corresponds to the selection of the entries on the diagonal. ■

*Proposition 6.* If the bipartite graph $G = (E \cup X, A)$ has a unique perfect matching $C$, the derived causal graph $G_c$ is acyclic and so is the causal dependence path linking all the variables.

The proof is trivial from the construction of $G_c$ from $G$. In other words, only the $MCS$s with multiple variables give rise to several possible perfect matchings, each of which provides a possible causal order around the loops.

*Corollary.* Consider that all the possible perfect matchings of the bipartite graph $G = (E \cup X, A)$ are $C_1, \ldots, C_n$ and that they include a common submatching, then the common submatching induces the same acyclic causal subgraph in any $G_{ci}, i = 1, \ldots, n$. The non common parts of the perfect matchings $C_1, \ldots, C_n$, induce strongly connected subgraphs in $G_{c1}, ..., G_{cn}$, respectively (they hence correspond to $MCS$s with multiple variables).

**PROOF.** Corollary comes directly from proposition 5. ■

*Remark.* Given that the causal ordering of Iwasaki and Simon is given by $G_{co}$ which is obtained from $G_c$ by aggregating all the nodes corresponding to $SCCs$, it comes from the above propositions and corollary that their causal ordering is unique and independent of the perfect matching it has been derived from.

In order to fulfil requirement 1 and given that all possible interpretations around the loops are equivalent for prediction purposes, our problem reduces to the one of finding one (any) perfect matching in $G$ and deriving the causal graph $G_c$. In other word, we do not need to exhibit the $MCSs$. In terms of mechanisms and influences, the perfect matching associates one variable to one equation, defining which variable is to be determined from which equation. Every equation $e_i$ can then be interpreted as a mechanism which determines the value of its matched variable $x_i$ as a function of the other variables appearing in the equation. In other words, the matched variable $x_i$ is viewed as causally dependent from (influenced by) the other variables in the equation $e_i$. The algorithm that we have devised, Causalito, provides:

- The assignment variable-equation $(x_i, e_i)$ for $i = 1, \ldots, n$ (perfect matching problem);
- The list of influences acting within the system (with associated delay times and activation conditions if any);
- The correspondence between equations and influences.

For determining the perfect matching, we use the Ford and Fulkerson [4] algorithm which finds a maximal flow through a weighted oriented graph.

#### *5.1.1 Dynamic systems*

Ca~En accepts equivalently differential equations or recurrent equations, so does the Causalito algorithm. The system must be self-contained and as in 4.2, it must be put in canonical form[3]

*Canonical recurrent equations.* If some variable appears $p$ times in the equation with different temporal labels, it is considered as $p$ different variables. Following the intuition, the causal links

[3] Similarly to continuous time differential equations, a recurrent equation of any order can be put in the form of a set of first order recurrent equations, i.e. there is only one variable with temporal label $(t+1)$, the other variables having a temporal label $t$, and this variable is the only term appearing on the left hand side of the equation.

are generated consistently with the chronological order imposed by the temporal labels. They are hence drawn from the variables appearing on the right hand side of the equation to the variable on the left hand side, which appears as the one whose value is determined by this mechanism. The causal links are then labelled by a *delay* (delay of the corresponding Ca~En influence) equal to the difference of the temporal labels of the variables that they relate.

*Canonical differential equations.* As opposed to Iwasaki and Simon approach, we do not distinguish a variable from its derivative. The differential causal links are drawn from the variables on the right hand side of the equation towards the primitive variable of the derivative of the equation, which appears as the one whose value is determined by this mechanism. When the primitive variable of the derivative appears explicitly in the equation, this results in a causal link which loops around the primitive variable. These loop links are then labelled by a delay time equal to 1.

#### *5.1.1 Mixed systems*

Given a mixed structure in which both the dynamic part and the static part are self-contained, the causal ordering is simply obtained by applying the causal ordering for dynamic structures to the dynamic part and the one for static structures to the static part. From the implementation point of view, this is obtained by applying the whole model structure to the Ford and Fulkerson algorithm, after having *forced* the entries corresponding the matching for the dynamic part (the variable to be matched to a differential equation is the primitive variable of the derivative of the equation ; idem for recurrent equations). Exogenous variables are accounted for in the static part by as many exogenous equations. The model structure is given by the occurrence matrix $M$ of the system in which some non null entries can be marqued as explained below:

$$m_{ij} = \begin{cases} 1 & \text{if variable } j \text{occurs in equation } i \\ * & \text{if variable } j \text{occurs in equation } i \text{ and must be forced} \\ 0 & \text{otherwise} \end{cases}$$

The algorithm returns the table after having modified some of the entries according to the perfect matching found:

$$m_{ij} = \begin{cases} 1 & \text{if variable } j \text{ is matched to equation } i \text{ by the perfect matching} \\ * & \text{if variable } j \text{ has been forced to equation } i \\ 2 & \text{if variable } j \text{ simply occurs in equation } i \\ 0 & \text{otherwise} \end{cases}$$

From the returned table $M'$, Causalito produces the causal structure and precises which influences (causal links) correspond to every equation. Delays are generated as explained in 5.1.1.

### *5.2 Multiple operating mode systems*

Most of the real systems have several operating modes for they include automatic switches and/or processes that have a different behaviour depending on the operating range. Surprisingly, the problem of causal ordering for such problems has never been discussed so far in the literature. The equational model of such systems is formed of a set of equations among which some have associated conditions defining their operation range. The global causal order associated to such systems hence includes some causal links with associated conditions as well.

This section proposes a formalisation of the problem of generating the causal structure associated to multiple mode systems and provides an algorithm for *generating the causal structure in an incremental way*, taking sequentially into account the different circuit configurations in a local manner. This allows us to optimise the procedure in the sense that, for every operating mode, only the minimal causal sub-graph is re-evaluated. This is obviously much more efficient than a brute force approach which would consist of generating a new graph for every different mode.

*Consistent operating mode.* An operating mode is said to be consistent iff it is logically and physically consistent, i.e. the set of logical conditions defining the operating mode are consistent and they are simultaneously realisable (from a physical point of view).

*Self-containment.* A multiple-mode mixed $S$ system is self-contained iff its static part and its dynamic part are self-contained in every consistent operating mode.

*Incremental causal ordering.* Our algorithm takes as input the equational model structure of S plus a *condition vector*:

- The model structure is given by the occurrence matrix $M$ as in 5.1.1.
- The condition vector V has as many components as equations:

$$v_i = \begin{cases} t & \text{if equation } i \text{ has no condition} \\ C & \text{if equation } i \text{ is submitted to condition } C \end{cases}$$

*Working hypothesis.* The conditions defining the different modes of some physical component (represented by one or several equations) must define

a partition of the parameter subspace spanned by the parameters appearing in these conditions, i.e. they must be mutually exclusive and cover the whole subspace. In logical terms, $(C_i \wedge C_j)$ is false for any $i$ and $j$, and $\bigvee_{i=1}^{n} C_i$ is true. If the above hypothesis would not hold, this would indicate that the whole model has a domain validity restricted to a subspace.

The number of different operating modes for the system can be obtained by multiplying the number of different modes for every physical component. However, one must be careful that among these modes, some may not be consistent. We define a consistent operating mode as a configuration.

The general scheme of the algorithm is as follows: beginning with an initial configuration $C_{onf0} = (C_1 C_2 \ldots C_n)$ and following the method presented in 5.1.1. Causalito generates a first causal graph $G_0$ in the form of Ca~En influences plus their associated activation conditions (the influences corresponding to equations submitted to some condition $C$ have the activation condition $C$). Causalito then switches to another condition and *determines the minimal causal sub-graph which needs to be re-evaluated and the new sub-graph to be added.* $G_0$ is updated and so on until all the configurations have been considered.

*Algorithm.* Assume that the system $S$ in a given configuration — assumed to be the initial configuration without loss of generality — consists of $n$ equations $E_0$ and $n$ variables $V_0$ and consider a configuration change, defined by the fact that a set on conditions change truth values. Let's define the "macro-conditions" $C$ and $C^*$ as the union of the conditions which change truth value from 1 to 0, and from 0 to 1 respectively. Note that this comes back to replacing the equations conditioned by conditions in $C$ by conditions in $C^*$, and that, since the system is self-contained, the occurrence matrix variables/equations is always a square matrix.

Consider the notations given in table 1.

Define $V'_C = V \cap V_C$ and $G'_{C0}$ as the subgraph obtained from $G_{C0}$ by deleting the edges entering the variable-nodes of $V_C$ and the nodes corresponding to $(V_C - V'_C)$ ; i.e. $G'_{C0}$ is the subgraph which remains after discarding $E_C$. Then we have the following result:

*Proposition 7.* The maximal subgraph of $G_{C0}$ that needs to be re-evaluated is the subgraph spanned by $(\Gamma(V'_C) \cup V'_C)$, where $\Gamma(V'_C))$ is the set of successors of the variable-nodes of $V'_C$ in $G'_{C0}$, and the sub-graph to be added is obtained from a perfect matching between $(V'_C \cup \Gamma(V'_C) \cup V_N)$ and $(E_{C*} \cup E_\Gamma)$, where $E_\Gamma$ is the subset of equations in $E_C$ which include variables of $(\Gamma(V'_C) \cup V'_C)$.

| | |
|---|---|
| $E_0$ | set of equations in the initial configuration ; |
| $V_0$ | set of variables occurring in $E_0$ ; |
| $C$ and $C^*$ | macro-condition determining the next configuration change ; |
| $C0$ | perfect matching for the initial configuration (condition $C$ true) ; |
| $G_{C0}$ | causal structure of $S$ in the initial configuration (derived from $C0$) ; |
| $E_C$ | set of equations conditioned by $C$; |
| $V_C$ | set of variables matched to $E_C$ by $C0$ ; |
| $E_{C*}$ | set of equations conditioned by $C^*$ ; |
| $V_{C*}$ | set of variables occurring in $E_{C*}$ ; |
| $V_N$ | set of variables occurring in $E_{C*}$ and not occurring in $V_0$ ; |
| $\bar{E}_C$ | set of equations neither submitted to $C$ nor to $C^*$ ; |
| $\bar{V}_C$ | set of variables occurring in $\bar{E}_C$ ; |
| $E$ | set of equations in the next configuration $(E = \bar{E}_C + E_{C*})$ ; |
| $V$ | set of variables occurring in $E$. |

Table 1: Notations

**PROOF.** This result is easily proved by reasoning about the occurrence matrix $M$ of $S$ in the initial configuration. A perfect matching, in particular $C_0$, associates one variable to one equation and it can therefore be represented on the occurrence matrix by the selection of $n$ entries with the property that there is one and only one selected entry per row and one and only one selected entry per column. Then, every non selected entry mij corresponds to an edge of the causal graph $G_{C0}$ drawn from variable $x_j$ towards the variable matched to the equation $e_i$. ■

We change $C$ into $C^*$. The new perfect matching $C$ can be decomposed in a first submatching Sub-$C_1$1 common to $C_0$ and a second submatching Sub-$C_2$ which is new. We want to determine the minimal set of variables and equations to be rematched (the ones which must be considered for finding Sub-$C_2$), guarantying at the same time that a perfect matching exists among them.

When changing configuration $C$ into $C^*$, the rows of $M$ corresponding to the equations in $E_C$ are replaced by the rows corresponding to the equations in $E_{C*}$. Let's reason with respect to the variables, knowing that a dual reasoning could be done with respect to the equations. In the new occurrence matrix, the entries which remain selected (matches) correspond to the variables in the set $(V - (V'_C \cup V_N))$, where $V_N$ is the set of variables newly introduced by $E_{C*}$.

Therefore, the variables which are not matched are $(V'_C \cup V_N)$.

*Matching* $V_N$. As the variables in $V_N$ only occur in $E_{C*}$, they can only be matched to equations in $E_{C*}$. Hence these matches do not require to modify any of the existing matches.

*Matching* $V'_C$. The variables of $V'_C$ may occur in equations of $\bar{E}_C$ (and eventually in equations of $E_{C*}$). If $x_j \in V'_C$ but does not appear in $\bar{E}_C$ i.e. it only appears in $E_{C*}$, then we are in the same case as for variables in $V_N$. If $x_j \in V'_C$ and occurs in the equation $e_i \in \bar{E}_C$ , then the entry $m_{ij}$ is candidate to be selected for matching $x_j$. But $e_i$ already has a matched variable $x_k$, given by the selected entry in the row $i$. Hence $x_k$ would in its turn need to be matched elsewhere. The entries in column $k$, $msk_0$, $s_i$, are other candidates for matching $x_k$. But the corresponding equations $e_s$ already have matched variables, and so on until no other candidates are found. In $G_{C0}$, $x_k$ is the successor of $x_j$ by the edge corresponding to $m_{ij}$ and the edges corresponding to the $m_{sk}$ bring to the successors of $x_k$, etc. This is illustrated in the figure 3.

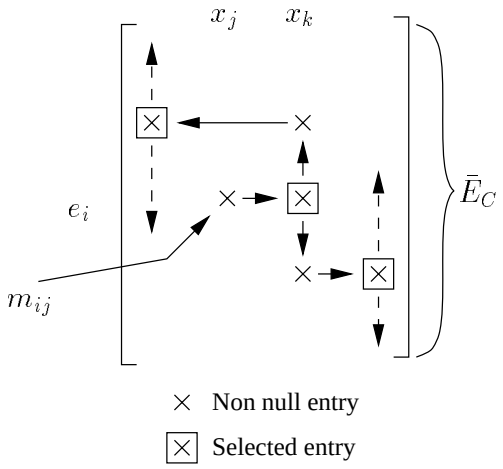



Figure 3: Matching

Therefore, the maximal sub-graph of $G_{C0}$ which needs to be re-evaluated is given by $(\Gamma V'_C \cup V'_C)$ and the new sub-matching Sub-$C_2$ must be searched for between the variables in $(V'_C \cup \Gamma(V'_C) \cup V_N)$ and the equations in $(E_{C*}) \cup E_\Gamma)$, where $E_\Gamma$ is the subset of equations of $\bar{E}_C$ which include variables of $(\Gamma(V'_C) \cup V'_C)$.

The algorithm is as follows:

**1 -** Choose the initial configuration vector. Find a perfect matching $C_0$ for $S$ in this configuration. Derive the corresponding causal graph GC0 and put the delay labels to the causal links.

**2 -** Change configuration by taking the negation of one condition $C$.

**2.1 - IF** $(V'_C \cup \bar{V}_C = \emptyset) \vee V'_C = \emptyset$

**THEN** Label the arcs entering the variables of $V_C$ with macro condition $C$ and find a perfect matching between $(V'_C \cup V_N)$ and $E_{C*}$.

**ELSE a -** Find in $G'_{C0}$ the set $\Gamma(V'_C)$ of all the variables which are successors of variables of $V'_C$ and label the arcs entering the variables of $(\Gamma(V'_C) \cup V_C)$ with condition $C$.

**b -** Determine $E_\Gamma$, the set of equations of which contain variables of $(\Gamma(V'_C) \cup V'_C)$.

**c -** Find a perfect matching between the variables of $(V'_C \cup \Gamma(V'_C) \cup V_N$ and the equations of $(E_{C*} \cup E_\Gamma$.

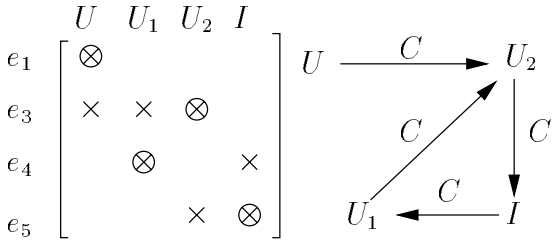
**2.2 -** Update $G_{C0}$ by adding the arcs corresponding to the perfect matching found and label them with macro condition $C^*$. Goto **2**.

The final labelled causal graph is a super-graph which includes all the causal graphs of S in the different configurations[4].

*Example.* Consider the very simple illustrative multiple-mode static system presented in section 4.1 and assume that the switch is connected to $R_2$ if a given condition $C$ is true and to $R_1$ otherwise. The equations are the following:

$$\begin{array}{llcll} (e_1) & U & = & U_0 & \\ (e_2) & U & = & R_1 I & \text{if } \neg C \\ (e_3) & U & = & U_1 + U_2 & \text{if } C \\ (e_4) & U_1 & = & R_2 I & \text{if } C \\ (e_5) & U_2 & = & R_3 I & \text{if } C \end{array}$$

Name the variables $x_1 = U$, $x_2 = U_1$, $x_3 = U_2$ and $x_4 = I$. Let's define $C$ as the initial configuration. Then the occurrence matrix concerns $e_1$, $e_3$, $e_4$, $e_5$ and $x_1$, $x_2$, $x_3$, $x_4$. A possible — this system includes a cycle — perfect matching $C_0$ is indicated by the selected entries and provides the following causal structure $G_{C0}$ (see section 4.1).

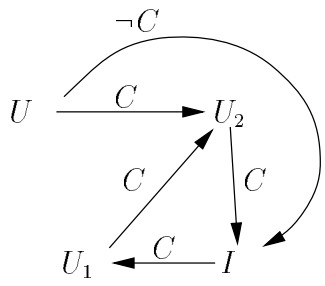



Let's now change $C$ into $C^* = \neg C$. We have $E_C = \{e_3, e_4, e_5\}$, $V_C = \{x_2, x_3, x_4\}$, $E_{C*} = \{e_2\}$, $V_{C*} = \{x1, x4\}$, $\bar{E}_C = \{e1\}$, $\bar{V}_C = \{x1\}$, $E = \{e_1, e_2\}$, $V = \{x_1, x_4\}$ and $V'_C = \{x4\}$. We are hence in

[4] This can be directly used by Ca∼En

the `THEN` case of `2.1`. Therefore, we must find a perfect matching between $\{x_4\}$ and $\{e_2\}$, which is trivial.

The final causal graph is:




Although this is not really computationally significant in the case of this simple system, the benefit of the incremental approach appears clearly as the new perfect matching had to be found between one variable and one equation instead of between two variables and two equations.

Note that the resulting causal structure would have been exactly the same if we had defined by the initial configuration $\neg C$ and that the same branch of the algorithm would have been followed. The next section presents the results obtained with CAUSALITO on the APU application.

## 6. Application of Causalito to the APU gas turbine fuel system

### 6.1 Presentation of the APU fuel system

The APU is a little turbine used as an auxiliary power supply in aircrafts. The one that we considered was designed by the company *Micro Turbo* for *Dassault Aviation* and used in Rafale fighter. Like all turbine systems, it is made of an air supply, a compressor, a combustion chamber, a turbine and an exhaust pipe. It is used on the ground or during flight time to produce electric or pneumatic energy. Our study focused on the APU fuel system which feeds and regulates the APU, providing the fuel from the aeroplane tanks to the injectors with the right pressure and flow, depending on the shaft speed and the aeroplane operating mode.

The APU fuel system is made of the following components (see figure 4) :

- an *inlet fuel filter* eliminates impurities (dust, ice-crystals, etc.) ;
- a *fuel shut-off valve* opens or closes the fuel system ;
- a *check-valve* enables to fill the circuit with fuel at the start ; item a pump provides desired flow and pressure;
- a second *filter* protects the fuel control valve ;
- a *fuel control valve* regulates the fuel flow as a function of the APU operating mode and of the running speed set point ;
- a *differential pressure control valve* maintains constant pressure between the fuel control valve input and output ;
- two *injector rings* spray fuel in the combustion chamber ;
- a *dividing valve* feeds the second injector ring under some pressure condition ;
- a *drainage system* empties the fuel out of the system when stopping.

The APU fuel system global model includes 22 equations for 18 internal variables and 4 exogenous ones. The variables appear in the figure 4. Three pressure conditions define the operating modes of this system:

$$\begin{array}{ll}(C_1) & P_2 - P_3 < 0 \\ (C_2) & P_{cp} - P_2 > 0 \\ (C_3) & P_{c1} - P_{c2} < t\end{array}$$

Three component models have been chosen as examples. For more information the reader can refer to [12].

*First inlet fuel filter*

Equation (1) : $Q = k_1 S_1 \sqrt{P_{gav} - P_1}$

where $Q_1$ is the fuel flow through the filter, $S_1$ is the pipe section, $P_{gav}$ and $P_1$ are pressures and $k_1$ is an intrinsic parameter.

*Pump*

Equation (3) : $Q_p = [k_3 N(1 - k_4(N - k_5))] - (k_6 P_3^{1.5}(1 - k_7))$ if $(C1)$

Equation (3') $Q_p = 0$ if $(\neg C1)$

where $Q_p$ is a flow, $N$ is the rotating speed, $k_3$, $k_4$, $k_5$, $k_6$ and $k_7$ are intrinsic parameters and $P_3$ is a pressure. The condition $(C_1)(P_2 - P_3 < 0)$ is *true* when the pump is functioning and *false* when the circuit is being filled (the check valve is opened).

*Dividing valve*

Equation (14) : $Q_{r2} = 0$ if $(C3)$

Equation (14') : $P_{c1} - Pc2 = k_{22}$ if $(\neg C3)$

where $Q_{r2}$ is the flow through the dividing valve, and $P_{c1}$ and $P_{c2}$ are the pressure in the first and second injector ring. The condition $C_3$ is *true*

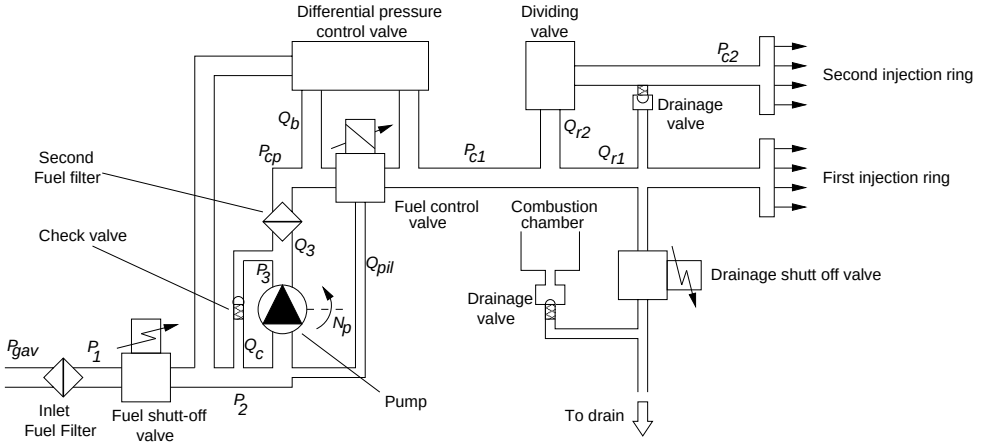



Figure 4: APU fuel system

when $P_{c1} - P_{c2} > \tau$ and it means that only the first injector ring is functioning ($\tau$ is the boundary pressure value for the opening of the dividing valve). When $C_3$ is false, the two injectors rings are functioning simultaneously.

Although none of the 8 possible operating modes are logically inconsistent, an analysis of the physical system shows that $\neg C_1$ implies $C_3$. We then have only six consistent operating modes (configurations).

### *6.2 CAUSALITO results*

*Input.* The input of the CAUSALITO module is a file containing the list of the variables, the list of equations, the *occurrence matrix* and the conditions associated with the equations. The model of the APU fuel system is hence given as follows (all the equations are not represented):

```
P1 P2 P3 Pcp dp Pc1 Pc2 Q1 Q3 Qb
Qpil Qinj Qr1 Qr2 Qp Qc eps S'
Pgav N ISV Pc
(1)   10000001000000000001000 t
(3)   00100000000000010000100 C1
(3')  00000000000000010000000 C1-
(5)   00110000100000000000000 t
(11)  01010000001000000000000 C2
(11') 00000000001000000000000 C2-
(12)  00000100000010000000001 t
(14)  00000000000000100000000 C3
(14') 00000110000000000000000 C3-
(22)  00000000000000000000001 t
```

We can easily identify the equations (1), (3), (3'), (14) and (14') as described before.

*Output.* The output of the CAUSALITO module is the following ('apu' is the name of the file containing the model):

```
Reading file apu: 22 variables,
26 equations, (4 exogenous equations are
added)
3 conditions, 8 operating modes.
Processing perfect matching for each
configuration:
C1 C2 C3... done.
-C1 C2 C3... done.
-C1 -C2 C3... done.
C1 -C2 C3... done.
C1 -C2 -C3... done.
-C1 -C2 -C3... done.
-C1 C2 -C3... done.
C1 C2 -C3... done.
Generating causal influences:
(partial results)
Equation (1): Q1 Pgav --> P1
Equation (3): P3 N --> Qp if C1 true
Equation (3'):
Equation (5): Pcp Q3 --> P3
Equation (11): P2 Pcp-->Qpil if C2 true
Equation (11'):
Equation (12): Qr1 Pc --> Pc1
Equation (14):
Equation (14'): Pc1--> Pc2 if C3 false
Equation (22) :
Generating causal graph: ...
```

The causal graph is presented in figure 5.

## 7. Conclusion

This paper presents the algorithm that has been developed for generating the causal structure of a Ca~En model from the available knowledge in form of a set of equations. The main originality of our algorithm compared to existing work is that it copes with systems that have several operating modes. To do so, it performs the generation of the causal graph in an incremental way.

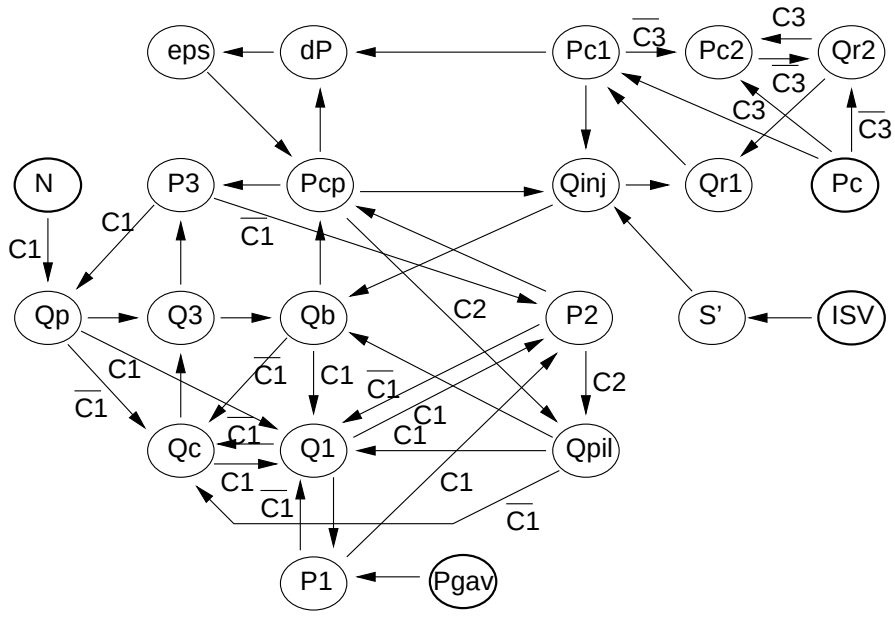



Figure 5: APU causal graph

The benefits of the incremental approach are clearly shown by the presented example. Indeed, if we consider that at each step corresponding to a configuration change the computational effort is proportional to the dimension of the perfect matching to be found, we successively evaluate gains of 12.5% (dimension 14 instead of 16) , 93.7% (dimension 1 instead of 16), 18.7% (dimension 13 instead of 16), 6.25% (dimension 15 instead of 16) and 31.25% (dimension 5 instead of 16). This indicates an average gain of 32.5%, which is significantly interesting.

However, there are still open questions that we are currently investigating. Is there an impact of the order in which the configurations of the system are considered on the resulting causal structure ? Which specific order would then result in a minimal causal structure (in terms of the number of causal links)? Moreover, we perceive that there might be some conditions about the connexity of the system and/or the dependency properties of the conditions defining the different configurations under which it would not be necessary to go through all the configurations to obtain the full global causal structure. The work is hence going on.

## References


[1] K. Bousson and L. Travé-Massuyès. Putting more numbers in the qualitative simulator caen. Hamburg-Harburg, Germany, 1994. Int. Conf. on Intelligent Systems Engineering.

[2] K. Bousson, L. Zimmer, and L. Travé-Massuyès. Causal model-based diagnosis of dynamic systems. New Paltz (NY), USA, 1994. 5th Int. Workshop on Principles of Diagnosis.

[3] de Kleer and Brown. Theories of causal ordering. *Artificial Intelligence*, 29(1-3):33–62, 1986.

[4] R.L. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian J. of Math.*, 8, 1956.

[5] S. Iwasaki and H. Simon. Causality in device behavior. *Artificial Intelligence*, 29(1-3):3–32, 1986.

[6] S. Iwasaki and H. Simon. Theories of causal ordering : reply to de kleer & brown. *Artificial Intelligence*, 29(1-3):63–67, 1986.

[7] S. Iwasaki and H. Simon. Causality and model abstraction. *Artificial Intelligence*, 67(1):143–194, 1994.

[8] B. Lucas. *Méthode d'aide à la modélisation par graphes de liaison et utilisation pour le diagnostic qualitatif de systèmes physiques.* PhD thesis, Paris 11 University, Orsay, 1994.

[9] N. Porté, S. Boucheron, S. Sallantin, and F. Arlabosse. An algorithmic view at causal ordering. Paris, France, 1986. Workshop on Qualitative Physics QR'86.

[10] P. Rose and A. Kramer. Qualitative analysis of feedback. Anaheim, CA, USA, 1991. 10th AAAI.

[11] J. Top and H. Akkermans. Computational and physical causality. pages 1171–1176, Sydney, Australia, 1991. 12th IJCAI.

[12] L. Travé-Massuyès and T. Escobet. Apu fuel system model. Technical report, LAAS - CNRS. TIGER Deliverable D523.3, ESPRIT Project 6862.

[13] L. Travé-Massuyès and R. Milne. Diagnosis of dynamic systems based on explicit and implicit behavioural models : An application to gas turbines in esprit project tiger. *Applied Artificial Intelligence Journal*, 10(3), 1996.