Indian Journal of Science and Technology, *Vol 8(14), 70702, July 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Survey on Fuzzy Petri Nets for Classification

S. Meher Taj[1*] and A. Kumaravel[2]

[1]Department of Mathematics, AMET University, Chennai - 603112, Tamil Nadu; taj.meher@gmail.com
[2]School of Computing, Bharath University, Chennai - 600073, Tamil Nadu; drkumaravel@gmail.com

Abstract

The aim of this study is based on Survey on fuzzy Petri nets for Classification. Petri Nets (PN) is excellent networks which have great characteristics of combining a well defined mathematical theory with a graphical representation of the dynamic behavior of systems. But, with the growth in the difficulty of modern industrial, and communication systems, PN found themselves inadequate to address the problems of vagueness, and imprecision in data. This gave rise to combination of Fuzzy logic with Petri nets and a new tool emerged with the name of Fuzzy Petri Nets (FPN). Much research has been done for FPN and a number of their applications have been expected, but their basic types and structure is still ambiguous. Result of this research, an effort is made to categorize the applications of FPN for classification according to their structure and algorithms. We identify the different types of Petri nets will improve future research on Petri nets. Hence in this study, due to these limitations we focus on establishing the FPN in the light of their classifications has been done.

Keywords: Discrete Event Systems, Fuzzy Logic, Fuzzy Petri Nets, Petri Nets

1. Introduction

Petri Net (PN) (also known as a place/transition net or P/T net) is one of several mathematical modeling languages for the description of Discrete Event Systems (DES). PNs are proved to be quite effective tool for graphical modeling, mathematical modeling, simulation, and real time control by the use of places and transitions. However, there was an intuitive need for a system, which would be able to address uncertainties and imprecision of the real world systems, because of increase in the complexity of industrial and communication systems. Fuzzy logic proved to be an appropriate complement because of its possibility nature to handle vague data.

Up till the date, numbers of ways have been proposed for combining PN with fuzzy logic, according to different applications. But with the increasing applications of these nets, there is an increase in the ambiguity about their types and structures. Almost in every new research on Fuzzy Petri Nets (FPN), researchers claim to have come up with new type of FPN. Therefore, for the ease of future researchers and engineers, it was essential to categorize FPN on the basis of some criteria. Owing to this fact, in current research FPN are classified according to their structures, and algorithms. Further, literature review of the applications of FPN has been done in the light of their classifications.

As PN can be timed and/or colored, similarly FPN can also be timed and/or colored to include the temporal effect and/or enhance their visibility. Like that of Neural Networks (NN), FPN can also do learning, and can be trained in order to get adapt to the changing situations. And as Fuzzy logic is being combined with PN to get FPN, in the same way FPN can be combine with other Artificial Intelligence (AI) tools, and mathematical models to become more efficient, and powerful.

On the basis of structures and algorithms FPNs have been classified as; Basic Fuzzy Petri Nets (BFPN), Fuzzy Timed Petri Nets (FTPN), Fuzzy Colored Petri Nets (FCPN), Adaptive Fuzzy Petri Nets (AFPN), and Composite Fuzzy Petri Nets (CFPN).

Section 1, 2, 3, 4 and 5 are allocated to each of the above mentioned type of the net, and their proposed applications. The research is being concluded in Section 6.

**Author for correspondence*

2. Basic Fuzzy Petri Nets

BFPN or simply FPN is a combination of PN and fuzzy logic. PN are graphical, and mathematical modeling tool usually used for discrete event systems. Detail description of PN and their properties can be found in[1]. The term "fuzzy logic" emerged in the development of the theory of fuzzy sets by Lotfi Zadeh in 1965. Fuzzy logic is a set of methodologies that functions effectively in an environment of imprecision and/or uncertainty. Detail reading of Fuzzy set theory, and Fuzzy logic can be found in[2]. Combination of these two gives a powerful tool to work efficiently in a real world system.

2.1 Grouping of Fuzzy Logic with Petri Nets

FPN was being proposed as 88 tuple in[3]. *FPN = (P, T, D, I, O, f, α, β)*

$P = \{p_1, p_2, \ldots, p_n\}$ was a finite set of places. $T = \{t_1, t_2, \ldots, t_m\}$ was a finite set of transitions. $D = \{d_1, d_2, \ldots, d_n\}$ was a finite set of propositions. *P«T«D = Φ*, $|P| = |D|$

I and *O* were the function of set of input and output places of transitions, where

I: $P \to T$ was the input function, a mapping from transitions to bags of places.

O: $T \to P$ was the output function, a mapping from transitions to bags of places.

f: $T \to [0,1]$ was an association function, a mapping from transitions to real values between zero and one.

α: $P \to [0,1]$ was an association function, a mapping from places to real values between zero and one.

β: $P \to D$ was an association function, a bijective mapping from places to propositions.

2.2 Modeling of Fuzzy Petri Nets

Basic steps involved in the modeling of FPN are; first extract information from experts and/or databases, and then fuzzify the information from crisp set to fuzzy set by defining membership function. Next step is to construct PN by designing rule base, and inference rules, and in the last, defuzzification of the results. Process is shown in Figure 1.

Example presented in[4] is shown in Figure 2. It represented the relationship between fuzzy logic, and PN. In that example, there was an antecedent condition that *"temperature is hot"*, and its consequent was *"humidity is low"*.

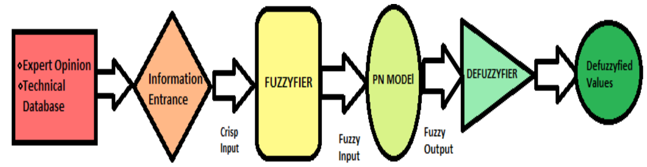


Figure 1. Methodology of modeling Fuzzy Petri Nets.

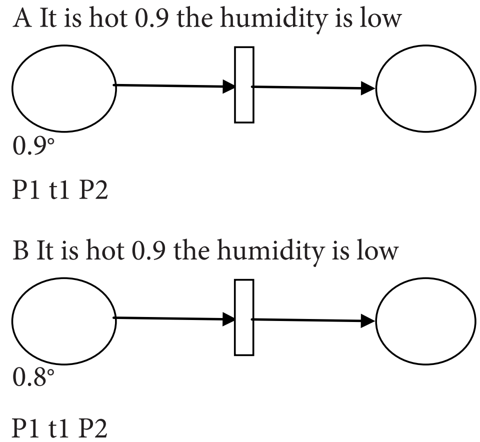


Figure 2. Firing of Fuzzy Petri Nets. **(a)** Before and **(b)** After[4].

P = {p1,p2}, T = {t1}, D = {it is hot, the humidity is low}
I*(t1) = {p1}, O(t1) = {p2}*
f(t1) = 0.9 *α(p1)* = 0.9, *α(p2)* = 0 *β(p1) = it is hot*
β(p2) = the humidity is low

I*F…THEN* rule in that example was expressed as I*F β THEN* β_2 *(CF)*, where β_1 was antecedent qualification, β_2 was consequent result, and *CF* was the "Certainty Factor", a larger *CF* value indicated a higher certainty of the rule. In the above example *"temperature is hot"* with the membership degree of 0.9 and *CF* for the truth ness of the result is 0.9, so the membership degree of the consequent that *"humidity is low"* was 0.81.

There may be many different ways of inference of rules according to the needs of the system like *MAX* 8operator or*t*8 norms or *s*8norms, and so on (Detail reading is available in[5]). Similarly there may be number of ways of combining PN and fuzzy sets. One way is may be to define fuzzy marking by denoting place token loads by means of fuzzy numbers. Another way may be to define fuzzy marking by attaching location of fuzzy set of places to each token. Still another way is to associate fuzzy variable with tokens. It is also possible to define fuzzy firing sequence which is more or less likely to be fire. Whatever the way of combining PN with fuzzy sets may be used, basic principles of PN had to be respected for the success of FPN[6].

2.3 Applications of Basic Fuzzy Petri Nets

Number of applications of FPN had been proposed in many different fields like communication, manufacturing, electronics, civil engineering, education, traffic control etc. Use of FPN included from process diagnostic, and control to knowledge based decision making. In this subsection some of the recent applications of FPN as anticipated in different research papers have been described.

Knowledge based decision making system for railway traffic control to assist dispatcher had been proposed in[7]. Possible actions for the dispatcher included; prolonged or additional stops at stations, crossing and overtaking, shifts or detours, and canceled or added trains. Knowledge based expert system was developed in three steps; knowledge acquisition, appropriate modeling of that fuzzy knowledge, and designing the rule base for inference purpose. Knowledge acquisition is the most important and basic step in any fuzzy modeling. Working methodology was; first classify conflicts, then define dispatching rules, then refine dispatching rules in lieu with conflicts, and constraints. In the last, appropriate dispatching actions were proposed. For small delays, trapezoidal distribution function was used as Fuzzy membership function. For modeling of rule base I*F THEN* rule were used; R = I*F* b *and* b_k *THEN* b_j and for combination of rules, arithmetic mean operator was used; $b_j=1/2(b_i+b_k) \times CF$.

In[3] reported abnormality dispatching rule on Taiwan railway system. They used the same approach as used in[7] and came up with promising results. There models were good initial models, but both research papers had used weighting factor of arcs as one. By fuzzification of the arc weighting, more real time results can be achieved.

Almost the same approach was used in[8] for optimum power supply of electricity in case of short circuit or any other abnormality. The study was done on distribution system of Taiwan Power Company (TPC). They used the *IF and THEN* rule base with *AND*, and exclusive *OR* operators for combination, and for the final place or output place they used the rule of winner take all in order to find the best solution. Instead of one type, they used three types of places as shown in Figure 3.

They called P_1 and P_2 as normal places, P_3 as distribution place, and P_4, P_5, P_6, P_7, P_8, P_9, P_{10} as fuzzy place. Although P_1, and P_2 were also fuzzy places but with membership degree of 1, and P_3 was simply a buffer place. They validated their model by comparing the results obtain by FPN with that of human experts.

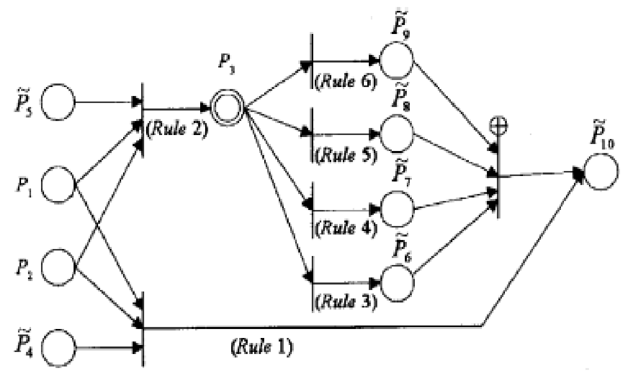
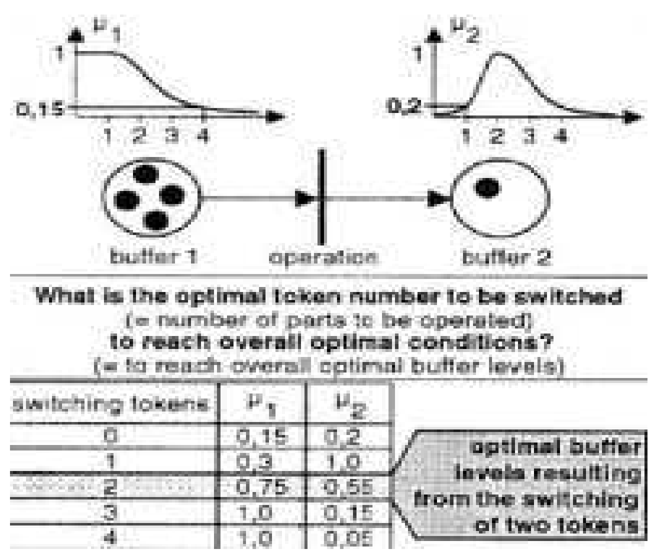


Figure 3. Knowledge representation for optimal power supply[8].

Way of dealing with contradictory requirements – like high variety, high volume, and high flexibility – in order processing, and scheduling with the help of FPN was described in[9]. Although that research paper did not explain much about the FPN model but gave a good idea of using FPN for keeping optimum WIP and for reducing scheduling complexity. Figure 4 explained that approach.

Figure 4 shows that optimal waiting parts at the buffer space of each work station can be achieved by applying different membership functions of optimal buffer space according to the requirements of that particular space. Here summation operation was used for *MAX* 8operator.

Prediction of failure in an industrial system by the use of FPN was proposed in[10]. In that research paper,



switching tokens	μ_1	μ_2
0	0,15	0,2
1	0,3	1,0
2	0,75	0,55
3	1,0	0,15
4	1,0	0,05

Figure 4. Firing of tokens depending on the fuzzy evaluated number of tokens for each place[9].

there was a case study on a paper mill and results were being compared with traditional methods and promising outcome was claimed to be found. FPN based approach for error detection and diagnosis in a complex computer[8] controlled system was used also in[11]. Use of FPN in computer integrated manufacturing engineering at local control level, supervisory control level, and between management and human/computer interface was being proposed in[6].

These were some of the applications of BFPN as proposed in scientific literature. Next section is dedicated to FTPN; their model, and their applications.

3. Fuzzy Timed Petri Nets

Time can be associated with PN in number of ways; time may be associated with transitions or with tokes or with arcs.

Usually time is associated with transitions and can be associated in two ways. By associating two numbers with transition, where one number represents the starting time of firing, and other represents the ending time of firing. Another way is by associating one number with the transition which represents the firing time of that transition. Similar is the case of FTPN for the association of time.

FTPN model where a fuzzy number was associated with transitions representing fuzzy time was proposed in[12,13]. Authors of that paper proposed rate of transfer of tokens from input place to outplace as constant during firing of transition. Also they introduced a threshold value for connecting arc, which was in fact the weight of the arcs. Their proposed model was a 58 tuple as;

$$FTPN = (P, T, Apre, Apost, w, d), \text{ where}$$

$P = \{p_1, p_2, \dots p_n\}$ was a finite nonempty set of places, $T = \{t_1, t_2, \dots t_m\}$ was a finite nonempty set of transitions, $A_{pre} = \{p \to t\}$ was a set of directed arcs which connected places with transitions, $A_{post} = \{t \to p\}$ a set of directed arcs which connected transitions to places, was a mapping to assign a weight to each arc,

$d: T \to E^1$ was a mapping to assign a firing time to each transition.

For each transition t, let $I(t)$and $O(t)$ be the input and output places, respectively. For each place p, let $I(p)$, and $O(p)$ be the input and output transitions, respectively.

Let $m_i: I \to E^i, I = 1, 2, \dots .n$ be a set of mappings. A fuzzy marking of a TFPN was a mapping

$m: I \to (m_1, m_2, \dots, m_n)$, $m(\tau) = (m_1(\tau), m_2(\tau), \dots, m_n(\tau))$, where each m_i was associated with place p_i.

A marked TFPN was 28 tuple (N, M_o) where N was a TFPN and $M_o = (m_1(0), m_2(0), \dots m_n(0))$, was its initial marking. Tokens and marks were different and marks could be split into many tokens.

A transition t_i of a TFPN was said to be fuzzy enabled for marking $m(\tau)$ at time τ if and only if P_{ij} $I(t)$, $m_{ij}(\tau) \geq w(p_{ij} \to t_i)$, $j = 1, 2, \dots k_i$

While a transition was firing, it would meet two situations.

- If one of the input places had no more tokens, then the firing would stop.
- If a new transition got enabled and the old one was continuing firing, the old one was said to be a continuing firing transition.

In either case, the net would reach a new marking, or a new state. In that state, three quantities come out:

- New marking,
- New enabled transitions,
- Continuing firing transitions.

3.1 Applications of Fuzzy Timed Petri nets

FTPN model in[12] was implemented for running voice and data over Asynchronous Transfer Mode (ATM) in order to compute the fuzzy time it took for the travel of voice and data over the system.

Witold Pedrycza, and Heloisa Camargo used FPTN in[14] to demonstrate the ageing effect on data with the passage of time. First they associated time factor with places then with transitions and then with both and analyzed the effect of each scenario on data. Data aging effect is shown in Figure 5.

Input places in their model were directly connected with the sensors, which served as a source of tokens in input places. When the transitions were fired, token in the output place increased with the same rate as tokens in the input place decreased. Authors proposed application of their model in classification, for example in classifying customers feedback acquired.

Also, FCPN had been used for real time process mentoring and control at the presence of uncertainties in[19]. Sensors served as real time input of tokens, further, *AND*, *OR*, and *NOT* operators were being used for Fuzzy inference. Experiments were done on diagnostic rules used for oil refinery with successful results.

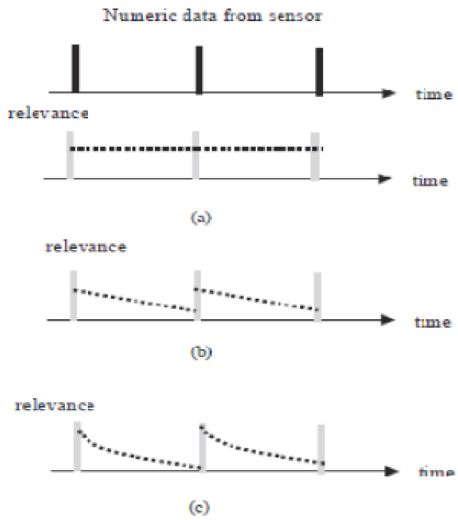


Figure 5. Aging effect. **(a)** No aging, **(b)** Linear effect of aging and **(c)** Exponential[14].

4. Fuzzy Colored Petri Nets

As the time can be associated with places or transitions of FPN, similarly color can be associated with places or arcs. Basic purpose of associating color with Petri nets is to enhance visibility and reduce complexity of the nets.

FCPN model in[16] was proposed to analyze the assembly operation with the level of difficulty. Basic structure of that model was similar to that of described in section II for BFPN with some additions. Places were divided into; PN normal places, PC control places, PT transition places and PI futile places.

$P = (P_N)»(P_C)\ (P_T)»(P\ )$, different color of each type of places. Also transitions were divided into two types; timed transitions, and selectional transitions. Selectional transitions were used on the basis of the lowest assembly sequencing rating index (ASRI). *T = (timed transition, selectional transition)*, different color for each type of transitions. Two type of arcs; inhibitor arcs and excitant arcs. *I = (inhibitor arcs, excitant arc),* each with different color.

4.1 Applications of Fuzzy Colored Petri Nets

An algorithm was developed[16] in order to first convert assembly graph into FCPN and then find the optimal assembly sequence with the help of inference. But that assembly graph was a complex architecture and complexity of the model was further increased when it was converted into FCPN model, although that was only for two subassemblies.

An automated monitor, diagnose and control system for nuclear power plants had been developed in[17] with the help of FCPN. Proposed system could monitor signals, diagnose statuses and generate control actions according to corresponding operating procedures without any human operator in case of emergency. Places were; jump place, basic place, simple place and complex place with different colors for each type of place. Also tokens were; normal execution, continuously execution and stop, each with different color. Thus movement of tokens showed the type of executing step.

Similarly, FCPN for fault diagnosis was modeled in[18]. FCPN helped to reproduce failure propagation process. Fuzzy colors were to characterize information to identify type of fault and the sequence of fault location.

5. Adaptive Fuzzy Petri Nets

As the name indicates, AFPN are those nets which have the ability to adapt according to their surrounding environment. This concept is new in PN and has come from Neural Networks (NN), where networks change their weights and learn by the passage of time and iterations.

AFPN model for dynamic knowledge representation and inference was being proposed in[20] which was a modified version of model in[4]. Weight *W* were added to arcs and threshold level *Th* to places such that;

Th: $P \rightarrow [0,\ 1]$ was the function which assigned a threshold value λi from zero to one to each place *i*. $Th = \{\lambda_1, \ldots \lambda_i, \ldots, \lambda_m\}$. For any transition *t,* if the CF associated with the tokens of all its input places were all greater than their thresholds, then the transition was enabled and fired instantly.

W: $I \rightarrow [1,\ 1]$and WO: $O \rightarrow [1,\ 1]$, were sets of input weights and output weights which assign weights to all the arcs of a net $W = W_I \rightarrow W_O$. $w_I \rightarrow W_I$ indicated how much a place (or an antecedent condition) impact a following transition or an event) connected by *wI*. A positive value meant a positive impact and a negative value meant a negative impact, For a transition *t,* assumed that $I(t) = \{p, p_2, \ldots, p_n\}$. The corresponding input weights to these places were $w_I, w_I, \ldots, w_I$. Further, supposed that $w_{I1} + w_{I2} + \ldots + w_I = 1$: Since all these conditions resulted in one consequent, the sum of their impacts was one. In the same way, $w_{Oj} \rightarrow W_O$ indicated how much a transition impacted its output places, when the transition fired.

Rule base, inference, and firing were almost similar to as already discussed in previous sections. For training of the net, net was translated into a standard neural network. $y(k)=W[(k)^T P(k+b]$.

Where k was time, input vector $P(k)=[\alpha(p_1)(k)M\ \lambda_1, \alpha(p_2)\ x\ (k)M\ \lambda_2]T$, the weight vector $W(k) = [w_1(k), w_1(k)]\ T$, bias $b = min(\lambda_1, \lambda_2)$,the output $y(k)$ was the CF of the conclusions. Widrow-Hoff learning law (Least Mean Square) was being applied as; $W(k+1) = W(k)+2\delta e(k)\ P(k)$, $e(k) = t(k)-y(k)$.

Where $t(k)$ was the goal output (teacher) and the weight vector $W(k)$ was calculated recursively. It was known that for a small enough positive constant δ. The updating law converged to real values after iterations and numbers of iterations were inversely proportion to the value of δ. Detail reading of NN can be found in[21].

An unsupervised learning model of an AFPN to be used in complex decision making situations like and automated car driving in practical traffic conditions was proposed in[25]. For learning algorithm Hebbian rule was adopted with neural decay of weights.

5.1 Applications of Adaptive Fuzzy Petri Nets

An online course generation process was proposed in[22] with the help of AFPN to facilitate professors, and students. In that model fuzzy membership function was associated with the number of times a specific topic was clicked and for how much time it was being viewed. That created an auxiliary place in the AFPN model. If a specific topic would exceed the threshold value of number of time being viewed and duration of view, it would become an auxiliary material. Thus, it would be on professor then to make that auxiliary material as permanent part of course or keep it as auxiliary by defining some constraint values.

An AFPN model with the capabilities of supervised learning for pattern classification was being developed in[23]. Model was divided into three layers as; input layer, intermediate layer, and output layer, where there were no transitions in the input layer. That model was being equated with 3 layered NN and had the advantage of clarity and explicitness of algorithm, even in the middle layer.

Training cycle of that proposed model required almost 150 iterations to substantially recognize the pattern. For pattern recognition 2D objects with different shapes were used. Criteria of recognition were; area, perimeter, maximum horizontal vertical length.

Similar kind of approach was adopted in[24] for using FPN as pattern classifier. In that model input places corresponded to the features of the patterns. Transitions built aggregates of the generic features and gave rise to their logical summarization. The output places mapped themselves onto the classes of the patterns while the marking of the places corresponded to the class of membership values. Figure 6 shows that topology of an AFPN, which is similar single layer NN.

6. Composite Fuzzy Petri Nets

This is the name given to that type of FPN which have been combined with other AI tools for different applications. Previous section has described the adaptability of FPN, this section will through light on the compatibility of FPN.

Genetic Programming (GP) and logic grammar was used to induce knowledge in[26] and was represented in the form of FPN, because of the expressiveness of FPN in representation of imprecise and vague data. In order to evolve the architecture of FPN from GP, cellular encoding was done. Cellular encoding was the process used for evolving the structure of NN, by dividing FPN into input layer, intermediate layer, and output layer. There were two types of functions used for manipulating; place manipulating function and transition manipulating function.

6.1 Applications of Composite Fuzzy Petri Nets

There may be number of ways of combining FPN with AI tools, to cover all of them is beyond the scope of this paper. In this subsection, focus would only be on the applications of CFPN.

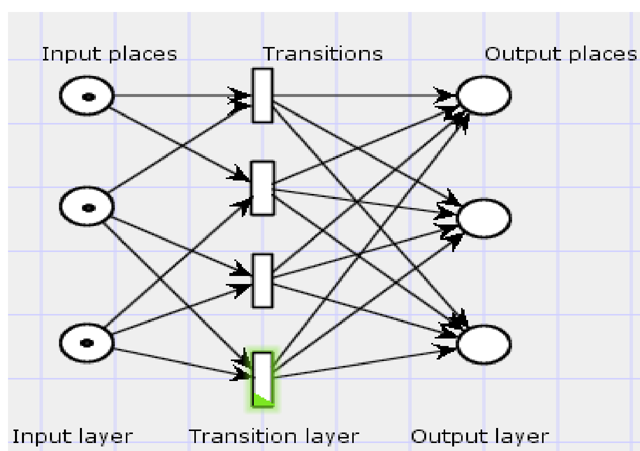


Figure 6. Single layer topology of AFPN [24].

A model was developed in[27] with the combination of NN and FPN for real time parts classification and sorting in an integrated manufacturing system. Methodology of the system is shown in Figure 7, where the Y channel added the fuzzy logical decision on the PN firing, which simulated the decision taken, to activate the resources of the real process. The NN part was applied to estimate the condition of the resources using a back[8] propagation scheme. Since the condition of the whole process was known, the availability of the resources and the firing decision of the tasks were being controlled.

In[28], a model was proposed for traffic control, where combination of FPN and NN were used. Authors designed many rules bases for traffic signals to be open or close according to real traffic conditions of a particular area and found that FPN alone was incapable of handling some conflicting and from surveys and questionnaires the results *i* of a questionnaire could refer to a different time period in comparison to some other data being available in a data base. Also they proposed the application in a dynamic system like industrial system where readings of sensors can be available at different sampling frequencies, for example temperature could be less frequent then readings of velocity.

TFPN were also used in real time distributed systems for the analysis of fault and timing[15]. G-nets were incorporated with fuzzy logic in that research. Where, G-nets were a Petri-net based framework for the modular design and specification of distributed information systems. The framework was an integration of PN theory with the object oriented software engineering approach for system design.

Those were the few applications of FTPN, next section will provide a brief overview of FCPN.

A similar kind of approach with least mean square error for the supervised learning was also used in[14] rule base. For example maximum time of closing of a particular

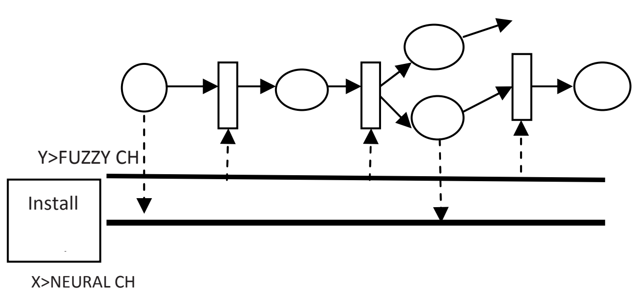


Figure 7. Real time neural networks and Fuzzy Petri nets extension[27].

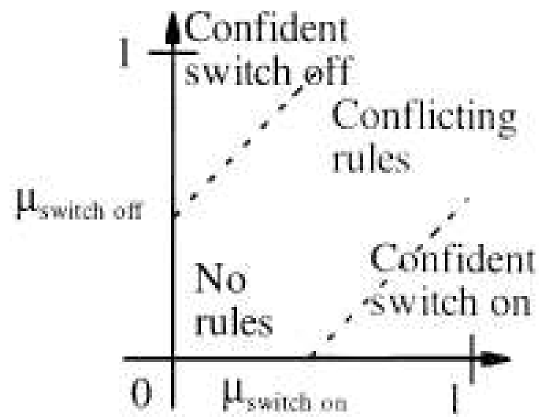


Figure 8. Confidence level of fuzzy control[28].

signal had been exceeded and at the same time there was still maximum load on the open signal, controversial which demanded the open of both the signals. Figure 8 shows a conflicting zone between confident switch on and confident switch off. In this zone both commands were applicable with CF between 0 and 1.

With the use of NN along with FPN, they were able to remove that conflicting rules zone from the graph and there were then only two zones, either confident switch on or confident switch off.

7. Conclusion

Models described in each type of FPN are shown for the purpose to have an idea of how a basic model of any type, e.g., TFPN or CFPN would look like, what should be their perimeters, what would be their rule base, and inference, how transitions get enabled and how these transitions may fire, because all these conditions may vary according to their usage.

Purpose of this research was to categorize FPN on the basis of their structure and algorithms and to through light on the applications of each type of those nets. It can be seen that FPN is a power full and effective tool not only to model and simulate real world conditions and scenarios but also to work as an expert in real time systems. Applications of FPN can be found in almost all fields of engineering and applied sciences and they can perform many different tasks; diagnosis, control, classification, scheduling, and even decision making like humans rather much quicker than humans.

Much research and experimentation have been done on FPN and still lot more would be done. Horizons are very broad for research in the field of FPN. Many industrial systems have been modeled by PN but not many have been modeled with FPN. Simple PN uses crisp set and cannot deals with uncertainty and vagueness, which are the basic parts of any real world system. From industrial engineering point of view, always there are demand fluctuations and machine breakdown. Both fluctuation and breakdown can be better modeled by FPN to simulate real world processes. In this regard, future research would focus on fuzzification of the PN model of dual Kanban pull system proposed in[29].

8. References

1. Richard Z, Meng Chu Z. Petri nets and industrial applications: A tutorial. IEEE Transactions on Industrial Electronics. 1994 Dec; 41(6):567–83.
2. Klir GJ, St. Clair U, Yuan B. Fuzzy set theory. Foundations and applications. London: Prentice Hall; 1997.
3. Cheng Y-H, Yang L-A. A Fuzzy Petri Nets approach for railway traffic control in case of abnormality: Evidence from Taiwan railway system. Expert Systems with Applications. 2009; 36(4):8040–8.
4. Chen SM, Ke JS, Chang JF. Knowledge representation using fuzzy Petri nets. IEEE Transactions on Knowledge and Data Engineering. 1990 Sep; 2(3):311–9.
5. Jang JSR, Sun CT, Mizutani E. Neuro Fuzzy and soft computing - A computational approach to learning and machine intelligence. London: Prentice Hall; 1997.
6. Cardoso J, Valette R, Dubois D. Fuzzy Petri net: An overview. Proceedings of 13th IFAC World Congress; 1996. p. 443–8.
7. Alexander F. A fuzzy knowledge-based system for railway traffic control. Eng Appl Artif Intell. 2000 Dec; 13(6):719–29.
8. Yang H-T, Huang C-M. Distribution system service restoration using fuzzy Petri net models. Electric Power and Energy System. 2002 Jun; 24(5):3395–403.
9. Walter E, Thomas H. Managing multiple product variants in assembly control with a Fuzzy Petri net approach. Annals of the CIRP. 1996; 45(1):45.
10. Rajiv KS, Dinesh K, Pradeep K. Predicting uncertain behavior of industrial system using FM - A practical case. Applied Soft Computing. 2008 Jan; 8(1):96–109.
11. Ting Y, Lu WB, Chen CH, Wang GK. A fuzzy reasoning design for fault detection and diagnosis of a computer-controlled system. Eng Appl Artif Intell. 2008 Mar; 21(2):157–70.
12. Zouhua D, Horst B, Mti S, Abraham K. Fuzzy timed Petri net definitions, properties, and applications. Mathematical and Computer Modelling. 2005 Feb; 41(283):345–60.
13. Zouhua D, Horst B, Oscar K, Mti S, Abraham K. Fuzzy timed Petri nets - analysis and implementation. Mathematical and Computer Modelling. 2006 Feb; 43(384):385–400.
14. Witold P, Heloisa C. Fuzzy timed Petri nets. Fuzzy Sets and Systems. 2003 Dec; 140(2):301–30.
15. de Figueiredo JCA, Angelo P. Faults and timing analysis in real time distributed systems: A fuzzy time Petri net based approach. Fuzzy Sets and Systems. 1996 Oct; 83(2):143–68.
16. Ben-Arieha D, Rajeev RK, Tiwari MK. Analysis of assembly operations' difficulty using enhanced expert high-level colored fuzzy Petri net model. Robotics and Computer Integrated Manufacturing. 2004 Oct; 20(5):385–403.
17. Seung JL, Poong HS. Development of automated operating procedure system using Fuzzy colored Petri nets for nuclear power plants. Annals of Nuclear Energy. 2004 May; 31(8):849–69.
18. Ouali MS, Ait-Kadi D, Rezg N. Fault diagnosis model based on Petri net with fuzzy colors. Computers and Industrial Engineering. 1999 Oct; 37(182):173–6.
19. Grantham KHP, Raymond T, Stephen SW. A Process-control and diagnostic tool based on continuous Fuzzy Petri nets. Eng Appl Artif Intell. Dec1995; 8(6):643–50.
20. Li X, Lara-Rosano F. Adaptive fuzzy Petri nets for dynamic knowledge representation and inference. Expert Systems with Applications. 2000 Oct; 19(3):235–41.
21. Martin TH, Howard BD, Mark B. Neural network design. Boston: PWS publishing Co; 1996.
22. Huang Y-M, Chen J-N, Huang T-C, Jeng Y-L, Kuo Y-H. Standardized course generation process using dynamic Fuzzy Petri nets. Expert Systems with Applications. 2008 Jan; 34(1):72–86.
23. Amit K, Uday KC. Reasoning unsupervised learning in a Fuzzy cognitive map. Information Sciences. 2005 Feb; 170(284):419–41.
24. Witold P. Generalized Fuzzy Petri nets as pattern classifiers. Pattern Recognition Letters. 1999 Dec; 20(14):1489–98.
25. Amit K, Uday KC, Paul PW. Supervised learning on a Fuzzy Petri net. Information Sciences. 2005 Jun; 172(3–4):397–416.
26. Man LW. A flexible knowledge discovery system using genetic programming and logic grammars. Decision Support Systems. 2001 Oct; 31(4):405–28.
27. Gracios CA, Marin EVS, Sanchez AD. Describing an IMS by a FNRTPN definition: A VHDL approach. Robotics and Computer Integrated Manufacturing. 2005 Jun; 21(3):241–7.
28. Henry JJ, Farges JL, Gallego JL. Neuro Fuzzy techniques for traffic control. Control Engineering Practice. 1998 Jun; 6(6):755–61.
29. Erik LJB. A new generic timed Petri net model for design and performance analysis of a dual Kanban FMS. Int J Prod Res. 2004; 42(4):714–40.