

Modeling and Analysis of a Machining Robot Cell with λ PN

Mehar Z¹, Nouredine R², Nouredine F³

^{1,2}Lab of Industrial Production and Maintenance Engineering, Institute of Maintenance and Industrial Safety, University of Oran2 Mohamed Ben Ahmed; P.B 1015 El M'naouer 31000 Oran Algeria

E-mail: mehar.zohra@univ-oran2.dz, nouredine.rachid@univ-oran2.dz.

³Production Engineering Laboratory, National School of Engineering in Tarbes, National Polytechnic Institute of Toulouse France

E-mail: farid.nouredine@enit.fr.

Abstract: This paper proposes a new approach for modeling and analyzing faults within a machining robot cell. Therefore we decided to use a hybrid analysis associating a Petri Net (PN) to a Fault Tree, resulting in a new technique called Lambda Petri Net (λ PN). This work has been implemented in the LabView environment (Laboratory Virtual Instrument Engineering Workbench). Lambda Petri Net showed its modeling power for the developed monitoring system. Lambda Petri Nets in fault analysis allow natural language descriptions of process entities. A graphical method is used to describe the relationships between conditions and events. Mathematical generic properties have been used to validate our whole research technique. The simulations and results obtained from the state of the operating system without and with fault are presented and discussed.

Keywords: Petri Net Lambda, Monitoring, Machining Robot Cell, Modeling, Fault Tree.

1. INTRODUCTION

The multiple reconfigurations and the complexity of modern production systems, such as machining robot cells, lead to the design of increasingly efficient monitoring support systems. Considering a robotic cell, we have been interested in modeling the changes in the system dynamics when one or more faults occur.

Figure 1 summarizes the monitoring components and focuses on our contribution to the diagnostic system of the machining robot cell with qualitative external inputs (Fault Tree). The output we will find the different possible causes associated with a fault location. The output will obtain the different possible causes associated with a fault location and degree of credibility and degree of severity for each cause. These degrees will help managers to evaluate and plan maintenance actions.

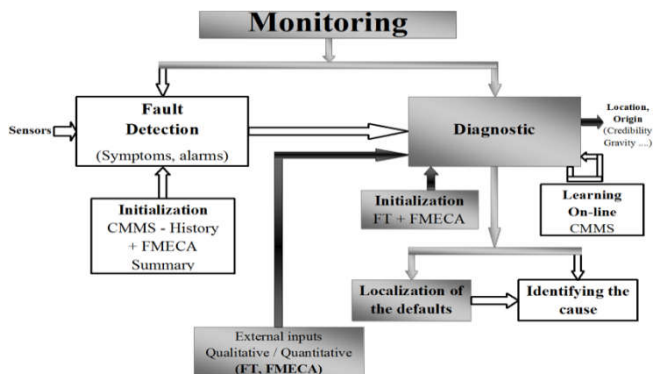


Figure 1. Block diagram of the monitoring system

2. λ PN FORMALIZATION

According to (David & Alla, 1992b), different PN have particular structures, i.e. they have characteristics and properties that most common networks do not have. State graph, event graph, conflict-free PN, free-choice PN, basic PN, pure PN, self-loop-free PN, Generalized PN, Capacity PN, Autonomous PN, Non-autonomous PN. There are also different types of high-level Petri Nets: synchronized, timed, interpreted, stochastic, colored, hierarchical, continuous, and hybrid.

We propose a Lambda extension of Petri Nets (λ PN) specific to the modeling and analysis of system monitoring activities. The uncertain knowledge associated with these activities requires specific reasoning and modeling methods adapted to the various failure rates.

In cooperation with the ordinary Petri Net and Capacity PN, which model the system to be monitored, this new tool makes it possible to carry out a complete diagnosis of the faults locations and degradations of the system. The lambda Petri Net approach provides more detailed information about the operating state of the monitored system.

Generally speaking, mathematical properties of a PN are: (Diaz, 2009)

- 1) A Petri Net (PN) is a couple $\{R, M\}$ where: R is a PN, denoted by a quadruple $R = \{P, T, f, M_0\}$ (1)
- ✓ M is an application from P to \mathbb{N} . M(p) equals the number of marks in a place $p \in P$.

- ✓ $F: (P \times T) \cup (T \times P) \rightarrow N_0$ Defines the set of directed arcs weighted by non-negative integer values.
- 2) The incidence matrix (W) of the Petri Net translates the global cost of firing a transition for each place.
- 3) Denoted by: $W = W^+ - W^-$ or $W = Post - Pre$ (2)
The (i, j) element of matrix W gives the balance for a place i of the firing of the transition j .
- ✓ The connecting arcs of Transitions to Places $Pre (P_j, T_j)$ can be represented in a matrix with
$$W^+(i, j) = Pre(P_i, T_j)$$
- ✓ The connecting arcs of Places to Transitions $Post (P_i, T_j)$ can be represented in a matrix with
$$W^-(i, j) = Post(P_i, T_j)$$
- 4) The marking vector of the Petri Net is constructed by the characteristic vector of the sequence \underline{S} which is formed by the number of occurrences of each transition. Let S be a firing sequence, then the state obtained after firing the transitions of S is obtained by
$$M_K = M_0 + (W \times \underline{S}) \quad (3)$$

Generic properties of a PN are: liveness, deadlocks, reversibility, repetitive components, effectiveness, reachability, and safety, (Hongwei et al., 2013).

We started this work with a thorough examination of the monitoring components; more precisely we focused on the diagnostic system with qualitative external inputs (Fault Tree (FT)). As an output, we will find the possible causes associated with fault location. These locations will help the managers to assess and plan maintenance actions. In the overall classification of monitoring methods and models, we have concentrated on monitoring methods with models, exactly on the methods by functional and material modeling (FT and PN).

We propose a new tool called Lambda Petri Net (λ PN). This Network describes the functioning of systems that are not autonomous. Their operation is conditioned by failure rates. A Lambda Petri Net consists of two parts: a static part and a dynamic one. The static part defines the structure of the lambda Petri Net, where the data is stored, and how the data interacts with each other. The dynamic part defines the initial state of the system. Indeed, the same lambda Petri Net will not have the same behavior depending on its initial state, so it is important to separate the two concepts.

The modeling of our diagnostic system can be done by different types of Petri Nets (ordinary PN, high-level PN), assuming that the possible faults are known a priori and modeled by specific mechanisms. Our approach deals with Lambda Petri Net modeling at the level of the transitions.

To model the monitoring function, we use an extension of the PN, which integrates through the lambda aspect the failure rate in the monitored system. The λ PN is oriented for modeling a base of failure rate rules which follows from the logical expression of the fault tree of the monitored system. The λ PN tool models the set of the logical reasoning of the FT, according to the specific concepts of a logical expression. The analysis aspect offers refined information at the level of each defect by the transfer of fault signals.

The main advantage of Petri Lambda Nets is their strong mathematical foundations. In addition, a great deal of software programs make it possible to simulate and analyze Lambda Petri Nets. Using lambda Petri Nets in industrial systems has several advantages in terms of reduced wiring and ease of monitoring and maintenance. Inputs and outputs in λ PN allow easy modeling and access to the markings of all places at any time. These elements make λ PN an effective and adequate tool for our modeling needs to get simulation support.

The disadvantages of Lambda Petri Nets is their modeling complexity, which leads to producing errors. When using λ PN, the delays are no longer negligible and must be taken into account, especially when the quality service of the Network changes over time which results in non-periodic activation moments.

3. CASE STUDY

3.1 Problem

Our case study has been conducted in collaboration with the Machining Robot Cell of the Production Engineering Laboratory of the National Engineering School in Tarbes (ENIT), France. Our machining robot cell in Figure 2 is composed of:

- i) KUKA KR120 robot ;
- ii) Pneumatic grinder ;
- iii) Tool adapted to the task to be performed ;
- iv) Clamped piece in a vise arranged on a grooved table ;
- v) The vision system camera (National Instruments Image Processing Software).



Figure 2. Machining robot cell

Many authors have been interested in the reliability field of cutting tools and in modeling the surface roughness of machined parts.

3.2 FT Analysis

The Fault Tree (FT) of the robot cell was the centerpiece of our PN-based strategy and is presented in figure 3. The analysis and research of the dreaded event in our FT also referred to as the tree top event, highlights the non-conforming piece (a), Figure 3, in the robot cell.

If we search for the cause of an undesirable event, it can be due to a fault on this very element or to a fault on any other element of the system.

We used CABTREE software 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 3.

- 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).

Logic gates can model the Boolean function F of the dreaded event of our FT. In our work, we used the "OR" logic gate. To illustrate our approach, we consider the logical equation F of the fault tree:

$$F = (b1 \text{ OU } b2) \text{ OU } (c1 \text{ OU } c2) \text{ OU } (d1 \text{ OU } d2) \text{ OU } (e1 \text{ OU } e2) \text{ OU } (f1 \text{ OU } f2)$$

The $(+)$ \Leftrightarrow OR operator represents the union of logical variables $\{a, b, c, d, e, f, b1, b2, c1, c2, d1, d2, e1, e2, f1, f2\}$.

$$F = \underbrace{\underbrace{(b1 + b2)}_b + \underbrace{(c1 + c2)}_c + \underbrace{(d1 + d2)}_d + \underbrace{(e1 + e2)}_e + \underbrace{(f1 + f2)}_f}_a \quad (4)$$

Such as:

$$\begin{aligned} b &= b1 + b2 \\ c &= c1 + c2 \\ d &= d1 + d2 \\ e &= e1 + e2 \\ f &= f1 + f2 \end{aligned}$$

$$F = [b + c + d + e + f] \quad (5)$$

$$F = a \quad (6)$$

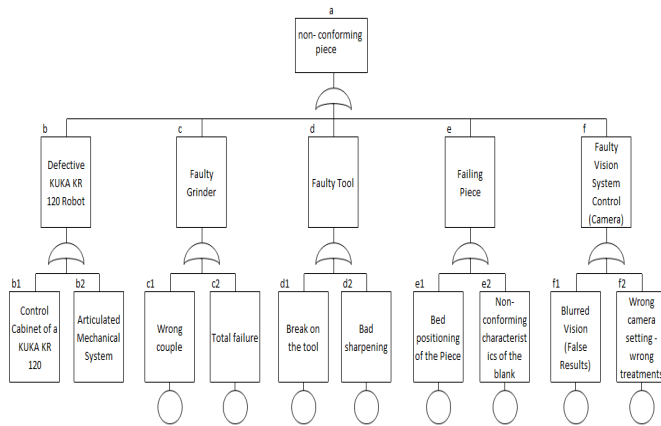


Figure 3. Fault Tree of the robot cell, corresponding to F

3.3 Analysis by λ PN

The machining robot cell is shown in figure 2. Its operation using the PN appears in figure 4. Our PN model is represented by 16 places and 26 transitions, respectively denoted from P1 to P16 and the transitions from T1 to T26. The number of tokens in P_i place represents the number of failures in the robot cell. The transition T_i located between the places P_i are transitions for all the actions performed by the robot cell, with all their connections. This transition is immediate. The number

of tokens in place P_i corresponds to the number of faulty elements in our robot cell. The weights of the arcs are indicated on the model next to the arcs. The absence of a firing means that the arc in question is implicitly weighted at 1.

Our PN can represent the successive assembly and disassembly of a single element ($P1$). Thus, there are two-state processes (Stop-Start), and the passage from one state to the other mobilizes a resource, symbolized by the token contained in the arcs, which was added between the places and the transitions. Containing a source place ($P1$) and a source transition ($T1$), this transition is always sensitized but with a capacity counter ($Cap(P1) = 5$).

Additional capacity has been given to the PN as in the generalized PN; there is no limitation on the number of tokens per place. Capacity Petri Net is a PN in which capacities that are strictly positive whole numbers, are associated with the places. The firing of an input transition of a place P_i whose capacity is a $Cap(P_i)$, is only possible if the firing does not lead to a number of tokens in P_i greater than $Cap(P_i)$.

In our example given in figure 4, firing $T1$ leads to 5 tokens in $P1$. $T1$ can thus no longer be fired.

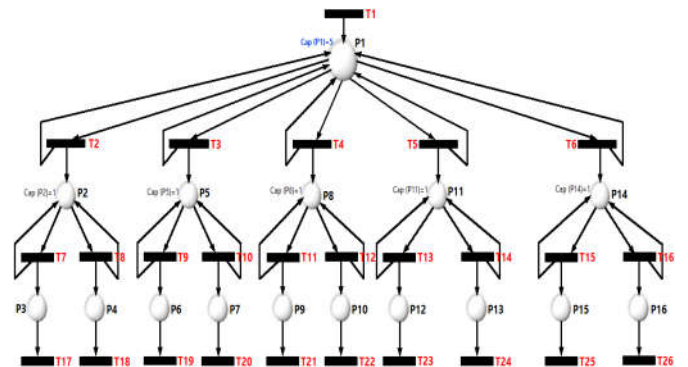


Figure 4. Petri Net before firing of transitions

Our objective is to control and automate the considered FT system using its PN model and using lambda failure rates. To do this, it is necessary to convert the PN model shown in Figure 6 into its equivalent λ PN model. The possible transformation from ordinary PN to high-level λ PN is shown in figures 5 and 6.

Our λ PN before firing of transitions is shown in figure 5.

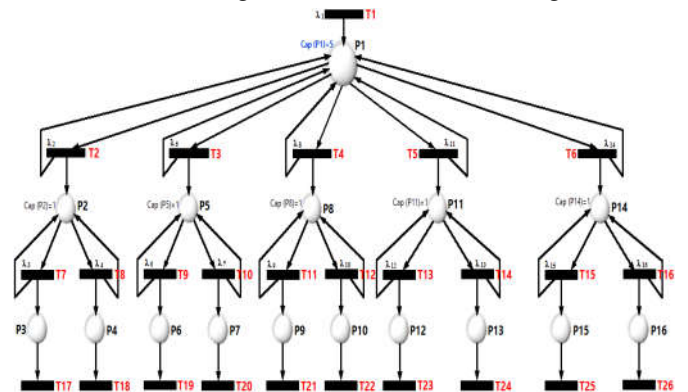


Figure 5. λ PN before firing of transitions

Our λ PN after firing of transitions is shown in figure 6.

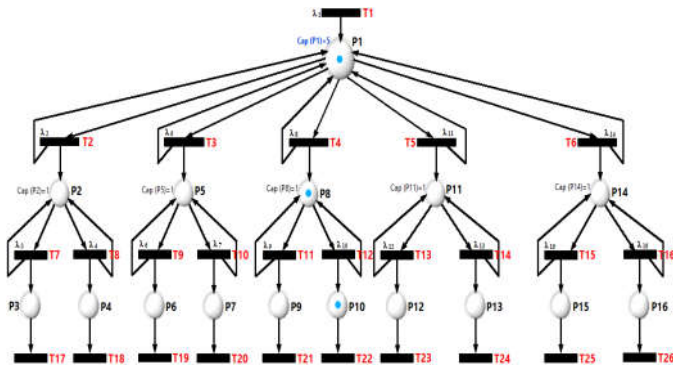


Figure 6. λ PN after firing of transitions

Transitions (T1, T2, ... T26) of the Petri Net are materialized by sensors. The messages used on the transitions are failure rates between 0 and 1. This communication type allows a relatively simple modeling of all states of the system dynamical behaviors. Table 2 describes the significance of each place and the different failure rates used in the Lambda Petri Net of our system are shown in table 3.

Table 2. Significance of the places

Places	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

4. LABVIEW IMPLEMENTATION

We have applied our fault tree transformation technique (FT) in a Lambda Petri Net (λ PN). We used the LabView environment platform for modeling and simulation. LabView is based on a graphical development environment of «National Instruments», and is mainly used for instrument control and industrial automation.

4.1 FT – LabView Implementation

We have proposed the implementation in the LabView environment of the Lambda Petri Net of the case study with failure rates of the different FT components. Our model is shown in the following figures 7 and 8.

1) The implementation-modeled AND- λ PN under LabView will be as follows, figure 7:

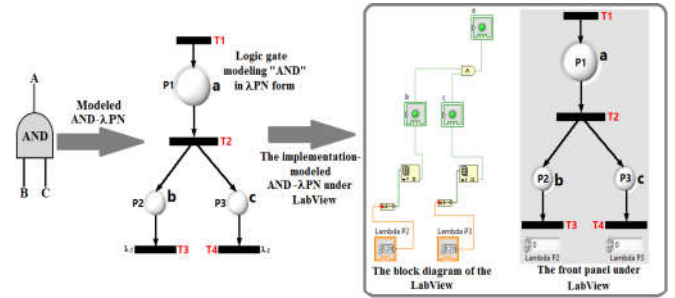


Figure 7. Transformation of the « AND » logical gate of the FT into the λ PN

2) The implementation-modeled OR- λ PN under LabView will be as follows, figure 8:

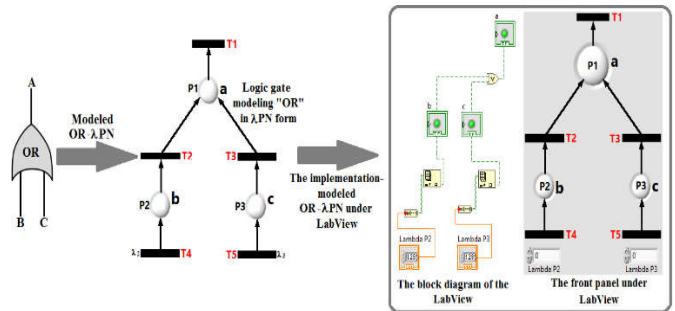


Figure 8. Transformation of the « OR » logical gate of the FT into the λ PN

4.2 λ PN - LabView Implementation

We have associated the λ PN to a LabView state machine. The resulting structure is shown in figures 9 and 11. The front panel is the user interface of VIs (Virtual Instruments) in our system. It is shown in figures 9 and 11 and describes an analysis application called Dominant Failure Mode. The λ PN: contains 16 places that are circular LEDs (Light Emitting Diodes) emitting 2 phases of light, white and gray, and 26 transitions. Each place presents an event of our FT and describes its state: inputs (commands) and outputs (indicators) of the program. String Indicators model these states and another String Indicator displays the state of our system (State of the non-conforming piece). This indicator is used to display Normal or Abnormal Operation.

Our application consists of a box that contains a Digital Indicator for the sum of the different failure rates of our FT. It also includes two other Digital Indicators, the first one to display the highest failure rate of level 1 to the FT and the second to display the highest failure rate of level 2 to the FT. 10 Numeric Controls and 5 Digital Indicators are materialized by failure rates. The different failure rates are values between 0 and 1.

The input variables are the failure rate of each component. They are labeled as «Numeric Control» in LabView and the results appear as «Digital Indicator» on the front panel.

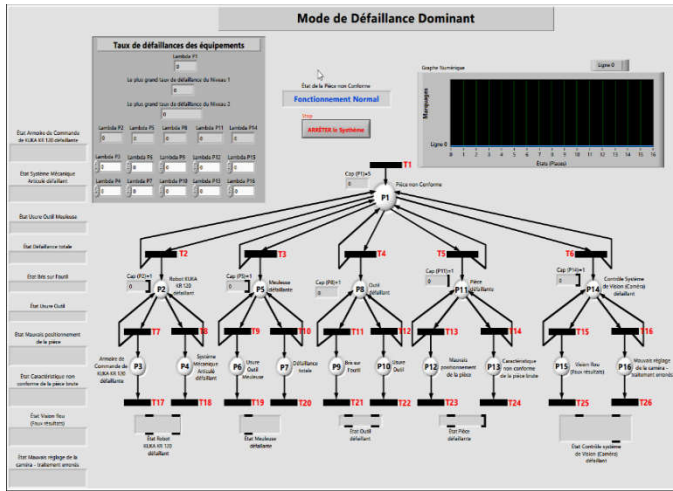


Figure 9. Front panel under LabView of the PN system - Modeling without fault

The block diagram, figure 10, represents the application program written in the form of a data flow diagram. This figure illustrates how commands and indicators are materialized by digital displays for a state 0 or 1 in the block diagram of LabView.

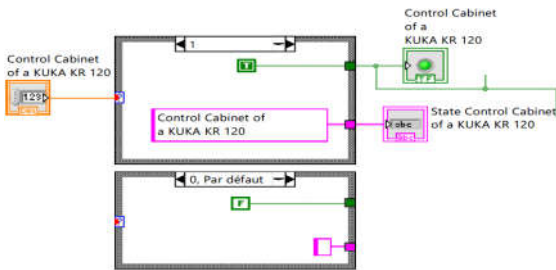


Figure 10. LabView block diagram of the λPN system

5. SIMULATION AND RESULTS

5.1 Failure rate

According to the hierarchical expertise of robustness, the failure data for each component is given in the following table 3. Vibrations are the most critical events. Vibrations in radial direction have the greatest value of failure rate ($\lambda_6 = 0.0009$ and $\lambda_{10} = 0.0009$).

Table 3. Failure rate of level 2 and dreaded event – FT - λPN

Level 2- FT - λPN			
λ_3	0.0001	λ_{10}	0.0009
λ_4	0.0002	λ_{12}	0.0007
λ_6	0.0009	λ_{13}	0.0006
λ_7	0.0005	λ_{15}	0.0003
λ_9	0.0008	λ_{16}	0.0004

According to equations (4), (5), and (6) and Table 3, we obtained the following failure rates shown in table 4. The faulty tool is the most critical event with a failure rate ($\lambda_8 = 0.0017$).

Table 4. Failure rate of level 1 and dreaded event – FT - λPN

Level 1- FT - λPN				
λ_2	λ_5	λ_8	λ_{11}	λ_{14}
0.0003	0.0014	0.0017	0.0013	0.0007
Dreaded Event - FT - λPN				
λ_1	0.0054			

5.2 LabView Simulation

The obtained simulations and results are shown in figure 11. We use tokens in the graph places to signal the state of each resource at a given moment, it is marked in gray.

At the second FT level and corresponding to its failure rate, the component « Bad sharpening » is in failure mode, therefore the corresponding state is activated.

According to the diagnostic characteristics of FT and equations (4), (5), (6), the dreaded event is $\lambda_1 = 0.0054$. After comparing all failure rates of level 1, the highest failure rate is $\lambda_8 = 0.0017$, so the highest level 2 failure rate is $\lambda_6 = \lambda_{10} = 0.0009$.

- So respectively, place (P10) is colored gray, and its signal is in state 1.
- If the place (P10) is faulty then (P8) is initially faulty and then (P1) is faulty. These two places are colored in gray, and their signal is in state 1. If there is a faulty place, the triggered diagnostic process makes the system fail. So the capacity $Cap(P8) = 1$ of place (P8) is displayed as 1 and the finite capacity $Cap(P1) = 5$ of place (P1) is displayed as 1.
- To carry out a deductive analysis in our λ PN, we proceed by firing transition (T1), and then place (P1) has a token. If there is a token in place (P1), then we have to go to transition (T4) directly. If (T4) is fired then place (P8) has a token. If place (P8) has a token, then (P10) also has a token after firing transition (T12). This diagnostic process is obtained through the return arcs building our Lambda Petri Net.

5.3 Marking

Our interest at this level lies in the control of the machining robot cell presented in figure 2. Its fault tree is given in figure 3, and is associated with the λPN of figure 5. The result after the firing according to the faults is the model shown in figure 6. We also defined the marking here to interpret the Lambda Petri Net simulation results. It allows us to understand the behavior of the system, identify the states of the system at different times and track its evolution by following transitions.

According to equation (1), we have:

- The initial marking of our λPN, corresponding to figure 8, is $M_0 = [0000000000000000]$.
- The marking of our λPN after the firing, corresponding to figure 10, is $M_1 = [1000000101000000]$.

$P = \{P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16\}$

$T = \{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15, T16, T17, T18, T19, T20, T21, T22, T23, T24, T25, T26\}$

5.4 Generic Properties of Lambda Petri Nets

In order to check the consistency of the model and detect possible undesirable behaviors we used the generic properties. Generic properties are fundamental characteristics of Petri Nets that allow the analysis of their behavior. These properties include Boundedness; Safeness; Liveness; Deadlock; Reversibility; Repetitive; conflictual; Reachability.

The mathematical properties resulting from the analysis of our λ PN allowed a behavioral and structural study which is essential to the validation of a specification. We obtained a marking $M_k = [1000000101000000] > 0$ then our λ PN is pure, and S is firing.

In this consistent verification, we used the incidence matrix and the marking vector to evaluate the behavior of the λ PN. The obtained results indicate that the model is consistent and fireable. The verification of the generic properties of our λ PN confirms the reliability of the model, which is essential to ensure the consistency and robustness of the monitoring system.

6. CONCLUSION

Modeling and simulation using Petri nets are powerful tools for assessing the performance of complex systems; these networks represent an efficient mathematical formalism for modeling system failures. This work presents the development of a diagnostic strategy for a machining robot cell using a Petri Net generated from an FT on this cell in the LabView environment. After transforming the fault tree into a Petri Net using equivalences, we determined the failure rates to consider on Lambda transitions. We performed simulations after implementation under the LabView environment to obtain results that prove to be fully satisfactory. Indeed, a consistent check of our λ PN to verify the model structure to exclude implementation errors has been carried out. This approach could contribute to improving the availability and performance of any industrial equipment.

REFERENCES

- Abdoulaye, D., Farid, N., Rachid, N., Bernard, K.-F., & Diakarya, B. (2019). Proportional hazard model for cutting tool recovery in machining. *Journal of Risk and Reliability*.
- David, R., & Alla, H. L. (1992b). *Du Grafset aux réseaux de Petri*. 2eme Edition revue et augmentée, Livre, France.
- Diaz, M. (2009). Fundamental Models, Verification and Applications. *ISTE & Wiley, N 978-1-84821-079-0*.
- Hongwei, L., Yin, W., Hyoun Kyu, C., Jason, S., Terence, K., Stéphane, L., Scott, M., & Spyros, R. (2013). Concurrency bugs in multithreaded software : Modeling and analysis using Petri nets. *Discrete Event Dynamic Systems*, 23, 157-195.
- Upadhyay, V., Jain, P. K., & Mehta, N. K. (2013). In-process prediction of surface roughness in turning of ti-6al-4v alloy using cutting parameters and vibration signals. *Measurement*, 1 (46), 154-160.
- Vaxevanidis, N. M., Kechagias, J., Fountas, N. A., & Manolakos, D. E. (2015). Evaluation of machinability in turning of engineering alloys by applying artificial neural networks. *The Open Construction & Building Technology Journal*, 8(1), 389-399.
- Xavior, A., & Adithan, M. (2009). Determining the influence of cutting fluids on tool wear and surface roughness during turning of AISI 304 austenitic stainless steel. *J Mater Process Technol*, 209, 900-909.