American Journal of Applied Sciences 2 (11):1547-1551, 2005
ISSN 1546-9239

Failure Diagnosis on Discrete Event Systems

[1]Sihem Kechida and Nasr Eddine Debbache
[1]Laboratory of Automatics and Informatics of Guelma LAIG, Algeria
[2]Department of Electronics, Faculty of Engineering Science, University of Badji Mokhtar, Annaba, Algeria

Abstract: The modern technology advances to a point where it is possible and extensively desirable to improve reliability and the technical process safety. This is achieved by computer implanted FDI procedures (Fault Detection and Isolation). However, the malfunction of actuators, sensors and of the process components, as well as erroneous actions of human operators can have some disastrous consequences in high risk systems such as: Spatial engines (Astronomy), aircrafts (Aviation), nuclear reactors and chemical plants. Thus, each failure or fault can lead to shutdowns or a rupture of service and consequently a plant output reduction. There is an improvement of consciousness and attitude to possible disaster provoked by failures that could enable a failure tolerating system development. Such system must maintain a optimal performance during normal operating conditions and must handle encountered critical situations during which the system's conditions are abnormal that is by performing of detection and diagnosis procedures and reconfiguration according to accurate software programs. In this study, we focus on the diagnosis of the flexible manufacturing systems which are described by a model based on the Petri nets. The basic idea consists of residuals generators resulting from the equation of marking evolution of the process and having appropriated structures to facilitate fault isolation.

Key words: Fault diagnosis, failure detection and isolation, discrete-event systems

INTRODUCTION

Supervision of industrial systems is the task to deduce, from observed variables of a process, if any component is faulty and if so, locate the faulty component. Supervision function objective is to increase productivity by a best scheduling of production tools availability. Principal elements of supervision are detection, location, diagnosis and error treatment.

The basic mechanism used for detection is to compare evolution of observed systems with those of model that opere synchronously. In the same way, the present trend is to found the isolation and diagnosis phases in profound model, which describe the system structure and/or behaviour[1].

System performance and reliability may be measured in terms of the number/frequency of failures occurring within the system over a period of time. Although probable causes of failures may be known before, accurate failure prediction is a difficult task. However, once a failure occurs, the cause of the failure can be identified accurately. A system failure may be classified either as external or internal failure. Depending on its effect on system operations, it can be classified as a soft or hard failure. Failure complexity degree may be judged by the down time of system due to that failure[2].

Failure can be described as a state when the system deviates from its given specification. Normally a reliability system study based on some parameters and some existing reliability model is essential. In many cases, some single reliability measures can be interesting. Such a measure is the mean time between failures (MTBF) or the failure rate of the system. Failure may occur due to two reasons: either due the ‘’ event ‘’ of doing an invalid operation, or by somehow reaching an invalid state operation[3].

PETRI NETS

Brief introduction: Petri Nets (PNs) are a graphical and mathematical modelling tool. The PN representation of a system consists of places and transitions (represented as a circle and rectangle, respectively, in a PN representation), with tokens flowing along the arcs interconnecting them. These tokens are used to simulate the dynamic and concurrent activities within the system. As a mathematical tool, PNs are used to describe the behavior of the system they represent, as state equations and algebraic equations.

A PN, which has all its arcs of weight I, is called an ordinary PN. A PN is called a definite capacity net when its places can hold an infinite number of tokens. When an upper limit to the number of tokens exists then it is called a finite capacity net. A rule with the capacity constraint is called the strict transition rule and without the capacity constraint is called the weak transition rule. More details on different types of PN structures may be found in René[4] and Ghoul[5].

Corresponding Author: Sihem. Kechida, Laboratory of Automatics and Informatics of Guelma LAIG, Algeria

Two types of properties can be studied using a PN model. They are:

* Those that depend on the initial marking, called the behavioral properties. These properties include: reachability, boundeness, liveness, reversibility, etc. and
* Those that are independent of the initial marking, called the structural properties.

Definition: A PN is a particular kind of directed graph, it is five-tuple < P, T, Pre, Post, M_0 > such that[6]:

P: is a finite and non empty set of places.
T: is a finite and non empty set of transitions.
Pre is an input function, representing weighted arcs connecting places to transitions called pre condition matrix of size (n, m).
Post is an output function, representing weighted arcs connecting transitions to places called post condition matrix of size (n, m).
M_0 is an initial marking

If a firing sequence D is applied to the marking M then the reached marking M' is given by the fundamental equation:

$$M(k+1) = M(k) + Post.D - Pre.D \qquad (1)$$

$$= M + C.\ D \qquad (2)$$

For $M \geq 0$; $D \geq 0$; and C is called incidence matrix.

Discrete event systems: The PNs are one of the useful for describing discrete event systems (DES). Indeed, they are well adapted for depicted dynamic behavior of the system. The des is a dynamic system defined by a discrete state space and an evolution based on succession of states and transitions. Transitions are associated to the set of events.

APPROACH FDI BASED ON THE PNS

From the PN system modeling point of view, FDI is a model-based approach. The difference between the FDI approach based on PNs and the above model-based approaches is that unlike PNs, the other approaches are not suitable for simulating the dynamic system behavior. Moreover, PNs inherently capture the various asynchronous, sequential and parallel interactions between the various system resources and operations with great ease. In Ramaswanny[7] PNs are shown to be useful for the detection of abnormal process behavior, or for the measurement of faults with very low time constants (faults with a low time constant will change the measurement signals to a minimum extent over a time period and therefore, would be undetectable) during the real-time monitoring of power plant systems.

Recently, searchers use reverse PNs to establish minimum and maximum times for applying PNs to fault detection/monitoring in automated systems.

The following reasons, therefore, justify the applicability of PNs for the modeling and analysis of systems with integrated error recovery:

* The marking of a PN represents the system state. Transitions result in state changes and they trigger a change in the PN marking corresponding to the occurred state transition.
* On-line error detection and recovery can be modeled and analyzed based on a-priori knowledge of the occurred/anticipated errors. Off-line error detection, isolation, identification and learning is greatly simplified with a PN system model, because the state of the system during the occurrence of a failure is preserved by the PN marking.
* A PN simulation can easily capture the dynamic system behavior and in addition they can be a highly useful interactive graphical tool in the modeling, analysis and performance evaluation of systems.

More specifically they are suitable in the design, analysis and real-time control of autonomous systems.

Failure classification: Flexibility and autonomy are two important system characteristics. Flexibility is related to the ability to adapt to new applications and may involve re–configuration of components and reprogramming. Autonomy refers to the ability of the system to make decisions according to external asynchronous events (asynchronous events are those that are beyond system control and are unexpected during the normal system operation), which may occur during a task execution. Failure recovery is necessary when a system failure occurs.

Failures are classified as follows.

External failure: it occurs because of external factors to the system. They are caused by human operator errors, power malfunction, dynamic changes in the workspace environment, etc…

Internal failure: It occurs due to factors directly associated with the system and may be classified as either hardware or software failure. Internal failure may also occur as a consequence of external factors, but they are accommodated as any other internal failure.

Function classification of supervision: In a process, we can separate products, production means and operations. For each part of the process can be associated a supervision method. The approaches can be moreover quantitative that qualitative.

At quantitative level, we compare realized product rates with planned ones. Any significant deviation can be a sign of one of several faults in the process evolution.

This technique can be used at different levels in supervision of flexible manufacturing system.

Products quality is also an important indicator of production failure. In manufacturing systems, this approach is based on metrological instrumentation.

Production means: It's a direct supervision of production tools, based on dedicated sensors (speed, temperature…).

Operation: the method consists in checking if each operation of the manufacturing process is being executed correctly.

Separate supervision method: For the monitoring systems based on PNs, the calls of the diagnosis systems are all done starting from two follows principles:

* It's impossible to fire a transition whose token quantity in its input places is not sufficient or if its spending time is not sufficient.
* In a given situation, it's impossible to take into account a given event after certain waiting (watchdog technique)

Te use of the watchdog technical is a simple mechanism, making it possible to easily detect the absent reports.

The solution is to introduce duration or an activation time to each transition. In the example presented in Fig. 1, it's enough to associate an activation time T to the transition, so that T has the same value of the watchdog Fig. 1.

In this paragraph, we describe a module intervening in supervisor (monitor) design, which its modelization is based on PN representation. The latter permit to modelize and visualize parallelism and synchronization behavior and resources share which characterize usually DES. Its evolution is controlled by integrated but separate supervisor[8].The representation corresponding to such model is given by Fig. 2.

Detection and diagnosis are integrated into the process model. A separate module represents them. The diagnosis is started automatically on detection of a failure. The two failure modes which can affect a DES are the following:

* If an activity is faulty, it is one of the necessary conditions to its realization was not good or one of the pre conditions was false (failure propagation). The conditions and pre conditions are related to the components of process: products, tools, production and transport resources.
* If an activity has not started again/has not acquitted because the pre conditions/post conditions were not executed.

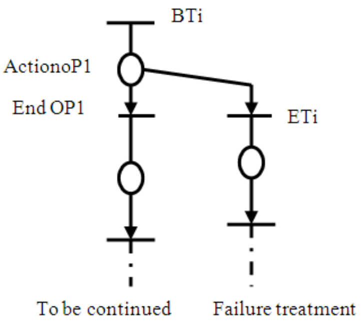


Fig. 1: Scheme of watchdog technique

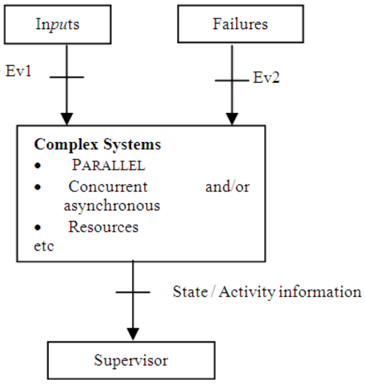


Fig. 2: Failure supervision in DES

Supervision system design is based on a PN model, which allows the additional places in order to supervise the dynamic sequencing of operation. The general structure is illustrated by Fig. 4.

If $M_s(k)$ is the marking vector, the evolution marking of supervisor must respect the famous relation defined by equation 2 and may be rewritten as follow:

$$M_s(k+1) = M_s(k) + C_s. D_s \qquad (3)$$

Where M_s, C_s and D_s have the same definition as for M, C and D.

Proposed algorithm: In this part, we develop an algorithm which detect and isolate failure transition in DES.

Step 1: Lets n number of places representing DES and s is the one of supervisor.

An enforce invariant condition is given by this equation:

$$M_s(k) = Q.M(k) \qquad (4)$$

where $M(k)$ and $M_s(k)$ are respectively marking vector of original model and supervisor.

And Q is a matrix with an appropriate dimension in N.

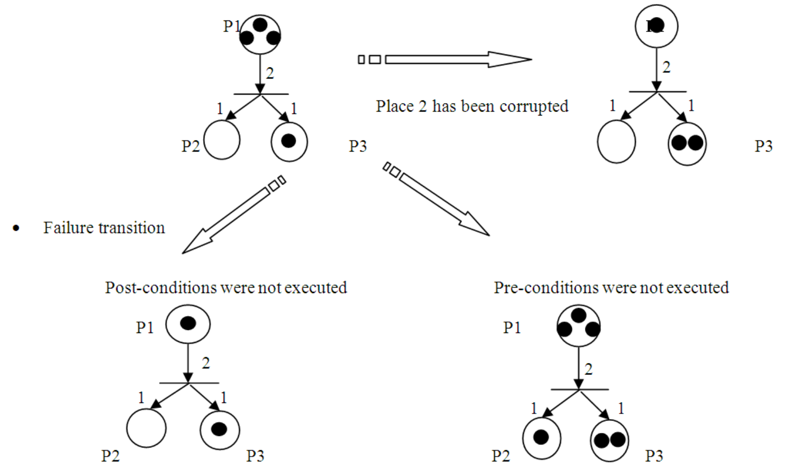

Failure place
P1
2
1
1
P2
P3
Place 2 has been corrupted
2
1
1
P3
Failure transition
Post-conditions were not executed
P1
2
1
1
P2
P3
Pre-conditions were not executed
P1
2
1
1
P2
P3

Fig. 3: Failures modelling in PNs

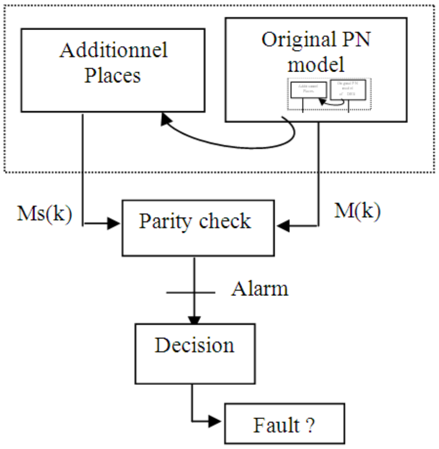


Fig. 4: Supervision scheme using separate PNS

Step 2: Define a new marking vector for global model noted $M_h(k)$:

$$M_h(k)=\begin{pmatrix}M(k)\\ M_s(k)\end{pmatrix}=\begin{pmatrix}I_n\\ Q\end{pmatrix}.M(k) \qquad (5)$$

with I_n is an identity matrix of n dimension.

Step 3: Fundamental marking evolution equation deduced of (2) is:

$$M_h(k+1)=M_h(k)+\begin{pmatrix}I_n\\ Q\end{pmatrix}C.D \qquad (6)$$

Step 4: Syndrome generation.
Define a matrix P as follows:

$$P = [-Q \quad I] \qquad (7)$$

Generate a vector S(k):

$$\begin{aligned}S(k) &= P.\,M_h(k)\\ &= [\,-Q \quad I\,]\,M_h(k)\end{aligned} \qquad (8)$$

that verify the following hypothesis:

$$\begin{cases} S(k)=0 \text{ no failure}\\ S(k)\neq 0 \text{ presence of failure}\end{cases} \qquad (9)$$

Step 5: Decision.
Determine the vector V defined from S(k):

If V(j) = -1 : pre-condition failure for transition Tj
If V(j) =+1 : post-condition failure for transition Tj

Remark: Detection and identification of multiple failures are dependent on information contained in vector S(k). However, an adequate choice of Q facilitates this task. In fact, Q is a nonnegative matrix with integer entries.

SIMULATION AND RESULTS

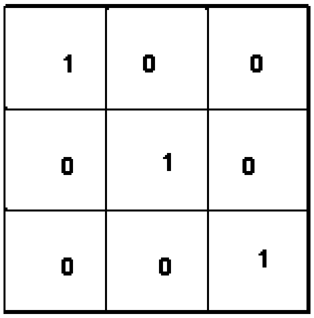
As the first test, we consider PN model formed by three places and three transitions which can represent a physical process i.e. filling and emptying of tank. In a manufacturing system, PN model must verify some proprieties like safeness, liveness and boundness.

The initial marking and incidence matrix are given by: M0 =[1 0 0]^T and

Incidence Matrix C =

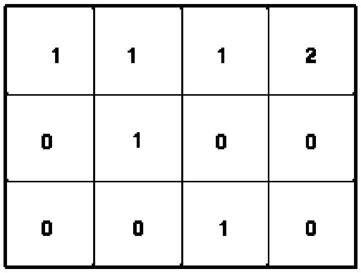
	T1	T2	T3
P1	-1.00000	0	1.00000
P2	1.00000	-1.00000	0
P3	0	1.00000	-1.00000

When fired transition is given in order: $T_1T_2T_3$ and we lead to this transition graph.



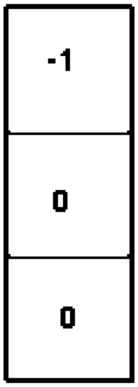
1	0	0
0	1	0
0	0	1

Fig. 5: Transition graph



1	1	1	2
0	1	0	0
0	0	1	0

Fig. 6: Marking graph



-1
0
0

Fig. 7: Decision vector

Parity vector S and decision vector V are equal to zero. When a failure occurs in transition T_1, we can show it by a marking $M_d = [1\ 1\ 0]^T$.

From simulation, we obtain S (means failure appearance) and V(=-1 a failure on pre-condition of T_1)

This example can represent a subnet of complex DES; however we can generalize this technique to detect failures on an FMS.

CONCLUSION

In this study, we have presented a new approach to analyze a DES using PN. Failures are a part of any system. The reason for analyzing the failures in a system is to expedite the repair process and hence improve the productivity of the whole production system.

Some general strategies may be proposed to deal with failures that occur in a DES into different categories. Moreover, identifying and rectifying failures is greatly simplified, when a thorough study and analysis of the failure has been done previously.

Thus, to detect we have chosen Q as a Pascal matrix and to isolate useful information is contained in decision vector. A simulated program has been developed in Matlab and obtained results were satisfied.

As a perspective, we can reduce dimension of Q enforcing a new criteria and we can extend this study to complex systems with shared resources and parallel process.

ACKNOWLEDGMENTS

The authors are grateful to Doctor Rachida Hadiby Ghoul for fruitful discussions

REFERENCES

1. Valette, R., 2000. Les RdP pour la détection et diagnostic .LAAS-CNRS Toulouse.
2. Sihem, K., D. Nasreddine, G.H. Rachida and T. Hicham, 2004. Residual generation synthesis for fault detection and isolation. WSEAS (World Scientific and Engineering Academy and Society).
3. Sihem, K., R.H. Ghoul and S. Hassainia, 2005. Failure diagnosis on flexible manufacturing systems modeled by Petri Nets. Al Azhar Univ. J., 8: 3.
4. David, R, and H. Alla, 1997. Du Grafcet aux réseaux de Petri Hermes.
5. Rachida, G.H., 2003. Modélisation et conduite des systèmes de production flexible par les réseaux de Pétri. Thèse de Doctorat d'état en automatique productique, Annaba, Algeria.
6. Hanzalek, Z., 1997. Algorithmes parallèles pour la commande distribuée. Une approche par réseaux de Pétri. Thèse de Doctorat en informatique industrielle U de Toulouse.
7. Ramaswanny, S., 1994. Hierarchical Time extended Petri nets (H E P Ns) for integrated control and diagnostics of multi level systems. Springs.
8. Wu, Y. and C. Hadjicostic, 2002. Failure identification in discrete event systems using encoded Petri Net states. Coordinate Science Laboratory and Department of Electrical and Computer Engineering. University of Illinois at Urbana-Champaign.