



**HAL**  
open science

## Comparison of Different Modeling Approaches in Simulation of Programmable Logic Controller

Pascal Meunier, Bruno Denis, Jean-Jacques Lesage

► **To cite this version:**

Pascal Meunier, Bruno Denis, Jean-Jacques Lesage. Comparison of Different Modeling Approaches in Simulation of Programmable Logic Controller. IFAC Conference Control System Design 2000, Jun 2000, Bratislava, Slovakia. pp.301-306, 10.1016/S1474-6670(17)37206-3 . hal-03745729

**HAL Id: hal-03745729**

**<https://hal.science/hal-03745729>**

Submitted on 4 Aug 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## COMPARISON OF DIFFERENT MODELING APPROACHES IN SIMULATION OF PROGRAMMABLE LOGIC CONTROLLER

P. Meunier, B. Denis and J.-J. Lesage

*Laboratoire Universitaire de Recherche en Production Automatisée,  
ENS de Cachan, 61 avenue du Président Wilson, F-94235 Cachan Cedex, France  
{meunier, denis, lesage}@lurpa.ens-cachan.fr - <http://www.lurpa.ens-cachan.fr>  
Tel: 33 1.47.40.27.62 Fax: 33 1.47.40.22.20*

**Abstract:** To design the control of automated systems, the designer must evaluate the performances of various solutions. Our research deals with the evaluation of distributed control architecture. We have chosen Colored Timed Petri Nets (CTPN) models for describe behaviors of each equipment of control and we use design/CPN simulation tool to assess response times performance. The aims of this paper are to build various Programmable Logic Controller (PLC) behavior models and to select the best according to our needs in performance evaluations. Various approaches for the creation of CTPN models are discussed. Some of them are based on synthesis techniques and some of them are based on an identification technique. Finally, a comparison method is presented. It will allow the designer to choose the right model for the simulation in the evaluation stage.

**Keywords:** Modelling, Petri-nets, Simulation, Control system analysis, Programmable logic controllers

### 1. INTRODUCTION

To design architectures of distributed control systems, the designer must deal with a phase of performance evaluation. Many studies are centered on the evaluations of functional structures of control (Kiencke *et al.* 1999) or components (for ex: API, fieldbus...) (List *et al.* 1998), or sometimes on the redundancy of components (Gulner and Theis 1999). The general goal of our research (Meunier and Denis 1997) is to evaluate response time performance of a whole architecture dealing with the functional and the material

point of view and using simulation of Colored Timed Petri Nets (CTPN) models. A library of behavior models of control equipments has been setup and design/CPN simulator is used to perform evaluations. The aim of this paper is to prepare a Programmable Logic Controller (PLC) behavior model as good as possible according to our needs of evaluation. In session [2], we describe the various approaches in design of CTPN models. Then, in session [3] we apply these approaches to the creation of PLC behavior models. Finally, in last session, we present a comparison method of these models.

## 2. MODELING APPROACHES

In this part, we describe the different ways to model an equipment behavior and the different ways to build a Petri net models. In order to illustrate these different ways, we study an industrial programmable logic controller (PLC) as example. This PLC is composed with:

- an input module (IM) which handles sensors signals and prepares them for the CPU module;
- a CPU module which executes users programs (we assume that no preemptive task is performed);
- and an output module (OM) which transforms CPU module outputs into actuator signals.

Behaviors of input and output modules, specifying by manufacturer, are describes as delays (noted respectively  $D_{IM}$  and  $D_{OM}$ ).

### 2.1 Different ways to get the behavior of an equipment

Design of dynamic behavior model requires informations on the equipment behavior. There are two ways to extract the behavior: synthesis-based approach and identification-based approach.

*Synthesis-based approach.* This approach consists in the analysis of manufacturer documentation. The behavior of PLC CPU is cyclic. The cycle is made up of (fig. 1):

- a Read input phase (R),
- a program EXecuting phase (EX),
- an Assignment output phase (A).

The cycle time depends of the program size. During the operating mode the program size is frozen and the cycle time is a constant value. This value could be obtained from the execution time of each statements of the user program or from the CPU rate speed given in kilo statement per second. Moreover, we consider that the PLC is periodic, the various phases of the cycle thus have fixed durations.

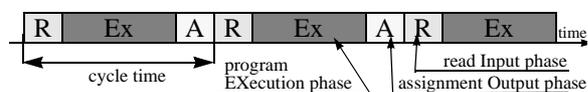


Fig. 1. Chronogram of CPU activity of a cyclic PLC according manufacturer documentation.

*Identification-based approach.* This approach consists in identifying the real equipment behavior. The equipment inputs are solicited and then, the measurement of its outputs provides us a view of its actual behavior. To study logical signals on industrial equipments, we built an experimental framework composed with signal generators and a logical analyzer. All of these devices are managed with a workstation across a GPIB network. Thus we are able to perform a large number of measurement with various kind of solicitation in batch mode. 1200 measurement of

cycle time of our CPU module are reported in the histogram of fig. 2.

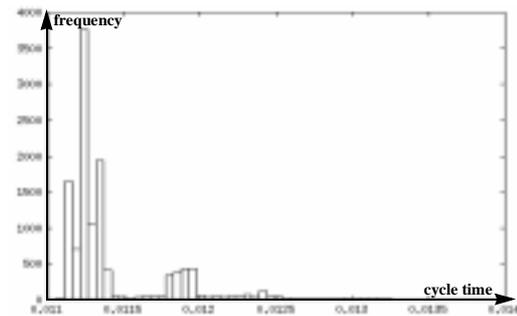


Fig. 2. Histogram of PLC cycle time measures

Measurement analysis gives the dynamic behavior of the PLC. This behavior is cyclic. We discover some periodicity disturbances. To model this behavior, we introduce new tasks in the CPU activity called internal management tasks (MT) (fig. 3). Management tasks of the API are defined as following:

- Period of the management task  $k$ :  $P_M(k)$ ,
- Duration of the management task  $k$ :  $D_{MT}(k)$

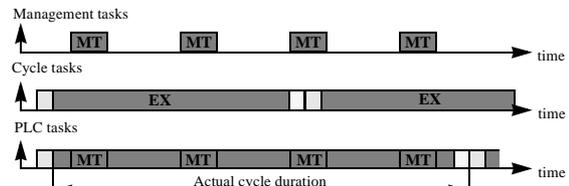


Fig. 3. Chronogram of CPU activity according identification

### 2.2 Different ways to build a Petri net model

Our models are made using colored timed Petri nets (CTPN) within the meaning of Jensen K (1997). We propose three ways to build a Petri net model. The structural, the extended transition and the stochastic model.

*Structural and deterministic model.* This manner of building a model is very current. Each aspect of the behavior of the modeled system is translated into places and transitions. The model reproduces accurately and in a deterministic way the behavior of the modeled system. Each change of state of the modeled system is thus modeled by the firing of a transition in the model. This type of model is simple to build but during the simulation much of transitions will have to be fired. Simulation is thus long and expensive in CPU time.

*Deterministic model with extended transitions.* Here the model also reproduces accurately and in a deterministic way the behavior of the modeled system. But some aspects of the behavior of the modeled system are translated by a list of instructions associated with certain transitions. The Jensen (1997) CTPN allows us to associate codes lines with the transitions. During simulation, this code is executed when the associated

transition is fired. This approach makes it possible to distribute the complexity of a behavior between a simple transition-place structure and simple algorithms associated with transitions.

*Stochastic model.* We build a model whose behavior is on average similar to that of the modeled system. Stochastic transitions are used in the model. During a simulation, the current values of the observed parameters are not significant. But the model is built in order to meet the average of these values to the modeled system.

### 3. MODEL DESIGN

In this session, all the different modeling approaches describe in the session [2] are studied.

#### 3.1 Structural model with synthesis based approach

The model is built according to two points of view. The first is the internal material point of view. It corresponds to the concept of cycle. And the second is the data point of view. It corresponds to the data states during the transmission through the PLC (fig. 4). This model is divided into three parts:

- on the part "a", the internal material point of view with the PLC CPU cycle,
- on the part "c", the data point of view with the data states in the PLC,
- on the part "b" are the elements which allow to synchronizing the two points of view.

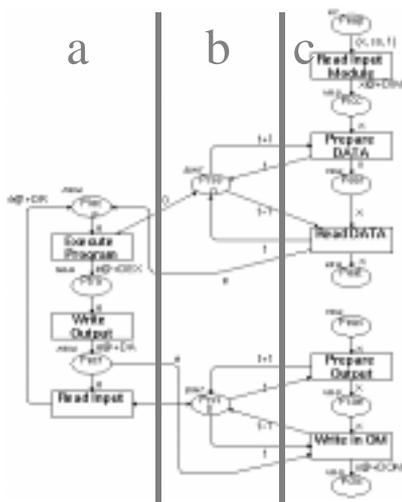


Fig. 4. CTPN structural model with synthesis based approach

The model obtained contains 12 places and 8 transitions.

#### 3.2 Model with extended transitions with synthesis based approach

The fig. 5 illustrates dynamic behavior define session [2.1, Synthesis-based approach]. We notice that the global behavior is a simple delay. The model will calculate this delay within a transition code according to the PLC input time. We know:

- the duration of the read Input phase ( $D_R$ )
- the duration of the program EXecuting phase ( $D_{EX}$ )
- the duration of the assignment Output phase ( $D_A$ )
- the values of the delays due to the output and input modules (respectively  $D_{IM}$  and  $D_{OM}$ ).

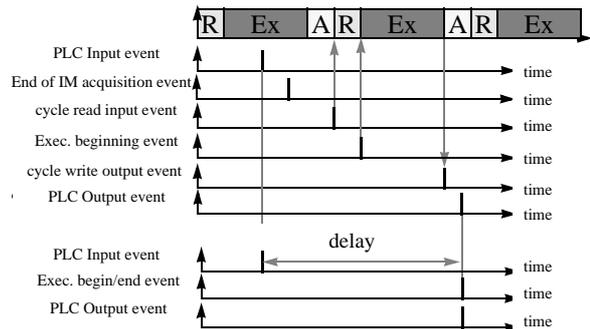


Fig. 5. detailed chronograms of CPU activity (from manufacturer documents)

After acquisition by the input module, we distinguish two cases. The cycle is or is not in the input read phase. For any event arrived at the date  $t$ , we calculate the relative position in the cycle (noted  $P_{CY}(t)$ ) after acquisition of the input module.

$$P_{CY}(t) = (t - D_{IM}) \bmod D_{CY} \text{ with } D_{CY} = D_R + D_{EX} + D_A$$

$$\text{if } P_{CY}(t) \leq D_I \Rightarrow \text{delay}(t) = D_R + D_{EX} - P_{CY}(t) - D_O + D_{IM} + D_{OM}$$

$$\text{if } P_{CY}(t) > D_I \Rightarrow \text{delay}(t) = D_R + D_{EX} + D_{CY} - P_{CY}(t) - D_O + D_{IM} + D_{OM}$$

The fig. 6 represents a model with extended transitions of the PLC behavior. Statements within the code zone correspond to the calculation of  $\text{delay}(t)$  function. The obtained model contains 2 places and 1 transition.



Fig. 6. CTPN model with extended transitions and with synthesis-based approach

### 3.3 Stochastic model with synthesis-based approach

The global behavior of a PLC is a delay (fig. 5). This delay is bounded and its limits are equal to:

$$\begin{aligned} \min &= D_{EX} - D_A + D_{IM} + D_{OM} \\ \max &= D_{EX} + D_{CY} - D_A + D_{IM} + D_{OM} \end{aligned}$$

The fig. 7 represents the stochastic model of the PLC dynamic behavior. The function  $Rint(\text{Min}, \text{max})$  returns a random integer between Min and max. The obtained model contains 2 places and 1 transition.



Fig. 7. CTPN stochastic model with synthesis-based approach

### 3.4 Structural model with identification-based approach

The design is made as describe in the session [3.1]. A new point of view is added. This point of view corresponds to the internal management tasks. The fig. 8 represents the dynamic behavior model of the PLC.

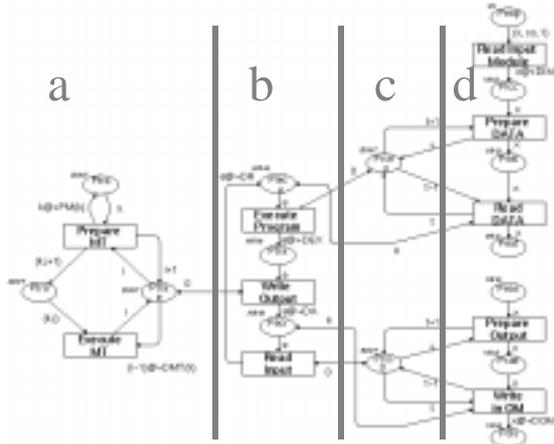


Fig. 8. CTPN structural model with identification-based approach

This model is divided into four parts:

- on the part "a", the internal point of view of the management tasks of the PLC,
- on the part "b", the internal material point of view with the PLC cycle,
- on the part "d", the data point of view with the data states in the PLC,
- on the part "c" are the elements which make it possible to synchronize the two points of view.

The obtained model contains 15 places and 10 transitions.

### 3.5 Model with extended transition with identification-based approach

The dynamic behavior of the PLC is a delay. But contrary to the session [3.2], its calculation is not simple. It is however possible to get the date of next the end of cycle knowing the previous date.  $EC_i$  is the last end of cycle date known, we consider the following variables (fig. 9):

- $AT_i$ , duration passed since the last management task
- $BT_i$ , lasted before the next management task

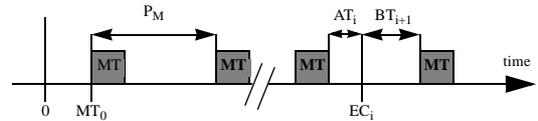


Fig. 9.  $EC_i$  Positioning compared to the management tasks

$$AT_i = (EC_i - MT_0 - D_{MT}) \text{ Mod } P_M$$

$$BT_i = P_M - D_{MT} - AT_i \text{ where Mod is the modulo operator.}$$

We obtain two cases described fig. 10 and fig. 11.

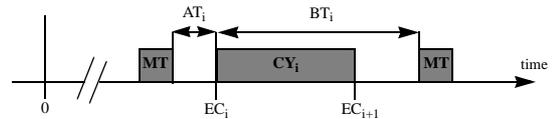


Fig. 10. Case  $BT_i > D_{EX}$

In this case, cycle  $CY_i$  has time to proceed before the next task of management. The calculation of the next end of cycle ( $EC_{i+1}$ ) is thus  $EC_{i+1}(EC_i) = EC_i + D_{CY}$

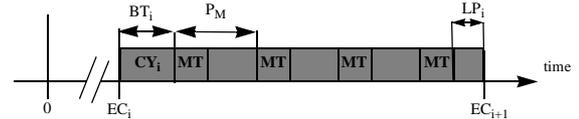


Fig. 11. Case  $BT_i \leq D_{EX}$

In this second case, the cycle does not have time to be carried out. One or more management tasks will be intercalated. We define the following variables:

- $I_{MT}$ , interval between two tasks of management
- $np_i$ , a number of pieces of  $CY_i$  of size  $I_{MT}$
- $LP_i$ , last piece of  $CY_i$

The next end of cycle ( $EC_{i+1}$ ) is calculated as following:  $EC_{i+1}(EC_i) = EC_i + BT_{i+1} + np_i * P_M + D_{MT} + LP_i$

$$\text{with } I_{MT} = P_M - D_{MT}$$

$$np_i = (D_{EX} - BT_i) \text{ Div } I_{MT}$$

$$LP_i = (D_{EX} - BT_i) \text{ Mod } I_{MT}$$

Where  $Div$  is the integer division.

We determine the date of end of cycle  $EC(t)$  which follows a date  $t$ . Two calculated dates are considered:

the last (L) and before last (BL). We use the algorithm described fig. 12.

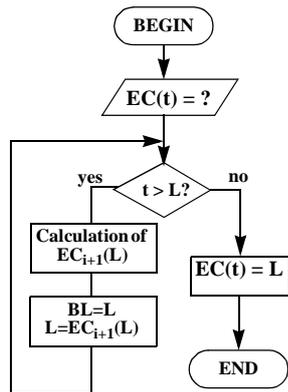


Fig. 12. EC(t) calculation algorithm

The fig. 13 represents the dynamic behavior model of the PLC. One can distinguish three codes. The "code 1" memorize the event arrival date (noted t). The "code 2" is the calculation of the algorithm described fig. 12. The "code 3" is the calculation of the transmission delay through the PLC (noted delay). The obtained model contains 4 places and 3 transitions.

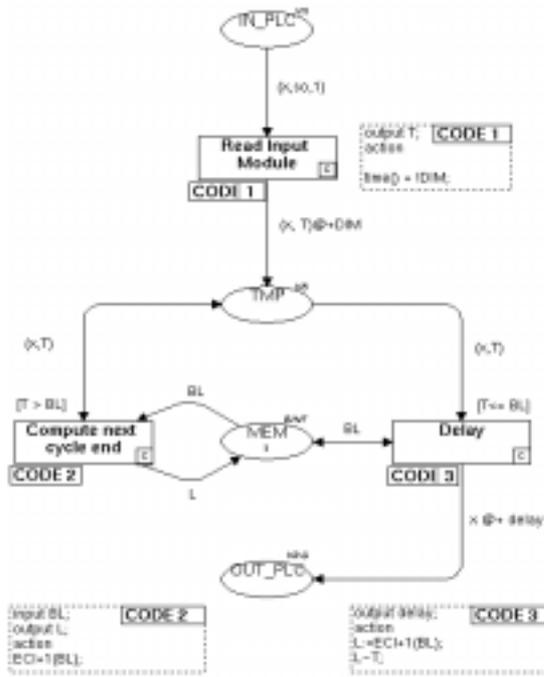


Fig. 13. CTPN model with extended transition and with identification-based approach

### 3.6 Stochastic approach with identification-based approach

According the study in the session [3.5], we can say that the PLC cycle will be interrupted  $n$  or  $n+1$  times. We defines  $n$  as follow:  $n = D_{CY} \text{ Div } I_{MT}$ . We consider that  $EC_i$  can take  $I_{MT}$  locations between two management tasks. During  $(I_{MT} - LP_i)$  the first values (line "a" fig. 14), the cycle is stopped  $n$  time. The line "b" fig. 14 corresponds to the limit. From this

limit to the  $I_{MT}$ th value (line "c" fig. 14), the number of interruption grows to  $n+1$ .

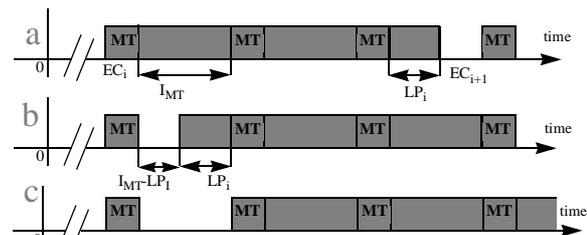


Fig. 14. Different locations of  $EC_i$  in the  $I_{MT}$  interval

The model fig. 15 looks like the session [3.3] model. Except that the values of Max and Min are raised  $k$  time  $D_{MT}$ . A random variable allows deciding if  $k$  is equal to  $n$  or  $n+1$ . The obtained model contains 2 places and 1 transition.



Fig. 15. CTPN stochastic model with identification-based approach

## 4. MODELS COMPARISON

In the session [3], we designed 6 CTPN models corresponding to the behavior of a PLC. In this part, we will present a comparison method. This method will be applied to the six models. This method helps the designer for the choice of the model.

### 4.1 Comparison method

Each model is simulated. The input excitations of the models will be the same for each one. During simulations, we evaluate the transmission delay through the PLC. After a simulation, we obtain 59660 values. Each model is compared with a real PLC. Measures of the real PLC are taken as reference. The values of the reference are obtained by experimentation on our framework. The identification duration takes eight hours. We compare the 6 models histograms with the reference histogram. The classes of the model histograms correspond to the classes of the reference histogram. The criteria used to compare the different histograms are:

- the minimum delay accuracy ( $\Delta_{min}$ )
- the maximum delay accuracy ( $\Delta_{max}$ )
- the percentage of value in common with the reference
- the simulation time ( $D_{SIM}$ )

We define the following variables:

- $D_{min}(R)$ ,  $D_{max}(R)$ , the lowest and biggest transmission time for the reference
- $D_{min}(M)$ ,  $D_{max}(M)$ , the lowest and biggest transmission time for the M model
- we define the  $\Delta_{min}(M) = \frac{D_{min}(R) - D_{min}(M)}{D_{min}(R)}$
- $\Delta_{max}$  is define as  $\Delta_{min}$

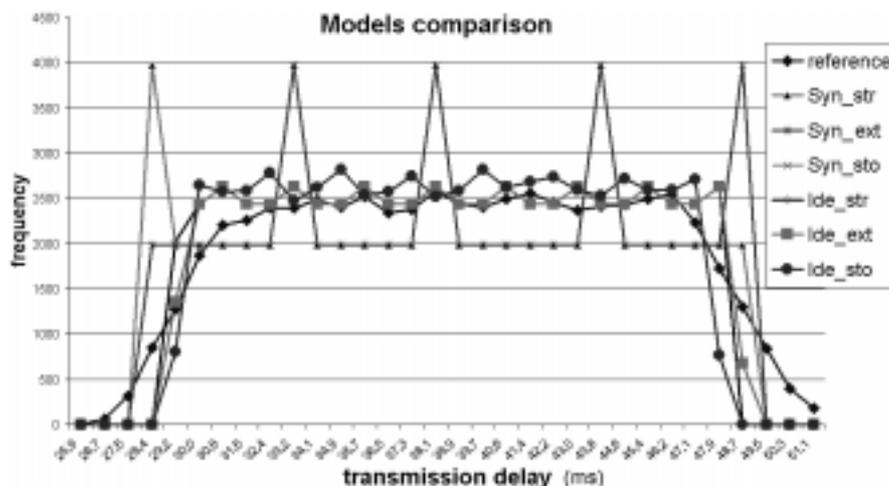
#### 4.2 Built models comparison

The fig. 16 represents the various histograms obtained after simulation and after actual PLC identification. For an easier graph reading, the histogram bars were replaced by lines. The Table 1 contains the analysis results of simulations. The best values are indicated in bold. With the Table 1, the designer can choose the model, which is appropriate according to his criteria. For example, if the criteria are the statistical accuracy and the simulation speed, the best model is the stochastic model with identification-based approach. On the other hand, the model with extended transitions and with synthesis-based approach would be interesting for a fast simulation and a limit accuracy without going beyond. But, if the criterion is the accuracy on the min and max boundaries, the best model is the stochastic model with synthesis-based approach. For our general aim relating the response time evaluation in the whole architecture, this is the best criterion.

Table 1 Results analysis

	Synthesis-based approach			Identification-based approach		
	Struct.	Ext. tr.	Stoch.	Struct.	Ext. tr.	Stoch.
$\Delta_{max}$	-15,52%	-12,82%	<b>+5,16%</b>	-16,42%	-13,75%	-17,82%
$\Delta_{min}$	-6,41 %	-9,1 %	<b>-0,75%</b>	-6,23 %	-10,7%	-11,79%
% in common	68,28%	68,28%	67,91%	75,27%	87,97%	<b>89,91%</b>
$D_{SIM}$	13h15'	<b>49'25"</b>	<b>49'25"</b>	13h55'	4h22'	<b>49'25"</b>

Fig. 16. Model histograms comparison



## 5. CONCLUSION

In this paper, we presented various approaches allowing to design behavior models for control system. After having illustrated these approaches, we presented a comparison method for the designed models. The presented comparison relates to alone models. But this model aims to be used as component of a larger model. The works in progress extend the method of comparison to study the influence of an equipment model on the behavior of a whole architecture model.

## 6. BIBLIOGRAPHY

- Guldner J. and I. Theis (1999). Comparison of redundancy for safety relevant automotive control system. In: *Proceeding of ECC'99*, Karlsruhe, Germany.
- Jensen K. (1997) Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Basic Concepts. Monographs in *Theoretical Computer Science*, Springer-Verlag.
- Kiencke U., R. Majjad and S. Kramer (1999). Modeling and performance analysis of a hybrid driver model. In: *Control Engineering Practice*, 7 (8), pp. 985-991
- List V., A. Chovin, P. Coat, G. Renard and P. Soeters (1998). Safety integration in distributed automation systems, fieldbus - dependability. In: *Proceeding of INCOM'98*, Nancy, France.
- Meunier P. and B. Denis (1997). Validation du comportement dynamique des architectures de conduite des systèmes de production par simulation. In: *Proceeding of MOSIM'97*, Rouen, France.

## COMPARISON OF DIFFERENT MODELING APPROACHES IN SIMULATION OF PROGRAMMABLE LOGIC CONTROLLER

P. Meunier, B. Denis and J.-J. Lesage

*Laboratoire Universitaire de Recherche en Production Automatisée,  
ENS de Cachan, 61 avenue du Président Wilson, F-94235 Cachan Cedex, France  
{meunier, denis, lesage}@lurpa.ens-cachan.fr - <http://www.lurpa.ens-cachan.fr>  
Tel: 33 1.47.40.27.62 Fax: 33 1.47.40.22.20*

**Abstract:** To design the control of automated systems, the designer must evaluate the performances of various solutions. Our research deals with the evaluation of distributed control architecture. We have chosen Colored Timed Petri Nets (CTPN) models for describe behaviors of each equipment of control and we use design/CPN simulation tool to assess response times performance. The aims of this paper are to build various Programmable Logic Controller (PLC) behavior models and to select the best according to our needs in performance evaluations. Various approaches for the creation of CTPN models are discussed. Some of them are based on synthesis techniques and some of them are based on an identification technique. Finally, a comparison method is presented. It will allow the designer to choose the right model for the simulation in the evaluation stage.

**Keywords:** Modelling, Petri-nets, Simulation, Control system analysis, Programmable logic controllers

### 1. INTRODUCTION

To design architectures of distributed control systems, the designer must deal with a phase of performance evaluation. Many studies are centered on the evaluations of functional structures of control (Kiencke *et al.* 1999) or components (for ex: API, fieldbus...) (List *et al.* 1998), or sometimes on the redundancy of components (Gulner and Theis 1999). The general goal of our research (Meunier and Denis 1997) is to evaluate response time performance of a whole architecture dealing with the functional and the material

point of view and using simulation of Colored Timed Petri Nets (CTPN) models. A library of behavior models of control equipments has been setup and design/CPN simulator is used to perform evaluations. The aim of this paper is to prepare a Programmable Logic Controller (PLC) behavior model as good as possible according to our needs of evaluation. In session [2], we describe the various approaches in design of CTPN models. Then, in session [3] we apply these approaches to the creation of PLC behavior models. Finally, in last session, we present a comparison method of these models.

## 2. MODELING APPROACHES

In this part, we describe the different ways to model an equipment behavior and the different ways to build a Petri net models. In order to illustrate these different ways, we study an industrial programmable logic controller (PLC) as example. This PLC is composed with:

- an input module (IM) which handles sensors signals and prepares them for the CPU module;
- a CPU module which executes users programs (we assume that no preemptive task is performed);
- and an output module (OM) which transforms CPU module outputs into actuator signals.

Behaviors of input and output modules, specifying by manufacturer, are describes as delays (noted respectively  $D_{IM}$  and  $D_{OM}$ ).

### 2.1 Different ways to get the behavior of an equipment

Design of dynamic behavior model requires informations on the equipment behavior. There are two ways to extract the behavior: synthesis-based approach and identification-based approach.

*Synthesis-based approach.* This approach consists in the analysis of manufacturer documentation. The behavior of PLC CPU is cyclic. The cycle is made up of (fig. 1):

- a Read input phase (R),
- a program EXecuting phase (EX),
- an Assignment output phase (A).

The cycle time depends of the program size. During the operating mode the program size is frozen and the cycle time is a constant value. This value could be obtained from the execution time of each statements of the user program or from the CPU rate speed given in kilo statement per second. Moreover, we consider that the PLC is periodic, the various phases of the cycle thus have fixed durations.

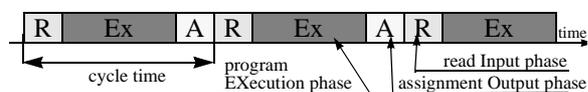


Fig. 1. Chronogram of CPU activity of a cyclic PLC according manufacturer documentation.

*Identification-based approach.* This approach consists in identifying the real equipment behavior. The equipment inputs are solicited and then, the measurement of its outputs provides us a view of its actual behavior. To study logical signals on industrial equipments, we built an experimental framework composed with signal generators and a logical analyzer. All of these devices are managed with a workstation across a GPIB network. Thus we are able to perform a large number of measurement with various kind of solicitation in batch mode. 1200 measurement of

cycle time of our CPU module are reported in the histogram of fig. 2.

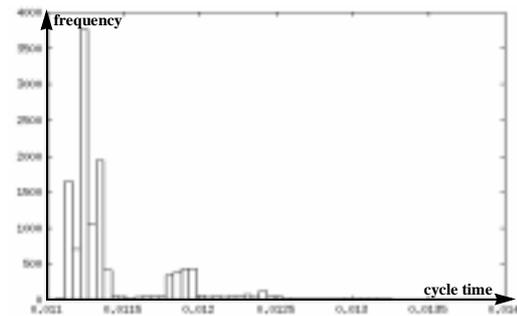


Fig. 2. Histogram of PLC cycle time measures

Measurement analysis gives the dynamic behavior of the PLC. This behavior is cyclic. We discover some periodicity disturbances. To model this behavior, we introduce new tasks in the CPU activity called internal management tasks (MT) (fig. 3). Management tasks of the API are defined as following:

- Period of the management task  $k$ :  $P_M(k)$ ,
- Duration of the management task  $k$ :  $D_{MT}(k)$

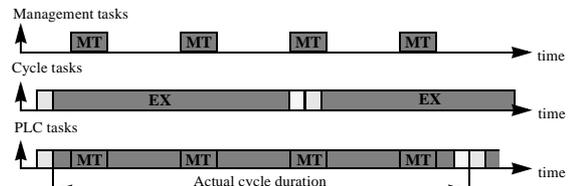


Fig. 3. Chronogram of CPU activity according identification

### 2.2 Different ways to build a Petri net model

Our models are made using colored timed Petri nets (CTPN) within the meaning of Jensen K (1997). We propose three ways to build a Petri net model. The structural, the extended transition and the stochastic model.

*Structural and deterministic model.* This manner of building a model is very current. Each aspect of the behavior of the modeled system is translated into places and transitions. The model reproduces accurately and in a deterministic way the behavior of the modeled system. Each change of state of the modeled system is thus modeled by the firing of a transition in the model. This type of model is simple to build but during the simulation much of transitions will have to be fired. Simulation is thus long and expensive in CPU time.

*Deterministic model with extended transitions.* Here the model also reproduces accurately and in a deterministic way the behavior of the modeled system. But some aspects of the behavior of the modeled system are translated by a list of instructions associated with certain transitions. The Jensen (1997) CTPN allows us to associate codes lines with the transitions. During simulation, this code is executed when the associated

transition is fired. This approach makes it possible to distribute the complexity of a behavior between a simple transition-place structure and simple algorithms associated with transitions.

*Stochastic model.* We build a model whose behavior is on average similar to that of the modeled system. Stochastic transitions are used in the model. During a simulation, the current values of the observed parameters are not significant. But the model is built in order to meet the average of these values to the modeled system.

### 3. MODEL DESIGN

In this session, all the different modeling approaches describe in the session [2] are studied.

#### 3.1 Structural model with synthesis based approach

The model is built according to two points of view. The first is the internal material point of view. It corresponds to the concept of cycle. And the second is the data point of view. It corresponds to the data states during the transmission through the PLC (fig. 4). This model is divided into three parts:

- on the part "a", the internal material point of view with the PLC CPU cycle,
- on the part "c", the data point of view with the data states in the PLC,
- on the part "b" are the elements which allow to synchronizing the two points of view.

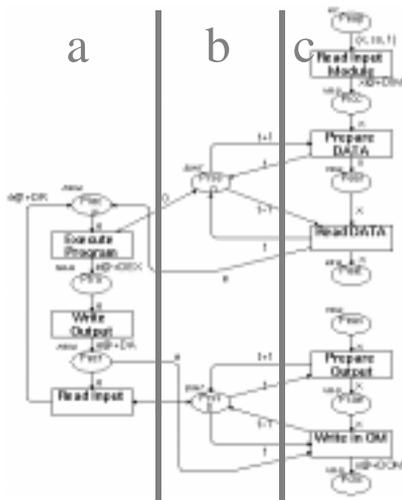


Fig. 4. CTPN structural model with synthesis based approach

The model obtained contains 12 places and 8 transitions.

#### 3.2 Model with extended transitions with synthesis based approach

The fig. 5 illustrates dynamic behavior define session [2.1, Synthesis-based approach]. We notice that the global behavior is a simple delay. The model will calculate this delay within a transition code according to the PLC input time. We know:

- the duration of the read Input phase ( $D_R$ )
- the duration of the program EXecuting phase ( $D_{EX}$ )
- the duration of the assignment Output phase ( $D_A$ )
- the values of the delays due to the output and input modules (respectively  $D_{IM}$  and  $D_{OM}$ ).

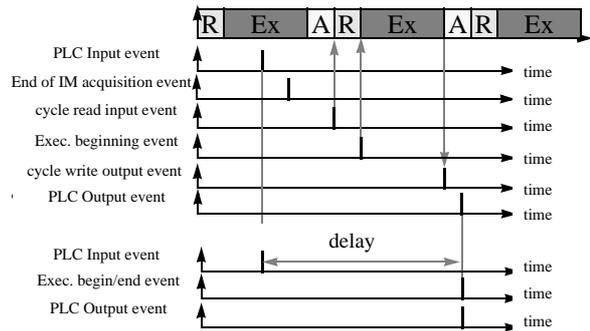


Fig. 5. detailed chronograms of CPU activity (from manufacturer documents)

After acquisition by the input module, we distinguish two cases. The cycle is or is not in the input read phase. For any event arrived at the date  $t$ , we calculate the relative position in the cycle (noted  $P_{CY}(t)$ ) after acquisition of the input module.

$$P_{CY}(t) = (t - D_{IM}) \bmod D_{CY} \text{ with } D_{CY} = D_R + D_{EX} + D_A$$

$$\text{if } P_{CY}(t) \leq D_I \Rightarrow \text{delay}(t) = D_R + D_{EX} - P_{CY}(t) - D_O + D_{IM} + D_{OM}$$

$$\text{if } P_{CY}(t) > D_I \Rightarrow \text{delay}(t) = D_R + D_{EX} + D_{CY} - P_{CY}(t) - D_O + D_{IM} + D_{OM}$$

The fig. 6 represents a model with extended transitions of the PLC behavior. Statements within the code zone correspond to the calculation of  $\text{delay}(t)$  function. The obtained model contains 2 places and 1 transition.

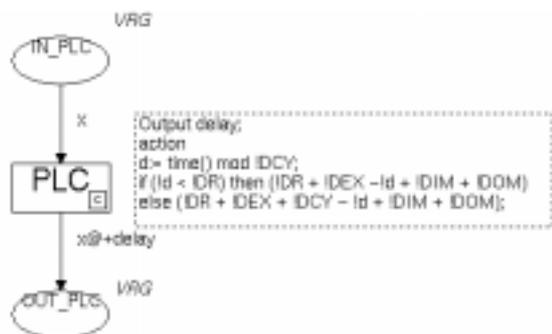


Fig. 6. CTPN model with extended transitions and with synthesis-based approach

### 3.3 Stochastic model with synthesis-based approach

The global behavior of a PLC is a delay (fig. 5). This delay is bounded and its limits are equal to:

$$\begin{aligned} \min &= D_{EX} - D_A + D_{IM} + D_{OM} \\ \max &= D_{EX} + D_{CY} - D_A + D_{IM} + D_{OM} \end{aligned}$$

The fig. 7 represents the stochastic model of the PLC dynamic behavior. The function  $Rint(\text{Min}, \text{max})$  returns a random integer between Min and max. The obtained model contains 2 places and 1 transition.

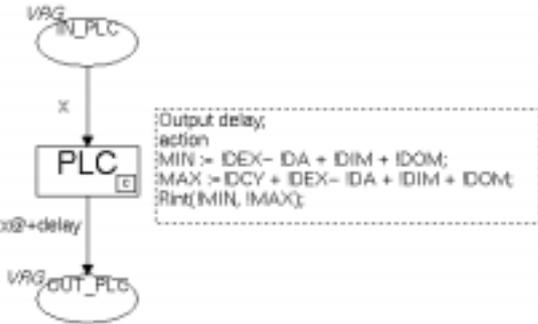


Fig. 7. CTPN stochastic model with synthesis-based approach

### 3.4 Structural model with identification-based approach

The design is made as describe in the session [3.1]. A new point of view is added. This point of view corresponds to the internal management tasks. The fig. 8 represents the dynamic behavior model of the PLC.

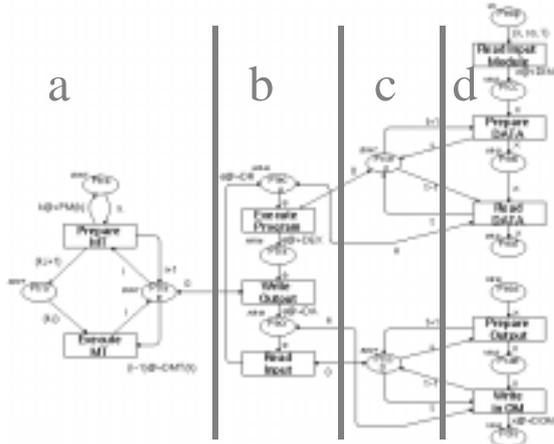


Fig. 8. CTPN structural model with identification-based approach

This model is divided into four parts:

- on the part "a", the internal point of view of the management tasks of the PLC,
- on the part "b", the internal material point of view with the PLC cycle,
- on the part "d", the data point of view with the data states in the PLC,
- on the part "c" are the elements which make it possible to synchronize the two points of view.

The obtained model contains 15 places and 10 transitions.

### 3.5 Model with extended transition with identification-based approach

The dynamic behavior of the PLC is a delay. But contrary to the session [3.2], its calculation is not simple. It is however possible to get the date of next the end of cycle knowing the previous date.  $EC_i$  is the last end of cycle date known, we consider the following variables (fig. 9):

- $AT_i$ , duration passed since the last management task
- $BT_i$ , lasted before the next management task

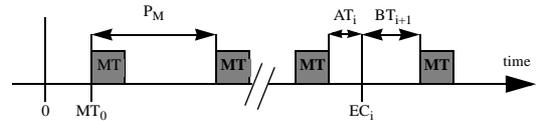


Fig. 9.  $EC_i$  Positioning compared to the management tasks

$$AT_i = (EC_i - MT_0 - D_{MT}) \text{ Mod } P_M$$

$$BT_i = P_M - D_{MT} - AT_i \text{ where Mod is the modulo operator.}$$

We obtain two cases described fig. 10 and fig. 11.

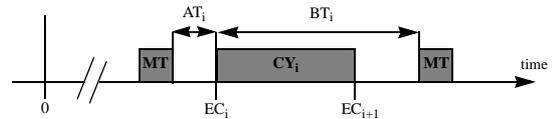


Fig. 10. Case  $BT_i > D_{EX}$

In this case, cycle  $CY_i$  has time to proceed before the next task of management. The calculation of the next end of cycle ( $EC_{i+1}$ ) is thus  $EC_{i+1}(EC_i) = EC_i + D_{CY}$

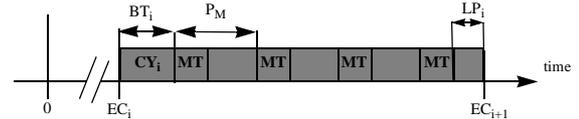


Fig. 11. Case  $BT_i \leq D_{EX}$

In this second case, the cycle does not have time to be carried out. One or more management tasks will be intercalated. We define the following variables:

- $I_{MT}$ , interval between two tasks of management
- $np_i$ , a number of pieces of  $CY_i$  of size  $I_{MT}$
- $LP_i$ , last piece of  $CY_i$

The next end of cycle ( $EC_{i+1}$ ) is calculated as following:  $EC_{i+1}(EC_i) = EC_i + BT_{i+1} + np_i * P_M + D_{MT} + LP_i$

$$\text{with } I_{MT} = P_M - D_{MT}$$

$$np_i = (D_{EX} - BT_i) \text{ Div } I_{MT}$$

$$LP_i = (D_{EX} - BT_i) \text{ Mod } I_{MT}$$

Where  $Div$  is the integer division.

We determine the date of end of cycle  $EC(t)$  which follows a date  $t$ . Two calculated dates are considered:

the last (L) and before last (BL). We use the algorithm described fig. 12.

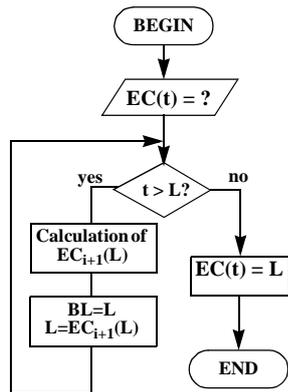


Fig. 12. EC(t) calculation algorithm

The fig. 13 represents the dynamic behavior model of the PLC. One can distinguish three codes. The "code 1" memorize the event arrival date (noted t). The "code 2" is the calculation of the algorithm described fig. 12. The "code 3" is the calculation of the transmission delay through the PLC (noted delay). The obtained model contains 4 places and 3 transitions.

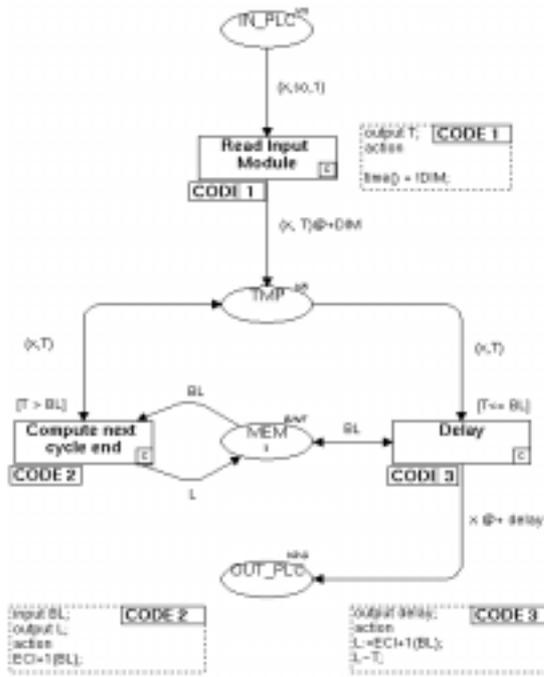


Fig. 13. CTPN model with extended transition and with identification-based approach

### 3.6 Stochastic approach with identification-based approach

According the study in the session [3.5], we can say that the PLC cycle will be interrupted  $n$  or  $n+1$  times. We defines  $n$  as follow:  $n = D_{CY} \text{ Div } I_{MT}$ . We consider that  $EC_i$  can take  $I_{MT}$  locations between two management tasks. During  $(I_{MT} - LP_i)$  the first values (line "a" fig. 14), the cycle is stopped  $n$  time. The line "b" fig. 14 corresponds to the limit. From this

limit to the  $I_{MT}$ th value (line "c" fig. 14), the number of interruption grows to  $n+1$ .

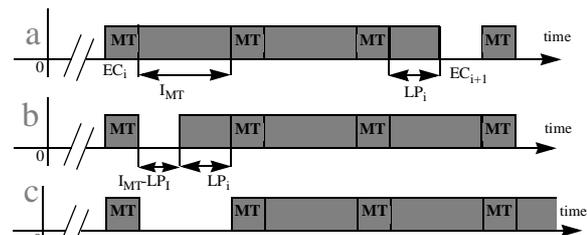


Fig. 14. Different locations of  $EC_i$  in the  $I_{MT}$  interval

The model fig. 15 looks like the session [3.3] model. Except that the values of Max and Min are raised  $k$  time  $D_{MT}$ . A random variable allows deciding if  $k$  is equal to  $n$  or  $n+1$ . The obtained model contains 2 places and 1 transition.



Fig. 15. CTPN stochastic model with identification-based approach

## 4. MODELS COMPARISON

In the session [3], we designed 6 CTPN models corresponding to the behavior of a PLC. In this part, we will present a comparison method. This method will be applied to the six models. This method helps the designer for the choice of the model.

### 4.1 Comparison method

Each model is simulated. The input excitations of the models will be the same for each one. During simulations, we evaluate the transmission delay through the PLC. After a simulation, we obtain 59660 values. Each model is compared with a real PLC. Measures of the real PLC are taken as reference. The values of the reference are obtained by experimentation on our framework. The identification duration takes eight hours. We compare the 6 models histograms with the reference histogram. The classes of the model histograms correspond to the classes of the reference histogram. The criteria used to compare the different histograms are:

- the minimum delay accuracy ( $\Delta_{min}$ )
- the maximum delay accuracy ( $\Delta_{max}$ )
- the percentage of value in common with the reference
- the simulation time ( $D_{SIM}$ )

We define the following variables:

- $D_{min}(R)$ ,  $D_{max}(R)$ , the lowest and biggest transmission time for the reference
- $D_{min}(M)$ ,  $D_{max}(M)$ , the lowest and biggest transmission time for the M model
- we define the  $\Delta_{min}(M) = \frac{D_{min}(R) - D_{min}(M)}{D_{min}(R)}$
- $\Delta_{max}$  is define as  $\Delta_{min}$

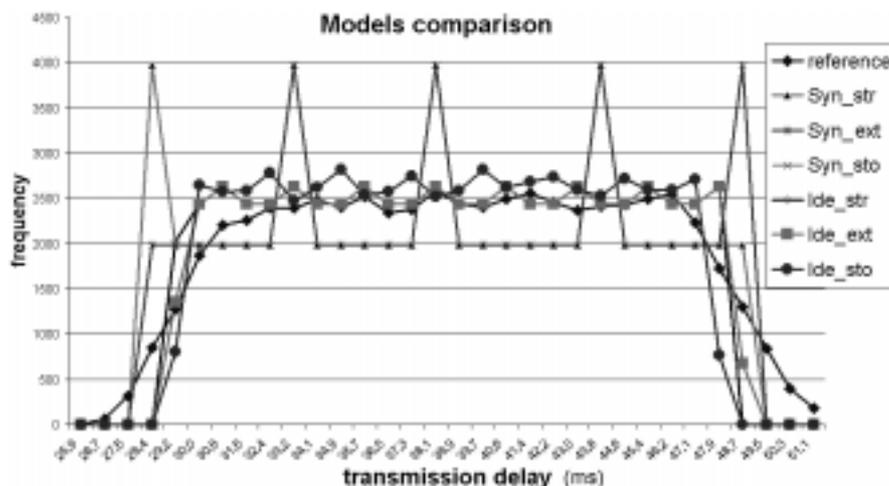
#### 4.2 Built models comparison

The fig. 16 represents the various histograms obtained after simulation and after actual PLC identification. For an easier graph reading, the histogram bars were replaced by lines. The Table 1 contains the analysis results of simulations. The best values are indicated in bold. With the Table 1, the designer can choose the model, which is appropriate according to his criteria. For example, if the criteria are the statistical accuracy and the simulation speed, the best model is the stochastic model with identification-based approach. On the other hand, the model with extended transitions and with synthesis-based approach would be interesting for a fast simulation and a limit accuracy without going beyond. But, if the criterion is the accuracy on the min and max boundaries, the best model is the stochastic model with synthesis-based approach. For our general aim relating the response time evaluation in the whole architecture, this is the best criterion.

Table 1 Results analysis

	Synthesis-based approach			Identification-based approach		
	Struct.	Ext. tr.	Stoch.	Struct.	Ext. tr.	Stoch.
$\Delta_{max}$	-15,52%	-12,82%	<b>+5,16%</b>	-16,42%	-13,75%	-17,82%
$\Delta_{min}$	-6,41 %	-9,1 %	<b>-0,75%</b>	-6,23 %	-10,7%	-11,79%
% in common	68,28%	68,28%	67,91%	75,27%	87,97%	<b>89,91%</b>
$D_{SIM}$	13h15'	<b>49'25"</b>	<b>49'25"</b>	13h55'	4h22'	<b>49'25"</b>

Fig. 16. Model histograms comparison



## 5. CONCLUSION

In this paper, we presented various approaches allowing to design behavior models for control system. After having illustrated these approaches, we presented a comparison method for the designed models. The presented comparison relates to alone models. But this model aims to be used as component of a larger model. The works in progress extend the method of comparison to study the influence of an equipment model on the behavior of a whole architecture model.

## 6. BIBLIOGRAPHY

- Guldner J. and I. Theis (1999). Comparison of redundancy for safety relevant automotive control system. In: *Proceeding of ECC'99*, Karlsruhe, Germany.
- Jensen K. (1997) Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Basic Concepts. Monographs in *Theoretical Computer Science*, Springer-Verlag.
- Kiencke U., R. Majjad and S. Kramer (1999). Modeling and performance analysis of a hybrid driver model. In: *Control Engineering Practice*, 7 (8), pp. 985-991
- List V., A. Chovin, P. Coat, G. Renard and P. Soeters (1998). Safety integration in distributed automation systems, fieldbus - dependability. In: *Proceeding of INCOM'98*, Nancy, France.
- Meunier P. and B. Denis (1997). Validation du comportement dynamique des architectures de conduite des systèmes de production par simulation. In: *Proceeding of MOSIM'97*, Rouen, France.