Fuzzy approximation of discrete timed Petri net

S. HENNEQUIN [†], D. LEFEBVRE [†‡], A. EL MOUDNI [†]
[†] Université de Technologie de Belfort-Montbéliard
Laboratoire d'Automatique, Mécatronique, Productique et Systémique,
90010 Belfort Cedex,
FRANCE
[‡] IUT Belfort-Montbéliard, GEII,
11 rue Engel Gros, 90018 Belfort,
FRANCE

Abstract: - This paper concerns the modeling of discrete timed Petri nets (tPNs) by means of fuzzy logic. A fuzzy multimodeling constituted by two fuzzy linear models, is worked out for each transition of the tPN. The accuracy of the model depends on the definition of the membership functions. As a result, we show that classical sets which are a particular case of fuzzy sets permit to obtain the exact modeling of tPN.

Key-Words: - fuzzy logic, timed Petri nets, multimodeling, manufacturing systems.
CSCC'99 Proceedings:- Pages 1411-1414

1 Introduction

Many authors have discussed and illustrated the application of Petri net (PN) models [1] to the analysis of discrete event systems (DESs) [3]. Indeed, they offer a good compromise between the graphic representation and the analytic description through mathematical equations that describe the evolution of systems. In order to evaluate dynamic performance of DESs, timed Petri nets have been introduced [1]. However, they are difficult to analyse and synthesize. In this context, the fuzzy logic theory seems to be a good alternative of the study of tPNs because it permits an intuitive modeling of systems based on expert appraisal (human expertise).

While the number of applications using fuzzy logic increased, few researches have been developed for manufacturing systems and PNs. In particular, certain results have been established for fuzzy modeling of manufacturing systems by Mahmood [7] and fuzzy control design was proposed by Ghabri [4] [6] and Genest [3]. In these works, the resulting fuzzy systems based on Mamdani model [8] are nonlinear. So, they are in certain cases computational combersome to analyse. Conversely, the Sugeno model [9], based on a set of linear equations, can form simply a global approximation of a system. In this method, sets of fuzzy rules are used to imply suitable local linear state space models.

Based upon this method, the main concern of this paper is to propose an approximation of tPNs by fuzzy logic. Then, we show that when the fuzzy sets tend to classical sets, we obtain exactly a T-timed Petri net (TtPN).

The paper is organized as follows. First, we briefly describe TtPNs. Thereafter, we describe a general methodology to construct the fuzzy model applied to tPNs. Then, by means of an elementary TtPN with three places and one transition, we study the fuzzy multimodel and point out that when its fuzzy sets tend to classical sets we have the exact modeling of discrete T-timed Petri nets.

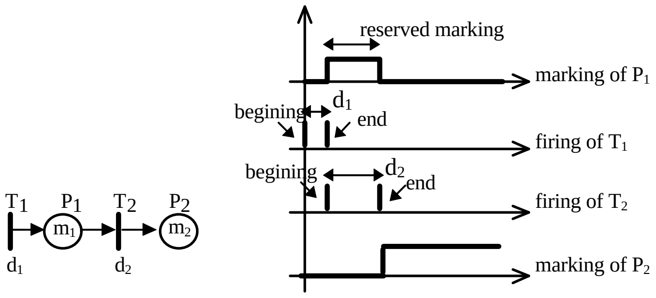
2 T-timed Petri nets

A timed Petri net [1] is generally defined as a Petri net such that constant times (delays) are associated either to places, the tPN is said to be P-timed Petri net, or to transitions, the tPN is called T-timed Petri net. It has been shown that these two models are equivalent. We can easily shift from one model to the other. In what follows, we briefly describe the T-timed Petri nets.

Let T be the set of the *m* transitions and P the set of the *n* places of a tPN. Let *M* be the marking vector given by $M = [m_1,...,m_n]^T$ where m_i which is an integer denotes the marking of the place P_i.

The T-timed Petri nets (TtPNs) represent the fact that the transitions $T_j \in$ T, $j \in \{1,\dots,m\}$, is enabled during a delay $d_j \in [0,+\infty)$. We shall consider the case where d_j is a constant value, but in general case d_j could be variable. Then, a token can have two states: it can be reserved during the firing of a transition or it can be unreserved. Only non reserved tokens are considered for enabling conditions. The decisions are not considered in this paper, i.e. the conflicts or the concurrency that may occur when a transition have several output places.

An example of TtPN constituted of two places and two transitions is given, Fig.1.a, and its evolution is showed, Fig.1.b.



a)T-timed Petri net b)Evolution of T-timed Petri

Fig.1: Example of T-timed Petri net

The fundamental equation [1] which describes the evolution of the discrete Petri nets is given by:

$$M_k = M_i + W.S, \qquad (1)$$

with S the firing sequence vector [1], W the incident matrix, M_i an initial marking which permits with S to obtain the marking M_k.

In the next section, we present all the parameters of the fuzzy multimodeling for T-timed Petri nets.

3 Fuzzy approximation of TtPNs

Let the TtPN, Fig.2, composed by two upstream places P_1 and P_2, one downstream place P_3 and one transition T_1. The fuzzy system is given next.

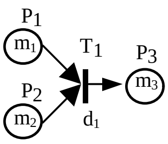


Fig.2: TtPN

We need two variables to describe the evolution of TtPN: the marking of the places which must be larger than one to fire the transitions, and the firing time which enable the firing of the transitions. We define the firing time as t_j=0 when the firing begins and t_j=d_j when the firing ends.

We describe the TtPN by a fuzzy Sugeno system [9] for which each transition has two type of inputs, the marking and the firing function, and one output the instantaneous firing vector S (1).

A fuzzy system is composed of four parts:

- Fuzzification procedure which transforms input vectors into fuzzy sets. The most commonly used is the singleton fuzzification. Then, we define two fuzzy sets for the marking m_i of the place P_i, $P_i \in P$ and $i \in \{1,...,n\}$, which are ″≥1″ and ″<1″ and two fuzzy sets for the time t_j wich are ″≥d_j″ and ″<d_j″, Fig.3. Their membership functions are given by:

$$\mu_{<1}(m_i) = \begin{cases} 1 \text{ for } m_i = 0, \\ 1\text{-}m_i \text{ for } m_i \in [0;1], \\ 0 \text{ for } m_i \in\,]1;+\infty[. \end{cases}$$

$$\mu_{\geq 1}(m_i) = \begin{cases} 0 \text{ for } m_i = 0, \\ m_i \text{ for } m_i \in [0;1], \\ 1 \text{ for } m_i \in\,]1;+\infty[. \end{cases}$$

$$\mu_{<dj}(t_j) = \begin{cases} 1 \text{ for } q(T_j) \in [0;\tau], \\ t_j/(\tau\text{-}d_j)\text{-}\ d_j//(\tau\text{-}d_j) \text{ for } t_j \in [\tau;d_j], \\ 0 \text{ for } t_j \in\,]d_j;+\infty[. \end{cases}$$

$$\mu_{\geq dj}(t_j) = \begin{cases} 0 \text{ for } t_j \in [0;\tau], \\ t_j/(d_j\text{-}\tau)\text{-}\tau/(d_j\text{-}\tau) \text{ for } t_j \in [\tau;d_j], \\ 1 \text{ for } t_j \in\,]d_j;+\infty[. \end{cases}$$

with $\tau \in [0;d_j]$.

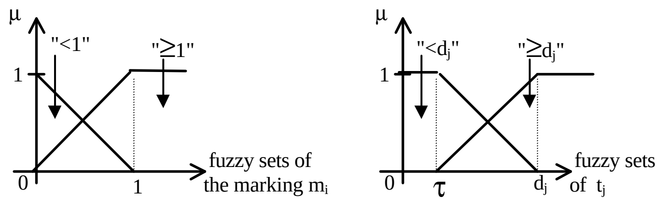


Fig.3: Fuzzy sets for the inputs

- Rulebase wich is a linguistic description of the studied system. The fuzzy rules for the transition T_j are given by:

R^k : If m_1 is ″<1″ or ... or t_j is ″<d_j ″ then S^k=0,
R^{k+1}: If m_1 is ″≥1″ or ... or t_j ″≥ d_j ″ is then S^{k+1}=C,

where m_i, i {1,..,n}, is the marking of the upstream place P_i, S^k is the instantaneous firing vector for the rule R^k, C is a vector where the unique component non equal to zero corresponds to the firing of the transition T_j.

- Inference engine which permits to interpret linguistic rules into mathematical relations. These relations are defined by the intersection between fuzzy sets which is represented by a T-norm [2], the union by a S-norm [2]. The minimum operator is chosen as the T-norm and the operator maximum is taken as the S-norm. Indeed, these operators have as a result only one input at each time and not a linear or nonlinear combination of the inputs.

- Defuuzification [2] which transforms the fuzzy outputs into output vectors. Since the conclusions of each rule of a Sugeno system are very simple, a complex defuzzification is not necessary and the output can easily be calculated as a weighted sum. Then, the result of the defuzzification is given by:

$$S_{res} = \frac{\min(\mu_{\geq 1}(m_1,...,t_j)).C + \max(\mu_{<1}(m_1,..., t_j).0}{\min(\mu_{\geq 1}(m_1,..., t_j) + \max(\mu_{<1}(m_1,..., t_j)}$$

with $\min(\mu_{\geq 1}(m_1,..., t_j)+\max(\mu_{<1}(m_1,..., t_j)=1$ which is explained by the definition of the fuzzy sets.

Then, we obtain 5 cases:

1. If $m_1=0$ or... $m_n=0$ or $t_j \in [0;\tau]$, then:

$S_{res}= \min(0,...,0,0).C+\max(1,...,1,1).0=0.$

2. If $m_1 \in [1;+\infty[$ or... $m_n \in [1;+\infty[$ or $t_j \in [d_j; +\infty[$, then:

$S_{res}= \min(1,...,1,0).C+\max(0,...,0,1).0=0.$

This case corresponds to the fact that the transition T_j is fired. This result is the same as for a T-timed Petri net.

3. If $m_1 \in [0;1]$ or... $m_n \in [0;1]$ or $t_j \in [\tau;d_j]$, then:

$$S_{res}= \min(m_1,..., m_n ,t_j/(d_j-\tau)-\tau/(d_j-\tau)).C + \max(1-m_1,..., 1-m_n ,t_j/(\tau-d_j)- d_j//(\tau-d_j)).0.$$

The transition T_j could not be fired because m_i are inferior to one. This case is impossible.

4. If $m_1 \in [0;1]$, or...or $t_j \in [d_j; +\infty[$, then:

$S_{res}= \min(m_1,...,1).C+ \max(1-m_1,..., 0).0=m_iC.$

This case is impossible because the transition T_1 is not enabled if the marking m_1 is inferior to 1.

5. If $m_1 \in [1;+\infty[$, or...or $t_j \in [\tau;d_j]$, then:

$$S_{res}= \min(1,..., t_j/(d_j-\tau)-\tau/(d_j-\tau)).C + \max(0,..., t_j/(\tau-d_j)- d_j//(\tau-d_j)).0$$

$$= t_j/(d_j-\tau)-\tau/(d_j-\tau)).C.$$

This case leads to the fact that the transition T_1 is enabled but not fired.

In the following, we study two particular cases which are $\tau = 0$ and $\tau = d_1$ in order to show that the exact model of T-timed Petri net is obtained when the fuzzy sets tend to classical sets.

3.1 Case $\tau = 0$

Then, we obtain the fuzzy sets of the time t_j given in figure 4.

We have the defuzzify value S_{res} given by (2) and we obtain if $m_1 \in [1;+\infty[$ and … and $t_j \in [0;d_j]$:

$S_{res}= \min(1,...,t_j/d_j).C+ \max(0,...,1-t_j/d_j).0= t_j/d_jC.$

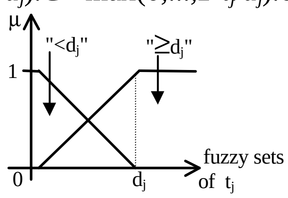


Fig.4: Fuzzy sets for t_j with $\tau = 0$

The other cases are the same as given above.

We can conclude that this fuzzy multimodel could not give exactly a T-timed Petri net (figure 6). Indeed, the marking is a linear function of the time and looks like a continuous approximation of discrete T-timed Petri nets.

3.2 Case $\tau = d_1$

In this case, the fuzzy sets "$\geq d_j$" and "$<d_j$" tend to classical sets [6], figure 5, on which we can simply apply binary logic.

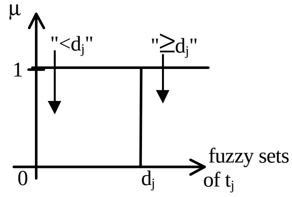


Fig.5: Fuzzy sets for t_j with $\tau = d_j$

We have also the same cases as before unless when $m_1 \in [1;+\infty[$ and … and $t_j \in [0;d_j]$. Indeed, we obtain:

- if $t_j < d_j$ then:
  $S_{res}= \min(1,...,0).C+ \max(0,...,1-t_j/d_j).0=0.$
- if $t_1 \geq d_1$ then:
  $S_{res}= \min(1,...,1).C+ \max(0,...,0).0=C.$

As we can see our fuzzy sets equal to classical sets approximate exactly a TtPN model, figure 6.

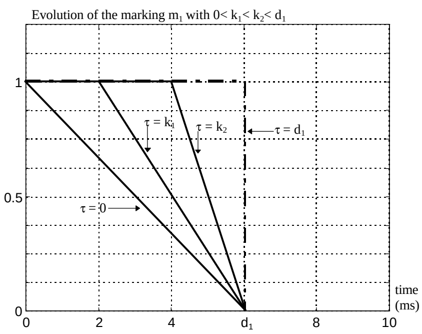


Fig.6: Results of the fuzzy approximation for TtPN

All these results can be easily generalized to a TtPN with m transitions and n places where the transitions have several input places and only one output places (TtPN without conflict or concurrency). The fuzzy approximation can also be applied to P-timed Petri nets with a different definition of the time. Indeed, the time in this cases is denoted by t_i and corresponds to the delay d_i associated with the places of the P-timed Petri net.

4 Conclusions

In this paper, we have realized a fuzzy multimodeling of T-timed Petri nets. We have shown that when the fuzzy sets tend to classical sets we obtain exactly a T-timed

Petri net. In this case, we reduce the computation times and the space memory by means of binary logic. It is interesting to note that the fuzzy multimodel has linear local models which permit a simpler controller design.
Finally, we would like to point out that the results presented here may be generalized to all timed Petri nets due to the equivalency of T-timed Petri nets and P-timed Petri nets. The only difference is the definition of the time t_j which for a P-timed Petri net is associated with the places.

References:

[1] R. David, H. Alla, "*Du Grafcet aux réseaux de Petri*", 2nd edition Hermes, Paris, 1992

[2] D. Dubois, H. Prade, "*Fuzzy sets and systems: theory and applications*", Academic Presse, 1980

[3] L. Genest, "*Outils d'aide à la décision pour le pilotage d'atelier*", PhD., 1995

[4] M.-K. Ghabri, P. Ladet, "Application of fuzzy control to a water bottling line", *proceedings IEEE on syst., man and cyber.*, 1994, pp. 812-817

[5] M.-K. Ghabri, "*Sur la modélisation et la commande des systèmes flexibles de production*", PhD., 1995

[6] A. N. Kolmogorov, S. V. Fomin, "*Introductory real analysis*", Richard Silverman editor, 1970

[7] W. Mahmood, G. Vaxhtsevanos, "Fuzzy linguistic modeling and control of manufacturing systems", *proceedings INRIA/IEEE.*, 1995

[8] E. H. Mamdani, S. Assilian, "An experiment with in linguistic synthesis with a fuzzy logic controller", *international journal of man and machine studies*, Vol.7, 1975, pp. 1-13

[9] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE transactions on syst., man and cyber.*, Vol.SMC-15, No.1, 1985, pp. ?-?