Hierarchical causal model for complex plant supervision

Sylviane GENTIL[1], Pascal DZIOPA[2], Jacky MONTMAIN[2]

[1]LAG - CNRS-INPG-UJF - BP46 - 38402 Saint Martin d'Hères Cedex FRANCE
[2] LGI2P - URC EMA-CEA - site EERIE - 30035 Nîmes Cedex 1 FRANCE

Abstract : The work presented in this paper is devoted to intelligent on line supervision tools. The proposed approach attempts to structure information about the process with multiple hierarchical models. This method has been applied on a nuclear fuel reprocessing plant. The models used are represented by graphs which combine causality and dynamics. The conception with a top-down analysis of a hierarchy of functional causal models of the process is discussed. Then a method for constructing a higher-level graph online at the operator's request from the most detailed graph, is presented.

Key-words : Multi-models approach - Causal Modeling – Graph Hierarchy - Supervision. Proc.pp..2451-2456

Introduction

In response to the increasing complexity of modern automated processes, the activities of the human operator responsible for overall process supervision have been redefined. In the event of a system malfunction or a change in the operating mode, the operator's activities include data monitoring and information seeking, pattern recognition, diagnosing, planning and acting on the system.

In order to favor human-machine cooperation at all times, the various reasoning models used by the operator may be assisted by different models of the supervised facility.

In industrial applications, control engineering is restricted to the use of models, usually in the form of many differential equations, to predict the behavior of the process. These models are restricted to numerical descriptions and are practically useless for diagnosis or action advice tasks.

It may be sufficient in some cases only to know that a relation exists among variables [1]. In other cases, it is desirable to specify the orders of magnitude of the phenomena. In fault diagnosis during transient system behavior, the operator must interpret new events that dynamically modify the process behavior.

Temporal reasoning and qualitative reasoning are fundamental processes leading to decision-making [2]. As diagnosis refers to the process of identifying the cause of an event, causal interpretation, in the sense of temporal precedence of events, is a fundamental tool for representing cognitive mechanisms of human understanding of physical phenomena [3]. The approach discussed in this paper thus focuses on dynamic causal modeling of processes.

A good way to cope with complex systems is to structure the information, in an abstraction hierarchy [4]. The multiple models of the hierarchy must be homogeneous to generate a coherent description of the system behavior.

The model developed in this paper is functional, not structural, as defined by Iwasaki [5]. This choice appears better suited for long-term planning, reasoning about the evolution of physical phenomena, perception and interpretation of a malfunction.

Section 2 describes the conception of causal functional models, represented by graphs. Section 3 discusses a method for developing a top-down graph hierarchy. A method is then indicated for constructing a higher-level graph online from the most detailed graph, at the operator's request, while conserving the semantics of the latter. An example of top-down construction of a causal graph hierarchy for a nuclear process is provided and used to illustrate the discussion.

2 Conception of Causal Graphs

A graph is a knowledge representation structure consisting of nodes interconnected by arcs. A graph is "causal" if the semantics of the arcs

represent the property of causality, i.e. if an input node of an arc is one of the causes of the output node. In the following discussion, the nodes represent process variables which are not necessarily measured ; the arcs represent the functions relating the variables.

2.1 Application plant

The process unit under consideration is a pulsed column facility, highly instrumented for safety reasons. The facility comprises two head-to-tail coupled pulsed columns and their feed systems. The two main product streams contain an organic phase and an aqueous phase.

A pulsed column is a liquid-liquid extractor used to separate uranium and plutonium, from fission products. The input aqueous stream includes two flows (CQ0500 and QG100); it is mixed with the organic solution (CQ1010) in the shaft of the column. The output acid solution (QE120) contains only the fission products. The output organic solution (QG600) is removed by overflow. Extraction requires suitable mixing of the aqueous and organic phases estimated by BETAE, the column retention, an unmeasurable variable useful for supervision, reconstituted from other measurements.

The mixture is subjected to a periodic pulsation pressure (CPRE801). The physical boundary between the two phases (NIRE), is regulated by modulating the aqueous outlet flow rate (QE120). A causal graph was constructed representing the column hydraulic balance, using variables CQ0500, QG100, CQ1010, QG600 and QE120. The balance of the aqueous phase and of the solvent, the balances between the inflow rates and the solvent outflow rate and between the aqueous phase outflow rate and the overflow rate result in a graph with six arcs (Fig. 1).

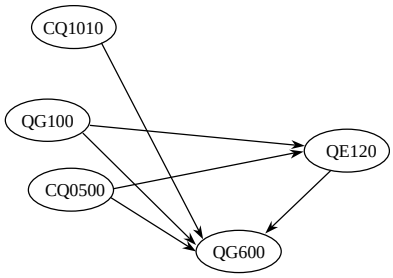

Fig.1 The mass balance graph

2.2 Arc Dynamics

In addition to causality, represented by the direction of the arc, each arc is assigned a semantic. The arc represents the transfer function between the input variable and the output variable, and the causal graph becomes the equivalent of the process block diagram [6].

In the following discussion, the arcs of the graph will be assigned very simple transfer functions to provide an approximate description of the process behavior in the nominal operating mode. That is largely sufficient for supervision purposes.

Three types of classic transfer functions are used: the Strejc function (1) [7], the differentiator (2) and the integrator (3). Parameters are the gain g, the delay *d*, the time constant *T* and the order *n*.

$$F(s)=\frac{g\cdot e^{-d\cdot s}}{(1+T\cdot s)^n} \qquad n\in\{0,1,2,3,4\} \quad (1)$$

$$F(s)=\frac{g\cdot s\cdot e^{-d\cdot s}}{(1+T\cdot s)^n} \qquad n\in\{0,1,2\} \quad (2)$$

$$F(s)=\frac{g\cdot e^{-d\cdot s}}{s} \quad (3)$$

3 Graph Hierarchy

3.1 Definition of Graph Hierarchy

The operator monitors the process by regularly observing a relatively small set of variables, which defines the high-level graph in the hierarchy. When the operator observes the situation is no longer normal, (s)he will focus on a particular subsystem to verify hypotheses or identify means of action if the problem is correctly understood. (S)he may also simply display more variables to understand the situation and to follow its evolution.

A graph G_2 at a lower hierarchical level than graph G_1 must therefore contain the nodes of G_1 together with a number of other nodes used to take more phenomena into account. Consequently, the detailed graph may contain new sources (e.g. a regulation setpoint), new sinks (e.g. additional measurements), or intermediate variables (for a more detailed perception of some phenomena); in the last case, an arc in the higher-level graph is broken down into a path in the lower-level graph.
As the graph arcs represent transfer functions, this hierarchical breakdown procedure consists in revealing internal variables by breaking down a complex transfer function into a product of elementary transfer functions. These internal variables must be meaningful to the human operator.

3.2 Conception of a Graph Hierarchy

A top-down approach is adopted to construct the causal graph hierarchy *a priori*. The construction is initialized by graph G_0, including a small set of variables used for normal supervision. These variables correspond to normal overall operation of the facility, reflecting mass balances or energy balances, for example. The highest level graph built for the nuclear fuel reprocessing plant represents the column hydraulic balance (Fig.1).

Additional variables are then introduced. A complex function may be broken down into elementary functions ; an arc is thus replaced by a path. The representation may also focus on a particular function, such as regulation. In this case, at a high level of abstraction, it is assumed that a regulated variable is equal to its setpoint. In order to monitor transients or diagnose regulation malfunctions, the arcs relating the setpoint and the disturbances to the regulated quantity and to the action must be developed.

In the example, the second level of the hierarchy includes the retention (BETAE) and the pressure (CPRE801). The second-level graph contains 7 nodes and 11 arcs (Fig.2).

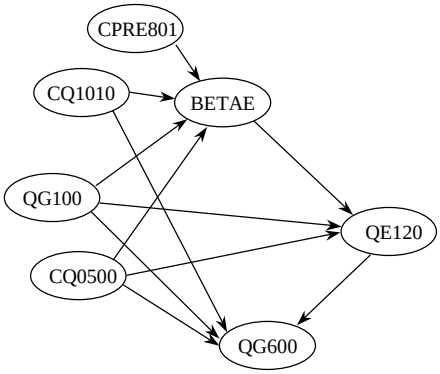

Fig.2 The second level

The third hierarchical level includes the column interphase level (NIRE) regulation. CNIRE is the setpoint ; the level is controlled by QE120. The column inflow rates disturb this control variable. Internal model variables (identified by DELTAQE) are created to allow for the disturbance causality (Fig.3).

Adding increasing numbers of variables, an increasingly detailed graph is obtained step by step until the graph contains all the variables useful for supervision purposes. This corresponds to the step of knowledge extraction and representation necessary for the development of any intelligent system.

3.3 On-line Graph Construction

The situation is very different when this graph is used online by a human operator. In order to construct a graph online, it is necessary to identify the paths of the low-level graph that will become arcs in a higher-level graph. This process must also ensure that no relevant information from the detailed path is lost.

3.3.1 Iterative Reduction of a Graph

The operator generally begins with graph G_0 and wishes to focus on a particular subsystem according to the context : i.e. either display additional measurements or detail a function. It is thus impossible to specify any *a priori* relevant hierarchical levels. Construction of the detailed graph is context-dependent. This approach is at the heart of human-machine cooperation, and attempts to favor operator intelligent behavior. The required graph is constructed from the definition of its hierarchical level (current graph variables and additional operator-requested variables), and is obtained online from the lowest-level (most detailed) graph.

After validating the structure of the desired graph, by identifying any circuits, all the non relevant nodes must be eliminated. When several nodes are to be eliminated from a graph, they must be deleted one at a time in an iterative process [8].

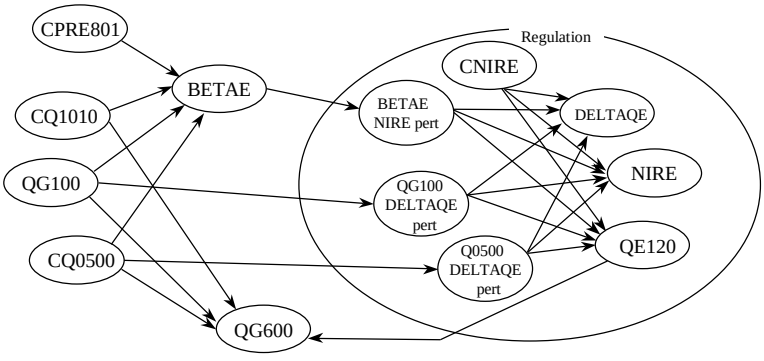

Fig.3 The third level graph with interphase level regulation

3.3.2 Path-Seeking in a Graph

Path algebra is highly effective in identifying the properties of a graph. This method uses a matrix representation of the graphs along with a suitable sum $\oplus$ and product $\otimes$. It can be used to find the number of paths between two graph variables, or to enumerate them, by a simple matrix product [9].

In order to enumerate paths of a graph, the arcs must be assigned semantics, referred as "names" in the following (Fig.4). The "name" may be a sign, a character or a transfer function. Enumeration, for each path length, yields aggregation of semantics, corresponding to the aggregation of the arcs in the path.

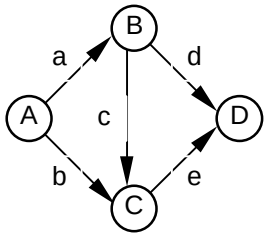

Fig.4 a graph example with named arcs

The graph in Fig.4 is represented by matrix N_V when the variables are arranged in the order given by vector V (4). The matrix of names N_V contains all the path names of length 1 in the graph. For example, $N_V(2,1) = a$ represents the arc of semantic a from node A to node B.

$$N_V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \boxed{a} & 0 & 0 & 0 \\ b & c & 0 & 0 \\ 0 & d & e & 0 \end{bmatrix} \qquad V = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \quad (4)$$

The paths of length n are obtained by calculating the n^{th} power of matrix N_V. It is important to note, however, that the concatenation is not commutative, as the arcs are directed. Thus, in the product of matrix $B = (b_{ij})$ by matrix $C = (c_{ij})$, the element of the resulting matrix A, $a_{ij} = \sum_k c_{kj} b_{ik}$. Similarly, $N_V^n = N_V \times N_V^{n-1}$

By calculating the square and cube of N_V, four paths containing two arcs and one path containing three arcs can be named (5,6). There are no paths with four or more arcs, because all elements of the fourth power of N_V are zero. For example, the path of length 3 between A and D is "ace".

$$N_V^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ ac & 0 & 0 & 0 \\ ad+be & ce & 0 & 0 \end{bmatrix} \quad (5)$$

$$N_V^3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \boxed{ace} & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

A circuit is identified simply when a non-zero element appears on the diagonal of a matrix. Circuits have to be treated carefully, because it is not possible to suppress all the nodes they contain, without loosing the meaning of causality [8].

The total number of paths is obtained by the limit of the sum N_V^* of all the powers of matrix N_V (7). Element $N_V^*(4,1)$ indicates three possible paths between A and D.

$$N_V^* = \sum_{n=1}^{\infty} N_V^n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ b+ac & c & 0 & 0 \\ \boxed{ad+be+ace} & d+ce & e & 0 \end{bmatrix} \quad (7)$$

This method is capable of finding the semantics of all the paths relating two variables in a graph, while respecting the direction of the arcs and thus the causality.

The elimination of a node from a graph is based on the product of the graph matrix and a matrix containing only the influence of the node to be eliminated. This matrix product is used to calculate a matrix in which the influence of an eliminated node is taken into account by the links between the remaining nodes. An iterative algebraic method has been defined [8].

3.3.3 Operators on Transfer Functions

As the arcs are assigned transfer functions, compatible operators $\otimes_F$ and $\oplus_F$ must be defined. As these functions are described by four parameters (gain, time constant, delay and order), a direct combination (in the sense of conventional transfer function calculations) of only two of them would already involve eight parameters. This accumulation of parameters is contrary to the required simplicity of the model, and must

therefore be reduced without excessive effect on the precision of the simulations.

In order to evaluate the four parameters of the approximate transfer function, the response of the classical transfer function sum (and product) and the response of the approximation to a step input have been analyzed. The integral error between the responses has been minimized.

3.3.3.1 Defining the product Å $_F$

The product f of two transfer functions f_1 and f_2 corresponds to the reduction of a path made of two arcs in series to obtain one single arc. It is clear that, for arcs in series, the total delay is the sum of the delays of each arc $d=d_1+d_2$, and the gain is the product of the gains of each arc $g=g_1.g_2$. It is then sufficient to determine the corresponding time constant T and the most suitable order n.

$$T = \frac{n_1 \cdot T_1 + n_2 \cdot T_2}{n} \tag{8}$$

$$\begin{cases} n_1 \cdot T_1 >> n_2 \cdot T_2 & \Rightarrow \quad n = n_1 \\ n_1 \cdot T_1 << n_2 \cdot T_2 & \Rightarrow \quad n = n_2 \\ n_1 \cdot T_1 \approx n_2 \cdot T_2 & \Rightarrow \quad n = n_1 + n_2 \end{cases} \tag{9}$$

When at least one of the functions f_1, f_2 is integral or derivative, the product is integral or derivative unless the second function is of the opposite type. The product of two differentiators yields a negligible output ; this approximation is perfectly plausible, considering the signals actually recorded for industrial processes (generally step or ramp signals). Moreover, it appears unnecessary in practice to use values of n greater than 4 for Strejc functions or greater than 2 for differentiators.

3.3.3.2 Defining the sum Å $_F$

A causal graph gives the cause-effect relations between two variables ; it is therefore impossible for two parallel arcs to originate from the same variable and lead to the same variable. It is possible, however, for two parallel paths to meet. As the graph reduction is iterative, the approximation of the combination of two parallel paths ends with the approximation of a sum of two transfer functions.

The integrator is considered dominant ; differentiators and Strejc functions can be disregarded when one is present. Similarly, a differentiator is systematically ignored in the presence of a Strejc function. Only the sums of transfer functions of the same type should be examined. It is easily observed that two paths with different delays cannot be combined : if a system gives two responses with a time lag to the same variable, it cannot normally be considered as a single relation. In such a case, either one of the paths must be ignored in favor of the other or the graph structure must be revised to prevent the combination of these two paths.

Now consider two paths with identical delays. The overall gain is equal to the sum of the gains $g=g_1+g_2$. The sum of two first orders $n_1 = n_2 = 1$ has three different shapes, in addition to the trivial case of the difference of two strictly identical transfers. The results are similar for higher-order transfers and for the differentiators. If the gains are both of the same sign, the response is similar to a first-order one (10).

$$\begin{aligned} T &= \frac{Max(n_1 \cdot T_1, n_2 \cdot T_2)}{n} \\ g &= g_2 + g_1 \\ n &= Min(n_1, n_2) \quad g_2 \cdot g_1 < 0 \end{aligned} \tag{10}$$

If the gains are of different signs and different absolute values, the characteristic “hump” of a non-negligible zero appears. This phenomenon affects only part of the transient, and is therefore disregarded (11). The sum of these Strejc functions is approximated by a Strejc function of the lowest order. This method retains only the simplest dynamics.

$$\begin{aligned} T &= \frac{n_1 \cdot g_1 \cdot T_1 + n_2 \cdot g_2 \cdot T_2}{n \cdot g} \\ g &= g_2 + g_1 \\ n &= Min(n_1, n_2) \quad g_2 \cdot g_1 > 0 \end{aligned} \tag{11}$$

If the gains are opposite, the response is similar to a second-order differentiator (12). However, if the overall gain g_1+g_2 is zero for the sum of differentiators, the resulting behavior is negligible.

$$\begin{aligned} T &= \frac{Max(n_1 \cdot T_1, n_2 \cdot T_2)}{2} \\ g &= n_1 T_1 g_2 + n_2 T_2 g_1 \\ n &= 2 \end{aligned} \tag{12}$$

Two integrators are approximated simply by considering them as an integrator with a gain equal to the sum of the gains g_1+g_2.

3.4 Application

Under steady-state conditions, when the pulsed column facility is operating in the nominal mode, the operator merely supervises the hydraulic balance, the retention and the interphase level. The default graph should display the following variables: CQ0500, CQ1010, QG100, QE120, QG600, BETAE and NIRE. The detailed graph is reduced to obtain the required graph (Fig.5).

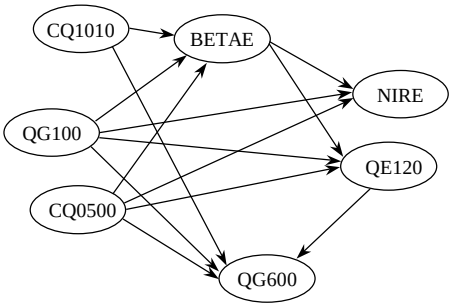

Fig.5 The graph used under steady state conditions

4 Conclusion

The supervision of complex processes requires co-operative assistance tools. Causality is one of the fundamental aspects of operator reasoning. Time-driven reasoning is also of use in supervision, and is combined in this work with causal reasoning by adding time parameters to the arcs of the directed graph. The transfer function is the code that has been attached to an arc. Considering the variety of operator tasks required for satisfactory process operation, it appears unreasonable to use a single graph to represent an entire plant ; hence the proposed construction and use of a directed graph hierarchy.

The construction of this hierarchy *a priori* involves a top-down approach similar to the analysis process used by a human being. First a very high-level graph structure is established, indicating only the cause-effect relations between the most relevant variables for the process. More detailed graphs are then constructed by adding lower-level variables. In order to ensure consistent temporal dynamics in all the graphs, only the parameters of the most detailed model in the hierarchy are estimated once and for all.

During supervisory system operation, the required model is constructed online at the initiative of the operator, who selects the relevant variables in the routine situation. Path algebra provides an elegant tool for representing a graph by a matrix, counting the paths, listing them by name, or identifying loops. A procedure to eliminate each of the non-relevant nodes from the detailed graph in an iterative manner is used to obtain the required model.

The transfer functions of the detailed model must then be merged online to obtain the parameters of the desired model. In order to obtain the matrix product required by path algebra, the sums and products of the transfer functions were redefined to provide approximations conserving the delays, static gains and response times.

The work discussed in this paper is a preliminary approach to hierarchical modeling of complex processes for supervision purposes. This work assumes the operator is capable of considerable autonomy in choosing the abstraction level. This assumption is valid except under conditions of stress that may impair an operator's judgment. It will no doubt be necessary to add guidelines to variable selection.

References :

[1] J. Shiozaki, H. Matsuyama, K. Tano and E. O'Shima, Fault diagnosis of chemical processes by the use of signed directed graphs, *International Chemical Engineering*, Vol. 25, No. 4, 1985, pp. 651-659.

[2] P. Cacciabue, F. Decortis, B. Drodzdowicz, H. Masson and J.P. Nordvik, Cosimo : a cognitive simulation model of human decision making and behavior in accident management of complex plants, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 5, 1992, pp. 1058-1074.

[3] J. Rasmussen, Diagnostic reasoning in action, *IEEE Transactions on Systems Man and Cybernetics*, Vol. 23, No. 4, 1993, pp. 981-991.

[4] J. Rasmussen, The role of hierarchical knowledge representation in decision making and system management, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 2, 1985, pp. 234-243.

[5] Y. Iwasaki, Reasoning with multiple abstraction models, *4th International Workshop On Qualitative Physics*, Lugano (C.H.), pp.183-194, 1990.

[6] L. leyval, S. Gentil and S. Feray-Beaumont, Model based causal reasoning for process supervision, *Automatica*, Vol. 30, No. 8, 1994, pp. 1295-1306.

[7] V. Strejc, Détermination approchée des caractéristiques de régulation d'un processus à réponse apériodique, *Automatisme*, Vol. V, No. 3, 1960, pp. 109-111.

[8] P. Dziopa, *Représentations Multi-Modèles pour la supervision*, Ph.D.thesis : Institut National Polytechnique de Grenoble, 1996.

[9] M. Gondran and M. Minoux, Graphes et algorithmes, *Collection de la Direction des Etudes et Recherches d'Electricité de France*, Eyrolles, Paris, 1995.