Research Journal of Applied Sciences, Engineering and Technology 5(20): 4843-4849, 2013
DOI:10.19026/rjaset.5.4330
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 22, 2012 Accepted: November 03, 2012 Published: May 15, 2013

**Research Article**

# Modeling and Simulating of Uncertain Quality Abnormity Diagnosis

Shiwang Hou and Mengqun Li
Department of Mechanical Engineering and Automation North University of China,
Taiyuan 030051, China

**Abstract:** There is much fuzzy uncertain information during the diagnosis of quality abnormity. The effective utilization model of that can provide important decision-making support. In this study, we consider three main types of fuzzy production rules, which can be used in fuzzy quality abnormity diagnosis problem and their presentation models are constructed by use of Fuzzy Reasoning Petri Nets (FRPNs). Considering of the graphic representation and logic structure of FRPNs, we propose the method for simulating model using Matlab toolbox state flow. By establishing a corresponding relationship between FRPNs rules and state flow block diagram, three simulating models for the three corresponding FRPNs' basic structure are developed. Finally, we give an application case of the proposed model. Taking place truth degree data of FRPNs as input, the diagnosis process and results can be shown dynamically in the state flow simulating model under Matlab environment. The result illustrated that the method proposed can give reliable information for process maintenance and abnormal causes' location.

**Keywords:** Modeling, quality abnormity diagnosis, simulation modeling

## INTRODUCTION

Constantly steady product quality is always the goal that manufacture pursues. In order to improve the product quality and ensure the process satisfy the given performance index, the quality abnormity or fault during manufacturing process must be detected, diagnosed and eliminated. From the view of quality control, modern manufacture is a complex system composed by man, machine, material, method, measure and environment, the produce and spread of quality abnormity is a typical dynamic process. On the one hand, a abnormal state may incur a series of successive states. This is the concurrence of quality abnormity. On the other hand, an abnormal state may be incurred by many causes, i.e. many original abnormal causes spread along different ways may incur the same system abnormity. There is great uncertainty and fuzziness during the process of abnormity detect and diagnosis.

In order to deal with vague or fuzzy information, Fuzzy Petri Nets (FPN) have been introduced and focused on by many researchers. As a model of knowledge-based systems, FPN are used for fuzzy knowledge representation and reasoning. Looney (1988) reviewed reasoning by means of transformations of the truth states by rule matrices and adopted Fuzzy Petri Nets through the application of Boolean Matrices to simulate actual situations. Fuzzy logic networks are used to modify Petri Nets, allowing rule-based decision systems to be represented and executed. The fuzzy simulation theory adopted "MIN" operator to handle "AND" problems and "Max" operator to handle "OR" problems. Chen *et al*. (1990) adopted a Fuzzy Petri Net Model to represent the fuzzy production rules of a rule-based system. This model could represent expert knowledge and support fuzzy reasoning. Mirko and Makajic-Nikolic (2004) and Daniel *et al*. (2003) proposed that Fuzzy Petri net based on the fuzzy production rules can express and deal with ambiguity knowledge well and reason according to the confidence of transition and find the causes of the failure. Also, it has been used in modeling and analyzing fault diagnosis systems for gas turbines, CNC machine tools, transformers, communication safety and manufacturing systems (Xiao-Guang *et al*., 2000; Jing *et al*., 2004; Jian-Yuan *et al*., 2003; Li-Xin *et al*., 2003; Riascos *et al*., 2004; Rangel *et al*., 2005; Han and Daoquan, 2002).

Fuzzy Petri net is a good tool for discrete event dynamic systems modeling and analyzing. It can describe fuzzy knowledge and carry out concurrent logic reasoning. The spread of abnormity is that a great disturbance or abnormal cause induces the other abnormal state. In the abnormal state spread model, transition represents the necessity that the result must be true if the precondition is true. The abnormity diagnosis is the process to locate the assignable causes based on the detected abnormal phenomena. It is a backward reasoning process along the opposite direction as the abnormity spread. So causal analysis based on the Petri

**Corresponding Author:** Shiwang Hou, Department of Mechanical Engineering and Automation North University of China, Taiyuan 030051, China

net model of abnormal state spread can provide important information for abnormity diagnosis.

Firstly, this study proposed a fuzzy reasoning Petri nets model that can denote the produce and spread process of quality abnormity. Then simulating method of the model by use of the state flow under Matlab environment was developed.

## FUZZY PRODUCTION RULE AND ITS PRESENTATION WITH FRPNS

**Type of fuzzy production rule:** Fuzzy production rule is a method to describe the uncertain and inaccuracy knowledge. It contains four types as follows:

- $R_i(c_i)$: IF $p_1(\theta_1)$ and $p_2(\theta_2)$ and … and $p_{k-1}(\theta_{k-1})$ THEN $p_k(\theta_k)$ $\theta_k = \min\{\theta_1, \theta_2, \cdots, \theta_{k-1}\} * c_i$
- $R_i(c_i)$: IF $p_1(\theta_1)$ THEN $p_2(\theta_2)$ and … and $p_k(\theta_k)$ , $\theta_j = \theta_1 * c_i, j = 2, 3, \cdots, k$
- $R_i(c_i)$: IF $p_1(\theta_1)$ or $p_2(\theta_2)$ or … or $p_{k-1}(\theta_{k-1})$ THEN $p_k(\theta_k)$ , $\theta_k = \max\{\theta_1, \theta_2, \cdots, \theta_{k-1}\} * c_i$
- $R_i(c_i)$: IF $p_1(\theta_1)$ THEN $p_2(\theta_2)$ or … or $p_k(\theta_k)$ , $\theta_j = \theta_1 * c_i, j = 2, 3, \cdots, k$

where $\theta_j$ denotes the truth degree of precondition place or result place in fuzzy rule $R_i$ and $c_i$ denotes the confidence level of fuzzy rule $R_i$.

The fourth type rule is avoided in the rule base since it is hard to obtain certain implication. The discussion following is mainly for the first three types.

**Presentations of rules:** Fuzzy Reasoning Petri Nets represent the syntactic structure of knowledge system based on rules. It uses place to denote proposition and each place contain a token taking value in (0, 1) to signify the truth degree of proposition. Transition firing is used to represent the rule reasoning process. The confidence level of proposition is determined by the corresponding transition. The arc between place and transition denotes the cause-effect relation between.

**Proposition and rules:** The first three types rules' presentation by FRPNs is shown as Fig. 1to 3.

**Process of negative propositions:** The supplementary arc is introduced considering the negative propositions contained possibly in the rules. It is denoted by adding small circle in the end of traditional arc. For example:

R (0.8): IF $p_1$(0.8) and $p_2$(0.9) and (not $p_3(0.7)$ ) THEN (not $p_4(\theta_4)$

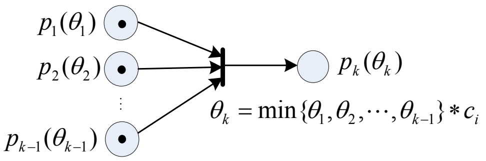

Fig. 1: Presentation of the type I fuzzy reasoning rule

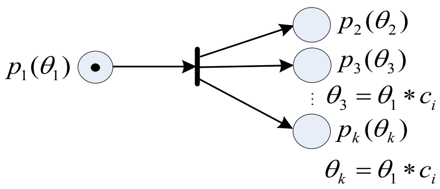

Fig. 2: Presentation of the type fuzzy reasoning rule

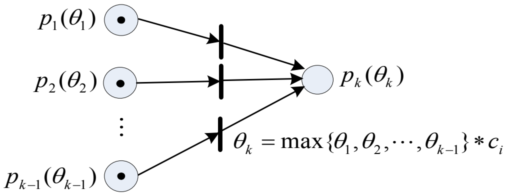

Fig. 3: Presentation of the type Ⅲ fuzzy reasoning rule

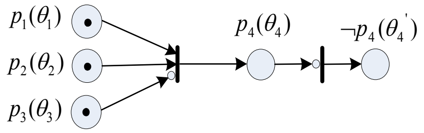

Fig. 4: FRPNs presentation of the negative propositions

Its presentation with FRPNs is shown as Fig. 4. The negative propositions of propositions $p(\theta)$ is denoted as p(θ), where θ = 1-θ.

## DEFINITION OF FRPNS MODEL

Fuzzy reasoning Petri nets is defined as a 8-tuple: (P, R, I, O, H, θ, γ, C).
where P = {$p_1$, $p_2$,…, $p_n$} is a finite set of places, R {$r_1$, $r_2$,…, $r_n$} is a finite set of transition rules; I: P×R → {, 1} is input function representing the mapping from precondition places to transition rules; O: R× P→ {0,1] is output function representing the mapping from transition rules to result places; H: R× P→ {0, 1} is a matrix of supplementary arc reflecting the link situation of supplementary arc between proposition places and transition rules; $\theta_i$ ∈ [0, 1], i = 1, 2,…, n denotes the truth degree of place $p_i$ and its original state is denoted by $\theta^0$; γ = ($\gamma_1$, $\gamma_2$,…, $\gamma_n$); is vector of marking, where $\gamma_i$ ∈ {0, 1}, 1, 2,…, n denotes the presence or absence of

token in place $p_i$ and its original state is denoted by $\gamma^0$. C = diag ($c_1$, $c_2$,…, $c_m$) is a diagonal matrix of confidence level and $c_i \in \{0, 1\}$, I = 1, 2,…, m denotes the confidence degree of rule.

## RUNNING RULE FOR FRPNS

Two operators as following are adopted as FRPNs carrying out the reasoning rules:

- $\oplus : A \oplus B = D$ , $A = (a_{ij})_{m\times n}$ , $B = (b_{ij})_{m\times n}$
  $D = (d_{ij})_{m\times n}$ , satisfying $d_{ij} = \max\{a_{ij}, b_{ij}\}$
- $\otimes$ ： $A \otimes B = D$ ， $A = (a_{ik})_{m\times p}$ ， $B = (b_{kj})_{p\times n}$
  $D = (d_{ij})_{m\times n}$ , satisfying $d_{ij} = \max\limits_{1\le k\le p}(a_{ik} \cdot b_{kj})$

The running process of FRPNs includes rule enabling and firing:

- A necessary and sufficient condition for rule enabling is that all the precondition places are marked;
- Under the marking state $\gamma$ , a new marking state $\gamma^{'}$ is produced after the rule $r_j$ was fired, satisfying $\gamma^{'}(p) = \gamma(p) \oplus O(p, r_j), \forall p \in P$ . The truth degree vector transform from $\theta$ to $\theta^{'}$, satisfying $\theta^{'}(p) = \theta(p) \oplus c_j \cdot \rho_j \cdot O(p, r_j), \forall p \in P$ , where, $\rho = (\rho_1, \rho_2, \cdots, \rho_m)^T$ is control vector and satisfy $\rho_j = \min\{x_i\}$ , $x_i = \begin{cases} \theta_i, & \text{when } I(p_i, r_i) = 1 \\ 1-\theta_i, & \text{when } H(p_i, r_i) = 1 \end{cases}$.

In the FRPNs model, all rules can be fired simultaneously. Firing vector $\mu = (\mu_1, \mu_2, \cdots, \mu_m)$ is used to denote the firing state of rules, $\mu_j$ takes value 1 or 0 determined by whether or not rule $r_j$ is fired.

- When a group of fuzzy rules are fired, marking vector and truth-degree vector change with the following equations:

$$\gamma^{'} = \gamma \oplus [O \otimes \mu] \quad (1)$$

$$\theta^{'} = \theta \oplus [O \cdot C] \otimes \rho \quad (2)$$

## SIMULATION MODEL CONSTRUCTION WITH STATEFLOW

In order to validate the running process and reasoning result of the FRPNs model constructed as above, appropriate simulating tools must be adopted. For its particular graphic representation and logic structure, traditional Petri net software, such as PNK (Wu, 1997), CPN, VPNT (Dennis and Wassim, 1989) and Visual Object ++, is hard to simulate and analyze its performance.

**Stateflow overview:** Simulink/State flow is a tool package of software Matlab, which is a graphic simulating tool for the finite state machine and can be used to design, analyze event-driven system. For complex monitoring problem, user can utilize graphic tools to realize the states transition. The monitoring logic produced by state flow can be embedded in simulink model and the two can be connected seamlessly.

State flow is a tool to design and simulate complex response system and event driven system. It combines theory of infinite state machine, flow chart and state transition in special manners. It adopts object-oriented programming idea, i.e. property, event and method, to describe the physical model in graphic manner. The characteristic of state flow as follows:

- **State and transition compose the basic framework of state chart:** Under the environment of state flow toolbox, state is represented with block diagram and each state includes four actions as follows, entry, during, exit and on event name. They denote the action when entering the state, during the state, exiting the state and event happening with specific name, respectively. Transition is denoted by array line, showing the ways of transition. Its label format is: event [condition] {condition action} /transition action. Event specifies an event that causes the transition to be taken, provided the condition, if specified, is true. Specifying an event is optional. The absence of an event indicates that the transition is taken upon the occurrence of any event. Condition specifies a boolean expression that, when true, validates a transition to be taken for the specified event trigger. Enclose the condition in square brackets ([]). Condition action follows the condition for a transition and is enclosed in curly braces ({}). It is executed as soon as the condition is evaluated as true and before the transition destination has been determined to be valid. If no condition is specified, an implied condition evaluates to true and the condition action is executed. Transition action executes after the transition destination has been determined to be valid provided the condition, if specified, is true. If the transition consists of multiple segments, the transition action is only executed when the entire transition path to the final destination is determined to be valid. Precede the transition action with a backslash.

- Object hierarchy, concurrent states and historic transition record are supported under state flow. So complex system can be stratified, simulated simultaneously and transition target state can be selected depending on historic record.
- Temporal logic can be simulated by use of time operator as follows: before, after, at and every.
- Monitoring logic can be constructed by state flow block diagram. Combined with the function of conditional execution system of Simulink, the subsystem can be activated or inhibited selectively and the dynamic simulation domain of Simulink is expanded.

Based on above-mentioned characteristic, State flow is suitable for simulating fuzzy reasoning Petri nets model based on event-driven. FRPNs model of fuzzy quality abnormity diagnosis is discrete event system and its simulating programmer can be developed by use of State flow (Michaela *et al*., 2005; Davidrajuh, 2008; Ji-Ping *et al*., 2006; Jinsong *et al*., 2007). Under state flow environment, the correctness of state flow block diagram can be checked by setting breakpoint use debugger tool and its execution process can be displayed dynamically.

**Stateflow overview:** Based on finite-state machine theory, state flow model can transform from one state into the other and execute a series of actions if the firing condition is true. The state flow block diagram is graphical representation of finite-state machine. In state flow block diagram, state, transition, state action and marking label of transition are used to represent the place, directed arc and transition rule of Petri net respectively. According to the transition relation and monitoring logic, state flow simulating model of FRPNs can be constructed by use of above elements.

Corresponding with the basic structure of FRPNs aforementioned, its construction rules of state flow model are as follows:

**Rule I:** As for type I FRPNs structure, there is only one transition after output arc of many places and only one place after output arc of transition. The input places of transition are denoted by a group of concurrent super states, $P_1$~ Pk-1 and each super state contain two mutually exclusive states: Normal and Abnormal. The transition between these two states is determined by the truth degree $a_j$ of place. State entry action En can be executed when transition happens between the two states and the state attribute is assigned a value with the truth degree $a_j$ of place, i.e., en: pk-CL = ak, event E is also triggered at the same time. The output places of transition are denoted by concurrent super states, $P_k$ and it contains two mutually exclusive states: None and on, which represent whether or not there is assignable cause respectively. The event E is the trigger of the transition of the two states. The result of logic expression of all input states is the condition of the transition. The value of rule confidence is assigned and the truth degree of all input places is minimized in "Transition action". As shown in Fig. 5.

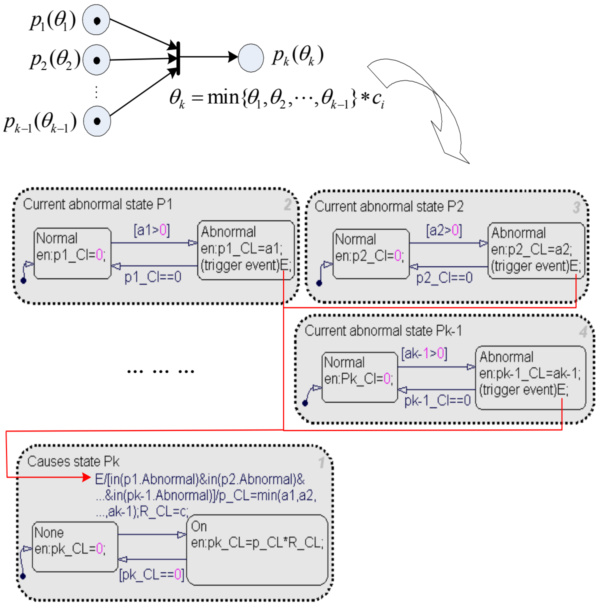

Fig. 5: State flow simulating model of type fuzzy reasoning rule

**Rule II:** As for type II FRPNs structure, there are many places after output arc of transition and only one place before input arc of transition. The input place of transition is denoted by a concurrent super state, $P_1$ and it contains two mutually exclusive states: None and On. The transition between these two states is determined by the truth degree $a_1$ of place. State entry action En can be executed when transition happens between the two states and the state attribute is assigned a value with the truth degree $a_1$ of place, i.e., en：p1-CL = a1, event E is also triggered at the same time. The output places of transition is denoted by a group of concurrent super states, $P_2$~$P_k$ and each contain two mutually exclusive states: None and On. The event E is the trigger of the transition of the two states. The result of logic expression of input state is the condition of the transition. The value of rule confidence is assigned in "Transition action". As shown in Fig. 6.

**Rule III:** As for type III FRPNs structure, there is only one place after output arcs of many transitions and only one place before input arc of each transition. The input places of transition are denoted by a group of concurrent super states, P1~ Pk-1 and each super state contains two mutually exclusive states: Normal and Abnormal. The transition between these two states is determined by the truth degree $a_j$ of place. State entry action En can be executed when transition happens

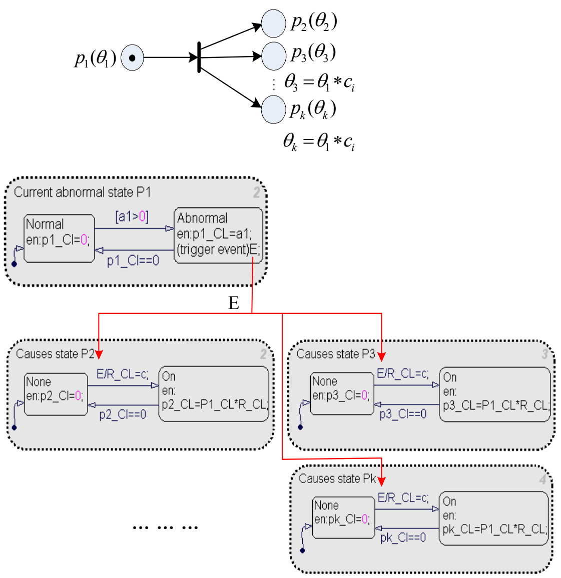

Fig. 6: State flow simulating model of type □ fuzzy reasoning rule

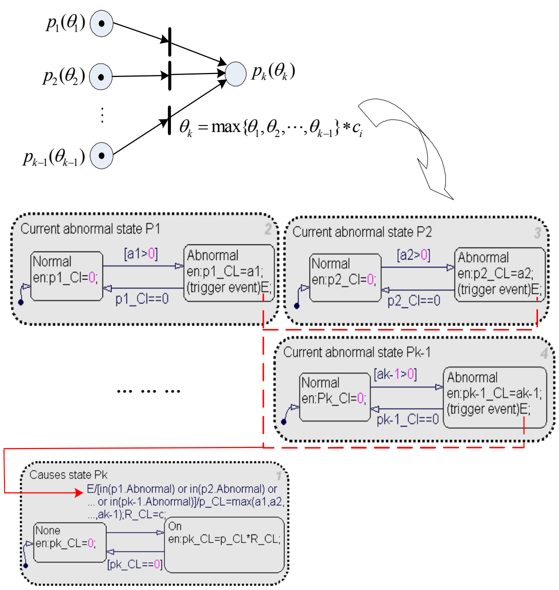

Fig. 7: State flow simulating model of type fuzzy reasoning rule

between the two states and the state attribute is assigned a value with the truth degree aj of place, i.e., en: pk-CL = ak, event E is also triggered at the same time. The output places of transition are denoted by a concurrent superstate, Pk and it contains two mutually exclusive states: None and On. The event E is the trigger of the transition of the two states. The result of logic expression of all input states is the condition of the transition. The value of rule confidence is assigned and the truth degree of all input places is maximized in "Transition action". As shown in Fig. 7.

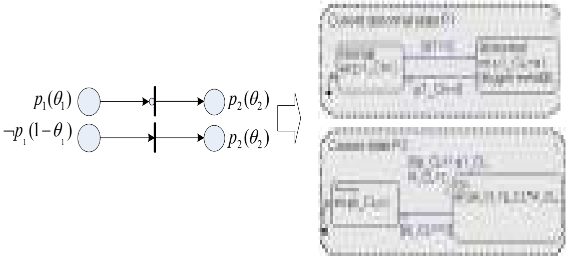

Fig. 8: State flow simulating model of negative proposition FRPNs

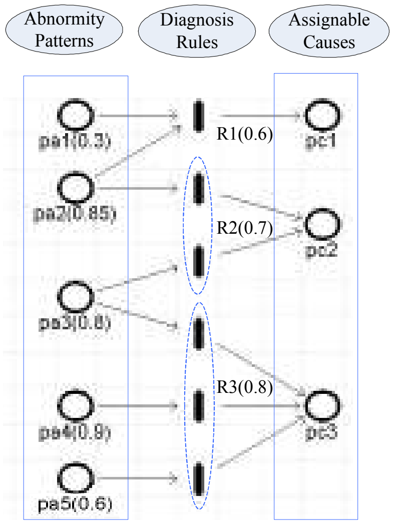

Fig. 9: Diagram of diagnosis FRPNs case

**Rule IV:** As for the negative proposition, it can be transformed into positive proposition by complement operation of place truth degree. Then FRPNs model contained supplementary arc can be transformed into the other three types. As shown in Fig. 8.

## APPLICATION STUDY

Consider of a simple diagnosis case with FRPNs including some rules as follows:

R1 = (0.5): If pa1 (0.3) and pa2 (0.85) then pc1
R2 = (0.7): If pa2 (0.85) or pa3 (0.8) then pc2
R3 = (0.6): If pa3 (0.8) or pa4 (0.9) or pa5 (0.6) then pc

The diagram of this case is shown as Fig. 9.

According to the method mentioned in section V, the case can be simulated by state flow model as Fig. 10 under Matlab environment. The diagnosis process can be shown dynamically and give the result at AC state finally. After simulating, the result is: f-pc3 = 0.54, f-pc2 = 0.59 and f-pc3 = 0.15. It shows that because pc3 has the highest contribution degree to current abnormity and should be checked prior to others.

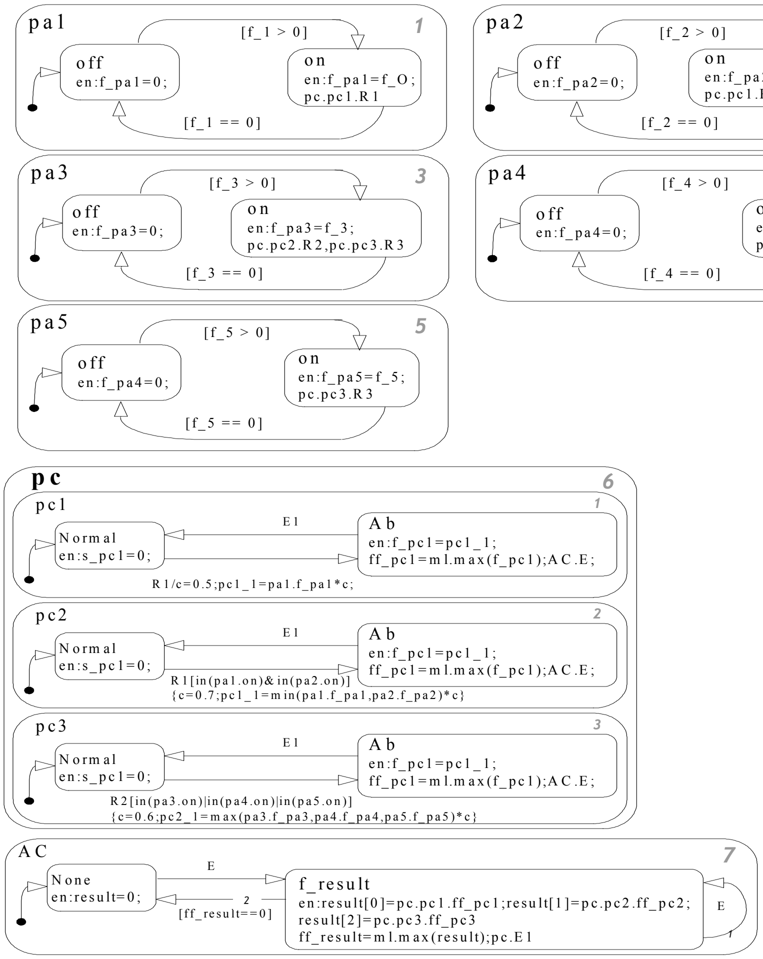

Fig. 10: State flow model of case

## CONCLUSION

In this study, fuzzy quality abnormity diagnosis model is constructed by use of FRPNs and its simulating model is established using state flow. The application result shows that the method proposed can dynamically show the diagnosis process following the given FRPNs and give the contribution degree of every assignable cause to the current process quality abnormity. The further study can be focused on the evaluation of the abnormity fuzzy membership degree based on control chart, so a complete diagnosis model can be constructed which takes control chart data as input, calculates its fuzzy abnormity degree and then output to the model developed in this study.

## ACKNOWLEDGMENT

This study was funded by China postdoctoral science foundation (NO.2011M501272). The authors would like to acknowledge the many helpful suggestions of reviewer's .We also thank the Editor of this Journal.

## REFERENCES

Chen, S.M., J.S. Kea and J.F. Chang, 1990. Knowledge representation using fuzzy petri nets. IEEE T. Knowl. Data Eng., 2: 311-319.

Davidrajuh, R., 2008. Developing a new Petri net tool for simulation of discrete event systems. Proceedings of 2nd Asia International Conference on Modeling and Simulation (AMS), Kuala Lumpur, pp: 861-866.

Daniel, R., M. Eugenia and Z. Noureddine, 2003. Fuzzy Petri nets for monitoring and recovery. Proceedings of the 2003 IEEE International Conference on Robotics and Automation, pp: 4318-4323, ISBN: 0-7803-7736-2.

Dennis, S.B. and M.H. Wassim, 1989. LQG control with an H∞ performance bounded: A Rickety equation approach. IEEE Trans. Auto. Cont., 34: 293-3051.

Han, Z. and X. Daoquan, 2002. Hybrid Petri nets for modeling of manufacturing systems. J. Tsinghai Univ. Sci. Technol., 42: 1429-1432.

Jing, S., Q. Shi-Yin and S. Yong-Hua, 2004. Fault diagnosis of electric power systems based on fuzzy Petri nets. IEEE T. Power Syst., 19(4): 2053-2059.

Jinsong, Z., L. Qiqiang, G. Qingqiang and W. Zhaoxia, 2007. A simulation method of controlled hybrid Petri nets based on Mat lab Simulink State flow. Proceedings of IEEE International Conference on Automation and Logistics, Jinan, pp: 2432-2436.

Ji-Ping, T., X. Wen-Yan, Y. Gen-Ke and W. Hao, 2006. A simulation method of Petri Nets based on state flow. Comput. Simul., 23(12): 96-100.

Li-Xin, J., X. Jun-Yi and R.U. Fang, 2003. Fuzzy Petri net based formalized reasoning algorithm with applications. J. Xi'an Jiao Tong Univ., 37(12): 1263-1266.

Looney, C.G., 1988. Fuzzy Petri nets for rule-based decision making. IEEE T. Sys. Man Cy., 18(1): 178-183.

Michaela, M., P. Konstantin and P. Octavian, 2005. A new approach to hybrid system simulation: Development of a simulink library for Petri Net Models. J. Cont. Eng. Appl. Inform., 7(4): 55-62.

Mirko, V. and D. Makajic-Nikolic, 2004. Fuzzy Petri net based reasoning for the diagnosis of bus condition. Proceedings of 7th Seminar on Neural Network Applications in Electrical Engineering, pp: 225-22, ISBN: 0-7803-8547-0.

Rangel, I.R., A.R. Trevino and E.L. Mellado, 2005. Building reduced Petri net models of discrete manufacturing systems. Mathemat. Comput. Model., 41: 923-937.

Riascos, L.A.M., L.A. Moscato and P.E. Miyagi, 2004. Detection and treatment of faults in manufacturing systems based on Petri nets. J. Braz. Soc. Mech. Sci. Eng., 26(3): 280-289.

Wu, N.E., 1997. Robust feedback design with optimized diagnostic performance. IEEE Trans. Auto. Cont., 42(9): 1264-12681.

Xiao-Guang, H., G. Mei-Mea, W. Yong-Hong and W. Shellie, 2000. Application of Fuzzy Petri nets in fault diagnosis of gas turbine. J. Aero. Power, 15: 311-313.