ADVANTAGES AND DRAWBACKS OF THE LOGIC PROGRAMM SYNTHESIS USING SUPERVISORY CONTROL THEORY

E. NIEL, L. PIETRAC, L. REGIMBAL
Laboratoire d'Automatique Industrielle INSA de Lyon
niel,pietrac,regimbal@lai.insa-lyon.fr

Abstract: Concurrent engineering, re-engineering and reactivity become more and more real in the design concept. Particularly in the design of automatic control efficiency is attempted in terms of quality, quickness, validation and dependability at the different development steps. Nevertheless more an industrial process is automatically controlled and more it is important to manage the performance of the control design. Several investigations are conducted today in the use of assistance tools in the design of proper control laws which will be applied to Programmable Logic Controllers (PLC). In another way, attentions emerged from the synthesis concept using language formalisms. The proposed idea here is to combine these two approaches, i.e. to persuade the designer that synthesis is able to assist the design of logic program. In that way, advantages and drawbacks of the logic program synthesis using supervisory control theory are here discussed. This paper includes two main parts, the first one is devoted to a brief description of the supervisory control particularly described in terms of ability in determining some important properties, the second one based on an applicative example leads to a discussion on the transfer ability of the theory. Copyright © 2001 IFA

Keywords: supervisory control, logic program, PLC, synthesis, assistance design

1. INTRODUCTION

The synthesis of a controller has as advantage to propose a formal modelling allowing to characterise systematically a set of properties to validate the respect as the constraints required in the prescription. These properties decline in properties of liveliness, safety, controllability and for needs of respect as performance (in safe functioning among others), of properties of stability, and optimisation. The synthesis generates a set of correct control law trajectories corresponding to the prescription. The implantation of the final control requires then "to select" one of these trajectories according to performance criteria. System's behaviour is described using a coordination of simple events (beginning and end of activity, presence of pieces, …). The underlying models will then be of the domain of the Discrete Events Systems (DES). In such a representation, the state reached is determined from its previous state and from the cause which provoked its evolution. The appearance of this event can have been foreseen in the specifications of nominal functioning or can be unexpected in the case of breakdowns for instance.

2. MAIN EXPECTED PROPERTIES OF DISCRETE EVENT SYSTEM

In a general way the main characteristics of the DES are parallelism, synchronisation and competition.

- parallelism expresses the fact that several activities can act simultaneously and independently in different parts of the system.
- synchronisation indicates the fact the fulfilment of certain activities requires the simultaneous availability of several resources or the simultaneous check of several conditions.
- competition expresses the mutual exclusion, for which conflicts can be avoided.

Modelling's techniques are based on the "state-transition" concept. They decline essentially in 3 additional tools, Petri nets, finite states machines and Markov chains. The complementarily bases on the various perceptions of modelling, but by preserving the concept state-transition, all make reference to the notion of language.

Expected properties

DES carried for a long time a lack of conceptualisation and formalism comparable to the continuous domain, theoretical advances recovered more from the analysis (performance assessment, design) that of the control. In this paper will treat briefly the properties and the problem of the generation of the control laws for the DES; and at last of the control laws into PLC's programs. The purpose being here to introduce the reader not to the exhaustiveness of the theory (numerous works are available) but to make him aware in the possibilities offered.

At first, one can assert that the ability to react of the DES results from the faculty of certain events to be observable and controllable. Also, the modelled DES should present natural properties of liveliness and safety insuring a correct behaviour :

- **liveliness**: notion which characterise the capacity of the model to be authorised constantly and from any state to run an evolution towards quite other state including the initial state (this property covers also the properties of accessibility, non-blocking, and of reversibility).

- **safety**: notions which characterise the capacity of the model to be forbidden from any state to run an evolution towards any hazardous state (by supposing a possible action on the events driving it *i.e.* the events considered having to be controllable then or to be pre-emptive).

- **controllability**: notions which characterise the capacity of the model to impose a behaviour from any current state. In a logical model, the control will consist of a set of license of event occurrence (event on which it is possible to act *i.e.* controllable event) to hold only the behaviour wished by the prescription.

- **pre-emption** generally implies the existence of a temporal model, and confers to the capacity to force an event occurrence before the other one (reference to temporal lower borders or to ticks of clock).

Observability and controllability are properties bound to the events or to event sequences generated by the system.

- **observability** involves to the model the possibility of monitoring the evolution of the system i.e. the capacity to express its current state distinguishing two trajectories starting from the same state by instance.

We have already introduced the notion of controllability, we remind that a control has no action's capacity on the uncontrollable events, it only undergoes its occurrence.

- **diagnosability** is the capacity of a system to recognise the occurrence of certain unobservable events. In a faulty state it allows by forward and backward approach to localise and diagnose initiative events resulting in perturbation.

Stabilisability and optimisation decline both the performance properties (Brave 1990)

- **stabilisability** is the capacity (including also uncontrollable events occurrence) of a system to join a set of states and to remain there infinitely.

- **optimisation** has for objective to determine a system and its control leading to the best performances (under cost and delay criteria).

These properties can be performed only by means of supplementary analytical or experimental techniques like synthesis.

Synthesis

The original theory bases on the use of finite state machine models, for which there is an explicit distinction between the process and its control. Supervisory control (Ramadge, Wonham 1989) is an approach of the theory of the systems unifying general concepts. It is a mathematical theory based on the formal languages that allows to model DES and to resolve the problems of control by means of standard algorithms. An order will impose a specified behaviour respecting the set of prediscussed basic constraints (liveliness, safety, ..). The main impact of this approach is to obtain a controller synthesis defining a set of trajectories (behaviour of the process bound to its control and respecting the specifications which will be also called control laws). The synthesis is based on a model of the process given at the level of a logical description, the behaviour of the DES is specified by an orderly list of events, without taking into account time between two consecutive events. A typical objective will be to avoid by supervision that an unwished sequence of events occurs.

Two major points of interest appear from this theory:

- the use of the **formal languages** - under theoretical results, they will characterise stability, observability, controllability, liveliness, *etc*....

- the **model** - describes the behaviour of the physical system to be controlled. The behaviour is defined by a generated language, the widely used generative models are finite state machines, Petri nets, *etc*....

3. SUPERVISORY CONTROL OF DES

The supervisory control theory was introduced by Wonham in 1987. In this approach the time does not intervene, the study takes place at a logical level. The process is modelled as a DES which evolves spontaneously by generating events. The functioning can be described by a set of sequences of events which constitutes the formal language on the alphabet of the events. The controller is coupled with the process, it is also a DES which evolves according to the events $\phi^{(i+1)}$ generated by the process on *i* logical time. It can modify the process behaviour by authorising or by forbidding the occurrence of controllable events given then a $\phi^{(i)}$ event list to the process (fig 1).

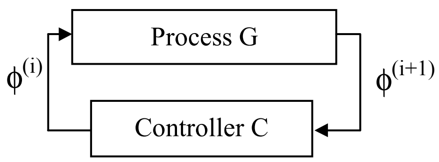

Fig 1 : coupled process to its controller

A process coupled with a controller can be perceived as a system whose inputs are a list of authorised events. This driven process is able to generate an event, if this event is generated by the process itself and if authorised by the controller. Fundamentally the observation of the process by the controller is asynchronous, *i.e.* no relation with any external clock. The occurrence of an event can drive the controller in a new state and at once the list of the authorised events is supplied in the process. One calls functioning in closed-loop, the functioning of the process coupled with its controller.

Finite state machines are used to model a system functioning as an input/output relationship. There are numerous models; determinist machines, model acceptors, Moore and Mealy machines. We shall consider the case where the model possesses a number of finite states, accepting then a regular language. Each system can be represented by its state transition graph. Such a system is called determinist in the sense that from any state two identical outputs will drive to two different states. Models not respecting this property are said non-determinists, they allow however to model in a easier certain way systems. Formally, a language will be defined by a chain of events, a chain is said accepted by a system if, leaving its state initial and receiving successively the symbols in entrance, the system is driven in a final state.

Principle of the supervision

The role of the controller consists in authorising (or forbidding) the occurrence of events in a process. In opposition to the classical automation concept supported by programmable logic controller (PLC) it can not force events to occur. It follows that the controller can only restrict the functioning of the process. Some events generated by the process can not be forbidden, they will be called uncontrollable events. *A contrario* controllable events are events which can be forbidden at any time. As the uncontrollable events can not be forbidden by the controller, one requires that the list of the authorised events contains all the uncontrollable events. If process and controller are respectively modelled by acceptors G and C (figure 2) then a global acceptor language L(C/G) is obtained by making the synchronous product of L(G) and of L(C). If one notes C/G the model established by the process G coupled with its controller C, L(C/G) represents then the functioning in closed-loop.

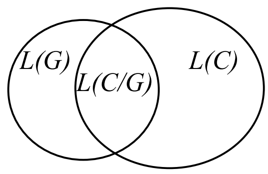

Fig 2 : L(C/G) composition

Being given a process G and one specification of functioning, one wishes to synthesise a controller C so that the system in closed-loop C/G respects the specification. It is not always possible to restrict by supervision the functioning of a process when uncontrollable events occur, the existence of a controller C such as $L(C/G) = L_D$ lies in the concept of controllability. Formally one can define the controllability from the prefix-closure of a language: let $\overline{K}$ be a language prefix-closed, $\overline{K}$ is said controllable with regard to a language L if $\overline{K}\Sigma u \cap L(G) \subseteq \overline{K}$. In this definition, K represents the language of specification and L the language of the process. So, K is controllable with regard to a language L if for any chain ω of $\overline{K}$ and for any event τ uncontrollable of Σ_u, the chain ωτ belongs in L, implies that it belongs also in $\overline{K}$.

Existence conditions

Let L(G) be the language of the process and L_D a wished functioning included in L(G). What is the condition so that a controller C exists such as $L(C/G) = L_D$? For a language L_D prefix-closed, not empty and included in the language L(G) some process, there is a controller C such as $L(C/G) = L_D$ if and only if L_D is with regard to L(G).

Controller synthesis

Let us consider a process G and a wished functioning L_D included in L(G). If L_D is not controllable then there is no controller S such as $L(S/G) = L_D$. In that case it is necessary to look for a more restrictive solution, *i.e.* L_D's subset which is a controllable language and prefix-closed, let $C(L_D)$ be this language. The class $C(L_D)$ of the sub-languages of controllable and prefix-closed L_D is closed under the union of the languages. Then it exists a bigger element of the set L_D, such a language will be called supreme controllable, noted $supC(L_D)$. From the models acceptors G of the process and A_{spec} of a specification of functioning, Kumar's algorithm allows to verify the controllability of the language of specification $L(A_{spec})$ and in case the language $L(A_{spec})$ is not controllable, to synthesise a model acceptor D' of the supreme controllable $supC(L_D)$ of the wished functioning.

When a single controller is coupled with the process, the supervisory control is said centralised, when there are several controllers coupled with the same process, it is said modular and will be according to horizontal or vertical decomposition. Extensions can be done in order to retain an abstracted but powerful coordinator (Chafik 2000a). The formal approach of the proposed hierarchical-decentralized structure allows to check that the combining of decentralized and hierarchical concepts does not affect the hierarchical consistency. The normality property of decentralized supervisors is conserved at the two hierarchical levels.

This structure (figure 3) supposes that there is not conflict between local supervisors at each level of the hierarchy. However when this supposition is not true, we propose an extension of the hierarchical-decentralized structure to a hierarchical coordination structure whose aim is to manage conflicts of local supervisors.

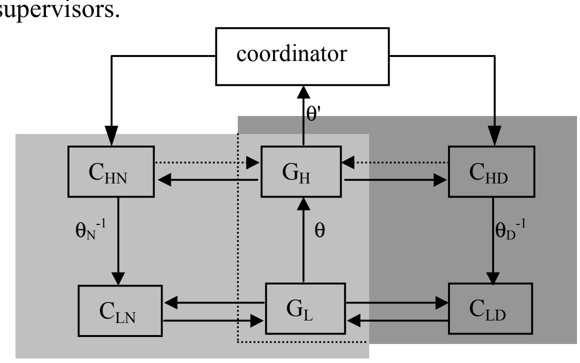

Fig 3 : Conflict resolution by a coordinator

4. APPLICABILITY OF THE SUPERVISORY CONTROL THEORY

Finally and to get closer to the operational control, the last part will deal with the applicability of the formalism of synthesis of safe control trajectories towards industrial processes. The appropriation of this theory is not easy in reference to the hypothesis allocated to the manipulated events and to the concept of modelling itself. Advantages and drawbacks of the current propositions of transposition will be discussed.

Process description

The used process installation moves two types of pieces from one entrance to two exits. The movement of the pieces is shown on the figure 4. When a piece of type P1 is detected on the conveyor C0, it is elevated to the first floor. Here the cylinder V0 stops. The air cylinder V1 pushes the piece P1 onto C1, it is evacuated and V1 returns to its initial position. This is followed by the return of V0 to its initial position to allow the treatment of a new piece (classical L cycle).

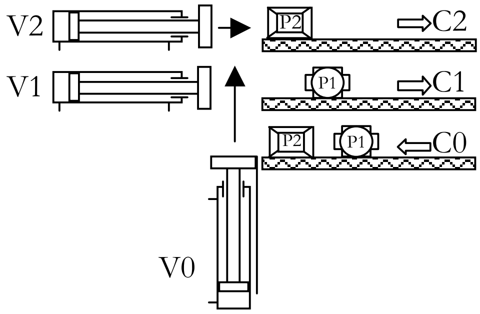

Fig 4 : process

When a piece of type P2 is detected on the conveyor belt C0, the cylinder V0 elevates it to the second floor. The air cylinder V2 pushes the piece P2 onto C2. Then the two cylinders V0 and V1 return to their own initial positions.

Model of the process

Using finite states machines as model supports, we now propose a graphical representation (fig 5, Fig 6, Fig 7) of the states of the cylinders models to give a complete model of the process.

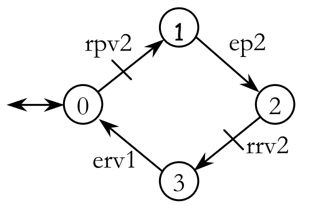

fig 5: Air cylinder V2 model

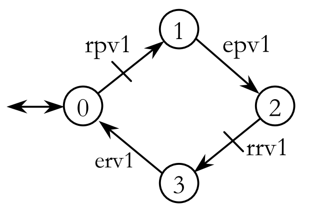

fig 6: Air cylinder V1 model

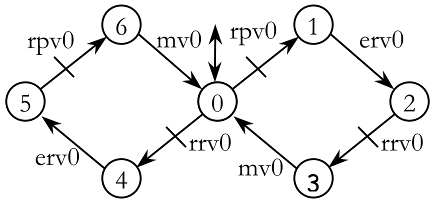

fig 7 :Air cylinder V0 model

Nomenclature

rpvx : request for push out to air cylinder x
epvx : end of push out to air cylinder x
rrvx : request for return to air cylinder x
ervx : end of return to air cylinder x
mv0 : middle position of air cylinder V0

The process has two sensors P1 and P2. This information appears only when the cylinder V0 is completely returned to its initial position. For their integration, we propose the following model.

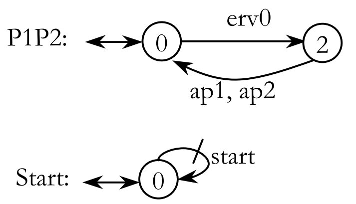

Fig 8 : external sensors modelling

Nomenclature

ap1 : occurrence of P1
ap2 : occurrence of P2
erv0 : end of V0 return
start : start of cycle

This representation is necessary so that the process can use the information of the start button and of the P1P2 sensors.

The global system

Now all the elements of the installation are known, we may create the total process in order to define its global behaviour. The synchronous composition of **V1**, **V2**, **P1P2** and **Start** gives a model with **224 states** and **1136 events**. Although only the nominal functioning mode is retained, we may notice that with few rudimental elements we obtain a complex state representation.

The constraints

We impose three process constraints, corresponding to the description which could be interpreted as restricted liveness (specific cycle for which the process is defined). They determine the cycles for each type of piece and the start up of the system. Every constraint is modelled by a finites states machine. To facilitate the reading, we use the writing {Σ - set of events} where Σ represents all the events generated by the process and {Σ- set of events} all the events pulling a change of state or prohibitions (in bold).

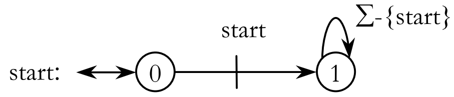

Fig 9 : constraint on start of cycle

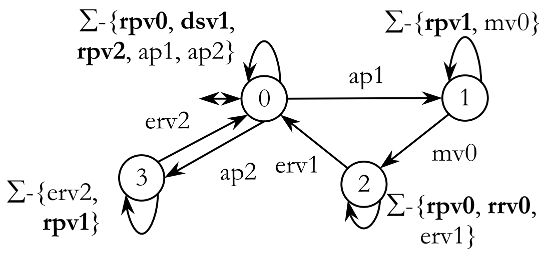

Fig 10 : Constraint on piece of type P1

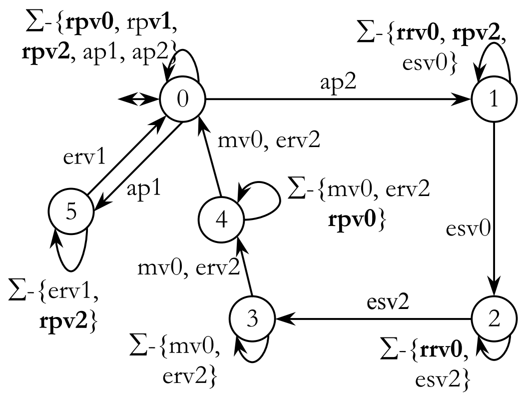

Fig 11 :Constraint modelling on piece of type P2

Figure 9 represents the constraint of cycle start of the system. Figures 10 and 11 impose on the global process the movements of the pieces.

The supreme controllable sublanguage:

The supreme controllable sublanguage, obtained composing process and constraints represents all the allowed trajectories of control. This machine contains 24 states and 29 transitions (figure 12).

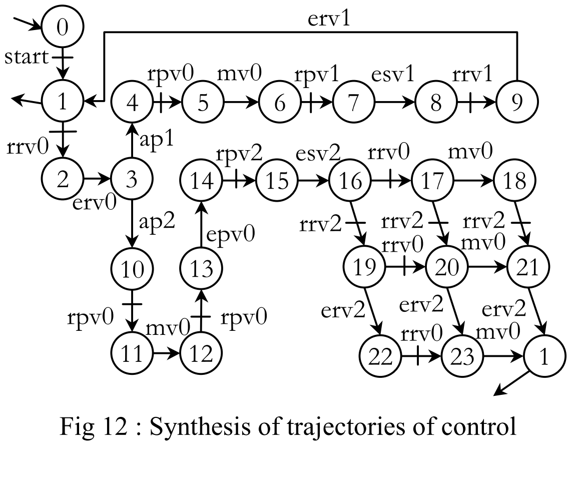

Fig 12 : Synthesis of trajectories of control

From the initial state it's possible only to start with controllable event of start of cycle. We notice the two possible trajectories of the pieces starting at the state n°3 relative to figure 12. The first trajectory follows the states 1 to 9 and corresponds to the cycle of the piece of type P1. We may distinguish a second trajectory including states 1, 2, 3, and 10 to 23. This trajectory of control corresponds to the cycle of the piece of type P2.

Obtained logic program

The main problem to transfer the model of the figure 12 into a logic program is the following: how to translate controllable / uncontrollable events suite issued from the controller synthesis model onto action / reaction implanted in PLC? The first answer is to choose the optimal control trajectory (Ndjab 1999)(Charbonnier 1996). Our proposition is to associate action to controllable event and reaction to uncontrollable event. From the graphical representation figure 12, we propose one conversion to a PLC language (Functional Chart) shown by the figure 13.

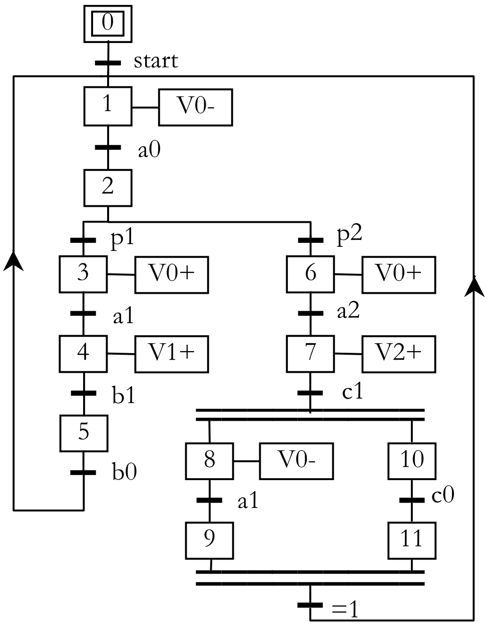

Fig 13 : representation of the machine language

This example describes the difficulty of modelling the global process. Its modelling depends on degrees of freedom of every elements it is composed of. Indeed, during the synchronous composition of the global process, the number of state increases considerably. To avoid this combinatorial explosion, we must choose the number of states of every elements with precaution. We note that on figures 10 and 11 the difficulty of interpreting the specifications of constraints of the system. The translation and the reading of automaton is also complex. Furthermore we should be aware of possible conflicts between the constraints specifications applied to the same process.

This example gives us a correct result in one mode of functioning. To apply this theory to other modes, it's necessary to use extensions like, for example, the decentralisation (Chafik 2000b).

5. CONCLUSIONS

Using supervisory control theory leads to obtain systematically a set of trajectories satisfying all predefined prescriptions. The attempted validation is based on a coherent and natural modelling approach using formal language where major control properties are strictly insured. In this spirit, synthesis represents a strong concept which could be used in other applicative areas such as communication, computing sciences, *etc* … If finite state machines represents a natural investigation to the language generation we show in this paper that despite to the difficulty in managing the state space, limits appear firstly in modelling and in prescription interpretation but also in the transfer phases i.e. the real applicative control law directly implanted in industrial PLC.

6. REFERENCES

Brave Y. (1990), M. Heymann. *Stabilization of discrete event processes*, International Journal Control, Vol. 5, 1990, pp. 1101-1117.

Chafik S. (2000a). *Proposition d'une structure de contrôle par supervision hiérarchique et distribuée : application à la coordination. PhD* thesis, INSA Lyon, 2000.

Chafik S. and Niel E. (2000b). *Hierarchical-decentralized solutions of supervisory control. 3rd* MATHMOD, Vienna, Austria, February 2000. Vol. 2, p. 787-790.

Charbonnier F. (1996). *Commande supervisée des systèmes à événements discrets. PhD* thesis, Institut National Polytechnique de Grenoble, 1996.

Ndjab H. (1999). *Synthèse de la commande des systèmes à événement discrets par Grafcet. PhD* thesis, University of Reims, 2000.

Ramadge P. and Wonham W. (1989). *Control fo discrete-event systems*. IEEE transaction on automatic control, 1989. Vol. 77, no. 1, pp. 81-98.