Synthesis and Implementation of Local Modular Supervisory Control for a Manufacturing Cell

Max H. de Queiroz, José E. R. Cury
LCMI-DAS - Federal University of Santa Catarina,
Florianópolis SC 88040-900 Brazil - Phone: 048 331-7025
max,cury@das.ufsc.br

Abstract

The Supervisory Control Theory is applied to a manufacturing cell commanded by a programmable logic controller (PLC). By means of local modular approach – a methodology that exploits modularity of the plant and of the behavioral specifications models – optimal supervisors are obtained without state-space explosion. For the purpose of simplifying the ladder diagram implementation in the PLC, reduced supervisors are implemented in a three level structure that executes the modular supervisors concurrent action and interfaces the theoretical model with the real system. The flexible and productive manufacturing cell behavior, after the control system practical implementation, and the final PLC code readability and flexibility are positive quality indicators to the applied methodology.

1. Introduction

The Supervisory Control Theory (SCT), initiated by Ramadge and Wonham [13] is a powerful framework for the synthesis of control for Discrete-Event Systems (DES). Although the SCT has obtained a wide acceptance in academy and some experimental applications [1][2] have elucidated its potential, the extensive use of SCT in industry remains a promise.

There are many reasons for the lack of industrial applications of SCT, including the computational complexity and state-space explosion in the synthesis process. The number of states of the global system model grows exponentially with the number of subsystems. For that reason the complexity of computing optimal supervisors, although polynomial in the size of the global system, is a barrier for real problems that involve a large number of subsystems. Advances in computational hardware and complexity reduction techniques [3][6][16] can help overcoming this barrier in many applications. However, the physical implementation of supervisors with a large number of states renders the control program immense, unreadable and, thus, untrustworthy.

In previous works [10][11][12], the authors have proposed a local modular approach that extends the results of Wonham and Ramadge [16] to avoid state-space explosion in composite systems. This approach reduces the computational complexity of the synthesis process and the size of supervisors by exploiting modularity of specifications and the decentralized structure of composite plants. Instead of a monolithic supervisor for the entire plant, a modular supervisor is obtained for each specification, taking into account only its local plant (affected subsystems). A necessary and sufficient condition named local modularity assures that the set of local modular supervisors has the same performance as the monolithic supervisor. As a consequence of the limited size of the resulting supervisors, the exponentially complex algorithm for optimal reduction of supervisors as introduced by Vaz and Wonham [14] becomes feasible. When local modularity holds, this approach has the advantage of providing more flexibility, computational efficiency and safety to the control application.

This paper presents the results of an application of the local modular approach to the control program synthesis for a real manufacturing cell commanded by a programmable logic controller (PLC). Several new challenges and solutions arise from the process of linking the theory to reality, concerning aspects from the open-loop system modeling to the final structure of the control program.

To guide the physical implementation of the control system from abstract supervisors, the authors propose a three level program structure that plays the set of reduced local modular supervisors concurrently, commands the evolution of decentralized subplants and acts as an interface between the theoretical model and real control signals. This control structure differs from the implementation scheme proposed by other works [1][2][4][7][9] for playing the supervisors exactly as previewed by the SCT while the interface levels resolve the differences between the abstract model and the real system. This characteristic allows a modular implementation of plant and of supervisors, what renders the control program clean and clear.

The paper sequence is the following: Section 2 presents the manufacturing cell. Section 3 explains the synthesis of local modular supervisors. In Section 4, a generic control system structure is proposed and implemented in ladder diagram for the manufacturing cell.

2. The Manufacturing Cell

The manufacturing cell (Figure 1) is composed by a four stages circular table (M_0), where metallic pieces are drilled and tested, and by other four operational devices: an input conveyor (M_1), a drilling machine (M_2), a test device (M_3) and a robotic manipulator (M_4). A programmable logic controller commands the cell according to the following sequence:

1. the conveyor advances until a part is settled in P1;
2. the table rotates 90°;
3. the part is drilled in P2;
4. the table rotates 90°;
5. the part is tested in P3;
6. the table rotates 90°;
7. the robotic manipulator removes the part in P4 from the circular table.

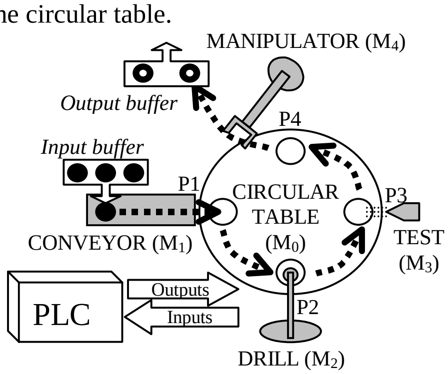

Figure 1. Manufacturing Cell

There are no sensors informing presence of parts in positions P1, P2, P3 and P4. The original control program, furnished by the cell manufacturer, allows the sequential operation of only one piece at a time. This restriction in the control logic avoids the following problems that may arise in the concurrent operation of multiple pieces:

- operation of conveyor, drill, test or manipulator at the same time as the rotation of the table;
- overflow of pieces in P1;
- rotation of the circular table when the parts in P2, P3 and P4 aren't drilled, tested or removed, respectively;
- drilling, testing and manipulating without pieces in P2, P3 and P4, respectively;
- drilling or testing twice the same piece;
- rotation when the circular table is empty.

However, this control law is very inefficient once the devices are mostly inactive while they could operate concurrently. Hence, the aim of this work is to apply the Supervisory Control Theory to the synthesis and implementation of a new control program for the PLC that avoids the described problems in a minimally restrictive logic.

3. Synthesis of Supervisors

The described manufacturing cell, as most of important problems involving DES, is composed by multiple concurrent subsystems. In this context, the main goal of supervisory control is the coordination of such subsystems so that the global system obeys a series of individual and joint specifications. The results of Queiroz and Cury [10][11] extend the framework of Wonham and Ramadge [16] to avoid state-space explosion by exploiting modularity of specifications and decentralization of plant. By this local modular approach, the plant is represented as a set of asynchronous generators and the specifications are expressed locally over the affected subplants. Optimal supervisors are computed from the local specifications and are finally reduced by a minimization algorithm. The main aspects of this approach and its application to the manufacturing cell are briefly described in the following subsections. The computations involved in the synthesis procedure were performed with CTCT [15] in a personal computer Pentium 233.

3.1. Modelling

In the framework of Ramadge and Wonham [13], the system is supposed to spontaneously generate discrete events $\sigma \in \Sigma$ classified as controllable ($\sigma \in \Sigma_c$), when the event can be disabled by the control system, or uncontrollable ($\sigma \in \Sigma_u$). Let Σ^* be the set of all finite chains of elements in Σ, including the empty chain ε. A language is a subset of Σ^*. The prefix closure of a language L is denoted by $\overline{L}$. The behavior of a DES may be modeled by languages that, when regular, are represented by generators. A generator is a quintuple $G = (\Sigma, Q, \delta, q_0, Q_m)$, where Σ is the event set, Q is a set of states, $q_0 \in Q$ is the initial state, $Q_m \subseteq Q$ is the subset of marked states and $\delta : \Sigma \times Q \to Q$, the transition function, is a partial function defined in each state of Q for a subset of Σ. Then, G is characterized by two languages: its closed behavior $L(G)$ and its marked behavior $L_m(G)$. The operator $\|$ represents synchronous composition of languages or generators. Generators are illustrated by state transition diagrams that are directed graphs where nodes represent states and labeled arcs represent events. Marked state nodes are double lined and an arrow points the initial state. Intercepted arcs indicate controllable events.

For purpose of control logic design, internal procedures of the manufacturing cell can be abstracted in the modelling process to discrete events that allow for a consistent expression of specifications and of the open loop system behavior. Although the operation of each device of the manufacturing cell corresponds to a sequence of specific commands, the described problems don't occur internally in the operational procedures of each subsystem, but in the lack of coordination between the start and the end of operations. Thus, to synthesize the coordination logic, we consider that the operation of subsystems M_i, $i=0,...,4$, starts with controllable events α_i, $i=0,...,4$, and ends with uncontrollable events β_i, $i=0,...,4$. Then, for $i=0,...,4$, the devices M_i are modeled by generators G_i, with event set $\Sigma_i = \{\alpha_i, \beta_i\}$, shown in Figure 2.

G_i, $i = 0,...,4$:

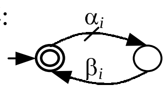

Figure 2. Generator for M_i, $i = 0,...,4$

This abstraction allows using simple models for (high-level) supervisory control, while the complete models, representing the operational procedures, are programmed for each device on a low level of control. If necessary, these low-level control logics could also be obtained by formal synthesis procedures. Of course, we need to assure hierarchical consistency [5][17] between the operational procedures and the high-level models, what is evident for the current problem.

Taking into account the decentralized structure of the system, the plant is then represented as a set of completely asynchronous subsystems, named product system [13]. The (composite) plant models all chains of events that could happen in the open-loop system, including undesired ones. To calculate the control logic that avoids the occurrence of undesired sequences, it is necessary to express the problem specifications in terms of languages representing the desired behavior. By the local modular approach, this behavior is represented by a set of local specifications [11].

The generator described in Figure 3a marks a generic specification $E_{gen,a}$ that prevents the table to rotate without at least a rough piece in P1, a drilled piece in P2 or a tested piece in P3. The four safety specifications that impede the table to rotate while the conveyor ($E_{gen,b1}$), the drill ($E_{gen,b2}$), the test ($E_{gen,b3}$) and/or the manipulator ($E_{gen,b4}$) are working are marked by the generator of index i in Figure 3.

a) $E_{gen,a}$: **b)** $E_{gen,bi}$, $i = 1,...,4$:

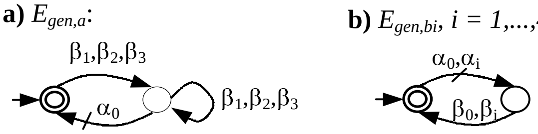

Figure 3. Generators for $E_{gen,a}$ and $E_{gen,bi}$, $i = 1,...,4$

The problems generated by the flow of multiple pieces are avoided by the specifications: $E_{gen,c1}$, related to the flow of rough pieces between P1 and P2; $E_{gen,c2}$, for the flow of drilled pieces between P2 and P3; and $E_{gen,c3}$, corresponding to the flow of tested pieces between P3 and P4. Since the specifications have the same structure, they are represented by the indexed generator of Figure 4.

$E_{gen,ci}$, $i = 1,2,3$:

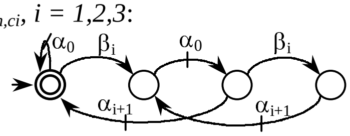

Figure 4. Generator for $E_{gen,ci}$, $i = 1,2,3$

For $x \in \{a, b1, b2, b3, b4, c1, c2, c3\}$, the local plant $G_{loc,x}$ is obtained trough the composition of subsystems sharing events with their respective generic specification $E_{gen,x}$. By this procedure, the following generators are achieved:

- $G_{loc,a} = G_0 \parallel G_1 \parallel G_2 \parallel G_3$;
- $(G_{loc,bi} = G_0 \parallel G_i)$, $i = 1,...,4$;
- $(G_{loc,ci} = G_0 \parallel G_i \parallel G_{i+1})$, $i = 1,2,3$.

For $x \in \{a, b1, b2, b3, b4, c1, c2, c3\}$, the local specification $E_{loc,x}$ is computed by the synchronous composition of the generic specification $E_{gen,x}$ with its respective local plant $G_{loc,x}$, i.e., $E_{loc,x} = E_{gen,x} \parallel L_m(G_{loc,x})$.

3.2. Synthesis of Local Modular Supervisors

In this paper, a supervisor is represented by a generator S, whose state changes are dictated by the occurrence of events in the plant G. The control action of S, defined for each one of its states, is to disable in G events that may not occur in the controlled system after an observed chain of events. The behavior of the controlled system S/G can be described by the generator $S \parallel G$. Hence, beyond restricting the plant behavior, the supervisor unmarks states. In other words, a task of the closed loop system is considered complete only when it is marked by the plant and by the supervisor. A supervisor S is nonblocking if $\overline{L_m(S/G)} = L(S/G)$. The necessary and sufficient condition for the existence of a nonblocking (marker) supervisor S that reaches a given specification $K \subseteq L_m(G)$ ($L_m(S/G) = K$) is the controllability of K [15]. K is said to be controllable if $\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$. The class of controllable languages contained in K has a supremal element $SupC(K, G)$.

When the control problem comprises multiple (m) specifications, it is possible to design a monolithic supervisor for the entire set of specifications or a modular supervisor for each specification. By the local modular approach, for $j=1,...,m$, nonblocking supervisors $S_{loc,j}$ are computed directly on the respective local specifications $E_{loc,j}$ so that $L_m(S_{loc,j}/G_{loc,j}) = SupC(E_{loc,j}, G_{loc,j})$. The local modularity – a generalization of non-conflict [16] – of the set of supervisors is verified when:

$\|_{j=1}^{m} \overline{L_m(S_{loc,j} / G_{loc,j})} = \overline{\|_{j=1}^{m} L_m(S_{loc,j} / G_{loc,j})}$.

This condition is necessary and sufficient to assure that the modular approach doesn't cause any loss of performance with relation to the monolithic approach [11].

For the described manufacturing cell, optimal nonblocking supervisors $S_{loc,x}$, $x \in \{a, b1, b2, b3, b4, c1, c2, c3\}$, are computed from the local specifications $E_{loc,x}$. The generator S = $S_{loc,a} \| S_{loc,b1} \| S_{loc,b2} \| S_{loc,b3} \| S_{loc,b4} \| S_{loc,c1} \| S_{loc,c2} \| S_{loc,c3}$ (with 151 states and 350 transitions) is trim. By consequence, the set of supervisors is locally modular (non-conflicting). For the eight modular supervisors we compute the control action, assigning to each state a set of events that must be disabled, that is, events that at the corresponding state may occur in the respective local plant and are not allowed by the supervisor generator.

3.3. Supervisors Reduction.

As a final step before the physical implementation, minimization of supervisors is taken into account, for a reduction in the number of states of a supervisor can represent memory economy and clarify the control logic.

For the supervisors $S_{loc,x}$, $x \in \{a, b1, b2, b3, b4, c1, c2, c3\}$, their respective reduced supervisors RS_x are obtained by the minimization algorithm of Vaz and Wonham [14]. Although this algorithm has exponential complexity, it becomes feasible for local modular supervisors that usually have a small number of states. Figure 5 illustrates the set of reduced local modular supervisors, computed with an implementation of the algorithm in Matlab, with their control actions represented as dashed arrows.

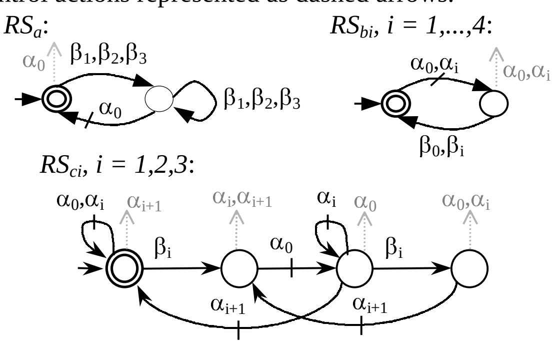

Figure 5. Reduced supervisors

The number of states of generators involved in the synthesis process is presented in Table 1. The computational efforts of the synthesis have been (polynomially) proportional to the small size of these generators.

Table 1. Number of states of generators

x	$E_{gen\ x}$	$G_{loc,x}$	$E_{loc,x}$	$S_{loc,x}$	RS_x
a	2	16	32	32	2
bi, i = 1,...,4	2	4	3	3	2
ci, i = 1,2,3	4	8	32	24	4

4. Implementation of the Control System

The synthesis process presented in the previous section provides a set of reduced local modular supervisors represented by finite-state machines. In each state of the supervisors, the control action is defined as the disablement of a set of events previously computed. Then, in theory, the physical implementation of the control system consists of playing the automata for the supervisors in parallel, according to the events read from the real system, and of sending disabling signals to the plant, according to the current state of the generators.

However, for the manufacturing cell, controllable events α_i (start of operation) represent commands that don't occur spontaneously as SCT presupposes and, hence, disabling signals don't correspond to real events. In this case, the control system must also execute the plant, commanding events that aren't disabled by supervisors and are allowed by the open loop behavior model.

Moreover, the plant model used for supervisors synthesis is an abstraction of the real system behavior. Commands α_i launch operational procedures that update the control system output signals according to the devices internal control logic. Further, uncontrollable events β_i don't correspond directly to input signals and model logical events (end of operation) generated by operational procedures. Therefore, the control system must also act as an interface between the theoretical model and the real input/output signals [1][4].

4.1. Control System Structure

Aiming to execute the modular supervisors, the product system and the operational procedures in a structure that avoids the composition of state machines, the control system is proposed to be programmed in a three level hierarchy, as illustrated by Figure 6. This structure, whose dynamics is explained in the following, may be implemented in PLC languages (Ladder Diagram, Grafcet, etc.), in PC languages (C++, Java, etc.) as well as directly in hardware (digital, electric, pneumatic or hydraulic circuits).

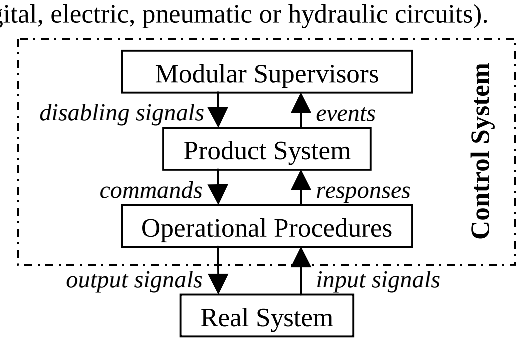

Figure 6. Control System Basic Structure

The set of reduced local modular supervisors is implemented in the highest level of the control system exactly as theoretically conceived by [13]. The program updates the active states of the supervisors according to their generators structures and to the events in the Product System level. A feedback map associates the active states to a set of disabling signals that control the Product System. It can be noticed that self-looping transitions, although fundamental for computing the control action, don't need to be programmed in the supervisors generators since they don't change their states and, by consequence, their control action.

The supervisors generators don't necessary contain the complete information of the plant behavior. Hence, to command the execution of modeled subplants, their state machines are implemented concurrently in the Product System level. The main function of this level is to execute commands by means of accepted controllable transitions. A controlled transition is considered accepted if its preceding state is active and the supervisors do not disable it. An uncontrolled transition is considered accepted if its preceding state and its corresponding event are active. The parallel evolution of the asynchronous subplants follows executed commands (controllable transitions) and responses (uncontrollable transitions) from the Operational Procedures, signalizing events to the supervisors.

The program needs to assure that the Modular Supervisors action is always updated before a new transition occurs in the Product System. This care also avoids more than one state change to be sent to the supervisors instantaneously.

When multiple transitions in the Product System are accepted at the same time, the program needs to make a choice based on a priority scheme or to pass this degree of freedom to a higher level of decision. In any case, it seems natural that uncontrollable transitions have more priority than controllable ones, for the states of the plant should be updated before executing a new command. If more than one response arrive from the Operational Procedures at once, any sequence of their corresponding transitions should lead the supervisors to the same state or the closed-loop behavior would depend strongly on the priority scheme. In [7] this issue is treated by testing interleave insensitivity of supervisors.

The Operational Procedures level works as an interface between the theoretical Product System and the Real System. In this level, the program interprets the abstract commands from the Product System as logical procedures that guide the operation of each particular subsystem. These low-level procedures generate the control system output signals and read the input signals, supplying the Product System with logical responses that reflect the occurrence of uncontrollable events.

4.3. Application to the manufacturing cell

The control system is implemented in the PLC in Ladder Diagram (LD) language [8]. State machines are programmed in LD by means of latched coils [2][7]. States and control system internal signals are assigned to PLC "flags". Then, at the highest level of the control system, the eight reduced supervisors RS_x, $x \in \{a, b1, b2, b3, b4, c1, c2, c3\}$, are implemented as concurrent state machines with disabling signals set according to the active states. Figure 7 illustrates a segment of LD implementation of supervisor RS_{c1}.

```
   s0  β1      s1    |  s2          d-α0 |
|--][-+-][--+--(S)-| |--][-+-----+--( )-|
|           |   s0 | |  s3 |             |
|           +--(R)-| |--][-|             |
|  s1  α0      s2  | |  s1          d-α1 |
|--][-+-][--+--(S)-| |--][-+-----+--( )-|
|           |   s1 | |  s3 |             |
|           +--(R)-| |--][-|             |
|  s2  β1      s3  | |  s0          d-α2 |
|--][-+-][--+--(S)-| |--][-+-----+--( )-|
|           |   s2 | |  s1 |
|           +--(R)-| |--][-|
```

Figure 7. Segment of LD for supervisor RS_{c1}: state machine (left) and disabling signals (right)

At the Product System level, the five subsystems G_i, $i=0,...,4$, are programmed as asynchronous state machines. Uncontrollable transitions are triggered by responses of operational procedures. Controllable transitions automatically launch operational procedures when not disabled by supervisors. Each transition also signalizes occurrence of events to the supervisors. By exploiting a particularity of the used PLC, each transition generates a jump to the first rung[1] of the supervisors to update their control actions before a new transition in the Product System occurs. The ordering of the rungs defines a priority hierarchy for the transitions. As an example, Figure 8 shows the implementation of G_1.

```
   g0  d-α1  c-start  |  g1  c-end    β1 |
|--][-+-]/[-+--( )-|  |--][-+-][--+--( )-|
|           |   α1 |  |           |   c3 |
|           +--( )-|  |           +--(R)-|
|           |   g1 |  |           |   g1 |
|           +--(S)-|  |           +--(S)-|
|           |   g0 |  |           |   g0 |
|           +--(R)-|  |           +--(R)-|
|           |  sup |  |           |  sup |
|           +---->>|  |           +---->>|
```

Figure 8. Ladder diagram for plant G_1: controllable (left) and uncontrollable (right) transitions

The operational procedures for the circular table, the conveyor, the drill, the test and the manipulator are also programmed as parallel state machines, started by Product System commands. The transitions read PLC input signals and update output signals, sending responses to the Prod-

1 This operation could also be programmed according to IEC 61131.

uct System. Figure 9 shows the operational procedure for the conveyor.

```
  c-start        c1     |  c2  I0.1   O0.1 |
 |--][-+-----+--(S)-|   |--][-+-][--+--(R)-|
 |  c1         O0.1 |   |           |   c3 |
 |--][-+-----+--(S)-|   |           +--(S)-|
 |           |   c2 |   |           |   c2 |
 |           +--(S)-|   |           +--(R)-|
 |           |   c1 |   |  c3        c-end |
 |           +--(R)-|   |--][-+-----+--( )-|
```

Figure 9. Operational procedure for the conveyor

After the implementation of the new control system in the PLC, the manufacturing cell has worked with great productivity and flexibility, operating from 1 to 4 pieces in parallel, according to the conveyor input. Also the control program was easy to debug and to upgrade by reason of its readability and flexibility.

5. Conclusion

A methodology for synthesis of supervisory control programs has been successfully applied to a manufacturing cell commanded by a PLC. The local modular approach, allied with a supervisor minimization algorithm, has been advantageous to the synthesis of optimal reduced supervisors, although the test for modularity (involving the composition of all supervisors) would be a tricky matter for more complex systems.

An original generic program structure has been proposed to the control system physical implementation. This structure allows playing the supervisors, to execute the plant and to interface with the real system in an intelligible manner. The new control system, implemented in ladder diagram, has improved the manufacturing cell behavior. The flexible and productive controlled behavior and the final PLC code readability and flexibility indicate the quality of the employed methodology.

Acknowledgements

The authors thank the financial support provided by CNPq under grants: 140.825/2000-2 (first author); 300.953/93-3 and PRONEX 015/98 (second author).

References

[1] Balemi, S., G.J. Hoffmann, P. Gyugyi, H. Wong-Toi and G.F. Franklin (1993). Supervisory control of a rapid thermal multiprocessor. IEEE Trans. on Automatic Control, 38(7):1040-1059.

[2] Brandin, B.A. (1996).The real-time supervisory control of an experimental manufacturing cell. IEEE Trans. Robotics and Autom., 12(1):1-14.

[3] Bryant, R.E. (1986). Graph-based algorithms for boolean function manipulation. IEEE Trans. Comp., vol. C-35, pp. 677-691.

[4] Charbonnier, F., H. Alla and R. David (1999). The supervised control of DEDSs. IEEE Trans. on Control System Technology, 7(2):175-187

[5] Cunha, A.E.C. da, J.E.R. Cury and B. H. Krogh (2002). An assume-guarantee reasoning for hierarchical coordination of DESs. Proc. of the WODES'02, Saragossa, Spain.

[6] Eyzell, J.M. and J.E.R. Cury (1998). Exploiting symmetry in the synthesis of supervisors for DES. Proc. ACC, Philadelphia, USA.

[7] Fabian, M. and A. Hellgren (1998). PLC-based implementation of supervisory control for discrete event systems. Proc. of the 37th CDC. Tampa, USA.

[8] ISO/IEC (1993). International Standard IEC 61131-3, Programmable Controllers – Part 3.

[9] Leduc, R.J. (1996). PLC Implementation of a DES Supervisor for a Manufacturing Testbed: An Implementation Perspective. M.A.Sc. Thesis, Dept. of Elec. and Comp. Eng., University of Toronto, Canada.

[10] Queiroz, M.H. de and J.E.R. Cury (2000a). Modular control of composed systems. Proc. of the ACC. Chicago, USA.

[11] Queiroz, M.H. de and J.E.R. Cury (2000b). Modular supervisory control of large scale discrete-event systems. In: Discrete Event Systems: Analysis and Control. Kluwer Academic Publishers, pp. 103-110. (Proc. WODES 2000, Ghent, Belgium)

[12] Queiroz, M.H. de and J.E.R. Cury (2002). Controle supervisório modular de sistemas de manufatura. Revista controle & automação, SBA. To appear.

[13] Ramadge, P.J. and W.M. Wonham (1989). The control of discrete event systems. Proceedings IEEE, Special Issue on DEDS, 77(1):81-98.

[14] Vaz, A.F. and W.M. Wonham (1986). On supervisor reduction in discrete-event systems. International Journal of Control, 44(2):475-491.

[15] Wonham, W.M. (2001). Notes on Control of Discrete-Event Systems. Dept. of Elec. and Comp. Eng., University of Toronto.

[16] Wonham, W.M. and P.J. Ramadge (1988). Modular supervisory control of DESs. Mathematics of control of discrete event systems, 1(1):13-30.

[17] Zhong, H. and W.M. Wonham (1990). On the Consistency of Hierarchical Supervision in Discrete-Event Systems. IEEE Trans. on Aut. Control, 35(10):1125-1134.