2009 Ninth International Conference on Intelligent Systems Design and Applications

Identification of Petri net models based on an asymptotic approach

M. E. Meda-Campaña, F. J. Lopez-Lopez, Cuauhtemoc Lopez-Martin, and Arturo Chavoya
Information Systems Department
University of Guadalajara
Zapopan, Jal; Mexico
{emeda, fjlopez, cuauhtemoc, achavoya}@cucea.udg.mx

Abstract— **The identification problem considered in this work, consists in compute an Interpreted Petri Net (IPN) model, in proportion as new output signals of the system are observed. The identification problem becomes complex when the complete state of the system cannot be fully measured. The state information that is not observed is inferred during the identification process allowing the computed model represents the observed system behavior. As the system evolves new information is revealed and the wrong dependencies are eliminated in order to update the computed model. Given this problem, in this paper are presented the needed algorithms to identify a class of Petri Nets (PN) known as state machines.**

Keywords: Identification, discrete event systems, modelling methods based on interpreted petri nets.

I. Introduction

Analogous to identification of continuous dynamic systems, identification of Discrete Event Systems (DES) consist in find out the mathematical model that describes the system behavior. In this work, the model is in general, an abstract machine that represents the evolution of internal states, inputs, and outputs of a DES.

This paper deals with the identification process from a passive point of view. This approach is best adapted to on-line operation; where the system is identified from the observation of its output signals. A succession of models is built as the system evolves in such way that the current model represents the observed system behavior; so, every new computed model acquires more details than the previous one, approaching to the actual model of the system; this strategy is called asymptotic identification [7][8].

Previous works on the matter have been published in the computer science community; the problem was first stated as the determination of a Finite Automata (FA) that accepts a given regular language. In [2] a FA is computed from positive samples (accepted words) of the language; it is shown that this problem is NP complete. There exists several works that presents polynomial identification algorithms to identify sub-classes of regular languages: in [1] the identification algorithm determines a FA using positive samples (accepted words) of a zero-reversible language. Finally a work dealing with Petri nets [4] presents an algorithm that builds a deterministic I-reversible automata (in PN terms); it also uses positive samples of the system language.

The identification approach, can be used in many different areas like in manufacturing systems, system communications, etc., when the model cannot be computed because there not exists enough information about the involved variables. Also the identification approach can be used in an adaptive control scheme, because the model is adapted to new observed conditions.

This article is organized as follows. After preparing the basic definitions and notations in section [2], we introduce the asymptotic identification problem and the asymptotic identification approach in section [3]. In section [4] is presented the identification algorithm for the class of Petri nets known as State Machines. Finally, the concluding remarks and discussion of future work follow in section [5].

II. Background

This section Introduces the definition of Interpreted PN (IPN) and related concepts on DES modeling used in this work. Fine PN surveys can be found in [3] and [6].

This work uses Interpreted Petri Nets (IPN) [5], an extension to the PN that can represent DES input and output signals.

Definition 1: An Interpreted Petri Net is the 5-tuple $Q = (N, \Sigma, \Phi, \lambda, \varphi)$ where N is a PN with initial marking M_0, $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_r\}$ is a finite set of elements σ_i called input symbols, $\Phi = \{\phi_1, \phi_2, ..., \phi_p\}$ is a finite set of elements ϕ_i called output symbols, $\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$ is a labeling transition function, where ε is the null symbol and $\varphi : \mathbf{R}(N, M_0) \rightarrow \{\Phi \cup \{\varepsilon\}\}^q$ is an output function; where $\mathbf{R}(N, M_0)$ is the set of reachable markings and q is the number of sensors associated to places in Q.

In an IPN, if a transition t_j is enabled and the input signal $\lambda(t_j) = a_i \neq \varepsilon$ is present then t_j must fire else if $\lambda(t_j) = \varepsilon$ then t_j can be fired.

Definition 2: A place $p_i \in P$ is said to be **measurable** if it has a sensor signal assigned, and **non measurable** otherwise. Non measurable places are depicted as dark circles.

This paper focuses on the case where $\Phi = \mathbb{Z}^+$ (the nonnegative integer numbers), and $\varphi : \mathbf{R}(N, M_0) \longrightarrow \{\mathbb{Z}^+\}^q$ is a linear function that can be represented by a $q \times n$ matrix $\varphi = [\varphi_{ij}]$, where the i-th row vector φ_i of φ is the transpose of the elemental vector e_j ($e_j[i \neq j] = 0$, $e_j[j] = 1$), if p_j is the i-th measurable place ($\varphi_i = e_j^T$), according to the order given by the place labeling.

The state equation of an IPN is completed as:

$$\begin{aligned} M_{k+1} &= M_k + Cv_k \\ y_k &= \varphi M_k \end{aligned} \tag{1}$$

where $y_k = \varphi M_k$ is a $q \times 1$ vector called output symbol. Notice that an output symbol is the marking vector of the measurable places in the marking M_k.

The incidence matrix C of an IPN Q can be decomposed as $C = \begin{bmatrix} \varphi C \\ \gamma C \end{bmatrix}$, where φC and γC represents the measurable and no measurable places of Q respectively, and γ is a linear function defined in a similar way than φ function but considering the non measurable places instead of the measurable places.

Definition 3: $^\bullet t_j$ denotes the set of all places p_i such that $I(p_i, t_j) \neq 0$ and $t_j^\bullet$ denotes the set of all places p_i such that $O(p_i, t_j) \neq 0$. $^\bullet p_i$ denotes the set of all transitions t_j such that $O(p_i, t_j) \neq 0$ and $p_i^\bullet$ denotes the set of all transitions t_j such that $I(p_i, t_j) \neq 0$.

Definition 4: A firing sequence of an IPN Q is a sequence $\sigma = t_i t_j ... t_k$ such that $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ...M_w \xrightarrow{t_k} ....$ The set of all firing sequences is called the firing language of Q, $\pounds(Q) = \{\sigma | \sigma = t_i t_j ... t_k$ and $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} ...M_w \xrightarrow{t_k} ...\}$.

Definition 5: Let Q be an IPN. Two transition t_i and t_j of Q form a dependency $[t_i, t_j]$ iff $\exists p_k$ such that $p_k \in t_i^\bullet$ and $p_k \in^\bullet t_j$. If p_k is a non measurable place, then $[t_i, t_j]$ will be called a non measurable dependency **(**NDep**)**, otherwise will be called a measurable dependency (MDep**)**. We use the notation $p_k = [t_i, t_j]$ referring to the fact that the place p_k form a dependency between t_i and t_j. The set of all possible dependencies is denoted as $Dep(Q) = Dep^m(Q) \cup Dep^u(Q)$, where $Dep^m(Q)$ is the set of all MDep and $Dep^u(Q)$ is the set of all NDep in Q.

Definition 6: Let Q be an IPN, a Dependency sequence $DSeq(t_x, t_y) = [t_x, t_a][t_a, t_b] \cdots [t_c, t_d][t_d, t_y]$ is a transition sequence $\sigma = t_x t_a t_b \cdots t_c t_d t_y$ from t_x to t_y that is accepted by Q.

III. Identification of IPN models

The model Q of a DES S_f is unknown in principle; the hypothesis of Q known is held just for proving the convergence of the proposed identification technique. In the remaining of this paper will be referred to a system S_f as its representation Q in IPN terms, and will be called system model or just system.

A. Problem definition

Problem 7: Let S_f be a DES that can be modeled by an IPN Q; and $\mathcal{M} = \{Q_0, Q_1, ...\}$ be the non empty set of all IPN. Then the asymptotic identification problem is defined as follows:

1) Select a functional $f : \{Q\} \times \mathcal{M} \longrightarrow \mathcal{R}^+$ indicating the similitude between Q and $Q_j \in \mathcal{M}$. A lower value of $f(Q, Q_j)$ indicates that Q and Q_j are more similar.
2) Find out a model sequence $Q_0, Q_1, ..., Q_k$, where $Q_i \in \mathcal{M}$ such as $f(Q, Q_i) \leq f(Q, Q_{i-1})$.

Through this paper the following functional will be considered.

Definition 8: Let S_f be a DES that can be modeled by an IPN Q, $\{\varphi C_Q\}$ be the set of columns of matrix φC of Q and $\{\varphi C_{Q_i}\}$ be the set of columns of matrix φC of the identified model Q_i. Let $Dep(Q)$ and $Dep(Q_i)$ be the set of dependencies of the system Q and the identified model Q_i. The error equation of the system and the identified model is defined as:

$$f(Q, Q_i) = |\{\varphi C_Q\} - \{\varphi C_{Q_i}\}| + |Dep^u(Q) - Dep^u(Q_i)| \tag{2}$$

The error equation 2 determines the number of system transitions and non measurable dependencies that are missed in the model.

B. Asymptotic identification approach

In this work, the systems considered to be identified are those that can be described by a live, bounded and cyclic IPN, and also it is considered that the systems are event detectable (a property stating that every occurrence of an event can be detected from the output signals of the system) instead of completely measurable (in which each place has a sensor signal assigned), because those systems represent more realistic cases.

Identification process

The identification process consists in compute the incidence matrix C of an unknown system Q as it evolves. This identification process can be decomposed in two main procedures:

1) Computation of the measurable part of the model Q_i: This procedure consists in compute the φC submatrix. Every time a new transition is detected, in the C matrix are computed the measurable places related with such transition.

2) Inference of the non measurable part of the model Q_i: This procedure consists in infer the γC submatrix. Every time a t-semiflow (cycle) is detected, if it is the case, the computed model is updated adding or removing non measurable dependencies.

Since all places and transitions must be detected from the output symbols of the DES, the IPN that describes this DES must exhibit the event detectability property. The characterization of this property is presented in the following definition.

Definition 9: An IPN Q described by the state equation (1) is event-detectable iff all φC columns are not null and different from each other.

By previous definition is possible to state that any transition t_i (representing an event i of the system) can be detected from consecutive output symbols $\varphi(M_i)$ and $\varphi(M_{i-1})$ as:

$$\varphi C_Q(\bullet, t_i) = \varphi(M_i) - \varphi(M_{i-1}) \tag{3}$$

If in Q there exists the following sequence of reachable markings: $M_i \xrightarrow{t_a} M_j \cdots M_u \xrightarrow{t_b} M_x \xrightarrow{t_c} M_y$ then Q could generates the output word $w_o = \varphi(M_i)\varphi(M_j)\cdots\varphi(M_u)\varphi(M_x)\varphi(M_y)$; hence each transition of the m-word $w = t_a \cdots t_b t_c$ generating w_o is computed using equation 3 as: $t_a = \varphi(M_j) - \varphi(M_i)$, $t_b = \varphi(M_x) - \varphi(M_u)$ and $t_c = \varphi(M_y) - \varphi(M_x)$.

Notice that using equation 3 it is possible to compute all the columns of φC matrix.

Definition 10: Let Q be an IPN. If $w_o = \varphi(M_0)\cdots\varphi(M_j)$ is an output sequence generated by Q; and $\sigma = t_1 \cdots t_k$ is the firing sequence detected when w_o is measured (using equation 3), then σ is a m-word iff $\varphi(M_i) = \varphi(M_j)$.

Previous definition states how to determine that a cycle in the system has occurred, notice however that these cycles not always are t-semiflows of the system (consider the case when $M_i \neq M_j$). This fact will lead to make wrong conjectures about how the non measurable places are connected in the unknown system model Q, however as the system evolves the new information will allow to update the model. Notice that the m-words are the t-semiflows of the output matrix φC of Q.

The incidence matrix of the system depicted on figure 1 has two t-semiflows: $X_1 = t_1t_2t_3t_4t_5t_6$ and $X_2 = t_7t_8t_9$, and four m-words $w_1 = t_1t_2$, $w_2 = t_3t_4$, $w_3 = t_5t_6$ and $w_4 = t_7t_8t_9$. Notice that X_1 is the concatenation of the m-words w_1, w_2 and w_3 i.e., $X_1 = w_1w_2w_3$, while the m-word w_4 is a t-semiflow of Q, $X_2 = w_4$. Then every t-semiflow of an IPN Q it says to have an m-word decomposition.

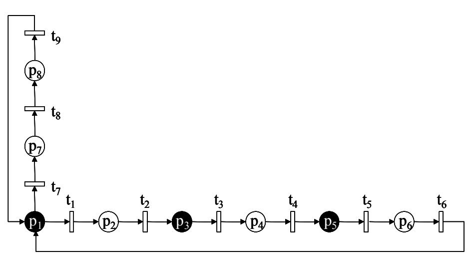

Fig. 1. Petri net with two t-semiflows and four m-words

When an m-word w_i is computed, all measurable dependencies related with its transitions are computed, hence the φC submatrix of a system model Q is computed correctly when all transitions of Q are fired. However the computation of the non measurable places (rows of matrix γC) is not as straight as the computation of the measurable places (rows of matrix φC). Some non measurable places must be inferred from several evolutions of the system. The non measurable places are computed according to: 1) preserve the firing order of the transitions in the current m-word and 2) preserve the order in which the m-words have been computed.

According to previous evolutions of the system and the new m-word computed, the non measurable places can be updated as follows:

-If there exists an NDep $[t_i, t_j] = p_k$ and it is needed to form another NDep $[t_x, t_y]$ using the same non measurable place p_k, then an arc is added from t_x to p_k and another arc is added from p_k to t_y.

-If an NDep $[t_i, t_j] = p_k$ must be removed and p_k belongs to another NDep, then the input and the output arcs of p_k related with t_i and t_j are removed; only in the case when p_k belongs to a single NDep, the place p_k can be removed.

If p_i and p_j are two non measurable places to be merged, then a new non measurable place p_k is computed such that $^\bullet p_k = ^\bullet p_i \cup ^\bullet p_j$ and $p_k^\bullet = p_i^\bullet \cup p_j^\bullet$. In the incidence matrix C_{Q_i} the rows i and j are added forming the row k, and the rows i and j are removed from C_{Q_i}.

For each NDep $[t_i, t_j] = p_k$, a vector $u_k = \begin{bmatrix} v_1 & \cdots & v_i & \cdots & v_j & \cdots & v_r \end{bmatrix}$ is computed; where r is the number of detected transitions and $v_i = 1$, $v_j = -1$ and $v_x = 0\ \forall x \neq i, j$.

To compute the incidence matrix C of Q, the columns $\varphi C_{Q_i}(\cdot, t_i)$ and u_{ij} vectors are arranged as follows:

$$C = \begin{bmatrix} \varphi C(\cdot, t_1) & \cdots & \varphi C(\cdot, t_r) \\ & u_j & \\ & \vdots & \\ & u_k & \end{bmatrix} \quad (4)$$

Asymptotic identification approach

Every model in the sequence of models defined in the asymptotic identification problem 7, is computed when the current output word observed is a new output word or if it is an already observed output word, the order in which it is observed provide new information to built a new model.

The next algorithm built a t-semiflow from the transition sequence computed from an observed output word of the system since when an m-word is computed only its measurable part is detected.

Algorithm 11: Computing the t-semiflow associated with an m-word.

Input: The m-word $w_i = t_1t_2...t_r$, which is determined when the output word $w_o = \varphi(M_0)\varphi(M_1)\cdots\varphi(M_k)$ of Q is observed. Such m-word $w_i = t_1t_2...t_r$ and columns $\varphi C(\cdot, t_1), ..., \varphi C(\cdot, t_r)$ can be computed using equation 3.

Output: The t-semiflow generating w_i.

1. If t_1 does not consume tokens from any measurable place, then an NDep $[t_r, t_1]$ will be added to the set $Dep^u(Q_1)$.- In this case, the new non measurable place belonging to $[t_r, t_1]$ must contain one token in the initial marking.

2. Let t_i and t_j any two consecutive transitions in the m-word $w_i = t_1t_2...t_r$, if there not exists a dependency $[t_i, t_j] \in Dep(Q_1)$ then, an NDep $[t_i, t_j]$ will be added to the set $Dep^u(Q_i)$.

3. Finally, if t_i occurs before t_j in the m_i-word $w_i = t_1t_2...t_r$ and $[t_i, t_j]$ could not be identified as an NDep then a MDep $[t_i, t_j]$ will be added to the set $Dep^m(Q_i)$.

This algorithm computes a non measurable place p_n needed to form the t-semiflow computed sequencing any two consecutive transitions t_i, t_j in w (forming an NDep) if they are not connected by a measurable place p_k i.e. if there not exists an MDep $[t_i, t_j] = p_k$. With this procedure it is constrained the firing of the transitions of w to the order in which they were computed. This algorithm can be used to compute the first IPN model for a system Q.

However not all the non measurable places computed as above could exist in Q since in any m-word $w_i = \cdots t_i t_j \cdots$ there could exists two consecutive transitions t_i, t_j that belong to different p-semiflows. In this case t_i and t_j are concurrent transitions, the elimination of a non measurable place $p_k = [t_i, t_j]$ is made when another m-word $w_j = \cdots t_j \cdots t_i \cdots$ is computed. Also if a non measurable place belongs to more than one t-semiflow then this place have more than one input and/or output transition and these places cannot be computed directly from the output system information. Next section deals with this problem.

IV. State machines identification

This section is devoted to the solution of the asymptotic identification problem for the class of State machines PN.

Definition 12: A state machine is a PN such as $|^\bullet t| = |t^\bullet| = 1$, i.e. each transition has only one input and one output place.

Computing a non measurable place having more than one input and /or output transitions is not straightforward as in the case of computing a non measurable place belonging to just one m-word which can be directly computed as stated algorithm 11.

In the case of state machines, an m-word cannot be verified if it is an actual t-semiflow of the system because the firing of a t-semiflow X_i does not mark or remove marks to the places belonging to another t-semiflow X_j. Hence, in addition to compute the non measurable places having more than one input and/or output transitions it is needed to compute the actual t-semiflows of the system.

In order to compute the actual t-semiflows of the system is defined the set $W = \{W_k\}$ composed by all the computed t-semiflows W_k, where each W_k is a concatenation of selected m-words. Notice then that each W_k is a t-semiflow at least in the φC matrix since every m-word is a t-semiflow of this matrix.

The first approach to compute an actual t-semiflow is sequencing the previous and current m-words. This procedure is illustrated in the following algorithm.

Algorithm 13: Computing an actual t-semiflow of the system

Input: $w_n = t_m \cdots t_n$ and the t-semiflow W_i if w_{n-1} belongs to W_i

Output: a t-semiflow W_i

If there not exists an NDep $[t_j, t_m]$ in $Dep(Q')$ and t_m has not an input measurable place, then remove from $Dep^u(Q')$ the non measurable dependencies $[t_j, t_i]$ and $[t_n, t_m]$, (where t_i and t_j are the first and the last transitions of W_i) and add to $Dep^u(Q')$ the following NDeps**:** $[t_j, t_m] = p_x$ and $[t_n, t_i] = p_y$, the marked place is then p_y. $W_i = W_i w_n$.

Proposition 14: Let Q_n be the model computed using the algorithm 13 then it is fulfilled that $f(Q, Q_n) \leq f(Q, Q_{n-1})$.

Proof: The error f is reduced each time when a column of φC or an NDep of Q is computed, and hence if the NDep computed with the above algorithm belongs to Q the error is reduced and if it is not the case the error f remains without change and in both cases the error $f(Q, Q_n)$ is equal or less than $f(Q, Q_{n-1})$. ■

Notice that if w_{n-1} and/or w_n is an actual t-semiflow of the system then the computed W_i is either a linear combination of $\overrightarrow{w_{n-1}}$ and $\overrightarrow{w_n}$ in C_Q or a linear combination in φC_Q. In fact, the computed W_i is an actual t-semiflow if w_n is the last m-word in the concatenation of a t-semiflow X_j of Q. Notice then, that the only non measurable places computed correctly are the places forming the dependencies between m-words belonging to an actual t-semiflow of the system, the other non measurable places are the places that belong to more than one t-semiflow, initially these non measurable places have only one input and one output transition, however when the m-word is computed in different order some non measurable places must be merged.

There exist also non measurable places that have more than one input and or output transitions that can be computed directly from the output. This computation is introduced in the next proposition.

Proposition 15: Let Q be an event detectable, live and bounded IPN, and let $K_a = \cdots t_i \cdots t_j \cdots$ and $K_b = \cdots t_r \cdots t_s \cdots$ be two t-semiflows of Q and let p_x and p_y be two non measurable places of Q belonging to both K_a and K_b such as $t_i, t_r \in p_x^\bullet$ and $t_j, t_s \in {}^\bullet p_y$:

case 1: Let $t_w = {}^\bullet p_x$ be the predecessor transition of t_i and t_r in K_i and K_j respectively, if p_i and p_j are the non measurable places computed such that $p_i = [t_w, t_i]$ and $p_j = [t_w, t_r]$ then p_i and p_j are the same place, i.e.$[t_w, t_i] = [t_w, t_r] = p_x$.

case 2: Let $t_v = p_y^\bullet$ be the successor transition of t_j and t_s in K_i and K_j respectively, if p_m and p_n are the non measurable places computed such that $p_i = [t_j, t_v]$ and $p_j = [t_s, t_v]$ then p_m and p_n are the same place i.e. $[t_j, t_v] = [t_s, t_v] = p_y$.

Proof: Case 1: In order to constrain the order of the computed transitions, we compute two non measurable places p_i and p_j to form the NDeps $[t_w, t_i]$ and $[t_w, t_r]$ of K_i and K_j respectively, then t_w has two output places p_i and p_j, by definition of t-semiflow $p_i^\bullet$ and $p_j^\bullet$ must belong to the same t-semiflow which is a contradiction since $t_i = p_i^\bullet$ and $t_r = p_j^\bullet$ belongs to different t-semiflows K_i and K_j respectively, hence to compute the place p_x of Q, p_i must be equal to p_j i.e. $[t_w, t_i] = [t_w, t_r]$.

Case 2: In order to constrain the order of the computed transitions, we compute two non measurable places p_m and p_n to form the sequences $[t_j, t_v]$ and $[t_s t_v]$ of K_i and K_j respectively, then t_v has two input places p_m and p_n, by definition of t-semiflow ${}^\bullet p_m$ and ${}^\bullet p_n$ must belong to the same t-semiflow which is a contradiction since $t_j = {}^\bullet p_m$ and $t_s = {}^\bullet p_n$ belongs to different t-semiflows K_i and K_j respectively, hence to compute the non measurable place p_y of Q, p_m must be equal to p_n i.e. $[t_j, t_v] = [t_s, t_v]$. ■

Notice that this proposition can be generalized for more than two t-semiflows.

Example 16: The m-words of the IPN depicted on figure 2 are $w_1 = t_1t_2t_3t_4t_7t_8$ and $w_2 = t_1t_2t_5t_6t_7t_8$. Notice that the m-words of the system are the same as its t-semiflows. In this example $p_x = p_3$, $t_w = t_2$, $t_i = t_3$ and $t_r = t_5$. The transitions t_2 and t_3 are consecutive transitions in w_1 and the transitions t_2 and t_5 are consecutive transitions in w_2. If we use the algorithm 11 to compute these non measurable dependencies then, two non measurable places p_i and p_j are computed as depicted on figure 3. Notice that in this IPN, the transitions t_3 and t_5 belongs to the same t-semiflow which is a contradiction since t_3 belongs to w_1 and t_5 belongs to w_2, hence p_i and p_j must be the same place, the union of these places results in the non measurable place p_3 of the system. The example for case 2 is similar to the case 1.

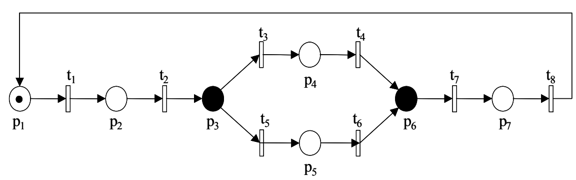

Fig. 2. Petri net with two t-semiflows.

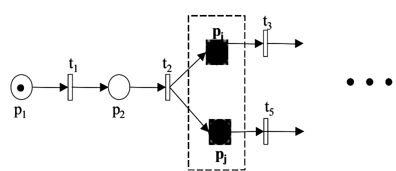

Fig. 3. Computation of non measurable places p_i and p_j using the algorithm 11 for the system depicted on figure 2.

However, in the case when p_1 is also a non measurable place, the above proposition could not be useful to compute the places p_3 and p_6. For this system we can compute four $m-$words: $w_1 = t_1t_2$, $w_2 = t_3t_4$, $w_3 = t_7t_8$ and $w_4 = t_5t_6$. Notice that t_2 and t_3, and t_2 and t_5 are not shared transitions in some m-word and hence the Proposition 15 case 1 cannot be used to compute the non measurable place p_3. Also, to compute the non measurable place p_6 using Proposition 15 case 2, the transitions t_4 and t_7, and t_6 and t_7 must be a shared transition in any m-word and it is not the case. The computation of this kind of places can be made using the next proposition.

Proposition 17: Let Q be an IPN and let $K_i = t_p \cdots t_i$, $K_j = t_j \cdots t_s$ and $K_m = t_k \cdots t_m$ be two t-semiflows of Q, if there exists transition sequences $\sigma_i = K_iK_jK_m$ and $\sigma_j = K_iK_m$ in Q (notice that σ_i and σ_j are also t-semiflows) then there exists a place $p_i = t_i^\bullet = {}^\bullet t_j = {}^\bullet t_k = t_m^\bullet$.

Proof: If σ_i can be generated by Q then there exists two dependencies $[t_i, t_k] = p_i$, and $[t_m, t_j] = p_j$ hence $t_i^\bullet = {}^\bullet t_k = p_i$ and $t_m^\bullet = {}^\bullet t_j = p_j$. As σ_j also can be generated by Q then there exists a decision place $p_i = t_i^\bullet$ such as t_j and t_k are in conflict, hence ${}^\bullet t_j = {}^\bullet t_k = p_i$, as σ_i states that there is a dependency between t_m and t_j this implies that $p_j = p_i$ since ${}^\bullet t_j = p_i$, as $p_j = [t_m, t_j]$ implies that $t_m^\bullet = p_i$ and hence $t_i^\bullet = {}^\bullet t_j = {}^\bullet t_k = t_m^\bullet = p_i$. ■

Notice that if the place p_i of the proposition above is a non measurable place then it could not be computed since no transition of t_i, t_j, t_k and t_m is shared. These non measurable places are computed when some computed non measurable places are merged. The places that are merged are those sequencing m-words. Since as mentioned before, the other non measurable places are those belonging to the same m-word and hence they are computed correctly.

Let p_i and p_j be two non measurable places to be merged, then a new non measurable place p_k is computed such that ${}^\bullet p_k = {}^\bullet p_i \cup {}^\bullet p_j$ and $p_k^\bullet = p_i^\bullet \cup p_j^\bullet$.

Next algorithm computes a model as the output words of the system are observed.

Algorithm 18: Asymptotic identification algorithm for state machine Petri nets.

---

input: The $Dep(Q')$ set
output: The updated model Q'

---

1) Compute the transition sequence $w_n = t_m \cdots t_n$ from the last observed output word generated by the system Q.
2) For any two consecutive transitions t_i and t_j in $w_n = t_m \cdots t_n$
3) if $[t_i, t_j] \notin Dep(Q_n)$
   a) If there exists $[t_i, t_k] = p_x \in Dep^u(Q')$ and $[t_k, t_j] = p_y \in Dep^u(Q')$ (Proposition 17) then
      i) merge p_x and p_y
      ii) update all the actual t-semiflows of Q' removing the t-semiflow w_j from all t-semiflow W_q in which w_j belongs i.e. $W_q = W_q/w_j$, where w_j is the m-word formed with the transitions from $p_x^\bullet$ to ${}^\bullet p_y$.
      iii) Form a new actual t-semiflow $W_{k+1} = w_j$.
   b) else if there exists an NDep $[t_i, t_k] = p_x \in Dep^u(Q')$ then add $[t_i, t_j] = p_x$ to $Dep^u(Q')$. Proposition 15 case 1.
   c) else if there exists an NDep $[t_k, t_j] = p_y \in Dep^u$ then add $[t_i, t_j] = p_y$ to $Dep^u(Q')$. Proposition 15 case 2.
   d) else add the NDep $[t_i, t_j] = p_r$ where p_r is a new non measurable place to $Dep^u(Q')$
4) End For
5) If $[t_n, t_m] \notin Dep(Q')$ then add $[t_n, t_m] = p_h$ to $Dep^u(Q')$
6) If $n = 1$, then $W_1 = w_1$ else if $W_k = W_1 = w_1$

7) If w_n has been already computed and it is the first m$_i$-word of any W_i, then mark t_j as $\mathbf{t}_f$ which implies that w_{n-1} is the last t–semiflow (m$_i$-word) of W_i which w_{n-1} belongs. $W_{k+1} = w_n$. Else if w_n has been already computed and there exists an NDep $[t_j, t_m]$ then $W_k = W_k w_n$. If this case is true then the following cases are omitted.
8) If t_m has not an input place or if w_n is not an already computed m-word
9) If $[t_j, t_m] \notin Dep(Q')$, where t_j is the last transition of W_k.
   a) If w_{n-1} belongs to another $W_j \neq W_k$ and w_n is computed by first time then: remove the NDep $[t_n, t_m]$. This implies that there exists an NDep $[t_j, t_k] = p_x$ then to form the dependency between t_j and t_m using the case 1 of proposition 15 we only need to add an arc from p_x to t_m and to form the dependence between t_n and t_y, where t_y is the first transition of W_j (the other t-semiflow which w_{n-1} belongs), we only need to add an arc from t_n to $^\bullet t_y$. $W_k = W_k w_n$.
   b) If w_n belongs to another $W_j \neq W_k$, then remove the arc from t_j to $^\bullet t_y$, where t_y is the first transition of W_j since there exists an NDep $p_n = [t_k, t_m]$ to form the dependency between t_j and t_m using the Case 2 of proposition 15 we only need to add an arc from t_j to p_n. $W_k = W_k w_n$.
   c) If t_j is marked as t_f and w_n is computed by first time, then remove the NDep $[t_n, t_m]$ and form the NDep $[t_n, t_m] = {^\bullet t_i}$, where t_i is the first transition of W_j which w$_{n-1}$ belongs. $W_{k+1} = w_n$.
   d) If there not exists a dependency $[t_j, t_m]$ but there exist a $DSeq(t_j, t_m)$ $B = w_{n-1} w_j w_n$, by proposition 17 is needed to remove the NDeps $[t_j, t_x]$ and $[t_y, t_m]$ where t_x and t_y are the first and the last transitions of w_j, add the NDeps $[t_j, t_m]$ and $[t_x, t_y]$ such as $[t_j, t_m] = [t_x, t_y] = p_q$. $W_j = W_j \backslash w_j$, $W_{k+1} = w_j$ and mark t_y as $\mathbf{t}_f$.
   e) else if form an actual t-semiflow. Algorithm 13.

else Built another t-semiflow $W_{k+1} = w_n$.

Theorem 19: Let Q be an event detectable and binary state machine and Q_{n-1} be the proposed model for Q. If w is the current m-word then a model Q_n can be built using the algorithm 18 such that $f(Q, Q_n) \leq f(Q, Q_{n-1})$.

Proof: The proof of this theorem is based on the propositions used to built the identification algorithm. Let Q_n and Q_{n-1} be the current and the previous computed models for Q, such that Q_{n-1} is updated from new information detected from Q. Notice that $f(Q, Q_i) = |\{\varphi C_Q\} - \{\varphi C_{Q_i}\}| + |Dep^u(Q) - Dep^u(Q_i)|$. Since the measurable part of any computed model Q_i is computed correctly then $|\{\varphi C_Q\} - \{\varphi C_{Q_n}\}| \leq |\{\varphi C_Q\} - \{\varphi C_{Q_{n-1}}\}|$ because in the model Q_n it is possible that new columns of $\varphi(C)$ are computed and in the worst case Q_n and Q_{n-1} have the same measurable information. Hence, we are focus on the non measurable part of Q to demonstrate that $|Dep^u(Q) - Dep^u(Q_n)| \leq |Dep^u(Q) - Dep^u(Q_{n-1})|$ and fulfill that $f(Q, Q_n) \leq f(Q, Q_{n-1})$.

The cases that could arise in order to form an NDep $[t_i, t_j]$ are illustrated in steps 3 and 9 and, if the non measurable places computed in steps 3 and 9 are places connected just with the t-semiflows computed, then these non measurable places are also non measurable places of Q hence $|Dep^u(Q) - Dep^u(Q_n)| < |Dep^u(Q) - Dep^u(Q_{n-1})|$, if is it not the case then the computed non measurable dependencies in Q_n do not belong to Q and $|Dep^u(Q) - Dep^u(Q_n)| = |Dep^u(Q) - Dep^u(Q_{n-1})|$ then it is fulfilled that $f(Q, Q_n) \leq f(Q, Q_{n-1})$. ■

V. Concluding Remarks

This paper addressed the problem of on-line identification for discrete event systems which are not instrumented completely and hence it is not possible to measure the entire system state from the output symbols. The identification approach used was a passive one. The obtained results implies that for the state machines PN class, the identified model will converge asymptotically to the model of the system as new information is observed, this is not true for other classes of PN like some classes greater than free choice PN. Current research deals with the extension of the presented results to more general classes of Petri nets, in the analysis of the input signal needed to identify a DES and in the definition of the identification problem when the input signals can be used in the identification process, relaxing the event detectability property.

Acknowledgment

We would like to thank to COECyTJAL for its support in the development of this work.

References

[1] Dana Angluin. *Inference of Reversible Languages*. Journal of the Association for Computing Machinery, Vol. 29, No. 3, July 1982, pp. 741-765
[2] E. Mark Gold. *Complexity of Automaton Identification from Given Data*. Information and Control, Vol. 37, pp. 302-320. 1978
[3] Manuel Silva. *Las Redes de Petri: en la Automática y la Informática*. Editorial AC. 1985
[4] Kunihico Hiraishi. *Construction of Safe Petri Nets by Presenting Firing Sequences*. Lectures Notes in Computer Sciences, 616, pp. 244-262. 1992.
[5] M.E.Meda & A. Ramírez. *Identification in Discrete Event Systems*. IEEE-Systems Man and Cybernetics Conference, San Diego Ca., USA, pp. 747-745. 1998.
[6] Tadao. Murata. *Synthesis of Decision-free Concurrent Systems for Prescribed Resources and Performance*. IEEE Trans. on Software Engineering, Vol. SE-6, No. 6, 525-530, 1980.
[7] M.E. Meda-Campaña, A. Ramírez-Treviño and E. López-Mellado. *Dynamical local properties for estimation and control of discrete event systems modeled by interpreted Petri*. 2000 IEEE International Conference on Systems, Man and Cybernetics. Nashville Tn, USA. October 2000.
[8] M.E. Meda-Campaña, A. Ramírez-Treviño and E. López-Mellado. *Asymptotic Identification for DES*. 2000 IEEE International Conference on Decision and Control. Sydney, Australia. December 2000.