Algerian Journal of Research and Technology

ISSN : 2543-3954

A.J.R.T Volume 6 (N° 1) (2022) 10 – 18

Selection and peer-review under responsibility of the scientific committee of The 1st International Symposium on INDUSTRIAL ENGINEERING, MAINTENANCE AND SAFETY IEMS'22.

Diagnostic by Fault Tree and Petri Nets of a Robotic Machining Cell

Z. Mehar[1*], R. Noureddine[2], F. Noureddine[3]

[1]Production and Maintenance Engineering Laboratory, Institute of Maintenance and Industrial Security, University of Oran 2 Mohamed Ben Ahmed, B.P 1015 El M'naouer,Oran, 31000, Algeria
[2]Production and Maintenance Engineering Laboratory, Institute of Maintenance and Industrial Security, University of Oran 2 Mohamed Ben Ahmed, B.P 1015 El M'naouer,Oran, 31000, Algeria
[3]Production Engineering Laboratory, National School of Engineering in Tarbes, National Polytechnic Institute of Toulouse, 47 avenue d'Azereix- BP 1629 - 65016 Tarbes CEDEX, France

**Corresponding author :E-mail:* *E-mail: mehar.zohra@univ-oran2.dz

Abstract. The objective of this work is to propose a diagnostic technical method for robot cell machining. The developed approach is based on a hybrid diagnostic FT-PN (Fault Tree - Petri Nets). This technique consists of the Petri Nets implementation in the LabView environment (Laboratory Virtual Instrument Engineering Workbench), based on the knowledge of the Fault Tree of the robot cell. The simulations and results are obtained from the state of the operating system without and with fault are presented and discussed.

Keywords. Diagnostic, Robotisation, FT, PN, LabView.

INTRODUCTION

In Industry 4.0, robotization has been imposed as an element of primary factor. The current performance of robots in terms of precision tracking trajectories allows us to consider robotization in machining and more specifically in operations such as surfacing, drilling, trimming, sawing, deburring, grinding, and polishing. Like any technological system, maintenance is mandatory and necessary to guarantee performance. In the case of failures, the minimization of the restart time is essential so as not to penalize the productivity of the cell concerned.

It is in this context we propose a diagnostic system based on Petri nets implemented from a fault tree.

The objective of the diagnostic system is to search for the causes and locate the organs that caused a particular anomaly which consist of two elementary functions: localization and identification, (Isermann and Ballé, 1997; Mahdaoui, 2012).
In the proposed approach, the diagnostic system input (Fig. 1) shows the external inputs of the qualitative or quantitative type that the operator can add to improve the diagnostic. As output, we will find the different possible causes associated with a degree of credibility and a degree of gravity for each of them. These degrees will help the manager now to evaluate and plan maintenance actions, (Palluat, 2006; Palluat et al., 2004).
In this work, we present a diagnostic method based on a deductive analysis from a Fault Tree of a robot cell machining followed by an implementation of a Petri Nets (PN).
In the following section, we present the Petri Nets formalism. It follows the study of the studied case (robotic cell), where the implementation of the FT in a PN in the LabView environment is realized. Finally, the simulations and results are presented.

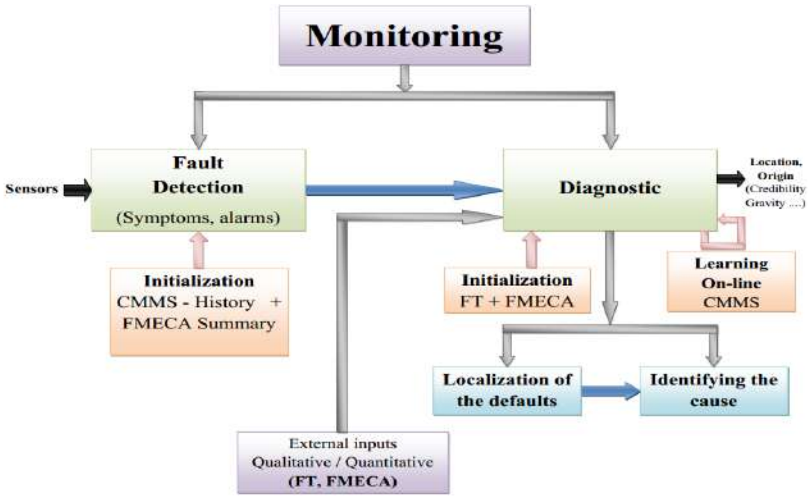

Fig.1. Diagnostic system.

PETRI NETS MODELING

Petri nets (PN) are graphical and mathematical tools (Brauer et al., 1985). They can describe existing relations between conditions and events, to model the behavior of dynamic systems with discrete events. This tool allows qualitative and quantitative analysis (Petri, 1962; Morère, 2002).
The Petri nets (PN) are composed of a set of places, a set of transitions, and a set of arcs that associate the input places with the transitions and transitions to output places weight integers associated with the arcs (Boucherit, 2019; Haiouni, 2010). The state of a Petri network is defined by its marking which associates with each place a positive or zero finite number, and is graphically represents by marks or tokens, (Petri, 1962; Morère, 2002; Cassez and Roux, 2003). There are different types of Petri nets (David and Alla, 1992): timed, interpreted, stochastic, colored, continuous, and hybrid.
The use of knowledge from a Fault Tree with the Petri Nets formalism through the reasoning must be done qualitatively but also quantitatively to provide effective analysis, like the case in industrial applications, (Palluat, 2006; Racoceanu, 2006). We, therefore, use the formalism that associates with each gate of the FT as a symbolization of the PN. So, the logical gate « OR » and the logical gate « AND » will be transformed according to figure 2 and 3:

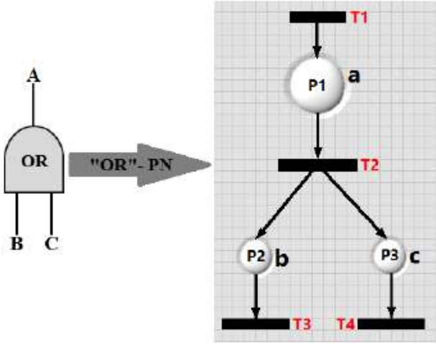

Fig.2. Transformation of the « OR » logical gate of the FT into the PN.

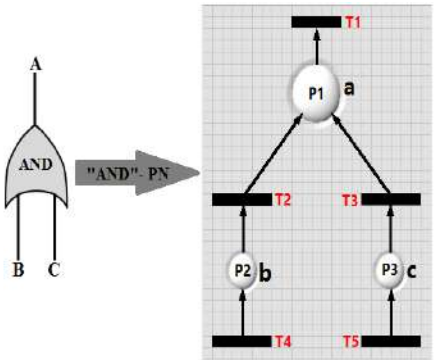

Fig.3. Transformation of the « AND » logical gate of the FT into the PN.

Case study - Robotic cell

The object of our study is the advanced robotic cell machining of the Production Engineering Laboratory of the National School of Engineering in Tarbes. This cell shown in figure 4 comprises:

- KUKA KR120 robot, (b),
- Pneumatic grinder, (c),
- Tool adapted to the task to be performed, (d),
- Clamped piece in a vise arranged on a grooved table, (e),
- The vision system camera (National Instruments Image Processing Software), (f).

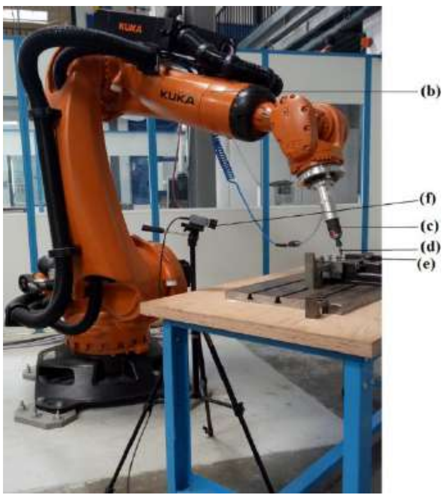

Fig.4. Robotic machining cell.

Analysis by Fault Tree

The FT is a method of deductive analysis used also in dependability, (Ghostine, 2008), (Noureddine et al., 2005). This is a method of analyzing the reliability, availability, and security of more widely used systems, (Ruijters et al., 2019; Dutuit et al., 2018). The diagnostic system we are developing they based on the learning base initiated from the Fault Tree (FT) of the robotic cell considered. This FT will be the centerpiece of our PN-based strategy and they are presented in figure 5.

The analysis and research of the dreaded event in our FT highlight the non-conformity of the workpiece (a) in the robotic cell.

CABTREE software we used to build and process our fault trees. We have limited our study to two levels, which show the first elementary elements, as shown in figure 5:

- First level: Defective KUKA KR 120 Robot, (b); Faulty Grinder, (c); Faulty Tool, (d); Failing Piece, (e); Faulty Vision System Control (Camera), (f).
- Second level: Control Cabinet of a KUKA KR 120, (b1); Articulated Mechanical System, (b2); Wrong couple, (c1) Total failure, (c2); Break on the tool, (d1); Bad sharpening, (d2); Bad positioning of the piece, (e1); Non-conforming characteristics of the blank, (e2); Blurred Vision (False Results), (f1); Wrong Camera setting, Wrong treatments, (f2).

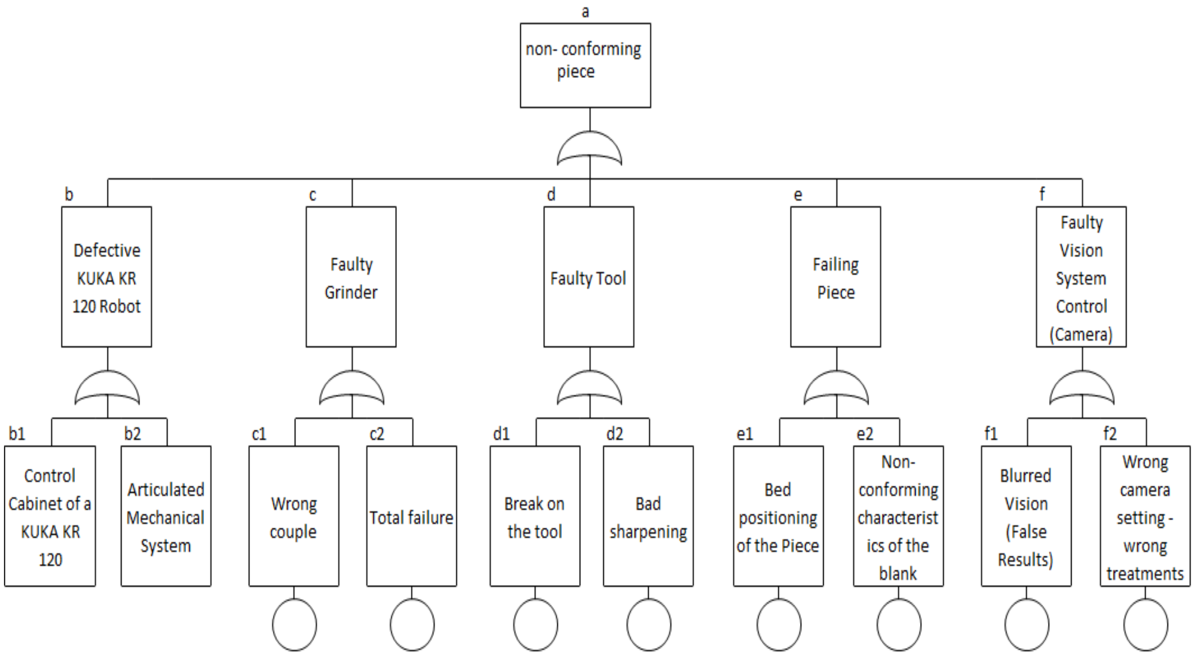

Fig.5. Fault Tree of the robot cell.

Modeling FT-PN

We have applied our fault tree transformation technique (FT) in a Petri net (PN); we used the LabView software platform for modeling and simulation. LabView is based on a graphical development environment of «National Instruments», and it is used mainly for instrument control and industrial automation (Decourt and Dordor, 2003; Frey et al., 2007; Sagar and Narayana, 2014).

We have implemented our PN into a LabView state machine; the structure obtained they shown in figures 6 and 9. These figures represent the front panel (the user interface of the virtual instruments, in our system, figures 6 and 9, of our PN that contains 16 places that are circular LEDs(Light Emitting Diodes) and 26 transitions. Each place presents an event of our FT and contains their status: inputs (commands) and outputs (indicators) of the program. The controls and indicators are materialized by digital displays for 0 or 1.

A digital graph displays the data as impulses, a binary number generator that will simulate the marking test (if there is a mark in a place, the signal displays the state 1 else the state 0). The X-axis is the state scale (places) and the Y-axis of the markings.

Our PN is limited to 1, alive, not resettable, not repetitive, graph of infinite markings. The initial marking, corresponding to figure 6, is m_0 = [0000000000000000].

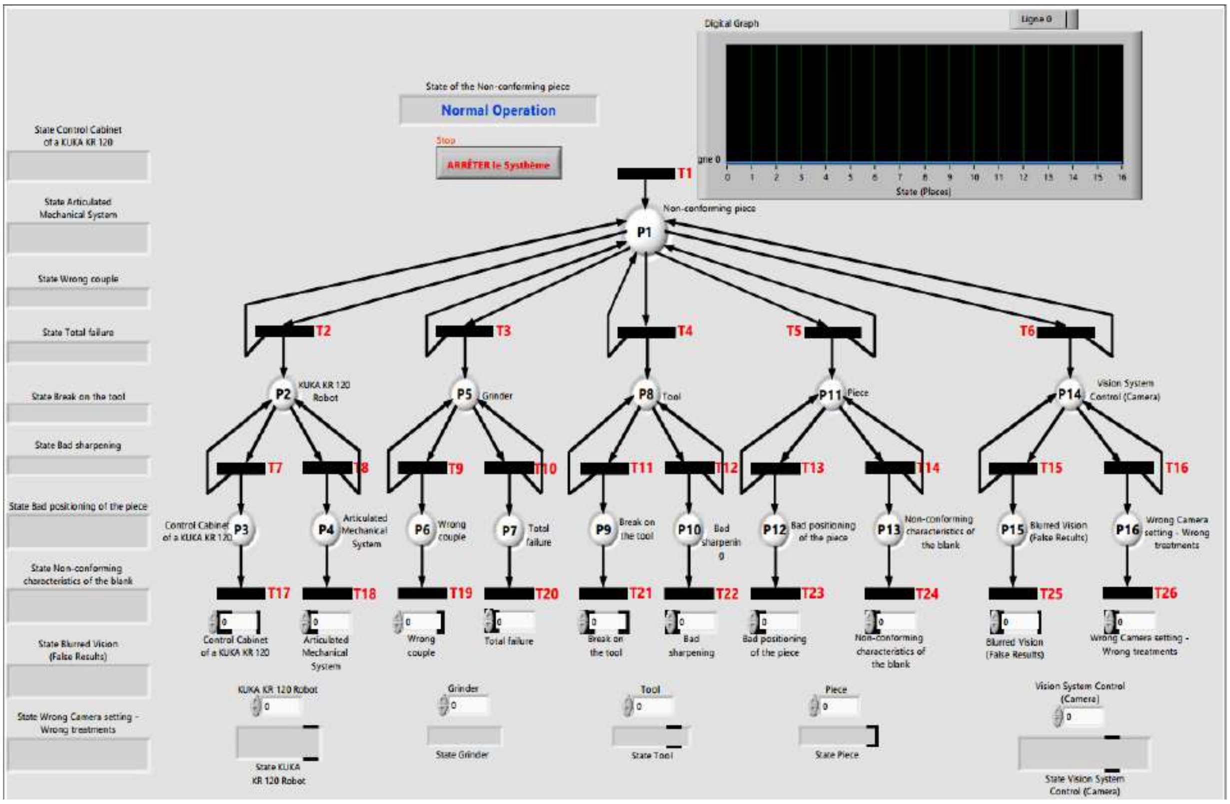

Fig.6. The front panel under LabView of the PN system - Modeling without fault

Block diagram, figure 7, represents a part of the application program developed in the form of a data flow diagram.

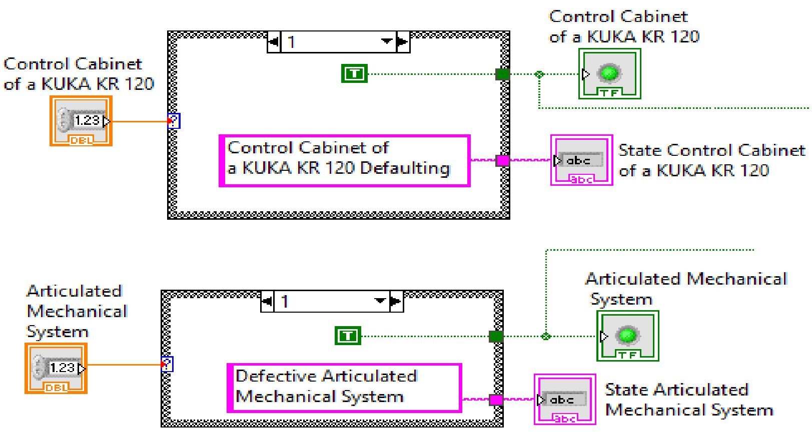

Fig.7. The block diagram of the LabView - part of the PN system.

The following figure (Fig. 8), illustrates how commands and indicators materialized by digital displays for a state 0 or 1, in the block diagram of the LabView.

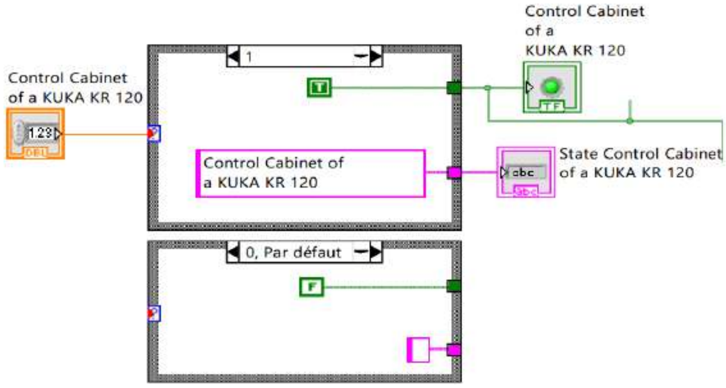

Fig.8. The block diagram of the LabView – is an example of a command.

The transitions (T1, T2, ..., T26) of our Petri net system are materialized by sensors. Table 1 describes the significance of each place.

Table 1. Significance of each place

Places	The significance of each place
P1	Non-conforming piece
P2	KUKA KR 120 Robot
P3	Control Cabinet of a KUKA KR 120
P4	Articulated Mechanical System
P5	Grinder
P6	Wrong couple
P7	Total failure
P8	Tool
P9	Break on the tool
P10	Bad sharpening
P11	Piece
P12	Bad positioning of the piece
P13	Non-conforming characteristics of the blank
P14	Vision System Control (Camera)
P15	Blurred Vision (False Results)
P16	Wrong Camera setting, Wrong treatments

Simulations and results

Simulations and results obtained are illustrated in figure 6. The corresponding marking is m_1 = [1100000000110000]; the state of the non-conforming piece is in operation with fault, the KUKA KR 120 robot is faulty in the first level of the FT, and the bad positioning of the piece, in the second level.

- The digital indicators of the KUKA KR 120 robot and the bad positioning of the piece are in 1: in « State KUKA KR 120 Robot » mark « Defective KUKA KR 120 Robot »; in « State Bad positioning of the piece » mark « Bad positioning of the piece ».

- So respectively, the place (P2) and the place (P12) are colored gray, and their signal is in state 1.

- If the place (P2) is faulty then (P1) faulty; If the place (P12) is faulty then (P11) is initially faulty and then (P1) is also faulty: If there is a faulty place, the triggered diagnostic process makes the failed system.

To do a deductive analysis for our PN, we proceed by crossing the transition (T1), and then the place (P1) has a token, if there is a token in the place (P1) then cross the transitions (T2) and (T5) directly. If (T2) is crossed then the places (P2) and (P1) have a token for each. Thus crossing (T5), the places (P11) and (P1) have a token for each. If the place (P11) has a token, then (P12) has a token also after crossing the transition (T13): This diagnostic process is obtained through the return arcs constituting our Petri net.

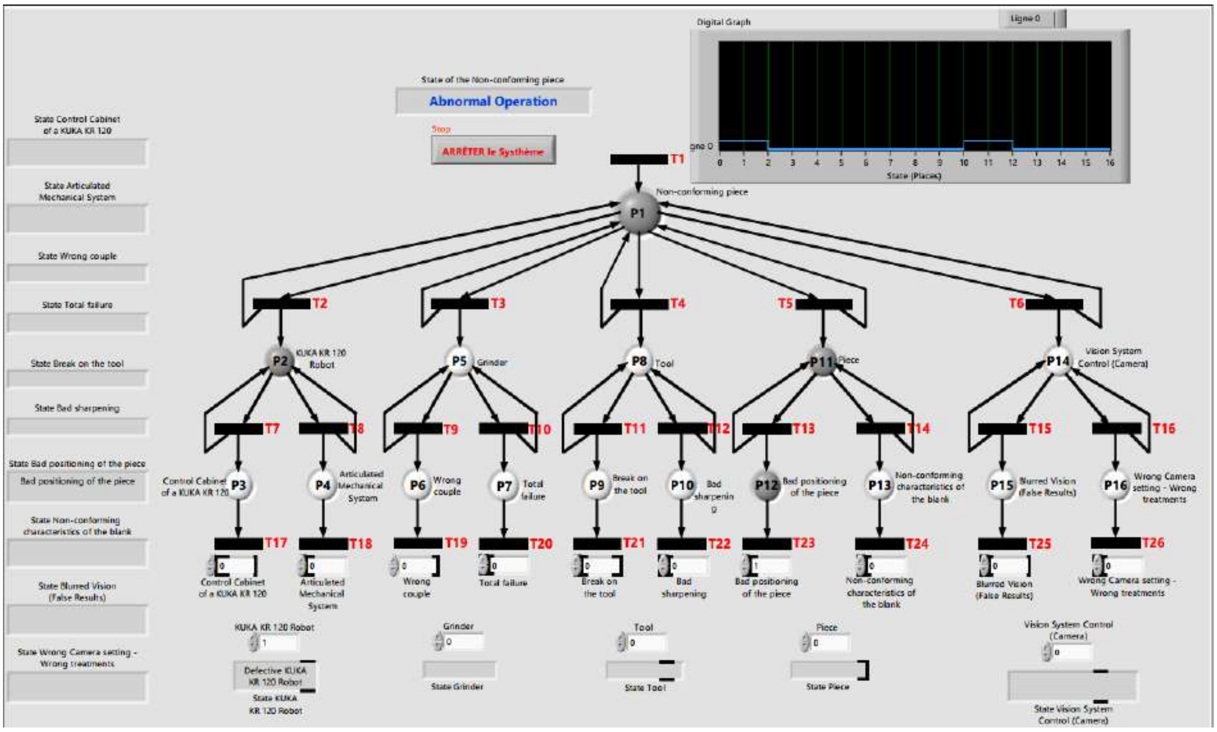

Fig.9. The front panel under LabView of the PN system - Modeling with fault.

CONCLUSION

This work presents the development of a diagnostic strategy for a robot cell machining using a Petri net generated from the realized knowledge of an FT on this cell, in the LabView environment. This first step, in the context of performing a diagnostic process on the cell, shows encouraging results.

In the following of this work, we plan to quantify the PN through the integration of failure rates in the FT on the one hand and, on the other hand, to develop an interface in the environment of Solid Works to concretize the proposed approach on the robot cell machining.

REFERENCES

Boucherit, A. (2019). Contribution à la conception d'architecture logicielle sur des systèmes critiques basés agent (Doctoral dissertation).

Brauer, W., Rozenberg, G., & Salomaa, A. (1985). Petri Nets-An Introduction. Ëpringer-Verlag, Berlin Heidelberg.

David, R., & Alla, H. L. (1992). Du Grafcet aux réseaux de Petri. Hermes.

Decourt, R., & Dordor, P. (2003). Introduction à LabVIEW.

Dutuit, Y., Signoret, J. P., Folleau, C., & Thomas, P. (2018, October). Détermination et calcul de la fréquence d'occurrence d'une coupe minimale-Un bref état de l'art. In Congrès Lambda Mu 21 «Maîtrise des risques et transformation numérique: opportunités et menaces».

Cassez, F., & Roux, O. H. (2003, October). Traduction structurelle des réseaux de petri temporels vers les automates temporisés. In Hermès Science, éditeur: 4ème Colloque Francophone sur la modélisation des Systèmes Réactifs (MSR 03) (Vol. 54).

Frey, D., Degryse, P. A., & Amalberti, J. L. (2007). Le langage de programmation Labview. IUT1 Grenoble Département GEII1.
Ghostine, R. (2008). Influence des fautes transitoires sur la fiabilité d'un système commandé en réseau (Doctoral dissertation, Institut National Polytechnique de Lorraine-INPL).
Haiouni, H. (2010). Approche mixte de modélisation par les réseaux de Petri et SMA.
Isermann, R., & Balle, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. Control engineering practice, 5(5), 709-719.
Mahdaoui, R. (2012). Contribution à la surveillance dynamique des systèmes de production évolutifs par les systèmes Neuro-Flous Temporels (Doctoral dissertation, Batna).
Morère, Y. (2002). Cours de réseau de Petri.
Noureddine, R., Benamar, A., & Noureddine, F. (2005). Intégration d'une technique de maintenance prédictive de l'outil de coupe dans une cellule d'usinage robotisée. In Conférence internationale sur la maintenance (PENTOM 2005).
Palluat, N., Racoceanu, D., & Zerhouni, N. (2004). Diagnosis aid system using a neuro-fuzzy approach. Advances in Maintenance and Modeling, Simulation and Intelligent Monitoring of Degradation.
Palluat, N. (2006). Méthodologie de surveillance dynamique à l'aide des réseaux neuro-flous temporels (Doctoral dissertation, Université de Franche-Comté).
Petri, C. A. (1962). Kommunikation mit automaten.
Racoceanu, D. (2006). Contribution à la Surveillance des Systèmes de Production en Utilisant l'Intelligence Artificielle (Doctoral dissertation, Université de Franche-Comté).
Ruijters, E., Budde, C. E., Nakhaee, M. C., Stoelinga, M. I. A., Bucur, D., Hiemstra, D., & Schivo, S. (2019). FFORT: a benchmark suite for fault tree analysis.
Sagar, S. C., & Narayana, M. (2014). Ambulance Controlled Traffic System Using RFID Technology with LabVIEW Simulation. International Journal of Engineering and Computer Science, 3(5), 6048-6052.