Supervisory control of discontinuous systems: an hybrid modelling based approach

Jean-claude PASCAL, David ANDREU[2]

LAAS-CNRS 7 avenue du colonel Roche 31077 Toulouse Cedex 4 FRANCE
[2] LIRMM 161 rue ADA 34192 Montpellier Cedex 5 FRANCE

Abstract : This paper focuses on an hybrid modelling based approach for supervisory control of discontinuous processes. We propose an hierarchically organised architecture for the control system : a hierarchy of local area networks for communication purposes and a hierarchy of models for control and monitoring purposes based on the necessity to take into account hybrid aspects at the different hierarchical levels. At each level, the structure of the control system is organised into two sub-models: a discrete-event one, called the control model, which specifies the control policies and an hybrid one, called the reference model, which described the expected process behaviour. The latter model, based on the coupling of Petri nets and differential algebraic equations, represents the (discrete) process configurations to which are associated subsets of equations describing the process continuous behaviours. This model of the industrial process, which is run in parallel and in real-time during the operations, mainly permits to detect process failures.

Key-words : Hybrid system modelling, Petri nets, Supervisory control.
CSCC'99 Proceedings: - Pages 2481-2486

1 Introduction

Discontinuous systems, like batch processes, closely link continuous and discrete aspects. The design of the control system must necessarily take this hybrid aspect into account. This leads to use hybrid modelling and has a consequence on the definition of the architecture of the control system.

The control is all the more delicate since the needs of production (great diversity of products) require a great flexibility of the system. Flexibility can be ensured by a modular and hierarchical approach. The standard ISA/SP88 [1] proposes a representation of the batch control system according to several hierarchical views. Based on this standard, few approaches [2], [3] are functional approaches built around a centralised data base. The hierarchy is not really respected. Others approaches [4], [5], [6] are purely and simply discrete approaches and only consider the continuous aspect at the process interface level. The hybrid aspect must be necessarily taken into account at the different hierarchical levels, mainly for monitoring and/or optimisation considerations.

The first part is devoted to the presentation of the hierarchical and modular decomposition of the control system. The accent is placed on the necessity to consider the hybrid aspect of the process on the whole hierarchy. In the second part, the communication architecture based on a hierarchy of local area networks is presented. The following part explains the design of the control system based on an hybrid modelling. For monitoring purposes, an hybrid model of the process is introduced. In the last part, the uses of this model are explained.

2 An hierarchical decomposition

To face the process complexity and to ensure flexibility, our approach is based on a decomposition in three hierarchical levels : supervision, coordination and local control. We apply the classic Discrete-Event Systems hierarchical and modular decomposition.

At the supervision level, the supervisory control drives the execution of the operation sequences described by the estimated production schedule at the scheduling level. It must manage in real-time plant resources and adapt the control when a deviation is detected. This adaptation can lead to modify process plan in interaction with the scheduling level, or to modify operating conditions in interaction with the local control levels.

Firstly, the supervisory control has to detect the end of each phase and to drive the production system from dynamic configurations to dynamic configurations in order to achieve the fabrication of the product. It manages configuration changes of the process (typically composed of a set of units connected by storage tanks) and establishes temporary links between units (so called process configuration). Indeed, in a batch operated plant the process equipment is not in continuous use, units are used sporadically with different status. At different instants of time, the plant is in different configurations, each configuration corresponding to a set of active connections between units.

The coordination level manages the configuration changes within the unit by establishing temporary links between equipments of the unit.

The local control level, the lowest level of the hierarchy, is constituted by local automation devices. Its continuous part consists of regulators (PID) and its discrete one to Programmable Logic Controllers (PLC).

It is evident that the supervision and coordination levels must have a discrete view of the process. The problem is to determine the levels which need the representation of the continuous aspect. At the higher level, the elaboration of control laws or the decision making for resource allocations often use non measurable variables (quality, concentration, …) calculated from the evolution of the continuous variables of the process. The quantities of matter to treat can change according to the equipment sizes. It is necessary to make mass balances implying continuous variables. For instance, for the transfer of a batch in a storage tank which is in a decanting phase, it is necessary, before to transfer, to ensure that the vessel volume constraint will be respected. This calculus is made from the current volume of the storage tank, by taking into account the volume of product to be transferred and the evolution rates of the loading and the decanting phases. This is an illustration of a supervisory purpose which needs to consider the continuous aspects. An other possible illustration lies on monitoring purposes.

The control system must include a monitoring function to detect process deviations in order to define a process recover. So, monitoring is based on the notion of a model of the industrial process which is run in parallel and in real-time during the operations : a reference model. This model represents the expected behaviour of the process, allowing to make a comparison with the real behaviour and permits the detection of process failures. The use of a process model for monitoring is usual in discrete-event systems [7], [8] as continuous ones [9], [10], [11].

In the case of batch systems, the process is hybrid. So then, the model must be hybrid. Firstly, the comparison mechanism must compare the occurrence date of the process event with a temporal window which gives the possible dates for the process event occurrence. This temporal window can not be beforehand defined by the estimated schedule as in manufacturing systems. Indeed, it depends on the real-time continuous evolution of the process : the operating durations can highly change from one batch to the next one (the sensitivity to the environment perturbations or the change of quality of the products frequently lead to corrective actions). Secondly, monitoring considerations impose also to monitor the values and the evolution rates of the continuous variables ; the volume of a storage tank which decreases while being in a loading phase traduces an abnormal functioning for instance. The hybrid aspect of the process thus requires the representation of the continuous phenomena. From these illustrations the use of an hybrid model at different levels of the hierarchy is pointed out. The control system is consequently structured in a hierarchy of hybrid models.

Before to present the modelling let us consider the architecture of the control system.

3 The communication architecture

In a batch plant, local automation devices (PID, PLC) are currently piloted by a supervisor and linked to a set of equipment. The frequent changes of the control strategy of the production, the resulting changes of configurations and the continuous nature of the product impose the dynamic coupling of discrete actions and continuous controls which are not necessarily linked to a particular equipment (for example, a transfer between two units may need to isolate the corresponding transfer line from the rest of the plant by disabling other lines). For this reason, it is necessary that the higher levels of the control system have the access to any variable of the process. So, the communication system must permit it.

The control of batch processes involves continuous controllers operating in a periodic way on sampled data and discrete controllers operating in an asynchronous way on sporadic events.

In order to guarantee the temporal consistency and spatial, the sporadic data flow is separated from the periodic one [12], [13]. So the proposed architecture is based on the use of two buses : a field bus (of "continuous nature" carrying continuous and discrete measures) and a cell network (of "discrete-event nature" carrying messages and events). The first one links regulators and PLC to the process and the second one links these automation devices to the higher levels (Fig.1).

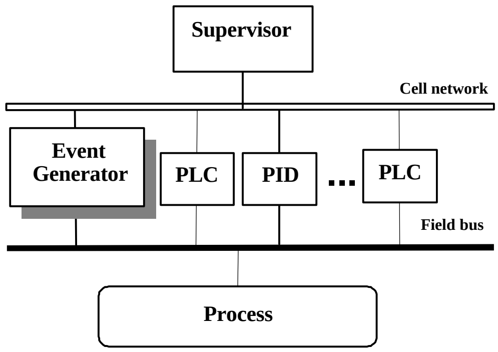

Fig.1 The communication architecture

A so strict hierarchy is not realistic. In batch systems, the production strategy is often referred to production volumes to be reached. This corresponds for the supervisory controller to the detection of state events (mainly corresponding to threshold crossings by continuous process variables). This detection depends on the evolution of continuous process variables which are available on the field bus. The supervisor can not directly access this bus. So the continuous to discrete-event conversion is located and necessarily done at the low level. A functionality is required to create a relation between the process on one hand and the supervisory controller on the other. Thus we have defined a new function : the Event Generator [13]. It receives all the sample values of the continuous state variables by means of the field bus but it only retransmits the significant ones to the supervisory controller via the cell network (Fig.1), according to event generation rules which define the events to generate (for instance, « generate an event when the volume measure reaches 3 m^3 »). The generated event (denoted process event) leads the supervisory controller to evolve in a new discrete state (corresponding to a new process configuration) and to detect new process events. The generating of these events implies a new set of continuous variables with associated event generation rules. So, the event generator must be dynamically programmed by the supervisory controller.

4 The modelling

We have pointed out the necessity of an hybrid modelling. So, the model must include both discrete-event and continuous aspects. Each of these « worlds » has a different mathematical framework for its description ; differential algebraic equations, state-space equations or Bond Graph are commonly used for the continuous aspect, and finite state machines, Grafcet or Petri nets for the discrete one.

Several approaches try to combine these two frameworks into one. It is the case for [17], [18], [19] which integrate the continuous aspect into Petri nets. Another type of approaches uses Petri nets associated with differential algebraic equations [17], [18], [19]. We think is better to combine the use of two well-established theories, each one being well suited to describe one aspect of the system, rather than to elaborate a complex new approach for each specific case. So, we have established a cooperation between the Petri net model of the discrete aspect of the system and the continuous model which is a set of differential algebraic equations. This has lead to the definition of a new model : Differential Predicate-Transition Petri net (DPTP) [20].

Thus, Petri nets are used to describe the discrete control at the different hierarchical and algebraic differential equations are used to describe the continuous behaviour (chemical or physical phenomenon) or to realise control algorithm (to evaluate decision criterion or to calculate control laws) or describe logical sensors (used to estimate a non measurable state variable).

At the different hierarchical levels, we separate the aspects that concern the control with those that concern the process as is currently the case in some approaches applied to manufacturing systems [7]. This permits notably a great modularity and the re-usability of the models for other plants. Thus, our system model is broken down into three views : a discrete view describing the control strategy, a discrete view corresponding to a qualitative representation of the process state (its configurations) and finally a continuous view of the process associated to each of these configurations. Consequently, the system model is composed by a control model and a hybrid model of the process. The latter model, called hybrid reference model, corresponds a qualitative view of the behaviour of the process independently of the control strategy. As for the control part, these models are hierarchically structured with an abstractive view of the process corresponding to the considered level (Fig 2).

At the considered level, the hybrid model of the process is composed of a set of hybrid reference models: one model by equipment (composing the unit) at the coordination level and one model by unit (composing the plant) at the supervision level. The discrete aspect of each model is represented by means of Petri nets. The places of the Petri net correspond to the configurations of the equipment (if we consider the coordination level). With each discrete state of the net (each place), a subset of differential algebraic equations is associated if any continuous phenomena takes place. This subset of equations describes the expected continuous process behaviour. The active equipment configuration is characterised by the location of the token in the net (the marking) and its continuous state, described by a subset of continuous variables, is carried by the token. The set of marked places of the nets (one net by equipment) corresponds to a logical description of partial states of the whole controlled unit.

When the control part evolves from an operating stage to another one, the reference model simultaneously evolves. A transition firing in the control part which corresponds to the execution of a given operation on the equipment, simultaneously results in the firing of the associated transition in the reference model [21]. The simultaneous evolution of the reference model with the control one ensures that the automatically selected subset of equations is consistent with the set of operations in progress.

The hybrid reference model is represented by means of DPTP nets [20].

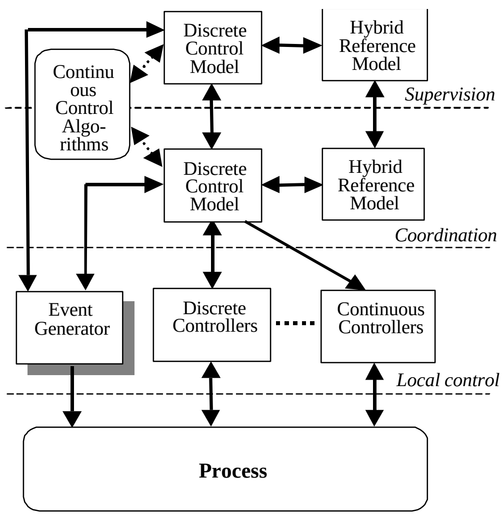

Fig 2. The modelling architecture

5 The uses of the reference model

The reference model is mainly used to detect abnormal behaviours of the process. The detection is based on the comparison between the effective process behaviour and the expected one simulated by the reference model.

Let us consider a storage tank on which two activities can be perform, simultaneously or not : the loading operation and the decanting one (Fig 3.). End-of-operation conditions depend on the volume to be transferred. The reference model is composed of four discrete states corresponding to the four possible configurations of the storage tank (Fig 4.). Equations associated to places of the net express the variation of the storage tank volume.

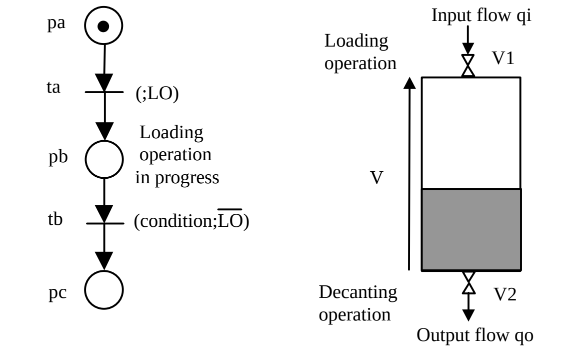

Fig 3. The storage tank example

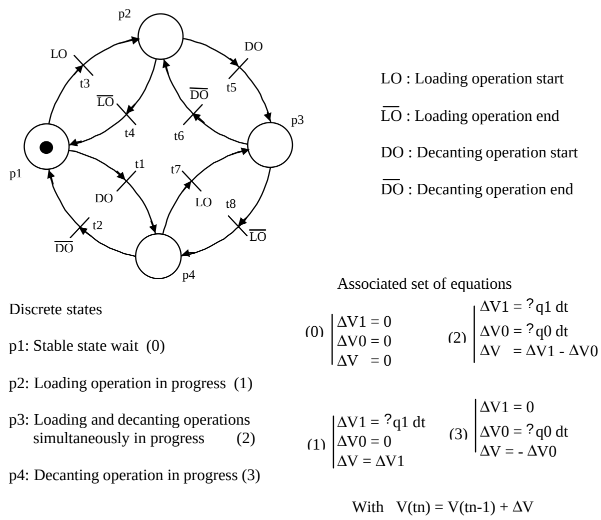

Fig 4. The storage tank reference model

Let us suppose that the supervisory control wants to perform a loading operation on the tank until its level reaches 2 m^3, with an input flow of 1 m^3/h.

On the control net, this corresponds to the firing of the transition ta and to the programming of the event generator in order it generates a process event (which indicates the loading-end-operation) when a threshold of 2 m^3 will be reached (if the tank is supposed initially empty). The place pb becomes marked with a token whose attributes (as previously said, we use predicate-transition Petri nets for which the token <y> carries a set of information) are the parameters of the operation to be executed : the volume to be transferred and the required input flow. So, the loading operation is in progress and the control net is waiting for the process event corresponding to the loading-end-operation. When the process event is sent by the event generator the supervisory control evolves in a new state. This is the functioning without the use of the reference model.

The starting of the operation must be only authorised if the process is in a configuration from which it is possible to start this operation and if the operation parameters are acceptable. Moreover, the supervisory controller evolution is linked to the process event occurrence. If the process event never occurs (because of a failure of the process) the supervisory controller can not more evolve. A failure of the process can also lead to receive a process event at a non consistent date.

Now, let us consider the use of the reference model. So, the supervisory controller wants to start the loading operation (firing of the transition ta). The reference model of the storage tank is in an inactive configuration The place p1 is marked with a token <x> whose attributes are pieces of information on the physical constraints of the tank (maximum volume, maximum input flow) as well as the value of the continuous state variables which characterise the tank (current volume) and the end-of-configuration event (the threshold value of the volume). The transition t3 being enabled the change to the loading operation is possible. Before to fire this transition, the required input flow attribute of the control net token <y> is compared with the maximum input flow attribute of the reference model token <x>. If this condition is satisfied, the transition ta and t3 are simultaneously fired. The control net goes on the loading operation in progress state (place pb marked) and it is waiting for the occurrence of the end-loading process event.

On the reference model, the attribute describing the end-of-configuration event on the token <x> is updated with the value of the volume to be transferred carried by the control net token <y>. The place p2 is marked (loading configuration). The subset of equations (1) is activated in order to calculate a temporal window which gives the possible dates for the occurrence of the loading-end-operation process event. So, the expected end-of-operation date is calculated. To this date is added a temporal slack corresponding to the transient phases of the operation in order to constitute the temporal window [d1,d2] (d1 being the earliest end-of-operation and d2 the latest one). The output transition t4 will stay enabled during this time interval.

If the process event sent by the event generator occurs out of the temporal window, the transition t4 can not be fired. The reference model and consequently the control net can not any more evolve (transition tb can not be fired). It is then possible to conclude that a failure happened. A message is then sent to a diagnosis module.

In the contrary case, it is possible to conclude that the operation has been successfully performed. The loading-end-operation produces the evolution of the control net in a new state (place pc marked). That automatically implies the change of configuration of the reference model (place p1 marked). Moreover, on the reference model the token <x> is updated with the value measured on the process corresponding to the current value of the volume. Thus, the reference model keeps a faithful view of the effective state of the process.

The reference model can be used for others objectives.

It is sometimes useful to simulate the process behaviour on a short time horizon. If in the previous case, the control is simultaneously applied to the process and to the reference model in order to detect abnormal behaviour of the process, the control is only applied to the model when it is used in emulation [21]. It is a kind of temporal projection of the process state. The objective is thus to verify that the control to apply respects process constraints. For instance, we could check if the command to be applied does not violate the constraint of maximal volume of a storage tank. That allows to avoid the occurrence of a failure whose cause could have been an erroneous command. This emulation principle can be also used to analyse the consequences of a decision before making the process recovery.

Another possible use is the validation of a production plan. Indeed, the end-of-operation temporal window can be used by the control net to make a comparison with the temporal window supplied by the scheduling level (this latter representing the estimated operation duration). Such an application of the reference model before to produce, could avoid to search for the cause of a temporal constraints violation which would certainly have occurred while the production is running.

Finally, the hybrid reference models can also be used to describe the plant equipments in order to resolve scheduling problems which need a precise representation (with an hybrid view) of the plant. In this context, a simulator based on a set of hybrid reference models to represent equipments in order to validate estimated production plans is under realisation by our colleagues of the Chemical Engineering Laboratory at Toulouse with who we have collaborated to develop the presented approach.

6 Conclusion

To face the complexity of discontinuous systems and to ensure flexibility, we have proposed a hierarchical and modular decomposition of the control system. The hybrid nature of these processes is taken into account by an hybrid modelling of the control system. The model of the control system is hierarchically structured and composed by a control model and an hybrid reference model which corresponds a qualitative view of the behaviour of the process independently of the control strategy. This last model is mainly used to detect process failures.

The modelling of the discrete-event aspects is based on the use of a same model at the different hierarchical levels : Petri nets. Their association with algebraic-differential equations permits to represent the hybrid aspects.

The communication architecture is composed by two buses : the first one with a continuous nature and the other one with a discrete-event nature interfaced by a event generator dynamically programmable by the supervisory controller.

References :

[1] ISA/SP88. Batch Control, Models and Terminology. *Technical report Draft 10*. Instrument Society of America, Research triangle Park, NC USA, 1994.

[2] L. Puigjaner, A. Huercio, A. Espuna. Batch production control in a computer integrated manufacturing environment. *Journal of Process Control*, Vol. 4, n° 4, pp. 281-290, 1994.

[3] Z.H. Liu, S. Macchietto. An advanced operations support system for batch plants. *Workshop on Analysis and Design of Event-Driven Operations in Process Systems*. Imperial College, London, 1995.

[4] E.C. Yamalidou, J.C. Kantor. Modelling and optimal control of discrete-event chemical processes using Petri nets. *Computer Chem Engng*, Vol 15, n° 7, pp. 503-519, 1991.

[5] H.M. Hanish, S. Fleck. A resource allocation scheme for flexible batch plants based on high level Petri nets. *CESA'96 IEEE-SMC IMACS Symp. on Discrete Events and Manufacturing Systems*, pp. 303-308, Lille, France, 1996.

[6] M. Tittus, M. Fabian, B. Lennartson. Controlling and coordinating recipes in batch applications. *34th conf. on Decision and Control*, pp. 2484-2489, New Orleans, LA, USA, 1995.

[7] M. Combacau, M. Courvoisier. A Hierarchical and Modular Structure for F.M.S. Control and Monitoring. *1st Inter. Conf. on Artificial Intelligence, Simulation and Planning in High Automaty Systems*, Tucson, Arizona, USA, 1990.

[8] L.E. Holloway, B.H. Krogh. Fault Detection and Diagnosis in Manufacturing Systems: A Behavioral Model Approach. *2nd Inter. Conf. on Computer Integrated Manufacturing*. RPC, may 1990.

[9] K. Bousson, L. Trave-Massuyès. A Computational Causal Model for Process Supervision. *IFAC Inter. Symp. on Artificial Intelligence in Real-time Control*, Delft, The Netherlands, 1992.

[10] L. Leyval, S. Gentil, S. Féray-Beaumont. Model-based Causal Reasoning for Process Supervision. *Automatica*, vol. 30, no. 8, p. 1295-1306. 1994.

[11] J. Montmain. Raisonnement Approximatif et Graphes Causaux pour la Detection et la Localisation de Défaillances. *Journées études GRECO Automatique "Sûreté, Surveillance, Supervision"*. Paris. 1994.

[12] P.F. Rosa, F. Vasques, R. Valette. The network transparency concept in fieldbus based distributed systems. *ISIE'94-IEEE Inter. Symp. on Industrial Electronics*, Santiago, Chile, 94.

[13] D. Andreu, J.C. Pascal, R. Valette. Events as a key of Batch Process Control System. *CESA'96 IEEE-SMC IMACS Symp. on Discrete Events and Manufacturing Systems*, p. 297-302. Lille, 1996.

[14] R. David, H. Alla. Continuous Petri nets. *8th European Workshop on Application and Theory of Petri nets*, pp. 275-294, 1987.

[15] J. Le Bail, H. Alla, R. David. Hybrid Petri nets. *European Control Conference*, pp. 1471-1477, Grenoble, France, 1991.

[16] I. Demongodin, N. Audry , F. Prunet. Batches Petri nets. *IEEE Inter. Conf. on Systems, Mans and Cybernetics*, pp. 107-117, le Touquet, France, 1993.

[17] R. Weiting, M. Sonnenschein. Extended High-Level Petri nets for modelling hybrid systems. A. *Sydow ed., IMACS Symp. on System Analysis and Simulation*, Vol 18-19, pp. 259-262, Berlin, Germany, 1995.

[18] C. Valentin-Roubinet. Modelling of hybrd systems : DAE supervised by Petri nets. The example of gas storage. *ADPM'94 Conf. on Automation of Mixed Process*. PP. 255-263, Brussels, Belgium, nov. 1994.

[19] D. Daubas, A. Pages, H. Pingaud. Combined simulation of hybrid processes. *IEEE Inter. Conf. on Systems, Mans and Cybernetics*, pp. 314-319, San Antonio, USA, 1994.

[20] R. Champagnat, P. Esteban, H. Pingaud, R. Valette. Modelling and Simulation of a Hybrid System through Pr/Tr PN-DAE Model. *ADPM'98 Conf. on Automation of Mixed Process: Dynamical Hybrid Systems*. Reims, France, 1998

[21] J.C. Pascal, D. Andreu. An hybrid reference model for on-line detection in discontinuous systems. *IFAC-IFP Workshop on On-Line-Fault Detection and Supervision in the Chemical Process Industries*, Lyon, France, 1998.