Revised October 23, 2002

System Failure Analysis Through Counters Of Petri Net Models

Angela Adamyan and David He*
Intelligent Systems Modeling and Development Lab
Department of Mechanical and Industrial Engineering
The University of Illinois at Chicago
Chicago, Illinois 60607
Tel: 312-996-3410; E-mail: davidhe@uic.edu; Fax: 312-413-0447

Abstract

Petri net is a powerful technique widely used in modeling and analysis of complex manufacturing systems and processes. Due to its capability in modeling the dynamics of the systems, Petri net has been combined with fault tree analysis techniques to determine the average rate of occurrence of system failures. Current methods in combining Petri nets with fault trees for system failure analysis compute the average rate of occurrence of system failures by tracking the markings of the Petri net models. The limitations of these methods are that tracking the markings of a Petri net represented by a reachability tree can be very complicated as the size of the system grows. Therefore, these methods offer less flexibility in analyzing sequential failures in the system. To overcome the limitations of the current methods in applying Petri net for system failure assessment, this paper expands and extends the concept of counters used in Petri net simulation to perform the failure and reliability analysis of complex systems. The presented method allows modeling the system failures using general Petri nets with inhibitor arcs and loops and employs fewer variables than existing marking-based methods and substantially accelerates the computations. It can be applied to real system failure analysis where basic events can have different failure rates.

Key words: Reliability, Petri nets, system failure, fault tree analysis.

* To whom correspondence should be addressed

1. Introduction

The reliability and safety analysis of complex systems and processes is becoming a more and more difficult task due to the rapid technology evolution and increasing complexity of the systems that, in turn, often causes the increase of occurrence of failures in the systems. A failure is defined as an event when a required function is terminated, or exceeds acceptable limits (IEC 50(191)). The average rate of occurrence of failures is defined as the number of times a failure has occurred per time k. In this paper, we refer the term *failure rate* to basic events and define it as a ratio of the probability density function of the basic event to the cumulative distribution function. For reliability and safety assurance, failures of the systems have to be traced and analyzed.

A variety of classical methods for reliability and safety analysis exist. These include reliability block diagram (RBD), Monte Carlo simulation, semi-Markov process, failure mode and effect analysis (FMEA), and fault tree analysis (FTA). Among the methods, FTA is a widely accepted and used technique for analysis of system failures. However, FTA has several drawbacks, for example, it represents only logic relations.

A recent development in reliability analysis is the application of Petri nets. The Petri net methodology of reliability modeling is similar to that of fault tree modeling. They both use graphical representation of the relations between conditions and events. The application of Petri nets to reliability engineering has been limited but a few examples can be found in reliability evaluation (Hura and Atwood, 1988; Bobbio 1988; Kumar and Aggrawal, 1993; Liu and Chiou, 1997), fault tolerant analysis (Viswanadham, 1987), safety analysis and reliability growth (Levenson and Stolzy, 1987; Yang and Liu, 1997), reliability of manufacturing systems (Jiang *et al*, 1995; Xiong and He, 1997)

Previous research demonstrated the superiority of Petri nets over FTA (Liu and Chiou, 1997; Yang and Liu, 1997). A Petri net model graphically symbolizes the cause and effect relationships among the events. Furthermore, it represents and analyzes dynamic behavior of the system and allows performing comprehensive failure and reliability analysis of the system. Petri net modeling provides the ability of assessing the quality and reliability impacts of unplanned failures and the sequence of these failures (Adamyan and He, 2002; He and Adamyan, 2001).

Traditionally, failure times of system components are assumed to follow the exponential distribution. A fundamental reason for the popularity of the exponential distribution and its widespread usage in reliability work is that exponentially distributed random variables are memory-less, which makes possible the computation of design data in a simple form. However, as early as in 1951, serious consideration began to be given to other life distributions and hence made the computation of reliability more complex.

Although useful, the current methods in applying Petri nets for system reliability modeling imply that the times between failures of different components are the same and constant. Second, the methods do not take into the account the possibility of loops and inhibitor arcs in Petri net modeling. The mentioned limitations of current methods in

reliability modeling using Petri nets do not allow representing the real-life systems completely. The addition of inhibitor arcs is an important extension of the modeling power of Petri nets, which gives them the same computational power as Turing machines (Peterson, 1981). Other Petri net-based methods use markings as state variables and compute the average rate of occurrence of the system failures based on marking transfer through the system. The marking-based methods are limited in applying Petri nets for reliability assessment of sequential failures.

The novel feature of this paper is that it expands and extends the concept of counters used in Petri net simulation to perform the failure and reliability analysis of complex systems. The presented method allows modeling the system failures using general Petri nets with inhibitor arcs and loops. It employs fewer variables than existing marking-based methods and substantially accelerates the computations. The method uses mean time to failure of basic events, therefore, can be applied to real system failure analysis, where basic events can have different failure rates and the failure times of the basic events can follow any distribution.

The remainder of the paper is organized as following. In Section 2, we provide a background of fault tree analysis, the general idea of Petri net modeling and conversion methods of fault trees to Petri nets. In Section 3, the development of the method is described. Section 4 demonstrates the proposed methodology on the example of nitric acid cooler with temperature feedback and pump-shutdown feedforward loops. Section 5 concludes the paper.

2. Background

In this section, the background of fault tree analysis, the basic idea of Petri net modeling, and the correlations between fault trees and Petri nets are provided.

2.1 Fault tree analysis

A fault tree arises from the logic diagram that is used to analyze the probabilities associated with various causes and their effects. FTA starts by identifying a problem (catastrophic accident or other undesirable result) and all possible ways that the problem (or failure) occurs. FTA has been widely used for obtaining reliability information about complex systems since 1960. The importance of FTA was pointed out in a safety study of the US Nuclear Regulatory Commission (1975). In addition, FTA is a powerful design tool that can help to meet product performance objectives.

Minimal cut set is a set of components, in which the repair of any failed component will result in functioning of the failed system. FTA is equivalent to the minimal cut set tree with all minimal cut set in an *AND*-structure. A minimal cut-*AND* structure is a set, in which the failed state of the output becomes true when all states of the inputs exist simultaneously. Therefore, it is very important to estimate the output of the minimal cut-*AND*-structure in order to quantify the top event of the fault tree.

2.2 Petri nets

Petri nets are widely used as a tool for analyzing system safety and reliability of the complex systems. They can be used as visual communication aid similar to flow charts, block diagrams, fault trees and networks. The use of Petri nets augments the ability of understanding the interaction between various effects. First developed by Adam Petri in the early 1960's, Petri nets have become a powerful and generic tool for modeling and simulation (Peterson, 1981; Holliday and Vernon, 1987; Murata, 1989; Ramaswamy and Valavanis, 1994; Liu and Chiou, 1997). The Petri nets where random delays are exponentially distributed are referred to as stochastic timed Petri nets (SPNs) (Zhou, 1995; Molloy, 1982, 1985; Florin *et al*., 1991). General-purpose software packages are available for solving SPN models, including GreatSPN (Chiola, 1987) and SPNP (Ciardo, 1989).

Incorporating Petri net modeling into system reliability and safety analysis provides an ability to assess the quality and reliability impacts caused by unplanned failures and their sequences. Petri nets have the ability to track system states and transitions between these states based on some triggering requirements and this ability allows to analyze combined failure modes and to predict their potential severity, as well as to estimate the probability of occurrence of failure modes. With this knowledge, engineers can put into place effective means to prevent the impacts of the failures.

Al-Jaar and Desrochers (1990) have demonstrated that Petri net modeling is superior over traditional Markov chain modeling in that the number of places and transitions increases slightly as the system complexity increases, whereas the number of states in the Markov chain increases exponentially. In addition, Petri net modeling provides a general and formal procedure to generate all possible states for analysis.

Baccelli *et al*. (1995) have demonstrated that Petri nets provide a powerful formalism to model various classes of discrete events. They can be used for qualitative and simulation purposes because they provide a better understanding of the dynamics of the system. The method presented provided substantial acceleration to the simulation.

Formally, Petri net is a directed bipartite graph defined by a 6-tuple $N = [T, P, A, M_0, I(t), O(t)]$, where $T = \{t_1, t_2, \ldots, t_n\}$ is a set of transitions, each transition representing an event or an action; and $P = \{p_1, p_2, \ldots, p_l\}$ is a set of places, where a place is used to represent either the condition for the event or the consequences of the event. Therefore, before building a Petri net model, the events and their conditions and consequences in a system are first defined, and then are represented by transitions and places in a Petri net. Each place can contain one or more tokens. Movement of tokens through the places in the constructed model simulates the process of the system and allows identifying and analyzing what can go wrong within the system. A place with (or without) a token indicates that the state represented by the place is true (or false). $A \subseteq \{T \times P\} \cup \{P \times T\}$ is a set of directed arcs that connect transitions to places and places

to transitions. M_0 is the initial marking of the system that represents initial state of the system. A marking M, can also be represented as a vector $M = \{m_1, m_2, \ldots, m_l\}$, where m_i is the number of tokens in place p_i. Figure 1 represents the Petri net with initial state M_0 = {2, 1, 0}, indicated by the number of tokens (black dots) in corresponding places.

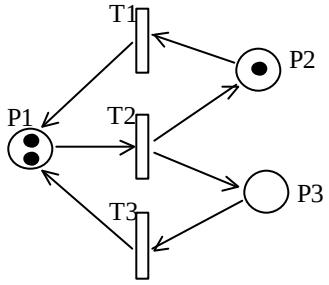

Figure 1. Petri net with three places and three transitions (n=3, l=3)

Places that represent the conditions of a transition are connected to that transition as input places, and places that represent the consequences of the events are connected to the transitions as output places. Respectively, $I(t) = \{p \mid (p,t) \in A\}$ is the set of input places of a transition t, and $O(t) = \{p \mid (t,p) \in A\}$ is the set of output places of a transition t. Directed arcs connect transitions to places and vice versa. A directed arc from a place to a transition is called an input arc, and an arc from a transition to a place is called an output arc.

In a Petri net, an action is represented by the 'firing' of a transition. The behavior of the Petri net is determined by following firing rules:

(1) Tokens in places with arcs towards a transition indicate that conditions are satisfied and the transition is ready to fire (event to occur).
(2) Upon firing, transition t consumes one token along each input arc.
(3) Upon firing, transition t produces one token along each output arc.

Whenever a transition is fired, tokens are taken away from the input places and appear in the output places of the transition. An arc with double arrows indicates that a place serves as both an input place and an output place. An arc with small circle instead of an arrowhead is an inhibitor arc. The inhibitor arc disables the transition when the input place has a token, and enables the transition when the input place has no token and other (normal) input place(s) have a token per arc.

If the firing of the transition results in a new marking $M¢$ from marking M, then $M¢$ is *immediately reachable* from M. Marking $M¢¢$ is *reachable* form marking M if it is reachable form any marking that is immediately reachable from M. The reachability tree is a graphical representation of all markings of a net starting from its initial marking. In other words, the reachability tree is the state diagraph in which each node represents the unique marking, i.e. state of the system, and edges represent the possible state transitions.

2.3 Transformation of fault trees to Petri nets

The Petri net graphical representation can be used to construct the cause and effect relationship among the events. Since boolean logic symbols are commonly used to account for failure causation, to convert a fault tree (FT) to a Petri net one needs to examine logic gates based on Petri net representation. According to enabling rules every logic gate can be represented by a Petri net model. Liu and Chiou (1997) have demonstrated that Petri nets in failure analysis can be used to replace logic gate functions in a fault tree. The transition of FTs to the Petri net representations allows performing thorough failure and reliability analysis of the systems, as well as provides efficient way for obtaining path sets and minimal cut sets.

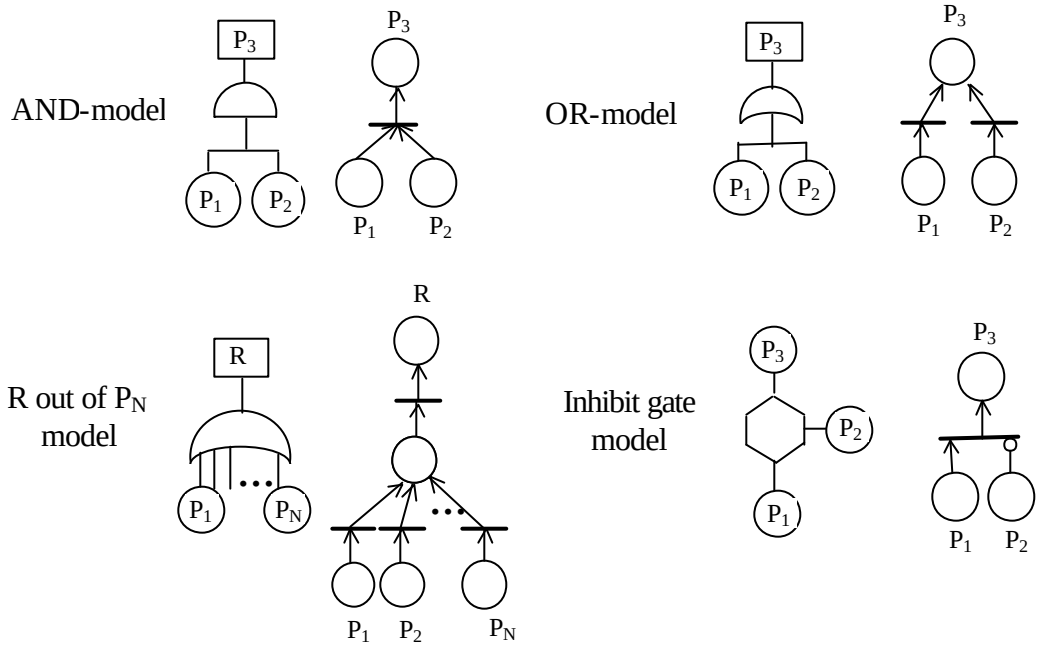

Figure 2. Correlations between fault trees and Petri nets

Some examples of the correlations between FTs and Petri nets are shown in Figure 2. For more detailed explanation of the transformation of fault trees to Petri nets readers can refer to (Yang and Liu, 1997).

3. The Method Development

In this paper, the method for computing the average occurrence of a failure is developed based on the concept of counters in Petri net simulation. A counter in a Petri net model represents the number of times a transition has fired in a certain period of time.

Baccelli and Canales (1991) introduced the concept of counters. They presented the novel approach for simulation of large Petri nets. They showed that analysis through counters is a powerful tool for discrete event simulation. The simulation was no longer based on the events occurring in the system but on evolution equations describing the

dynamics of the system. The method provides substantial acceleration of the simulation. However, their results are restricted to a small class of systems: the marked graphs. A marked graph is a subclass of the Petri net in which each place is either an input place or an output place of exactly one transition. In Baccelli *et al.* (1995), the authors extended the work to free choice nets (when a place could be an input to several transitions) and more general class of nets with a considerably greater power of description than the marked graphs. Unfortunately, the methods developed by Baccelli *et al* (1995) focused on only the simulation aspects of the Petri nets and described the evolution of the counters only for simple cases providing no information for general Petri nets that consist of inhibitor arcs and loops.

The approach presented in this paper expands and extends the concepts of counters presented by Baccelli *et al*., (1995) to reliability and failure analysis of complex systems. The systems that can be modeled with inhibitor arcs and loops, and consists of components with different times to failures.

To facilitate the method development we will use the same notations and definitions presented in Baccelli *et al*., (1995). The notations and definitions used throughout the paper are defined as follows:

${}^{\bullet}t = \{p \in P : (p,t) \in A\}$ is the set of all the input places of t,
$t^{\bullet} = \{p \in P : (t,p) \in A\}$ is the set of all the output places of t,
${}^{\bullet}p = \{t \in T : (t,p) \in A\}$ is the set of all the input transitions of p,
$p^{\bullet} = \{t \in T : (p,t) \in A\}$ is the set of all the output transitions of p,
$X_t(k)$ = number of times transition t has fired by time k.
$f_t(n),\ n \in N$ = duration of n^{th} firing of transition t, N is a set of positive numbers.

The firing times are restricted to be positive numbers. We assume that they all are bonded by K: $\forall t \in T,\ \forall n \in N,\ f_t(n) \in \{1,\ldots,K\}$.

Definition 1. *Serial Place*
A serial place is a place p such that $|\,{}^{\bullet}p\,| = |\,p^{\bullet}\,| = 1$.

Definition 2. R*outing Place*
A routing place is a place p such that $|\,{}^{\bullet}p\,| + |\,p^{\bullet}\,| > 2$.

There are several policies to solve the problem of conflicts when two transitions share a common input place. One of them is implementing the routing policy. In the routing policy, each routing place p is given a routing sequence $p^p(n)$: $N \rightarrow p^{\bullet}$, where $p^p(n)$ is the transition $t \in p^{\bullet}$ which receives the n^{th} token to enter place p. The routing sequence can be periodic or random. If a token is k^{th} to be routed to transition $t \in p^{\bullet}$, then this token is immediately consumed by the transition where it experiences the firing time $f_t(k)$.

Definition 3. *Routing function*
From the sequence $p^p(n)$ defined for each routing place p the routing function for every transition $t \in p^\bullet$ is:

$$\Pi^t(n) = \sum_{i=1}^{n} 1_{(p^\bullet(i)=t)} \tag{1}$$

$\Pi^t(n)$ gives the number of tokens that are routed to transitions t after n tokens have entered place $^\bullet t$

The marking at time k can be easily retrieved from the counters (Baccelli *et al.*, 1995). If let $M_p(k)$ be the marking of place p at time k, then

$$M_p(k) = \sum_{t \in {}^\bullet p} X_t(k) - \sum_{t \in p^\bullet} X_t(k) \tag{2}$$

The transition can be separated into two subsets:

- the set of transitions that belong to the output set of serial places (Figure 3a)
- the set of transitions that belong to the output set of routing places (Figure 3b).

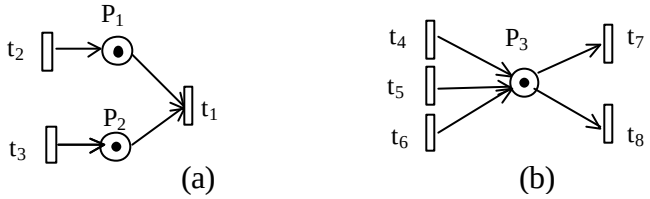

Figure 3. Two sets of transitions: (a) that belong to the output set of serial places and (b) that belong to the output set of routing places

Let T_1 be the set of transitions that belong to the output of serial places and T_2 the set of transitions that belong to the output set of routing places. For the sake of explanation of the method development, a Lemma from (Baccelli *et al*, 1995) is restated in the following.

Lemma 1: *For a transition* $i \hat{I}\ T_1$

$$X_i(k) = \Pi^i\left(N_{\bullet i}(k)\right), \tag{3}$$

where

$$N_{\bullet i}(k) = \sum_{j \in {}^\bullet i} \left(X_j\left(k - f_j\right) + M_0(j,i)\right).$$

And for transition $i \hat{I}\ T_2$:

$$X_i(k) = \min\left(X_j\left(k - f_j\right) + M_0(j,i)\right) \tag{4}$$

Next, the concept of a counter in Petri net simulation will be extended to determine the computation of the average failure occurrence rate of a top event in a system modeled by Petri nets. Based on Lemma 1 and the definitions provided above, the evolution of counters in a Petri net model is described below.

3.1. Transitions with serial places

Transitions with serial places can be clustered into three groups: transitions with a single place, transitions with hierarchical serial places, and transitions with serial places in a loop. The counters of the top transition of the corresponding Petri net structures are described next.

3.1.1. *Transition with a single place*

The transition with a single place is revealed in Figure 4.

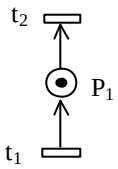

Figure 4. Transition with a single place

The counter of transition t_2 is computed as follows:

$$X_2(k) = X_1(k - f_1) + 1 \tag{5}$$

3.1.2. *Transition with hierarchical serial places*

The transition with hierarchal serial places is revealed in Figure 5.

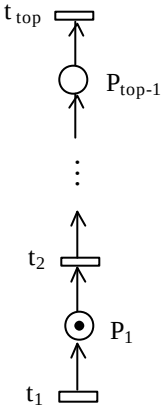

Figure 5. Transition with hierarchical serial places

The counter of the top transition in a hierarchical structure is derived from equation (5) as follows:

$$X_2(k) = X_1(k - f_1) + 1$$

$$X_3(k) = X_2(k - f_2)$$

$$\vdots$$

$$X_{top} = X_{top-1}(k - f_{top-1})$$

$$X_{top} = X_1(k - f_1 - f_2 - \ldots - f_{top-1}) + M_0(2,1) = X_1(k - \Phi) + M_0(2,1) \qquad (6)$$

where $\Phi = \sum_{i=1}^{top-1} f_i$ denotes the total delay time of the transitions.

3.1.3. *Transition with serial places in a loop*

Figure 6 reveals the transition with serial places in a loop.

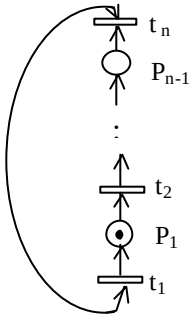

Figure 6. Transition with serial places in a loop

Based on (6), the counter of any transition in a loop structure can be derived as follows:

$$X_i = \left(k - \sum_{j=1}^{i-1} f_j - \sum_{s=0}^{m} \sum_{l=1}^{n} f_l \right),$$

where: m =1,2 …, ∞ such that $k - \sum_{j=1}^{i-1} f_j - \sum_{s=0}^{m} \sum_{l=1}^{n} f_l > 0$.

3.2. Transitions with multiple inputs

Transitions with multiple inputs can be divided into two categories: Petri net structures equivalent to the *AND*-gate of a FT and Petri net structures equivalent to the *OR*-gate of a FT.

3.2.1. *Equivalence to AND-gate of a fault tree*

Petri net structures resembling the *AND*-gate of a FT can be formed by single level of places or multiple levels of places. Each representation may have inhibitor arcs. Single

level *AND*-structures, multiple level *AND*-structures, and *AND*-structures with inhibitor arcs are presented next.

3.2.1.1 Single level AND-structure

The single level *AND*-structure is revealed in Figure 7.

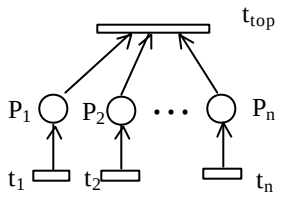

Figure 7. Single level input places in an *AND*-structure

According to Lemma 1, the counter of the top transition is computed as the minimum number of firings of its downstream transitions:

$$X_{top} = \min\left[X_1(k-f_1), X_2(k-f_2), \ldots, X_n(k-f_n)\right] = X_s(k-f_s) \qquad (7)$$

In equation (7), f_s is the largest among f_i, i = 1, 2, ... , n. In other words, the counter of the top transition is computed as the counter of the proceeding transition with the longest firing time.

3.2.1.2 Multiple level AND–structure

The multiple level *AND*–structure is revealed in Figure 8.

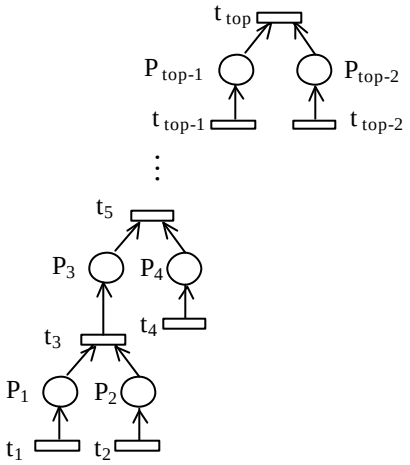

Figure 8. Multiple level input places in an *AND*-Structure

From (7) and in accordance with Figure 8, the counter of the top transition in the multiple level *AND*-structure is derived as follows:

$$X_3(k) = \min\left[X_1(k-f_1), X_2(k-f_2)\right]$$
$$X_5(k) = \min\left[X_3(k-f_3), X_4(k-f_4)\right]$$
$$\vdots$$
$$X_{top}(k) = \min\left\{\ldots \min\left\{\min\left[X_1(k-f_1), X_2(k-f_2), X_4(k-f_4)\right]\right\}, X_{top-1}(k-f_{top-1})\right\} =$$
$$\min\left[X_1(k-f_1-f_3-\ldots-f_{top-2}), X_2(k-f_2-f_3-\ldots-f_{top-2}),\right.$$
$$\left.X_4(k-f_4-f_5-\ldots-f_{top-2}), \ldots, X_{top-1}(k-f_{top-1})\right] =$$
$$= \min\left[X_1\left(k-\sum_{s=1}^{top-2} f_s\right), X_2\left(k-\sum_{s=2}^{top-2} f_s\right), \ldots X_{top-1}(k-f_{top-1})\right] \quad (8)$$

3.2.1.3 AND-structure with inhibitor arcs

The *AND*-structure with inhibitor arcs is revealed in Figure 9.

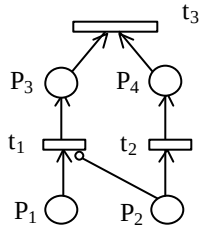

Figure 9. The *AND*-structure with inhibitor arcs

Based on the property of inhibitor arcs of Petri nets and equation (7), the counter of transition t_3 in Figure 9 is derived as follows:

$$X_3(k) = \min\left[X_1(k-f_1-f_2), X_2(k-f_2)\right] \quad (9)$$

3.2.2 *Equivalence to OR-gate of a fault tree*

Similar to the Petri net structures resembling the *AND*-gate of a FT, the structures resembling the *OR*-gate of a FT can be formed by a single level of transitions or multiple levels of transitions. Each representation may have also inhibitor arcs. The counters of the top transitions of the corresponding structures are presented next.

3.2.2.1 Single place

The *OR*-structure with a single place is revealed in Figure 10.

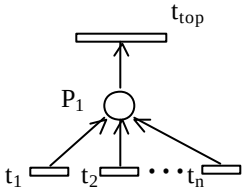

Figure 10. Single level place for *OR*-structure

According to Figure 10, the counter of the top transition in the *OR*-structure is the summation of the input counters:

$$X_{top} = X_1(k - f_1) + X_2(k - f_2) + \ldots + X_n(k - f_n) = \sum_{s=1}^{n} X_s(k - f_s) \qquad (10)$$

3.2.2.2 OR-structure with multiple levels

The *OR*-structure with multiple levels is revealed in Figure 11.

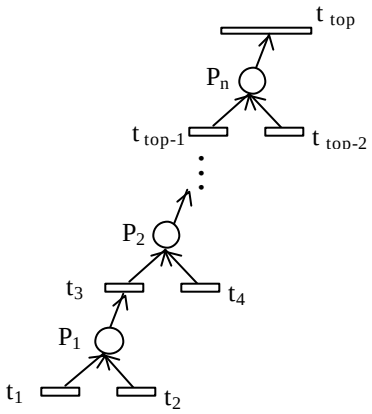

Figure 11. *OR*-structure with multiple levels of places

From (10) and in accordance with Figure 11, the counter of the top transition is derived as follows:

$$X_3(k) = X_1(k - f_1) + X_2(k - f_2)$$

$$X_5(k) = X_3(k - f_3) + X_4(k - f_4) = X_1(k - f_1 - f_3) + X_2(k - f_2 - f_3) + X_4(k - f_4)$$

$$\vdots$$

$$X_{top}(k) = X_1(k - \sum_{s=1}^{R} f_{2s-1}) + \sum_{u=s}^{R-1} X_{2s}(k - f_{2u+1}) + X_{top-1}(k - f_{top-1}) \qquad (11)$$

where $R = \dfrac{[(top - 2) + 1]}{2}$

3.2.2.3 Transition with inhibitor arcs

The *OR*-structure with inhibitor arcs is revealed in Figure 12.

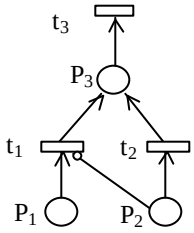

Figure 12. *OR*-structure with inhibitor arcs

Based on the property of inhibitor arcs of Petri nets and equation (10), the counter of transition t_3 in Figure 12 can be derived as follows:

$$X_3(k) = X_1(k - f_1 - f_2) + X_2(k - f_2) \tag{12}$$

3.3. Equivalence to inhibitor gate of a fault tree

In this subsection, we consider the Petri net structure resembling an inhibitor gate of a FT shown in Figure 13.

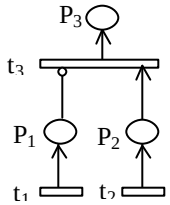

Figure 13. Petri net structure equivalent to an inhibitor gate of a fault tree

The counter of transition t_3 in a Petri net equivalent to an inhibitor gate of a fault tree is computed as following:

$$\text{For } k < f_1 \text{ if } \begin{cases} M_0(1,3) = 1, X_3(k) = 0 \\ M_0(1,3) = 0, X_3(k) = X_2(k - f_2) \end{cases} \tag{13}$$

$$\text{Otherwise } X_3(k) = X_2(f_1 - f_2) \tag{14}$$

3.4. Transitions with several input and output places

Petri net structures can have a transition with several inputs and outputs that do not belong to the *AND* or *OR* categories. Examples are routing places and complicated structures. Additional functions are introduced to deal with those cases.

3.4.1. *Routing places*

Figure 14 reveals the example of routing places, where a token in place P_3 can activate either transition t_7 or t_8.

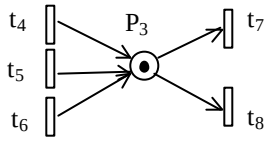

Figure 14. Routing places

For the set of transitions (t_7 and t_8 in Figure 14), the usual addition is used along with routing function (see Definition 3). To compute the number of firings of upstream transitions, the number of completed firings of upstream transitions must be added to the initial marking of the input place p, denoted by $N_p(k)$, and then the routing function must be applied. For the example in Figure 14, we can write (Baccelli *et al*, 1995):

$$\begin{aligned} N_3(k) &= X_4(k-f_4) + X_5(k-f_5) + X_6(k-f_6) + 1 \\ X_7(k) &= \Pi^7(N_3(k)) \\ X_8(k) &= \Pi^8(N_4(k)) \end{aligned} \tag{15}$$

3.4.2. *Complicated case of transitions*

Transitions that have in their input sets several places, which are not all serial, can also be solved (for example, transition t_{12} in Figure 15). Usual algebra and the *min* operator along with the canonical transformation of the Petri net would be necessary for calculating the counters of these types of transitions.

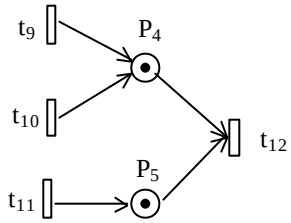

Figure 15. Complicated case of transitions in the Petri nets

Next, transformation to canonical form of Petri nets is presented (Baccelli *et al*., 1995). Let $F=(P, T, A, M_0)$ be a general Petri net. Each transition $t \in T$ verifies the following conditions:

- ${}^{\bullet}t \cap t^{\bullet} = \{p\}$ (i.e. each transition is recycled)
- $\forall n \in N$, $f_t(n) > 0$.

To obtain its canonical form $F_c = (P_c, T_c, A_c, M_{0c})$, the following transformation has to be performed. For every routing place p of F, we add a transition u and an empty place p_u between p and each output transition of p such as $p^{\bullet}=\{u\}$, $u^{\bullet}=\{p_u\}$ and $p_u{}^{\bullet}=\{p\}$ shown in Figure 16.

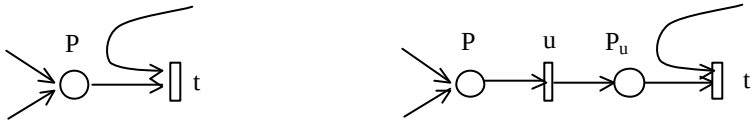

Figure 16. Canonical transformation

In the transformed Petri net we have only two sets of transitions, $T = T_1 \cup T_2$

- the added transitions $u \in T_1$ which verify ${}^{\bullet}u$ is a routing place,
- the original transitions $t \in T_2$ which verify ${}^{\bullet}t$ contains only serial places.

4. Example of Failure Analysis of Nitric Acid Cooler

In this paper, as an example, we use a heat exchanger (H.E.) system, which first appeared in Lapp and Powers (1977) and then was modified in Cheng and Yuan (2000) to demonstrate the application of the proposed method to model the dynamic behavior of the system failure. The function of the system depicted in Figure 17 is to cool a hot nitric acid stream before reacting it with benzene to form nitrobenzene. The water flow (10) is used to cool and control the temperature of the hot HNO_3 through heat exchanger (b). After sensing the temperature the control loops are to minimize the water flow in a proper amount by valve (e) to maintain temperature in flow (4) within targets according to the designed control mechanism.

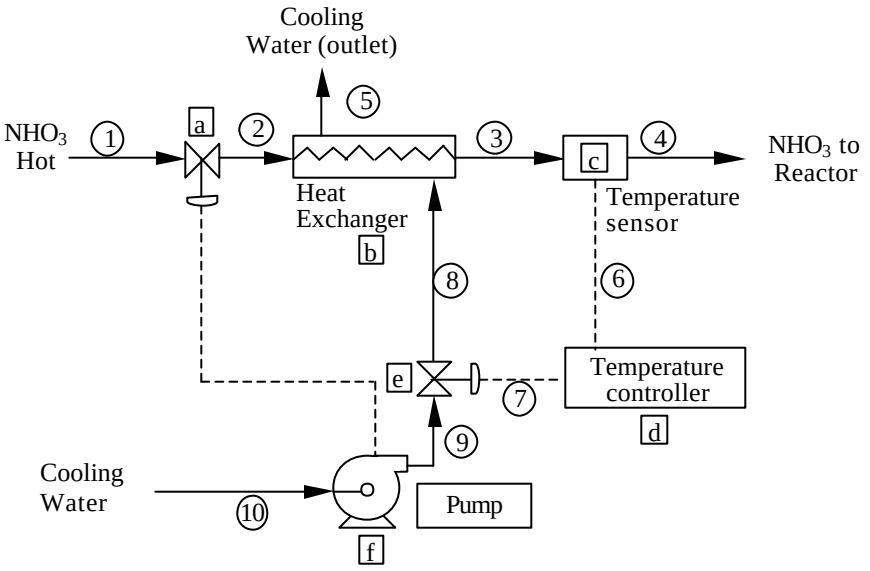

Figure 17. Nitric acid cooler with temperature feedback and pump-shutdown feedforward loops

One critical failure (top event) of the system is a high temperature in the nitric acid reactor feed since this could cause a reactor runaway. The following notations are used: T_i - temperature for the component i in the Figure 18; "+" and "-" denote direction of the deviation (positive or negative); 0, 1, 10 denote magnitude of the deviation (none, moderate, or very large); X - sensed variable. The system is assumed to have two states, *N* and *Ext-fire*. The states are shown and described in Table 1.

Table 1. Relation between heat exchanger input and put output

Heat Exchanger	Valve 1 (*Input*)	Output (2) *Corresponding response*
N	±10; ±1; 0	±10; ±1; 0
Ext-fire	0; +1; -1	+1; +10; 0

Nitric acid cooler (Figure 17) has feedforward loop indicated by dash line and feedback loop indicated by solid line. Each of such control loops is split into three portions: water supply, the *MP* (manipulating path) and the heat exchange (H.E.) So, the states of the control loops are completely specified by those of such three portions, as *C-N*, *C-STK* (*STK = stuck*), *C-REV* (*REV = reverse*), *C (x)*, *Water Supply-LO* (and *MP-N*) and *Water Supply-SH* (and *MPN*), which can be seen in Table 2. In Table 2, the malfunction that both water supply and *MP* are not normal is ignored.

Table 2. States and failure modes of the feedback control loop

States of the control loop according to its function	Description
C-N	*Water Supply-N* and *MP-N*
C-STK	*Water Supply-N* and *MP-STK*. This malfunction only happens when $X(x)$ for $x \in \{\pm 1; \pm 10\}$.
C-REV	*Water Supply-N* and *MP-REV*
$C(x)$, $x \in \{\pm 1; \pm 10\}$	*Water Supply-N* and *MP(x)*. This malfunction only happens when $X(0)$.
Water Supply-LO (*MP-N*)	This malfunction happens on *MP-N*.
Water Supply-SH (*MP-N*)	This malfunction happens on *MP-N*.

Note: $X(x)$ means the value of X is x; +1 = high; +10 = very high; -1 = low; -10 = very low; 0 = normal level

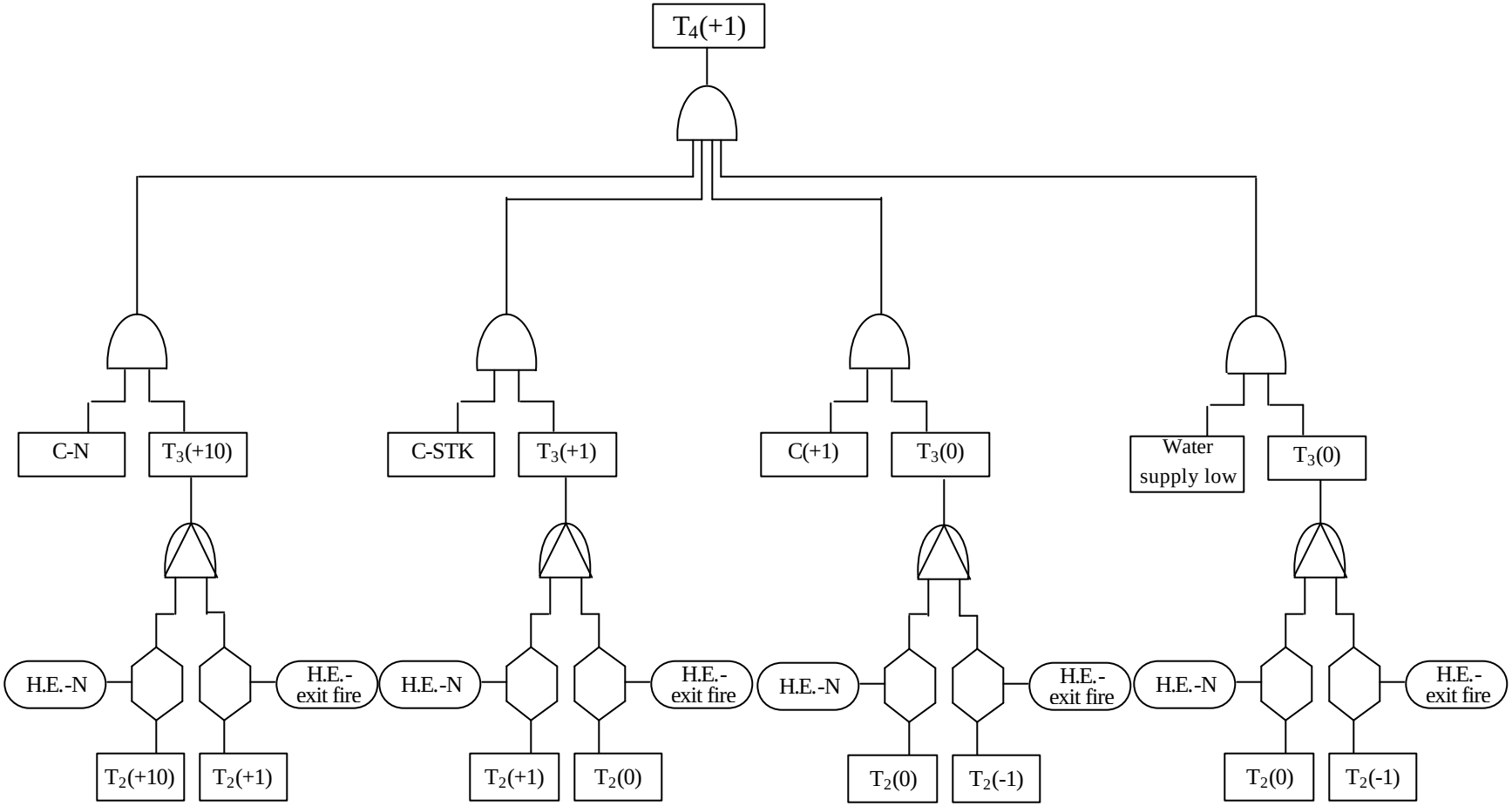

Figure 18. The fault tree for a high temperature of nitric acid reactor feed (Lapp and Powers, 1977)

The Petri net converted from the fault tree in Figure 18 is presented in Figure 19.

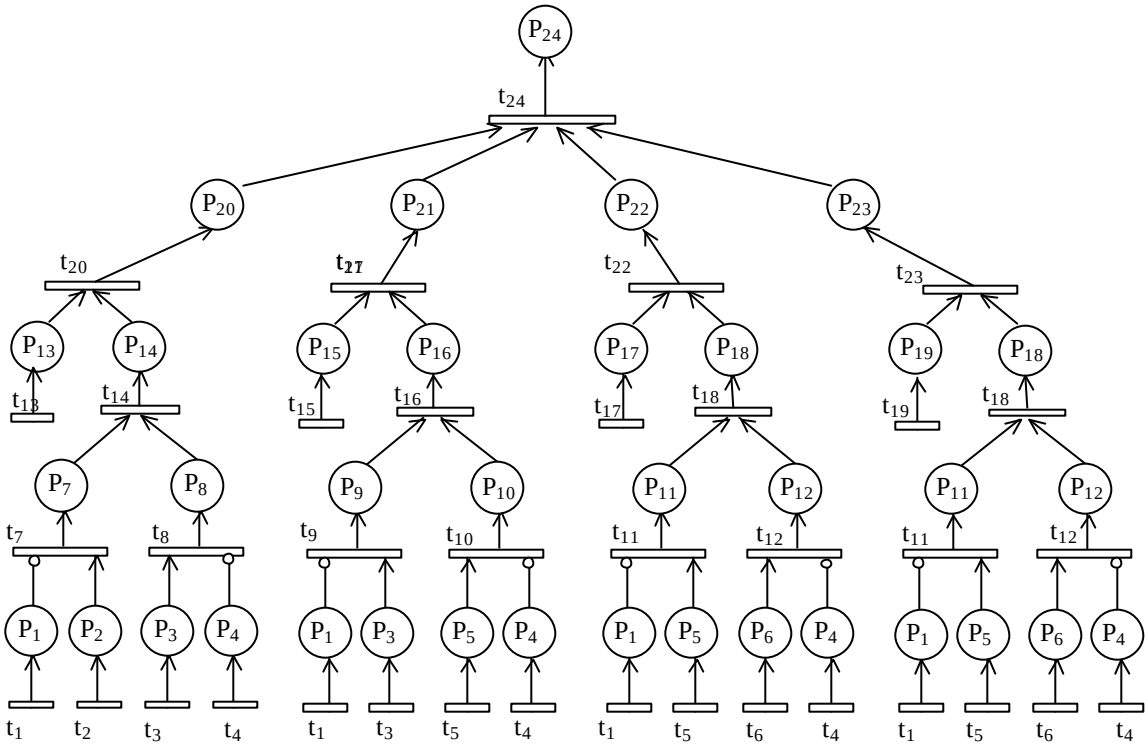

Figure 19. The corresponding Petri net model converted from the fault tree in Figure 18

Based on equations (5) to (15), the average rate of occurrence of the system failures when temperature is above the limits in the nitric acid reactor feed depicted in Figure 18, can be derived in the following. From equation (7), the number of times the transition T_{24} will be fired by time k is computed as:

$$X_{24} = \min\left[X_{20}(k-f_{20}); X_{21}(k-f_{21}); X_{22}(k-f_{22}); X_{23}(k-f_{23})\right]$$

$$X_{20} = \min\left[X_{13}(k-f_{13}); X_{14}(k-f_{14})\right]$$

$$X_{14}(k) = \min\left[X_{7}(k-f_{7}); X_{8}(k-f_{8})\right]$$

Based on equation (14)

$$X_{7}(k) = X_{2}(f_1 - f_2)$$

$$X_{8}(k) = X_{3}(f_4 - f_3)$$

$$X_{21} = \min\left[X_{15}(k-f_{15}); X_{16}(k-f_{16})\right]$$

$$X_{16}(k) = \min\left[X_{9}(k-f_{9}); X_{10}(k-f_{10})\right]$$

$$X_{9}(k) = X_{3}(f_1 - f_3)$$

$$X_{10}(k) = X_{5}(f_4 - f_3)$$

$$X_{22} = \min\left[X_{17}(k - f_{17}); X_{18}(k - f_{18})\right]$$
$$X_{18}(k) = \min\left[X_{11}(k - f_{11}); X_{12}(k - f_{12})\right]$$
$$X_{11}(k) = X_5(f_1 - f_5)$$
$$X_{12}(k) = X_6(f_4 - f_6)$$

$$X_{23} = \min\left[X_{19}(k - f_{19}); X_{18}(k - f_{18})\right]$$

Given the duration of time of basic events, the number of time that transition *i* has been fired by time *k* corresponding to basic event *i* can be computed as:

$$X_1(k) = \frac{k}{f_1},\quad X_2(k) = \frac{k}{f_2},\quad X_3(k) = \frac{k}{f_3},\quad X_4(k) = \frac{k}{f_4},\quad X_5(k) = \frac{k}{f_5},\quad X_6(k) = \frac{k}{f_6},$$

$$X_{13}(k) = \frac{k}{f_{13}},\ X_{15}(k) = \frac{k}{f_{15}},\ X_{17}(k) = \frac{k}{f_{17}}\ X_{19}(k) = \frac{k}{f_{19}}$$

Then the average rate of occurrence of the failure of the high frame contact voltage is:

$$l_{system} = \frac{X_{24}}{k}$$

Note that the average rate of occurrence of the failure computed above is dependent on the time interval *k* unless the system reaches its steady state.

Let us compute the average failure intensity of the system in 90 days (2160 hours) and assume that the firing duration of transitions are given in Table 3. The firing durations of transitions are computed based on the mean times to failures of basic components. Therefore, the failure times of basic events can follow any distribution.

Table 3. Transition and their durations for the example of resistance-grounded AC system.

Transition	Duration	Transition	Duration	Transition	Duration
T_1	2000	T_9	10	T_{17}	2000
T_2	1000	T_{10}	10	T_{18}	1500
T_3	1500	T_{11}	10	T_{19}	1500
T_4	2000	T_{12}	10	T_{20}	10
T_5	1000	T_{13}	2000	T_{21}	10
T_6	1000	T_{14}	10	T_{22}	10
T_7	10	T_{15}	1000	T_{23}	10
T_8	10	T_{16}	10		

The average rate of occurrence of the system failure is computed as:

$$l_{system} = \frac{1}{2160} = 4.63 \times 10^{-4}$$

Here, the illustrative example demonstrated that the counters of Petri net model could be used to model the dynamic behavior of the system failure, whereas fault trees could not. As we showed above, the method allows analyzing large systems and overcoming the state-space explosion since it does not employ Petri net conversion to Markov chain.

5. Conclusions

A variety of methods for system failure analysis exist. Among those methods are reliability block diagram, Monte Carlo simulation, semi-Markov process, failure mode and effect analysis, and fault tree analysis. However, the applications of exiting methods to system failure assessment are limited. For example, for complex systems, an analysis by FTA may produce hundreds of thousands of combinations of events that may cause system failure.

A recent development in system failure analysis is the application of Petri nets. Yang and Liu (1997) presented a Petri net-based failure analysis method. The authors used Petri nets to study dynamic behavior of system failures through marking transfer. The average rate of occurrence of the system failures is computed based on the number of tokens in the place that represents the top event (i.e. system failure). The limitation of the method is the assumption that the tokens in the basic places are generated with the same time between failures. This assumption implies that the basic events have the same time to failure, hence, does not reflect the real life situation. To overcome the limitations of current methods for system failure analysis, the authors introduced new variable r_i that denotes the factor to account for different periods among events and has to be computed for each place separately. In the proposed method, the failure rates of the basic events are not the same, therefore, there is no need for additional variables. This significantly simplified the solution process. Consequently, the proposed method can be applied to more complex Petri net models, for example those with different failure rates, without increasing the computational complexity.

The proposed methods use transitions to identify the sequences of the events in the Petri nets, i.e. the solution is given in terms of the sequence of the firing transitions. Therefore, to be able to compute the probabilities of sequential failures, we need information that is based on the firing rate of a transition rather than on the number of tokens in a place. The proposed method employs counters, the number of times transitions are fired, and hence can be applied for the computation of the probabilities of sequential failures. Other existing methods use markings instead of counters for system failure analysis, therefore, may not be applicable to sequential failure analysis.

We assume the counters to be the state variables of the system while the classical methods use markings as state variables. To be compatible with the classical methods, the presented method allows computation of markings for any state of the system. Since the

number of times each transition is fired is known, computing the marking of the given place is straightforward.

The advantages of the method presented in this paper over the method presented by Yang and Liu (1997) include: 1) fewer variables are needed; 2) less computational effort is required; and 3) the marking at time *t* can be easily retrieved from the counters while the opposite is not always true. Moreover, the method is superior to other existing failure analysis methods since: 1) it provides better understanding of the dynamic behavior of the system; 2) it can handle large systems by avoiding state-space explosion; and 3) it can deal with any distribution of failure times of basic events.

In summary, the novel feature of this paper is that it expands and extends the concept of counters used in Petri net simulation to perform the failure and reliability analysis of complex systems. The presented method allows modeling the system failures using general Petri nets with inhibitor arcs and loops and employs fewer variables than existing marking-based methods and substantially accelerates the computations. It can be applied to real system failure analysis where basic events can have different times to failures.

The method has been illustrated on the example of failure analysis of a nitric acid cooler with temperature feedback and pump-shutdown feedforward loops.

The method can be used as a comprehensive risk assessment method to provide managers with a tool for analyzing hazardous operations for improving safety of the workers and environment, as well as the overall safety of the processes. Data can also be used for helping the industry to meet safety requirements and to improve the efficiency of new manufacturing system implementations. The results obtained can contribute to the safety, environmental, and ergonomic aspects in designing and operating systems.

Acknowledgements

The authors of the paper are grateful to the editor and the anonymous reviewers for making helpful comments and suggestions on the revisions of the paper.

This research has been supported by the research grant EPA 82854101 from the US Environmental Protection Agency (EPA).

References

Adamyan, A. and He, D., 2002, "Analysis of Sequential Failure for Assessment of Reliability and Safety of Manufacturing Systems", *Reliability Engineering and System Safety*, Vol. 76, pp. 227 – 236.

Al-Jaar, R., Y., and Desrochers, A. A., 1990, "Performance Evaluation of Automated Manufacturing Systems Using Generalized Stochastic Petri Nets", *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 6, pp. 621 – 639.

Baccelli, F., and Canales, M., 1991, "Paralleled Simulation of Stochastic Petri Nets Using Recurrence Equations," *ACM Transactions on Modeling and Computer Simulation*, Vol., 3, No. 1, pp. 20-41.

Baccelli, F., Furmento, N. and Gaujal, B., 1995, "Parallel and Distributed Simulation of Free Choice Petri Nets," *Proceedings of Ninth Workshop on Parallel and Distributed Simulation*, France, pp. 3-10.

Bobbio, A., 1988, "System Modeling With Petri Nets," in: Colombo A., Saiz de Bustamante A, editors, "System reliability Assessment", *Proceedings of the ISPRA Course* held in Madrid.

Cheng, Y.-L. and Yuan, J., 2000, "On Structured Fault Tree Construction By Modulating Control Loops," *Reliability Engineering and System Safety*, Vol. 67, pp, 161-173.

Chiola, G., 1987, "A Graphic Petri Net Tool for Performance Analysis", *Proceedings of International Workshop on Modeling Techniques and Performance Evaluation*, France, pp. 323 – 333.

Ciardo, G., 1989, *Manual for the SPNP Package*, Duke University.

Florin, G., Fraize, C., and Natkin, S., 1991, "Stochastic Petri Nets: Properties, Applications, and Tools", *Microelectronics Reliability*, Vol. 31, No. 4, pp. 669 – 697.

He, D. and Adamyan, A., 2001, "An Impact Analysis Methodology for Design of Product and Process for Reliability and Quality", *Proceedings of the 6th Design for Manufacturing Conference*, Pittsburgh, PA, Sept., 2001.

Holliday, M. A., and Vernon, M. K., 1987, "A Generalized Timed Petri Net Model for Performance Analysis", *IEEE Transactions on Software Engineering*, Vol. SE – 13, No. 12, pp. 1297 – 1310.

Hura G. and Atwood J., 1988, "The Use of Petri Nets to Analyze Coherent Fault Trees," IEEE Transactions on Reliability, Vol. 37, No. 5, pp. 469-74.

IEC 50(191), 1990, *International Electrotechnical Vocabulary* (*IEV*), Chapter 191-*Dependability and quality of service*, International Electrotechnical Commission, Geneva.

Jiang, C., Diao, B. and Wu, F., 1995, **"**Study On Reliability Of Manufacturing System Based On Petri Net," *High Technology Letters*, Vol. 1, 2, pp. 25-30.

Kumar, V. and Aggrawal, K., 1993, "Petri Net Modeling and Reliability Evaluation of Distributed Processing Systems," *Reliability Engineering and System Safety*, Vol. 41, No. 2, pp. 167-176.

Lapp, S. A. and Power, G., J., 1977, "Computer Aided Synthesis Of Fault-Trees," *IEEE Transactions on Reliability*, Vol. R-26, No.1, pp. 2-13

Levenson, N. and Stolzy, J., 1987, "Safety Analysis Using Petri Nets," *IEEE Transaction on Software Engineering*, Vol. SE-13, No. 3, pp. 386-397.

Liu, T. S. and Chiou, B. S, 1997, "Application of Petri Nets to Failure Analysis", *Reliability Engineering and System Safety*, Vol. 57, pp. 129-142.

Long, W., Sato, Y., and Horigone, M., 2000*a*, "Quantification of Sequential Failure Logic for Fault Tree Analysis", *Reliability Engineering and System Safety*, Vol. 67, No. 3, pp 269-274.

Molloy, M. K., 1982, "Performance Analysis Using Stochastic Petri Nets", *IEEE Transactions on Computers*, Vol. 3, No. 9, pp. 913 – 917.

Molloy, M. K., 1985, "Discrete Time Stochastic Petri Nets", *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 4, pp. 417 – 423.

Murata, T., 1989, "Petri Nets: Properties, Analysis, and Applications", *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541 - 579.

Ramaswamy and Valavanis, 1994, "Extended Petri Net-Based Modeling, Analysis And Simulation Of An Intelligent Materials Handling System", *Journal of Intelligent and Robotic Systems: Theory & Applications*, Vol. 10, No.1, pp. 79-108.

US Nuclear Regulatory Commission, 1975, "An Assessment of Accident Risk in U.S. Commercial Nuclear Power Plants", *Reactor Safety Study* WASH-1400 (NUREG-75/014), Washington, DC.

Peterson, J. L., 1981, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Englewood Cliffs, NJ.

Viswanadham, N., 1987, *Reliability of Computer and Control System*, New York, Elsevier.

Xiong, H., and He, Y, 1997, "GSPN Based Reliability Modeling And Analysis Of CIMS," *Mechanical Science and Techno*logy, Vol 16, pp. 1103-1106.

Yang, S. and Liu, T, 1997, "Failure Analysis for an Airbag Inflator by Petri Nets, " *Quality and Reliability Engineering International*, Vol. 13, No. 3, pp. 139-151.

Zhou, M. and Zurawski, R., 1995, "Introduction to Petri Nets in Flexible and Agile Automation," in *Petri Nets in Flexible and Agile Automation*, Zhou (Ed), Kluwer, Norwell, MA, pp. 1 – 42.