U.P.B. Sci. Bull., Series D, Vol. 85, Iss. 4, 2023 ISSN 1454-2358

PETRI NET MODELING OF A MACHINING ROBOT CELL

Zohra MEHAR[1], Rachid NOUREDDINE[2*], Farid NOUREDDINE[3]

In this paper, we propose a new approach for modeling and analyzing faults in a machining robot cell. Therefore we decided to use a hybrid analysis associating a Petri Net (PN) to a Fault Tree and thus obtained our called Lambda Petri Net (λPN). This work has been implemented in the LabView environment (Laboratory Virtual Instrument Engineering Workbench). The 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, 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. The faults are supposed to be permanent, i.e. when a fault occurs, the system enters a degraded mode and will no longer return to a nominal mode without being repaired first. It can thus end up in a failure mode, in which the system is no longer operational.

In case of failures, minimization of restart time is essential to avoid penalizing the productivity of the concerned cell. We therefore propose and this is our main objective a diagnostic system based on the use of a Petri Net implemented from a fault tree. Petri Nets (PN) are intelligent diagnostic models, constitute a graphical and mathematical tool allowing qualitative and quantitative analysis. They are composed of a set of places (input), transitions (output), and arcs (integer)

[1] PhD student, Industrial Production and Maintenance Engineering Laboratory, Institute of Maintenance and Industrial Safety, University of Oran 2 Mohamed Ben Ahmed, Oran, Algeria, e-mail: mehar.zohra@univ-oran2.dz

[2] **corresponding author,* Prof., Industrial Production and Maintenance Engineering Laboratory, Institute of Maintenance and Industrial Safety, University of Oran 2 Mohamed Ben Ahmed, Oran, Algeria, e-mail: noureddine.rachid@univ-oran2.dz

[3] Ass. Prof., Production Engineering Laboratory, National School of Engineering in Tarbes, National Polytechnic Institute of Toulouse, France, e-mail: farid.noureddine@enit.fr

which are effective for modeling availability and for both production and security systems. The analysis is performed by simulations of behavioral and explanatory models. PNs are capable of managing all sorts of probabilistic distributions. The state of a Petri Net is defined by its marking and by a distribution of tokens in different places.

The most important contribution of this work lies in our designing and implementing of Lambda Petri Nets (λPN) which are a mathematical formalism that manipulates discrete variables, adequate to the framework of systems failure using tokens and failure rates. These failure rates are deduced from the analysis and modelling of the fault locations of each piece of equipment in a machining robot cell.

For this purpose, our contribution was established in three steps: In the first step, the analysis by FT of the machining robot cell was carried out. In the second step, the transfer of the logic gates from the FT to the PN was completed, followed by the system modeling into PN. The third step consisted in transforming the PN to the λPN by introducing the respective failure rates, according to hierarchical expertise of robustness. The failure data for each component were used to calculate the λPN of the system. The results indicate that the λPN of the system is strongly influenced by the most critical components. Indeed, components with the highest failure rates have the greatest impact on the system's failure probability. This approach would allow modeling the interactions between failures as events that can occur simultaneously or sequentially.

Generic properties of normative assessment have been applied. The mathematical properties allow the writing of incidence matrices and marking vectors. These qualitative and quantitative analyses were paramount to validate the system. Figure 1 summarizes the monitoring components and focuses on our contribution to the diagnostic system of the machining robot cell. We thus will obtain the different possible causes associated with a degree of credibility and degree of severity for each cause. These degrees will help managers to evaluate and plan maintenance actions.

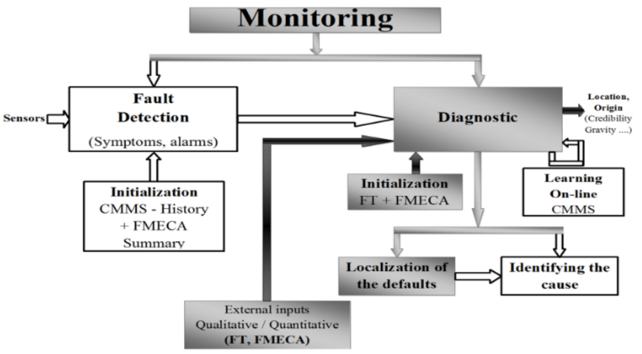

Fig. 1. Block diagram of the monitoring system

State of the art

[1] Propose a new approach for both modeling and failure analysis. They combine graphical representations provided by Petri Nets and fuzzy logic. They used Petri Nets in failure analysis, allowing to replace logic gate functions in fault trees. Fuzzy logic allows natural language descriptions of process entities and thus applies these rules to diagnose breakdowns.

[2] Identified two distinct formalisms for analyzing failure scenarios and availability of considered systems. These two formalisms are Generalized Stochastic Petri Nets (GSPN) on the one hand and Fault Tree Driven Markov Processes (FTDMP) on the other hand.

[3] Propose a methodology to assess the probability of electrical failure of a floodgate by using fault tree and the implementation of stochastic Petri Nets. The authors constructed the stochastic Petri Net with repair by modeling the different operating states of the valve.

[4] Present how Petri Net models have been developed for a wide variety of manufacturing systems. The modeled system was a robot cell for the preparation and analysis of metallographic samples. The model is very complex and with many elements. They achieved the simplification of the obtained model by switching to colored Petri Nets (CPN).

[5] They propose a Petri net model to validate search techniques in graphs, in order to improve the Worst-Case method through dynamic optimization of the number of involved agents.

[6] We propose an algebraic approach to study fault diagnosis classes in a labeled Petri net (LPN) system based on state estimation over a sliding window of length h, and fault detection is performed by solving an Integer Linear Programming (ILP) problem.

2. λPN Formalization

According to [7], different PNs 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 (λPN) 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: [5,8]

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 . 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: $W = W^{+} - W^{-}$ ou $W = \text{Post} - \text{Pré}$ (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 (Pj, Tj) can be represented in a matrix with $W^{+}(i,j) = \text{Pré}(\text{P}_i, T_j)$
- ✓ The connecting arcs of Places to Transitions Post (Pi, Tj) can be represented in a matrix with $W^{-}(i,j) = \text{Post}(\text{P}_i, T_j)$

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

$$M_k = M_0 + (\text{W} \times \underline{S}) \quad (3)$$

Generic properties of a PN are: liveness, deadlocks, reversibility, repetitive components, effectiveness, reachability, and safety, [5,9].

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 (λPN). 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 λPN 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 λPN 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 λPN allow easy modeling and access to the markings of all places at any time. These elements make λPN an effective and adequate tool for our modeling needs to get simulation support.

The disadvantages of Lambda Petri Nets is their modeling complexity, which lead to producing errors. When using λPN, 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. Transfer PN-FT

FT is a method of deductive analysis used in dependability, [10,11]. This method analyzes the reliability, availability, and security of more widely used systems, [12]. FT is the simplest and most used technique to assess reliability, [13-15].

According to [16,17], a fault tree in general is divided into two categories: coherent and non-coherent. A coherent fault tree consists only of logical operations “AND” and “OR”, while a non-coherent tree also contains other logical operations.

This article is mainly concerned with the application of Petri Net modeling to coherent fault tree analysis [18].

The use of knowledge from a Fault Tree with the Petri Nets principle through the reasoning must be done qualitatively but also quantitatively to provide effective analysis, like the case in industrial applications, [19,20].

A logical relationship exists between Petri Nets and fault trees. The formalism used in our work allows to implement the PN from the FT. We thus use a formalism that associates each gate of the FT as a symbolization of the PN.

1) The "AND" logic gate is as follows, figure 2:

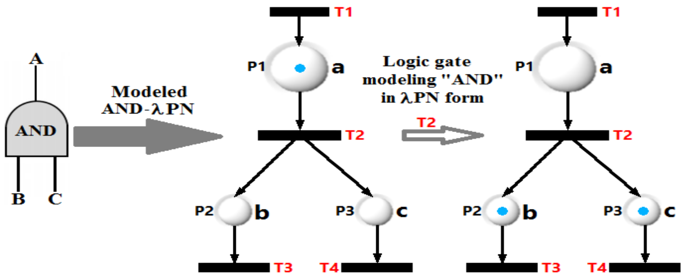

Fig. 2. Transformation of the « AND » logical gate of the FT into the PN

2) The "OR" logic gate is as follows, figure 3:

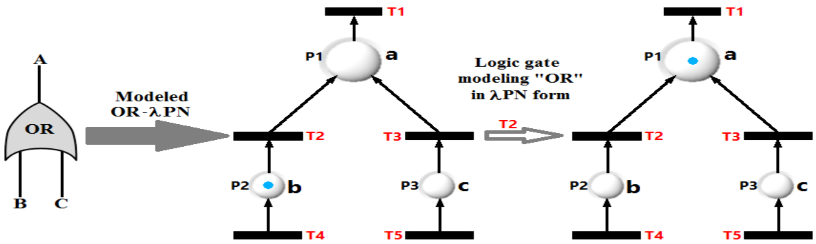

Fig. 3. Transformation of the « OR » logical gate of the FT into the PN

4. Case study

4.1 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 4 is composed of:

- A KUKA KR120 robot,
- An electro spindle,
- A milling tool,
- The part to be machined in its clamping feature,
- The lubrication system.

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

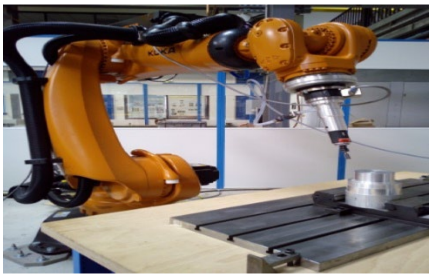

Fig. 4. Machining robot cell

[21] Have studied and modeled a new basic hazard function called GEV proportional hazard function. This generalized function with extreme values is obtained thanks to the Gumbel function and the property of being non-monotonic, an increasing then decreasing function, thus capable of taking into account the cutting tool behavior with different mechanisms of self-repair and self-recovery after a tiny breakage. The authors introduce parameters considering operating and environmental conditions, including vibration signals, material hardness, and lubrication/cooling. Their results show the impact of all these variables on the surface roughness of machined parts.

Vibration signal: Two types of vibration occur during machining that are sustained and self-sustained vibrations. Self-sustained vibrations are characterized by an absence of periodic external forces but depend on the chip formation process. On the other hand, sustained vibrations originate from periodic external forces such as those acting on the part during intermittent cutting.

[22] Have studied similar cutting tool conditions and the acceleration amplitude of tool vibrations in the axial, radial, and tangential directions. Table 1 shows the impact of experimental values of acceleration amplitude of vibration (Root Mean Square (RMS) value) in axial (Vx), radial (Vy), and tangential (Vz) directions on the surface roughness (Ra). The worst case is the variation in Vy(g).

Material hardness: Both a decrease in the used material hardness and the reliability of the applied heat treatment lead to the deterioration of the surface roughness.

[23] Have studied the influence of cutting tool conditions and material hardness on surface roughness. Their obtained results are given in Table 1.

Lubrication/cooling conditions: High cutting temperature during machining can lead to unacceptable surface quality. Compared to dry cutting, the lubrication/cooling conditions lead to a significant improvement in surface roughness.

[24] Show all the impacts of cutting conditions on surface roughness and tool wear in the turning process. The contribution of cooling/lubrication is around 14%.

[21] Studied the influence of vibrations, material hardness defects, and lubrication. The results are presented in Table 1.

We can deduce that all the contributions of the variables deteriorate the basic hazard function.

Table 1.

Machining parameters considered

Cutting conditions			
Cutting speed V_c (m/min)	Feed rate f (mm/rev)		Depth of cut d (mm)
250	0.1		0.5
Vibration impact on the surface roughness			
	V_x (g)	V_y (g)	V_z (g)	R_a (μm)
Experiment 1	0,37	3,92	1,63	1,51
Experiment 2	0,39	4,84	1,65	1,73
Material hardness impact on the surface roughness				
	Hardness (HB)		R_a (μm)	
Experiment 1	130		1,53	
Experiment 2	240		1,35	
Lubrication/cooling conditions				
Vibration signal (covariate X_1)	Material hardness (covariate X_2)		Lubrication (covariate X_3)	
$\varphi(\beta_1 X_1) = 1,15$	$\varphi(\beta_2 X_2) = 1,13$		$\varphi(\beta_3 X_3) = 0,86$	

4.2 FT Analysis

The Fault Tree (FT) of the robot cell was the centerpiece of our PN-based strategy and is presented in figure 5. 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 5, 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 5.

- First level: Defective KUKA KR 120 Robot, Faulty Spindle, Faulty Tool, Failing Piece, Malfunctioning Lubrication System (the lubrication item is mainly a pump).
- Second level: Faulty Feed rate (f), Faulty Depth of cut (d), Vibration in axial (Vx) directions, Vibration in radial (Vy) directions, Vibration in tangential (Vz) directions, Faulty Cutting speed (Vc), Break on the Tool, Bad positioning of the Piece, Failing Hardness, Low Flow, Bad Liquid of the lubrication pump.

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:

$$F = \begin{pmatrix} (b1\ OU\ b2\ OU\ b3\ OU\ b4\ OU\ b5)\ OU \\ (c1\ OU\ b3\ OU\ b4\ OU\ b5)\ OU\ (d1\ OU\ b3\ OU\ b4\ OU\ b5)\ OU\ (e1\ OU\ e2)\ OU\ (f1\ OU\ f2) \end{pmatrix}$$

The $(+) \Leftrightarrow OR$ operator represents the union of logical variables {a, b, c, d, e, f, b1, b2, b3, b4, b5, c1, d1, e1, e2, f1, f2}.

$$F = \underbrace{\left(\underbrace{(b1+b2+b3+b4+b5)}_{b} + \underbrace{(c1+c3+b4+b5)}_{c} + \underbrace{(d1+b3+b4+b5)}_{d} + \underbrace{(e1+e2)}_{e} + \underbrace{(f1+f2)}_{f}\right)}_{a}$$

(4)

Such as:

$b = b1 + b2 + b3 + b4 + b5$	$d = d1 + b3 + b4 + b5$	$f = f1 + f2$
$c = c1 + b3 + b4 + b5$	$e = e1 + e2$	

$$F = [b + c + d + e + f] \tag{5}$$

$$F = a \tag{6}$$

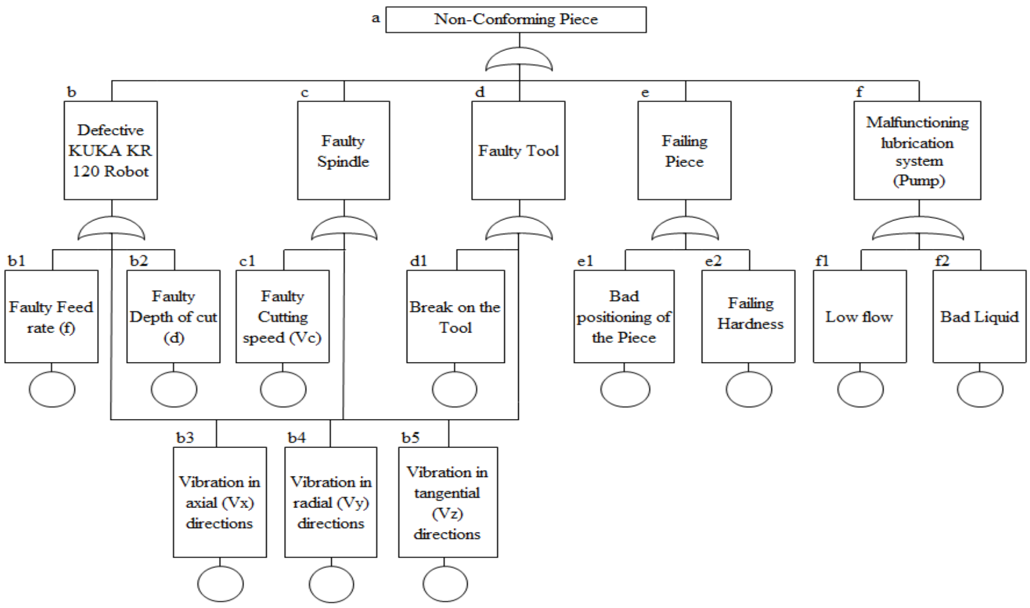

Fig. 5. Fault Tree of the robot cell, corresponding to F

4.3 Analysis by λPN

The machining robot cell is shown in figure 4. Its operation using the PN appears in figure 6. Our PN model is represented by 17 places and 28 transitions, respectively denoted from P1 to P17 and the transitions from T1 to T28. The number of tokens in Pi place represents the number of failures in the robot cell. The transition Ti located between the places Pi 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 Pi 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 capacity that are strictly positive whole numbers, is associated with the places. The firing of an input transition of a place Pi which capacity is a Cap(Pi), is only possible if the firing does not lead to a number of tokens in Pi greater than Cap(Pi).

In our example given in figure 6, firing T1 leads to 5 tokens in P1. T1 can thus no longer be fired.

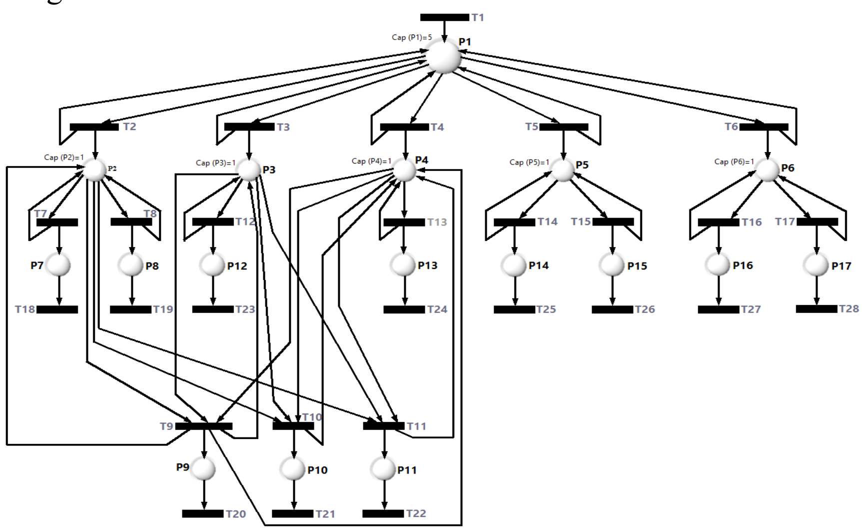

Fig. 6. Petri Net before firing of transitions

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 8 into its equivalent λPN model. The possible transformation from ordinary PN to high-level λPN is shown in figures 7 and 8.

Our λPN before firing of transitions is shown in figure 7.

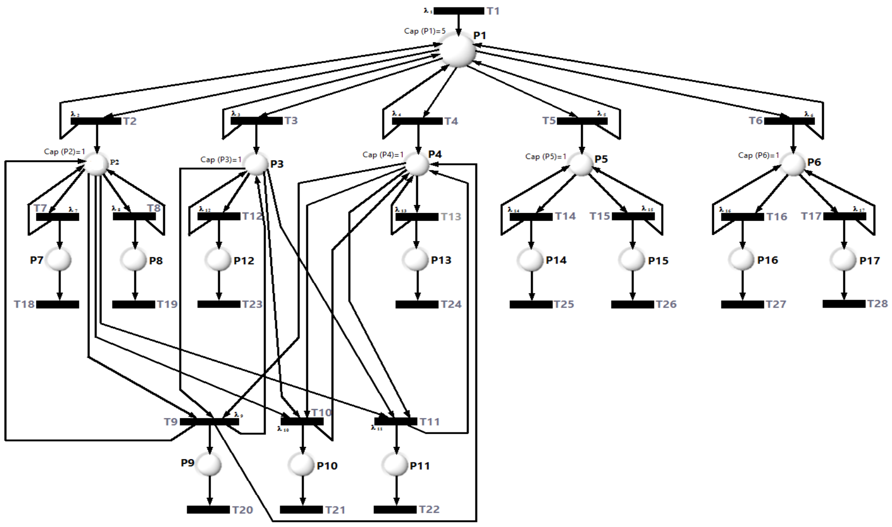

Fig. 7. λPN before firing of transitions

Our λPN after firing of transitions is shown in figure 8.

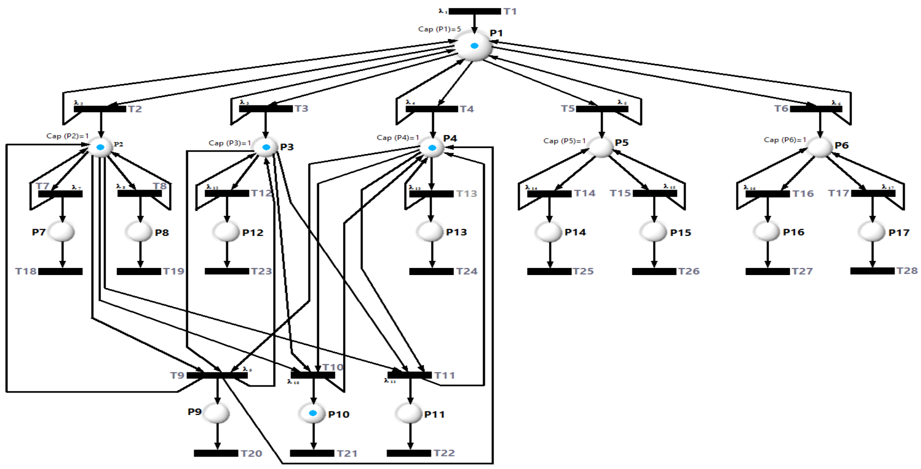

Fig. 8. λPN after firing of transitions

Transitions (T1, T2, ... T28) 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	Defective KUKA KR 120 robot

P3	Faulty spindle
P4	Faulty tool
P5	Failing piece
P6	Malfunctioning lubrication system (pump)
P7	Faulty feed rate (f)
P8	Faulty depth of cut (d)
P9	Vibration in axial (Vx) directions
P10	Vibration in radial (Vy) directions
P11	Vibration in tangential (Vz) directions
P12	Faulty cutting speed (Vc)
P13	Break on the tool
P14	Bad positioning of the piece
P15	Failing hardness
P16	Low flow
P17	Bad liquid

5. LabView Implementation

We have applied our fault tree transformation technique (FT) in a Lambda Petri Net (λPN). 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.

5.1 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 9 and 10.

1) The implementation-modeled AND-λPN under LabView will be as follows, figure 9:

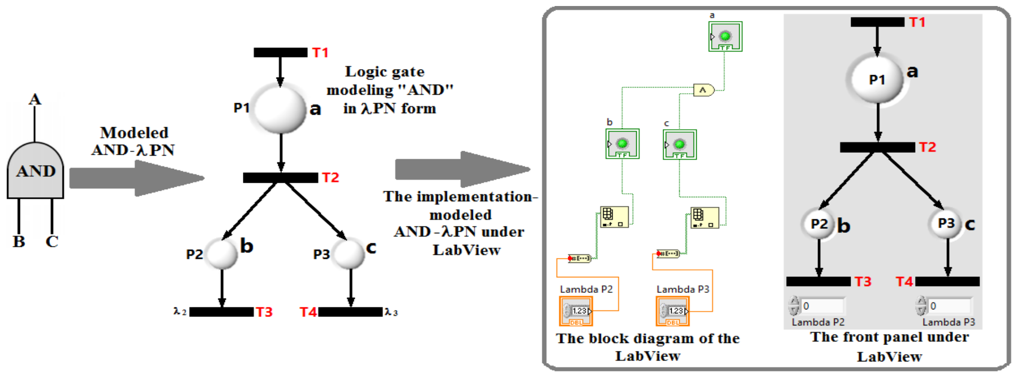

Fig. 9. Transformation of the « AND » logical gate of the FT into the λPN

2) The implementation-modeled OR-λPN under LabView will be as follows, figure 10:

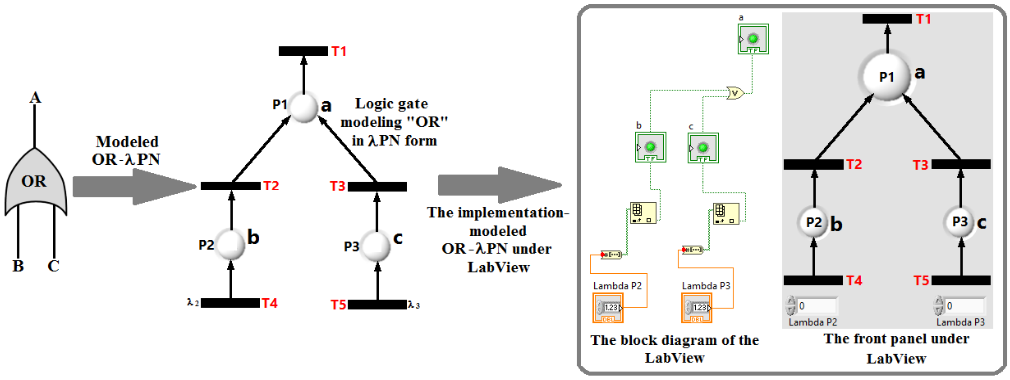

Fig. 10. Transformation of the « OR » logical gate of the FT into the λPN

5.2 λPN - LabView Implementation

We have associated the λPN to a LabView state machine. The resulting structure is shown in figures 11 and 14. The front panel is the user interface of VIs (Virtual Instruments) in our system. It is shown in figures 11 and 14 and describes an analysis application called Dominant Failure Mode. The λPN: contains 17 places that are circular LEDs (Light Emitting Diodes) emitting 2 phases of light, white and grey, and 28 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. 11 Numeric Controls and 5 Digital Indicators are materialized by failure rates. The different failure rates are values between 0 and 1.

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

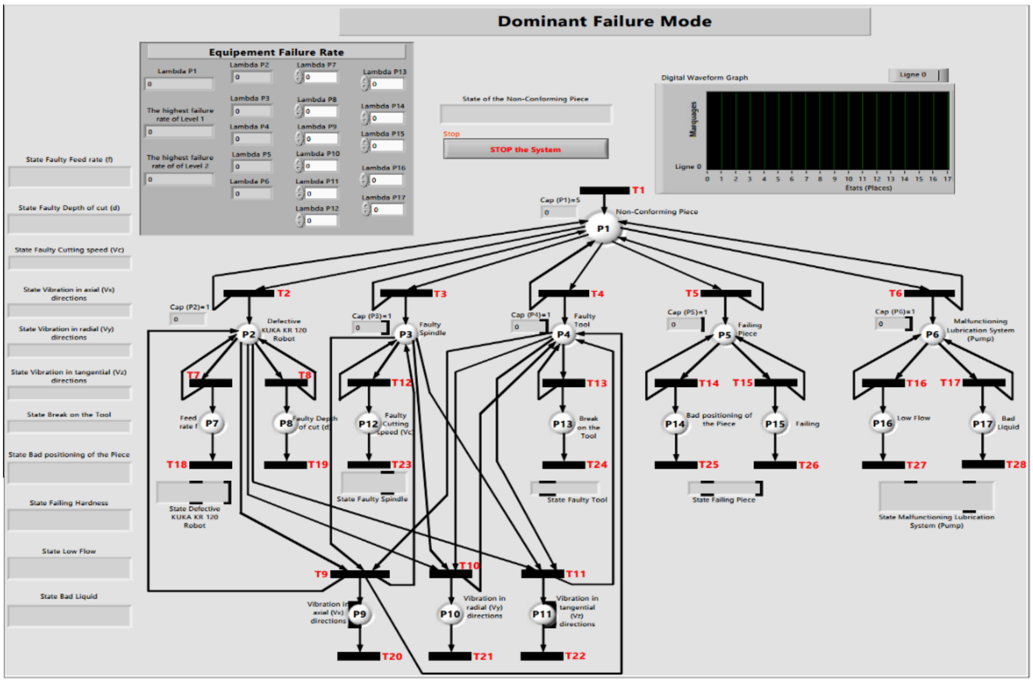

Fig. 11. Front panel under LabView of the PN system - Modeling without fault

The block diagram, figure 12, represents the application program written in the form of a data flow diagram. This figure illustrates how command and indicator are materialized by digital displays for a state 0 or 1 in the block diagram of LabView.

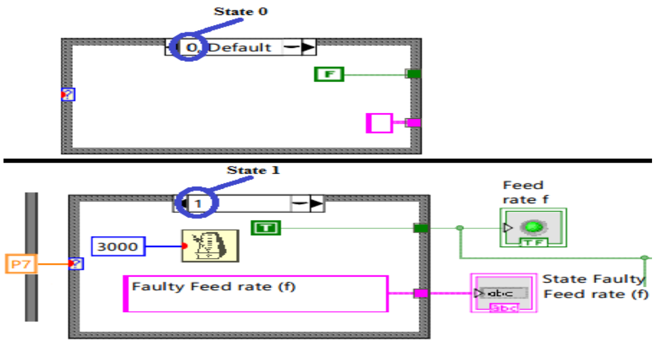

Fig. 12. LabView block diagram of the λPN system

6. Simulations and results

As shown in figure 13, the simulation of the proposed diagnostic system was carried out in three essential steps. First, we assessed the failure rates of all system components and then we used LabView to perform the simulation, allowing to observe the distribution of the system states. Finally, in the third step, we successively identified the marking of the Lambda Petri Net simulation, obtained its generic properties for automatic checking, determined its incidence matrix and obtained its marking vector needed to validate the model.

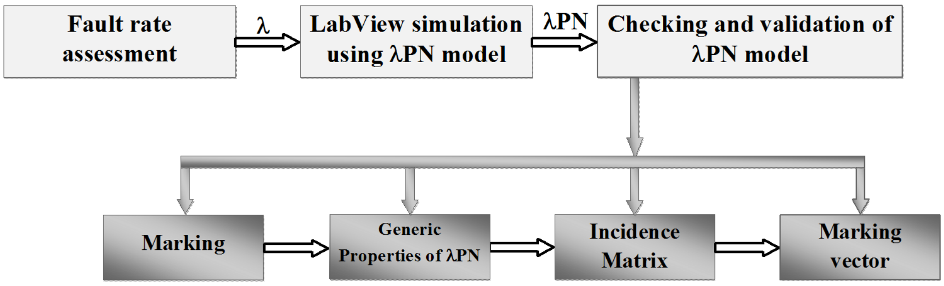

Fig. 13. Steps of the proposed diagnostic system

6.1 Failure rate

According to the hierarchical expertise of robustness and thanks to the values given in table 1, 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 ($\lambda_{10} = 0.006$).

Table 3.

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

Level 2- FT - λPN			
λ_7	0.0001	λ_{13}	0.0003
λ_8	0.0001	λ_{14}	0.0003
λ_9	0.0004	λ_{15}	0.0002
λ_{10}	0.0006	λ_{16}	0.0002
λ_{11}	0.0005	λ_{17}	0.0003
λ_{12}	0.0001		

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 $\lambda_4 = 0.0018$.

Table 4.

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

Level 1- FT - λPN				
λ_2	λ_3	λ_4	λ_5	λ_6
0.0017	0.0016	0.0018	0.0005	0.0005
Dreaded Event - FT – λPN				
λ_1		0.0061		

6.2 LabView Simulation

The obtained simulations and results are shown in figure 14. 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 « Vibration in radial (Vy) directions » 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 $\lambda_1 = 0.0061$. After comparing all failure rates of level 1, the highest failure rate is $\lambda_4 = 0.0018$, so the highest level 2 failure rate is $\lambda_{10} = 0.0006$.

- So respectively, place (P10) is colored gray, and its signal is in state 1.
- If the place (P10) is faulty then (P4) 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(P4) =1 of place (P4) 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 λPN, 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 (T10). This diagnostic process is obtained through the return arcs building our Lambda Petri Net.

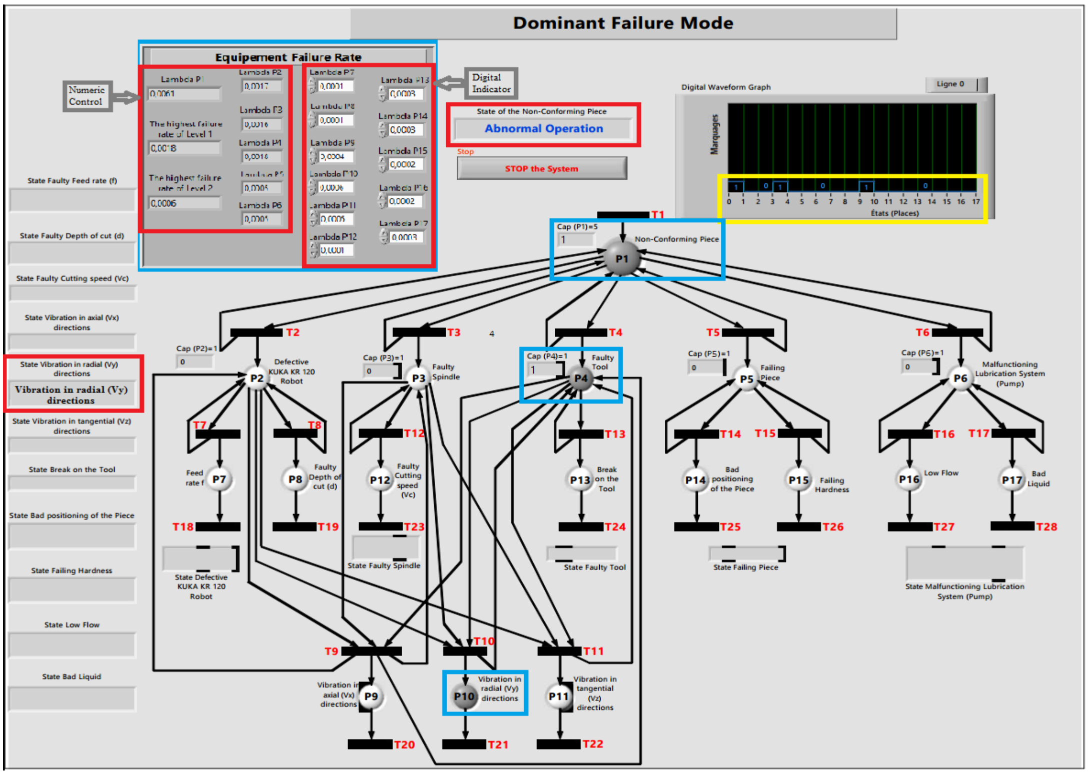

Fig. 14. Front panel under LabView of the λPN system - Modeling with fault

6.3 Marking

Our interest at this level lies in the control of the machining robot cell presented in figure 4. Its fault tree is given in figure 5, and is associated with the λPN of figure 7. The result after the firing according to the faults is the model shown in figure 8. 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 10, is M_0 = [00000000000000000].
- The marking of our λPN after the firing, corresponding to figure 12, is M_1 = [10010000010000000].

  P = {P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17}

  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, T27, T28}

6.4 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.

We obtained the following generic properties for the automatic checking of our system: our λPN is unbounded but safe. λPN is lively, deadlock-free, non-reversible, non-repetitive, conflict-free, reachable, and safe, with a graph of infinite markings.

6.5 Incidence Matrix

The incidence matrix represents the relationships between places and transitions in a Petri Net, allowing to calculate the marking of the system after a transition. The incidence matrix (W) of our Lambda Petri Net (λPN), after applying equation (3) is a matrix of 17 rows (places) and 28 columns (transitions), and is represented in the following matrix:

$$W = \begin{array}{c|cccccccccccccccccccccccccccc|}
 & 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 & T27 & T28 \\
P1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
P11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
P12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
P13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
P14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
P15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
P16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
P17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{array}$$

6.6 Marking vector

The marking vector represents the marking of the λPN in a compact form; it proves useful for the analysis of the results. The marking vector of our λPN is obtained after a firing sequence according to equation (3): $M_k = M_0 + (W \times \underline{S}_1)$.
So we use the M_1 marking for the calculation of the transition vector.
$\underline{S}$ = [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, T27, T28]
Consider sequence S_1= T1 T4 T12
So we can write: $\underline{S}_1$= [1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]^T

$$M_k = \begin{bmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix} + W \times \begin{bmatrix} 1\\0\\0\\1\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 1\\0\\0\\1\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix} + \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix} \times \begin{bmatrix} 1\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix} + \begin{bmatrix} 1\\0\\0\\1\\0\\0\\0\\0\\0\\3\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 1\\0\\0\\1\\0\\0\\0\\0\\0\\3\\0\\0\\0\\0\\0\\0\\0 \end{bmatrix}$$

The mathematical properties resulting from the analysis of our λPN allowed a behavioral and structural study which is essential to the validation of a specification. We obtained a marking M_k= [10010000030000000] > 0 then our 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 λPN. The obtained results indicate that the model is consistent and firable. The verification of the generic properties of our λPN confirms the reliability of the model, which is essential to ensure the consistency and robustness of the monitoring system.

7. 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 λPN 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.

The main future development we aim at lies in taking into account the system dynamics. This extension would allow to consider changes in the system's state over time, which could improve the accuracy of fault location.

Another research area of particular interest could be the design of an integrated diagnostic system combining different diagnostic formalisms, such as Petri nets, fault trees, and probabilistic models which lead to a more thorough fault location.

Acknowledgments

This work was supported by the doctoral training committee in electromechanics of the Institute of Maintenance and Industrial Safety of University of Oran 2 (Algeria) and by both research laboratories : Industrial Production and Maintenance Engineering (LGPMI) of University of Oran 2 (Algeria) and Production Engineering (LGP) of National School of Engineering in Tarbes (France).

R E F E R E N C E S

[1] *F. Lajmi, J. Ghabi, et H. Dhouibi*, « Applying Interval Fuzzy Petri Net to failure analysis », *International Journal of Engineering and Technology*, vol. 11, nº 1, 2020.

[2] *M. Talebberrouane, F. Khan, et M. Z. Kamil,* « Dynamic RAMS Analysis Using Advanced Probabilistic Approach », Dynamic RAMS Analysis Using Advanced Probabilistic Approach, Chemical Engineering Transactions, vol. 77, p. 241-246, 2019.

[3] *K. F. K. Donchi, G. Ftatsi Mbetmi, D. J.* Fotsa-Mbogne, et B. Djeumako, « Modélisation et estimation de la fiabilité dynamique d'un évacuateur de crues par réseaux de Petri stochastiques », 2022.

[4] *F. S. 1Blaga, A. Pop, V. Hule, et C. I. Indre*, « The efficiency of modeling and simulation of manufacturing systems using Petri nets », *IOP Conference Series: Materials Science and Engineering, The Annual Session Of Scientific Papers - IMT Oradea 2021, 2021*.

[5] *A. Djellal et R. Lakel*, « Petri Net Model of Pursuit-Evasion Game in a Graph », International Journal of Control and Automation, 2017.

[6] *A. Chouchane et P. Declerck*, « Diagnosis on a sliding window for partially Observable Petri Nets », *Kybernetika*, vol. 58, nº 4, p. 479-497, 2022.

[7] *R. David et H. L. Alla*, « Du Grafcet aux réseaux de Petri ». 2eme Edition revue et augmentée, Ed. Hermes Science Publications, 1992.

[8] *M. Diaz*, « Fundamental Models, Verification and Applications », ISTE & Wiley, N 978-1-84821-079-0, 2009.

[9] *L. Hongwei et al.*, « Concurrency bugs in multithreaded software: modeling and analysis using Petri nets », *Discrete Event Dynamic Systems*, vol. 23, p. 157-195, 2013.

[10] *R. Ghostine, J.-M. Thiriet, J.-F. Aubry, et M. Robert,* « A Framework for the Reliability Evaluation of Networked Control Systems *», Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea,* vol. 41, nº 2, p. 6833-6838, 2008.

[11] *N. Yel, R. Noureddine, et F. Noureddine,* « Model-based approach for fault monitoring in machining », Euro-*Mediterranean Seminar on Environment and Industrial Safety*, 2011.

[12] *E. Ruijters et al.*, « FFORT: A benchmark suite for fault tree analysis ». The 29th European Safety and Reliability Conference, Singapore, 2019.

[13] *M. Bouissou, Y. Dutuit, et S. C. Maillard*, « Reliability Analysis of a Dynamic Phased Mission System: Comparaison of two approaches », *MMR2004*, vol. 2, p. 1-19, 2004.

[14] *M. Talebberrouane et Z. Lounis,* « Safety assessment of flare system by fault tree analysis », Journal of Chemical Technology & Metallurgy, vol. 51, nº 2, p. 229-234, 2016.

[15] *M. Talebberrouane, F. Khan, et Z. Lounis*, « Availability Analysis of Safety Critical Systems Using Advanced Fault Tree and Stochastic Petri Net Formalisms », Journal of Lass Prevention in the Process Industries 44, 2016.

[16] *W. S. Lee, D. L. Grosh, F. A. Tillman, et C. H. Lie*, « Fault Tree Analysis, Methods, and Applications-A Review », IEEE Transactions on Reliability, vol. 34, nº 3, p. 194-203, 1985.

[17] *G. S. Hura et J. W. Atwood*, « The Use of Petri Nets to Analyze Coherent Fault Trees », IEEE Transactions on Reliability, vol. 37, nº 5, p. 469-474, 1989.

[18] *S. Kabir et Y. Papadopoulos*, « Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review », Safety Science, vol. 115, p. 154-175, 2019.

[19] *M. Monnin, D. Racoceanu, et N. Zerhouni*, « Overview on diagnosis methods using artificial intelligence application of fuzzy Petri nets », IEEE Conference on Robotics, Automation and Mechatronics, vol. 2, p. 740-745, 2004.

[20] *N. Palluat, D. Racoceanu, et N. Zerhouni*, « Diagnosis aid system using a neuro-fuzzy Approach », *A*dvances in Maintenance and Modeling, Simulation and Intelligent Monitoring of Degradation, 2004.

[21] *D. Abdoulaye, N. Farid, N. Rachid, K.-F. Bernard, et B. Diakarya*, « Proportional hazard model for cutting tool recovery in machining », Journal of Risk and Reliability, 2019.

[22] *V. Upadhyay, P. K. Jain, et N. K. Mehta,* « In-process prediction of surface roughness in turning of ti-6al-4v alloy using cutting parameters and vibration signals. », *M*easurement, vol. 1 (46), p. 154-160, 2013.

[23] *N. M. Vaxevanidis, J. Kechagias, et N. A. Fountas, D. E. Manolakos* « Evaluation of machinability in turning of engineering alloys by applying artificial neural networks. », The Open Construction & Building Technology Journal, vol. 8 (1), p. 389-399, 2014.

[24] *A. Xavior et M. Adithan,* « Determining the influence of cutting fluids on tool wear and surface roughness during turning of AISI 304 austenitic stainless steel. », J Mater Process Technol, vol. 209, p. 900-909, 2009.