SCIENCE SET
OPEN ACCESS PUBLISHERS

ISSN: 3067-2384
Research Article

World Journal of Sensors Network Research

λPN-Based Modeling and Analysis of a Machining Robot Cell

Zohra Mehar[1]*, Rachid Noureddine[2], & Farid Noureddine[3]

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

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

***Corresponding author:** Zohra MEHAR, Lab of Industrial Production and Maintenance Engineering, Institute of Maintenance and Industrial Safety, University of Oran 2Mohamed Ben Ahmed; P.B 1015 El M'naouer 31000 Oran Algeria

Submitted: 07 July 2025 **Accepted:** 18 July 2025 **Published:** 28 July 2025

https://doi.org/10.63620/MKWJSNR.2025.

Citation: *Zohra, M., Noureddine, R., & Noureddine, F. (2025). λPN-Based Modeling and Analysis of a Machining Robot Cell. Wor Jour of Sens Net Res, 2(4), 01-10.*

Abstract

This paper introduces a novel approach for modeling and analyzing faults within a machining robot cell. We propose a hybrid analysis thatcombines Petri Nets (PNs) with Fault Tree, resulting in a new technique called Lambda Petri Net (λPN). This work was been implemented in the LabView environment. The λPN demonstrated strong modeling capabilities 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: Fault Tree, Lambda Petri Net, Machining Robot Cell, Modeling, Monitoring.

Introduction

The increasing complexity and frequent reconfigurations of modern production systems, such as machining robot cells, necessitate 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 illustrates the monitoring components and focuses on our contribution to the diagnostic system of the machining robot cell. This system incorporates qualitative external inputs from a Fault Tree. The output will find the different possible causes associated with a specific 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.

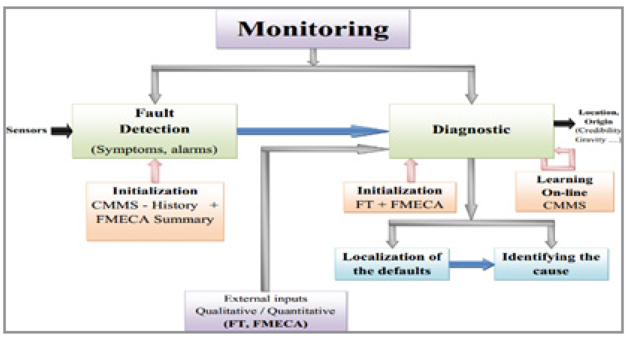


Figure 1: Block diagram of the monitoring system

λPN Formalization

According to, different PNs have particular structures, i.e., they have characteristics and properties that most common networks do not have [1, 2]. 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, including 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 fault 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: [1] [3]
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 ϵ P.
$F:(P\times T)\cup(T\times P)\rightarrow N_0$ Defines the set of directed arcs weighted by non-negative integer values.
The incidence matrix (W) of the Petri Net translates the global cost of firing a transition for each place.
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 (Pj, Tj) can be represented in a matrix with
$W^{+}(i,j)=Pre(P_i,T_j)$
The connecting arcs of Places to Transitions Post (Pi, Tj) can be represented in a matrix with
$W^{-}(i,j)=Post(P_i,T_j)$
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
$M_K=M_0+(W\times\underline{S})$ (3)
Generic properties of a PN are: liveness, deadlocks, reversibility, repetitive components, effectiveness, reachability, and safety, [4].

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 the 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, specifically 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 that follow from the logical expression of the fault tree of the monitored system. The λPN tool models the set of 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 through the transfer of fault signals.

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

Case Study

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

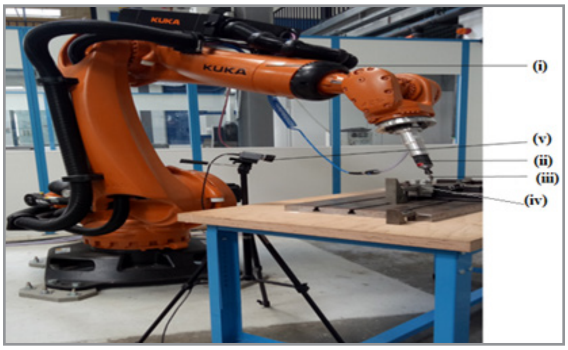


Figure 2: Machining robot cell

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

FT Analysis

The FT is a method of deductive analysis used also in dependability. This is a method of analysing the reliability, availability, and security of more widely used systems, [5, 6]. 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 in this very element or to a fault in 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 OU b2) OU (c1 OU c2) OU (d1 OU d2) OU (e1 OU e2) OU (f1 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}.

$$F=\underbrace{\left(\underbrace{(b1+b2)}_{b}+\underbrace{(c1+c2)}_{c}+\underbrace{(d1+d2)}_{d}+\underbrace{(e1+e2)}_{e}+\underbrace{(f1+f2)}_{f}\right)}_{a} \quad (4)$$

Such as:

b = b1 + b2

c = c1 + c2

d = d1 + d2

e = e1 + e2

f = f1 + f2

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

F=a (6)

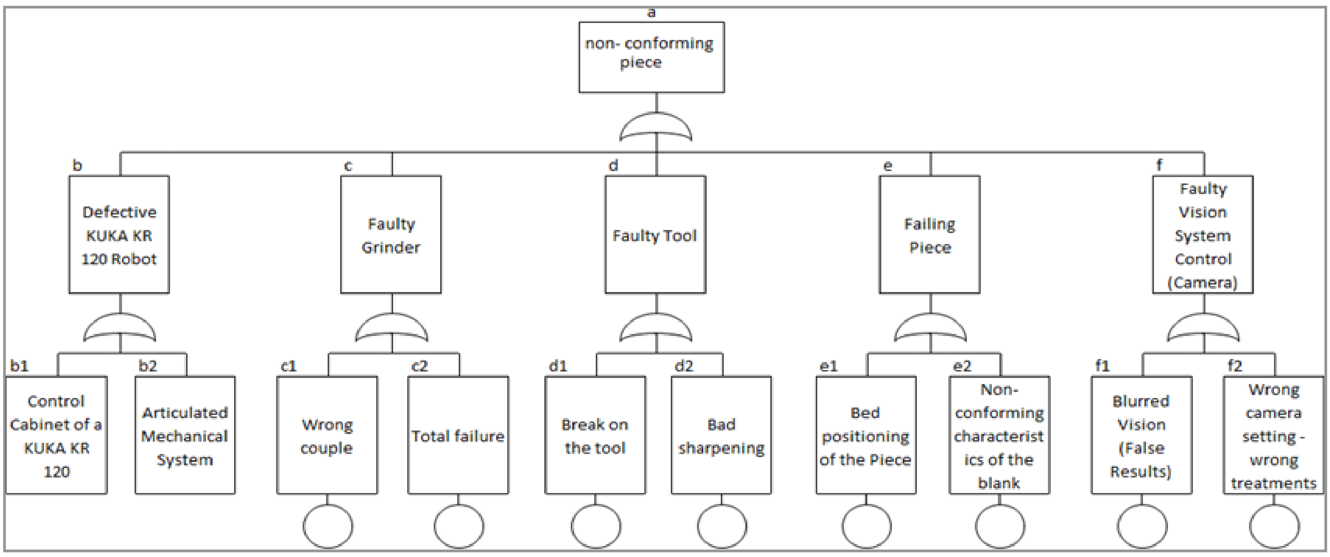


Figure 3: Fault Tree of the robot cell, corresponding to F

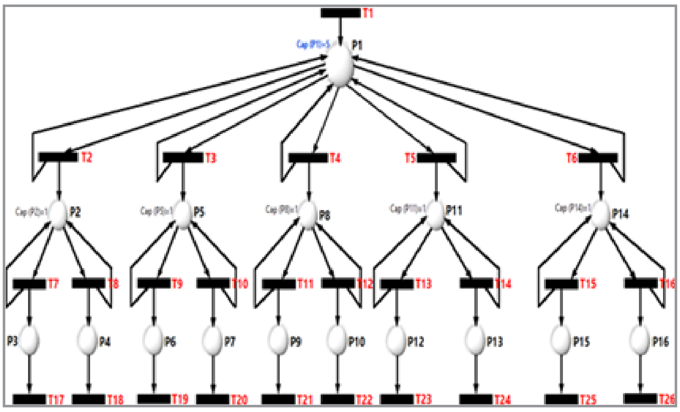


Figure 4: Petri Net before firing of transitions

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 λPN model. The possible transformation from ordinary PN to high-level λPN is shown in figures 5 and 6. Our λPN before firing of transitions is shown in figure 5.

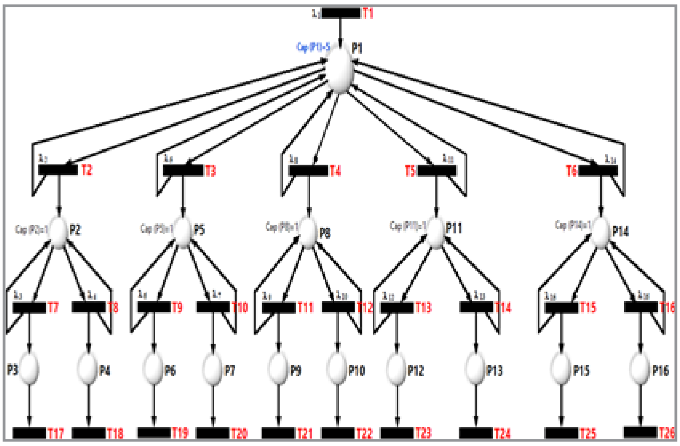


Figure 5: λPN before firing of transitions

Our λPN after firing of transitions is shown in figure 6.

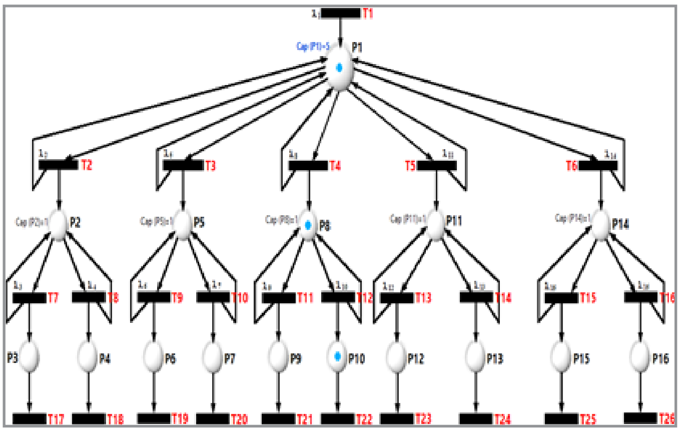


Figure 6: λPN after firing of transitions

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 1 describes the significance of each place and the different failure rates used in the Lambda Petri Net of our system are shown in table 2.

Table 1: 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

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.

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-λPN under LabView will be as follows, figure 7:

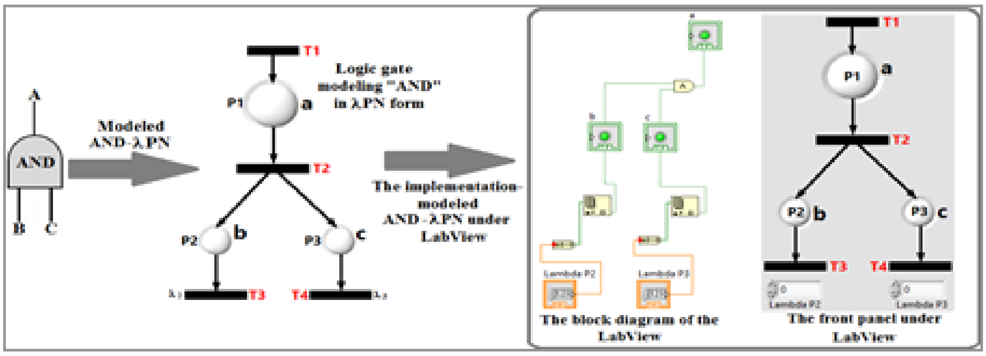


Figure 7: Transformation of the « AND » logical gate of the FT into the λPN

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

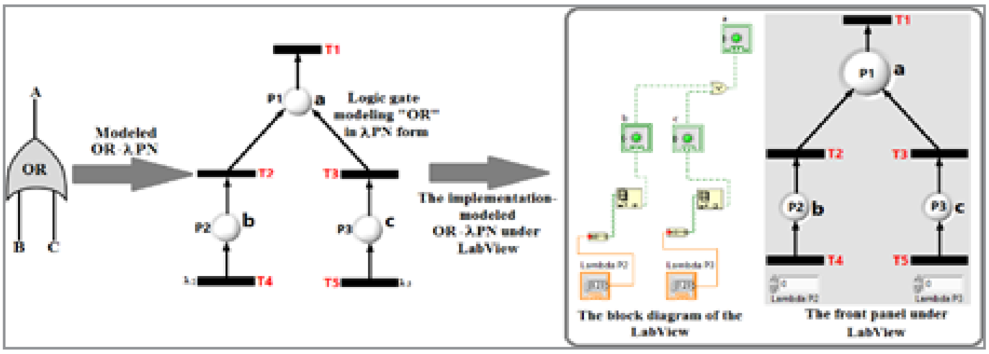


Figure 8: Transformation of the « OR » logical gate of the FT into the λPN

λPN - LabView Implementation

We have associated the λPN with 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 λPN: 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

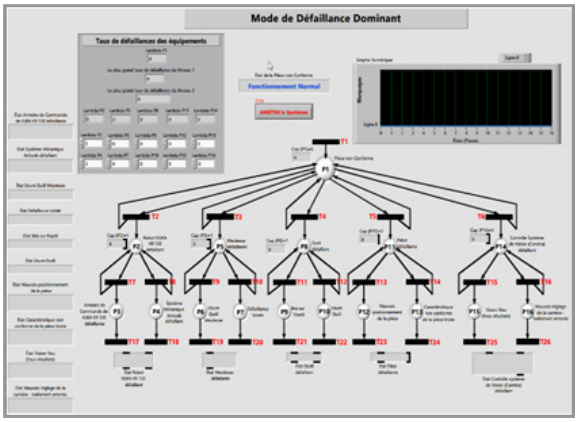


Figure 9: Front panel under LabView of the PN system - Modeling without fault

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.

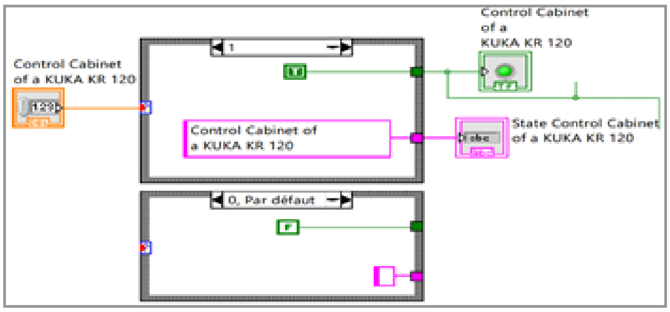


Figure 10: LabView block diagram of the λPN system

Simulation and Results

As shown in figure 11, 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 us 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.

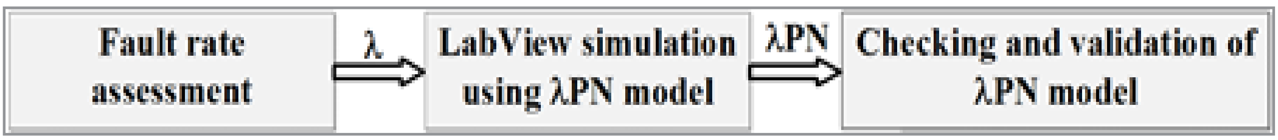


Figure 11: The Steps of the proposed diagnostic system

Failure Rate

According to the hierarchical expertise of robustness, the failure data for each component is given in the following table 2. Vibrations are the most critical events. Vibrations in radial direction have the greatest value of failure rate (λ 6 = 0.0009 and λ 10 = 0.0009).

Table 2: 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 2, we obtained the following failure rates shown in table 3. The faulty tool is the most critical event with a failure rate (λ_{8} = 0.0017).

Table 3: 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				
λ1		0.0054		

Lab View Simulation

The obtained simulations and results are shown in figure 12. 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 λ_{1} = 0.0054. After comparing all failure rates of level 1, the highest failure rate is λ_{8} = 0.0017, so the highest level 2 failure rate is $\lambda_{6} = \lambda_{19}$ = 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

Figure 12. Front panel under LabView of the λPN system - Modeling with fault

- 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.

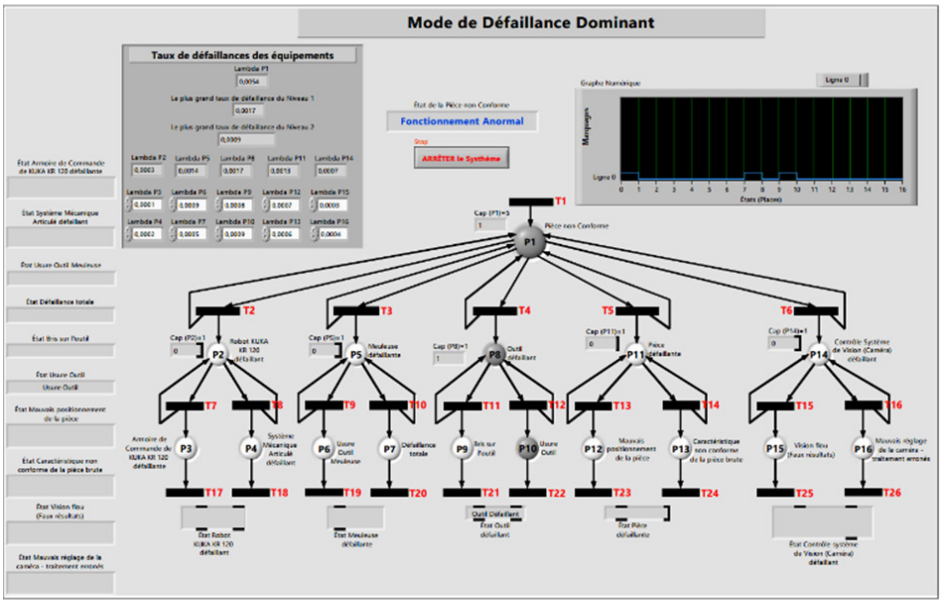
- To carry out a deductive analysis in our λPN, we proceed by firing transition (T1), and then place (P1) has a token. If there is a token in place (P1), then we have to go to transition (T4) directly. If (T4) is fired then place (P8) has a token. If place (P8) has a token, then (P10) also has a token after firing transition (T12). This diagnostic process is obtained through the return arcs, building our Lambda Petri Net.

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 M0 = [0000000000000000].
- The marking of our λPN after the firing, corresponding to figure 10, is M1 = [1000000101000000].



The marking of our λPN after the firing, corresponding to figure 10, is M1 = [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}

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, and 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.

Pre-Incidence Matrix λPN

Following equation (3), we obtained the Pre-incidence matrix (W+) of our Lambda Petri Net (λPN). This matrix, which has 17 rows and 28 columns, is represented by:

W⁺ =

	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
P1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P3	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
P4	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
P5	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P6	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
P7	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
P8	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P9	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
P10	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
P11	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
P12	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
P13	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
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
P15	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
P16	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

Post-Incidence Matrix λPN

Following equation (3), we obtained the Post-incidence matrix (W^-) of our Lambda Petri Net (λPN). This matrix, which has 17 rows and 28 columns, is represented by:

W⁻ =

	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
P1	1	1	1	1	1	1	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	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P3	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
P4	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
P5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P6	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
P7	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
P8	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P9	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
P10	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
P11	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
P12	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
P13	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
P14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	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	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

Incidence Matrix

Define abbreviations and acronyms the first time they are used in the text, even after they have already been defined in the abstract.

The incidence matrix represents the relationships between places and transitions in a Petri Net, allowing for the calculation of 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 =

	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
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
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
P3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
P5	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	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
P8	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
P9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
P11	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
P12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0
P13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0
P14	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
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	-1	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	-1

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): Mk = M0 + (W × S1).

So we use the M1 marking for the calculation of the transition vector.

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]

Consider sequence S1= T1 T4 T12

So we can write: S1= [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]T

$$M_k = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\1\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix} + W \times \begin{bmatrix}1\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\1\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix} + \left[\begin{array}{cccccccccccccccccccccccccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1
\end{array}\right] \times \begin{bmatrix}1\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} + \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\1\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\1\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

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 Mk= [1000000101000000] > 0 then our λPN 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.

Conclusion

Modeling and simulation using Petri nets are powerful tools for assessing the performance of complex systems; providing an efficient mathematical formalism for modeling and representing system failures. This article details the development of a diagnostic strategy for a machining robot cell, using a Petri Net generated from an Fault Tree within the LabView environment.

Our process involved transforming the fault tree into a Petri Net using established equivalences, then determining the failure rates applicable to Lambda transitions. We conducted simulations after implementing the system in LabView, to obtain results that proved 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.

References

1. Mehar, Z. (2025). Contribution à la mise en œuvre et surveillance d'une cellule robotisée de parachèvement de pièces (Doctoral dissertation). Université d'Oran 2 Mohamed Ben Ahmed, Institut de Maintenance et de Sécurité Industrielle.
2. David, R., & Alla, H. L. (1992). Du Grafcet aux réseaux de Petri (2nd ed.). France.

3. Mehar, Z., Noureddine, R., & Noureddine, F. (2023). Petri net modeling of a machining robot cell. UPB Scientific Bulletin, Series D: Mechanical Engineering, 85(4), 89–108.
4. Hongwei, L., Qiang, L., Xinjian, Z., & Jiaguang, S. (2013). Concurrency bugs in multithreaded software: Modeling and analysis using Petri nets. Discrete Event Dynamic Systems, 23, 157–195.
5. Mehar, Z., Noureddine, R., & Noureddine, F. (2022). Diagnostic by fault tree and Petri nets of a robotic machining cell. Algerian Journal of Research and Technology, 6(1), 10–18.
6. Ruijters, E., Guck, D., Drolenga, P., & Stoelinga, M. (2019). FFORT: A benchmark suite for fault tree analysis. In Proceedings of the 29th European Safety and Reliability Conference (ESREL), Singapore.