International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3, Issue No.3, pp : 227-230 1 march 2014

A Novel Approach to Reliability Analysis Using Petri Nets and Ladder Logic Diagrams

Majid Pakdel and Kazem Mazlumi
University of Zanjan, Zanjan, Iran
Email: pakdel@mail.ru, kmazlumi@znu.ac.ir

Abstract— ***In this paper the dynamic modeling of repairable and standby components of complex systems is analyzed using Petri Nets and ladder logic diagrams. The use of Petri Nets as modelling of a complex system avoids two basic disadvantages of Markov chain analysis. First, the Petri Net model does not grow in size as increasing the number of components and second, Markov chain has been limited to the exponential probability distribution. PetriLLD software is used for building Petri Net models and then the Petri Net models are converted to ladder logic diagrams for reliability analysis.***

Keywords— **Reliability, Petri Nets, Ladder Logic Diagrams**

I. Introduction

Petri Nets are a graphical tool for the formal description of the activities and dynamics in complex systems. Typical conditions that can be modeled by Petri Nets are synchronization, sequential events, concurrency and conflict. The theory of Petri Nets was introduced by C.A. Petri in 1962 [1]. Several textbooks on the Petri Net subject are available [2, 3, 4, and 5]. Petri Nets consist of four basic elements: places, transitions, tokens and arcs [6]. Places represent a state in the process, transitions represent the stochastic, or time-based nature of changes in the model. Transitions can be immediate, time-delayed, or time-delayed based on a probability distribution. Tokens represent objects or components in the model. Arcs are implemented between a transition and a place and they are not used between places or transitions. Places which arcs are entered to transitions are called input places and places which arcs are entered from transitions are called output places. In graphical form, places in a Petri Net may include discrete numbers of tokens. Token distribution in places forms a marking network. A transition is fired when there are enough tokens in input arcs. When a transition is fired it consumes these tokens and the tokens are included in all output arcs. Firing is a non interruptible procedure. When several transitions are activated at the same time any of them can be fired. If a transition is activated it may be fired or not fired. Since the firing operation is not determined and several tokens can be in anywhere in a Petri Net thus Petri Net modelling is suitable for concurrency behaviour of discrete systems. Firing of a transition in a Petri Net is shown in Fig. 1 and the condition when a transition could not be fired is depicted in Fig. 2.

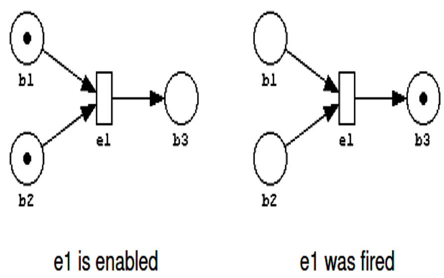

Fig.1. Firing of a transition in a Petri Net.

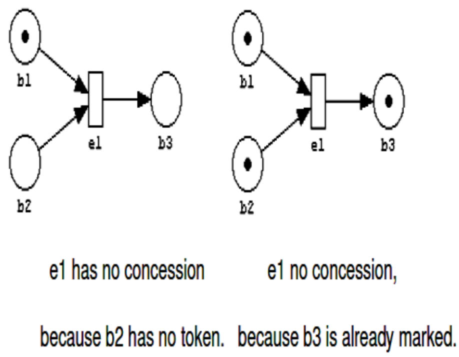

Fig.2. Conditions when a transition could not be fired.

II. Reliability Modelling Using PetriLLD

PetriLLD is a java based simple graphical tool that can be used to build a Petri net that represents the desired behavior. PetriLLD is built upon the Petri net formalism. Specifically, it uses a modified form of a basic sort of a net that only allows one token in a place. Petri nets are an excellent model for expressing concurrent behavior. For this reason, they are useful for modeling the behavior of discrete event systems such as those in manufacturing plants where there may be many operations occurring simultaneously [7]. Initial place (Idle) is shown with a yellow circle, input with yellow triangle, output with a yellow square, transition with a green rectangle and arc with a connection between a place or input to a transition or between a transition and an output as shown in Fig. 3.

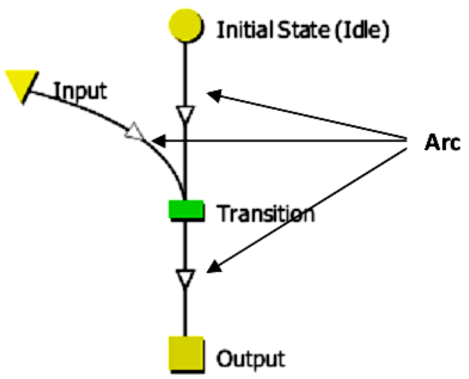

Fig.3. Petri Net elements in PetriLLD.

The proposed repairable complex system with standby components is shown in Fig. 4.

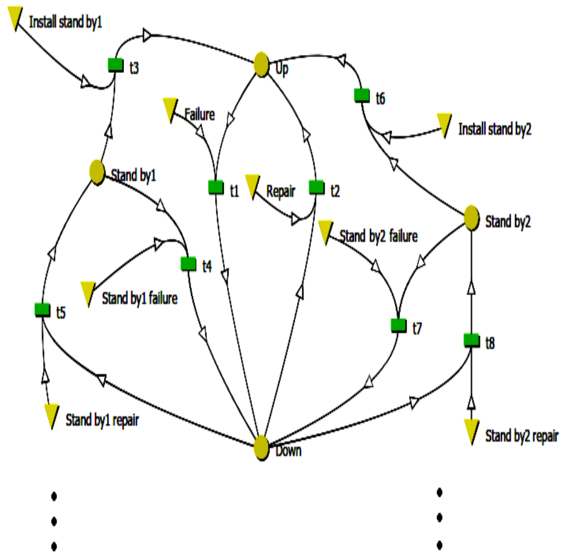

Fig.4. Proposed repairable complex system with standby components.

To convert the Petri Net diagram shown in Fig. 4 to a ladder logic diagram, the transition logic, place logic and output logic is written sequentially as shown in Fig. 5.

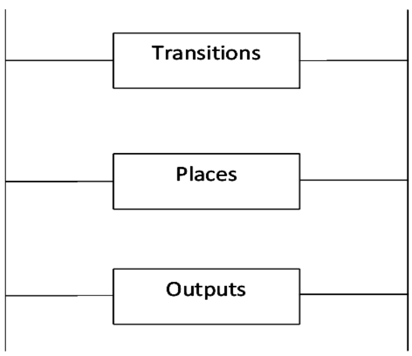

Fig.5. Sequence of ladder logic diagram.

For writing the transition logic, all places and inputs entered to the specified transition are considered as series elements with normally open (NO) contacts and the mentioned transition would be as the output of the ladder logic rung. For writing the place logic, all transitions entered to the specified place are considered as parallel elements with normally open (NO) contacts and all transitions leave the mentioned place are considered as series elements with normally closed (NC) contacts and the mentioned would be as the output of the ladder logic rung. The ladder logic rung of initial place is the same as other places logic but it also includes the normally closed (NC) contacts of other places. For writing the output logic, all desired places are considered as parallel with normally open (NO) contacts.

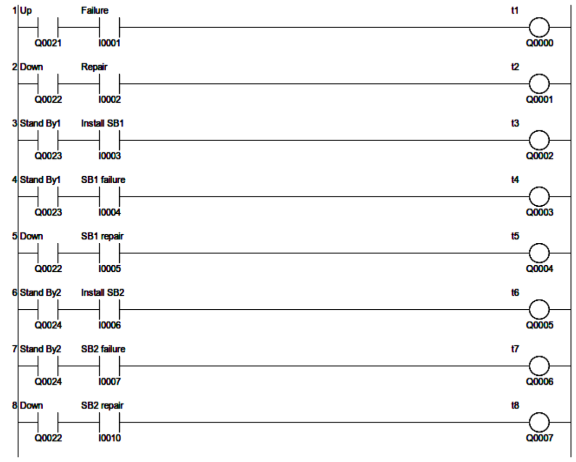

Fig.6. Transitions ladder logic.

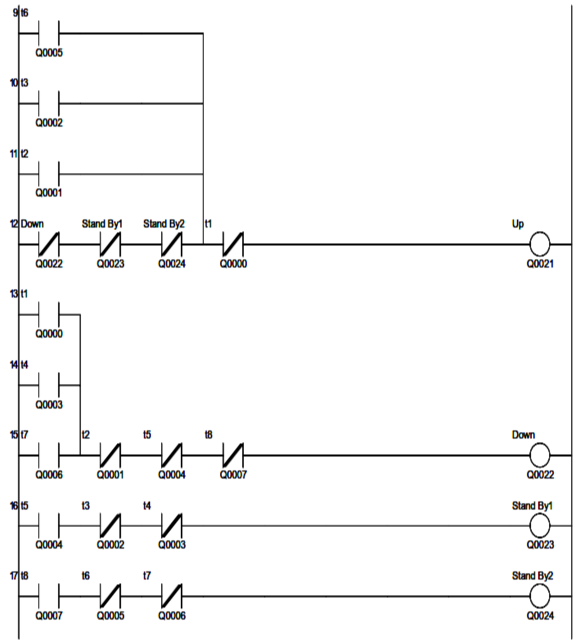

Fig.7. Places ladder logic.

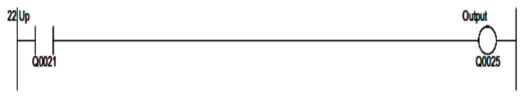

Fig.8. Output ladder logic.

Fig. 6, Fig. 7 and Fig. 8 depict the transitions, places and output ladder logics respectively. The reliability analysis now is straight using ladder logic diagram. So we can write from Fig. 6,

$$\begin{cases} P(t_1) = P(Up).\ P(Failure) \\ P(t_2) = P(Down).\ P(Repair) \\ \ldots\ldots\ldots \\ P(t_8) = P(Down).\ P(Stand\ By2\ Repair) \end{cases} \quad (1)$$

In equation (1), P(Failure), P(Repair), …, and P(Stand By2 Repair) are time dependent probabilities and can have any probability distribution functions. For example they may have the following probability density functions:

$$\begin{cases} f(t) = \lambda e^{-\lambda t} \quad \text{(Exponential distribution)} \\ f(t) = (\lambda t)^x . e^{-\lambda t}/x! \quad \text{(Poisson distribution)} \\ f(t) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\left(\frac{t-\mu}{\sigma}\right)^2}{2}} \quad \text{(Normal distribution)} \\ \ldots\ldots\ldots \end{cases} \quad (2)$$

Also, we know that:

$$\begin{cases} P(Defect) = Q(t) = \int_0^t f(t)\, dt \\ P(Success) = R(t) = 1 - Q(t) \end{cases} \quad (3)$$

Also, from Fig. 7 we have:

$$\begin{cases} P(Down) = (1-(1-P(t_1))(1-P(t_4))(1-P(t_7)))(1-P(t_2))(1-P(t_5))(1-P(t_8)) \\ P(Stand\ by1) = P(t_5)(1-P(t_3))(1-P(t_4)) \\ P(Stand\ by2) = P(t_8)(1-P(t_6))(1-P(t_7)) \\ P(Up) = (1-(1-(1-P(Down))(1-P(Stand\ by1))(1-P(Stand\ by2)))(1-P(t_2))(1-P(t_3))(1-P(t_6)))(1-P(t_1)) \end{cases} \quad (4)$$

Therefore, we can continue the following steps to reliability analysis of the proposed system. First, for each time step we find the probability of transitions using the equations (1), (2) and (3) then apply the transition probabilities to find the new values for place probabilities from equation (4), the newly obtained values for place probabilities can be used again for finding new values for transition probabilities from equations (1), (2) and (3) and these steps continue up to simulation time.

The results of simulation with exponential distribution function and the following values for failure, repair and installation rates for 20 hours of system operation are shown in Fig. 9.

Failure rate = 0.15 f/hr; repair rate = 2 r/hr;
Install SB1 rate = 5 repl/hr; SB1failure rate = 0.15 f/hr;
SB1repair rate = 2 r/hr; Install SB2 rate = 5 repl/hr;
SB2 failure rate = 0.15 f/hr; SB2 repair rate =2 r/hr;

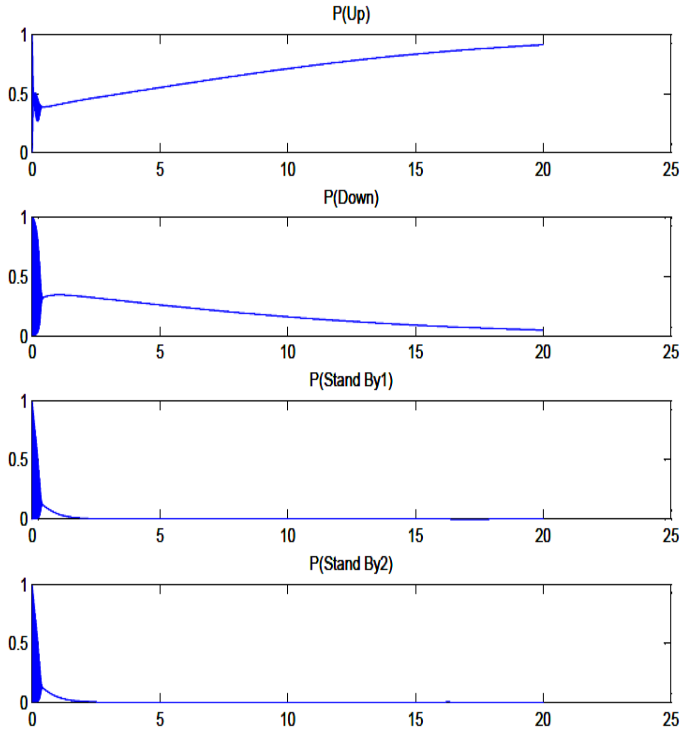

Fig.9. Results of simulation for 20 hours of system operation.

Therefore, from simulations it is obvious that by using Petri Net and ladder logic diagram first, the size of the model is easily controllable, regardless of the amount of tokens present. Once the model is constructed, the user can change the number of tokens, without affecting the places and transitions. Conversely, if Markov Chain used to model the system the resulted model would be extremely large. Second, Petri Nets allow the analyst to model dynamic events using other probability distribution functions than the exponential distribution. This distribution, widely regarded as appropriate for electronic components, is usually inappropriate for mechanical components and real world processes. Petri Nets give the user more flexibility in creating a model. Third, Petri Nets, used in a simulation, allow the user to actually observe the stochastic processes. The user can develop a model, and then observe the tokens as they move throughout the model in real or simulated time. This feature gives the user insight into the actual flow of the model and any potential conflicts. Finally, Petri Nets are easy to modify. Since the model itself is very simple, it is easy to change the values of transitions or the number of tokens without having to change the entire model itself. This characteristic is useful when the user needs to analyze several different cases [8].

III. Conclusion

In this paper the dynamic modeling of repairable and standby components of complex systems was analyzed using Petri Nets and ladder logic diagrams. The use of Petri Nets as modelling of a complex system avoids two basic disadvantages of Markov chain analysis. First, the Petri Net model does not grow in size as increasing the number of components and second, Markov chain has been limited to the exponential probability distribution.

PetriLLD software was used for building a Petri Net model then the Petri Net model was converted to ladder logic diagrams for reliability analysis and a simulation was done using typical parameters. Petri Nets offer an analyst another powerful tool to simply, and effectively model stochastic processes. Petri Nets can overcome to Markov Chains. The advantages that Petri Nets present over Markov Chains could make them the tool of choice for dynamic models.

IV. References

i. *C.A. Petri. Kommunikation mit automaten. Technical report, Doctoral Thesis, University of Bonn, 1962. (Available in English as: Communication with automata, Technical Report RADC-TR-65-377, Rome Air Development Center, Griffiss NY, 1966).*

ii. *J.L. Peterson. Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs, 1981.*

iii. *W. Reisig. Petri nets - An introduction. Springer-Verlag, 1982.*

iv. *G.W. Brams. R´eseaux de Petri: Th´eorie et pratique. Masson, 1983. (in French).*

v. *M. Silva. Las Redes de Petri en la Automatica y la Informatica. AC, Madrid, 1985.*

vi. *C. Girault and R. Valk, Petri Nets for Systems Engineering, Springer-Verlag, 2001.*

vii. *James Brusey, PetriLLD Tutorial, 2006.*

viii. *Captain Richard V. Melnyk, Petri Nets: An Alternative to Markov Chains http://theriac.org/DeskReference/ view Document.php ?id=80 & Scope =journal&year=2004&Q=1.*

.