applied sciences

MDPI

Article

Reliability, Availability, and Maintainability Assessment of a Mechatronic System Based on Timed Colored Petri Nets

Imane Mehdi [1,*], El Mostapha Boudi [1] and Mohammed Amine Mehdi [2]

[1] Quality Safety and Maintenance Laboratory, Mohammadia School of Engineers, Mohammed V University, Rabat 10090, Morocco
[2] Civil Engineering Laboratory, Mohammadia School of Engineers, Mohammed V University, Rabat 10090, Morocco
* Correspondence: imane_mehdi@um5.ac.ma

Abstract: The mechatronic industry is currently subject to huge changes challenging it to offer products matching individual customer requirements at competitive prices. The design of such products calls for sophisticated and complex components integration following different technologies. Since we are on the cusp of the Fourth Industrial Revolution, in which the world of mechatronic production, network connectivity, the Internet of Things, and cyber-physical systems are correlated, the complexity of these systems increases exponentially, and we are talking about advanced mechatronic systems. To assist these changes, various methods, sweeping all project phases, are used by business houses. Predictive dependability assessment in the earlier design stage is considered a powerful metric used to evaluate the performances of different kinds of mechatronic products before the production phase. Altogether, dependability analysis ties the design directly to the desired functionality, operability, and integrity of the system. This paper explores an approach to assessing the dependability attributes, reliability, availability, and maintainability (RAM), of repairable mechatronic systems based on timed colored Petri nets and a Monte Carlo simulation, integrating simultaneously diverse components technologies: mechanical, electronic, and software. The proposed approach is tested taking the case of a regenerative braking system. The methodology appears to be efficient for evaluating predictive RAM indicators (MTTFF, MTTR, MTBF...) for the whole system and for each individual component separately.

Keywords: dependability; reliability; availability; maintainability; Petri net; colored Petri nets; mechatronic systems; Monte Carlo simulation

check for updates

Citation: Mehdi, I.; Boudi, E.M.; Mehdi, M.A. Reliability, Availability, and Maintainability Assessment of a Mechatronic System Based on Timed Colored Petri Nets. *Appl. Sci.* **2024**, *14*, 4852. https://doi.org/10.3390/app14114852

Academic Editor: Benoit Lung

Received: 4 April 2024
Revised: 22 May 2024
Accepted: 29 May 2024
Published: 4 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Mechatronic systems are one of the cornerstones of the contemporary industrial development. They involve the concurrent and synergistic use of mechanics, electronics, automation, and information technology during products' design and manufacturing [1]. Our day-to-day activities and scheduling depend as never before on the smooth performance of such systems. Their design calls for a sophisticated components integration following different technologies, in view of producing a competitive product of the highest quality. The approach of joining these different technologies together seems quite simple but is indeed full of challenges, due to the considerable additional complexity arising from the integration process, as the difficulty of dealing with system dynamics, considering the existing interactions between functional units, and ensuring that the system can be exhaustively tested. Every change in this complexity level will, in fact, introduce new emerging issues depending on the intensity of the integration process and the final mechatronic system's functional scope.

However, the industry is currently subject to continuous large-scale changes triggered by several persistent global megatrends which will pose a significant challenge to the

future context of mechatronic industries. For instance, the advent of Industry 4.0, which could be defined as the technical integration of cyber-physical systems in production and supply chains, and the application of the Internet of Things in industrial operations [2], enables the integration of information technology with industrial mechatronic technology, by stressing the importance of cyber-physical systems and connectivity, which will, thereby, have a miscellaneous impact on the mechatronic industry and which will definitely impinge on the complexity of mechatronic systems. This is prompting mechatronics designers to rethink the way mechatronic systems are perceived, designed, manufactured, and utilized. In this new perspective, we are talking about advanced mechatronics systems, rather than mechatronics systems focusing on developing intelligent autonomous products and boosting product features by furnishing powerful applications for sensing and automated learning, diagnosis, detection, optimization, and configuration, requiring low or even no user interaction through applications [3].

Following this background, more customer quality requirements are established to further implement the tailored solutions needed for the market demand. One of the important requirements of mechatronic system development is dependability. Its evaluation at the early design stage is an important challenge related to quality of service, functional life, maintenance, safety, and all the relevant cost factors. In fact, the predictive reliability of a mechatronic system requires knowledge of all failure and repair mechanisms affecting its components, for each integrated technology, together with the associated reliability and repair laws. Thus, the required performance of mechatronic products cannot be achieved without interdisciplinary collaboration.

In light of this, mechatronic industries, and product designers in particular, are confronted with new requirements and highly conscious customers. The faster designers can identify risks in their mechatronic system concepts, the more likely they are to satisfy market demands. In short, the designer must be able to identify, evaluate, and prevent failures, losses, and hazards in their process and product definitions from the earliest design stages to achieve significant benefits considering the synergistic combination of mechatronic systems and the specificities for each component separately. Among the appropriate tools devoted to this purpose, we find Petri nets.

Several studies have focused on Petri nets. Petri net models were developed by [4] to assess the resilience of nuclear power plants, and they assess the ability of a system to respond to disruptive events and focus on both the speed and efficiency used to restore the system to its pre-event operating state. The method involves three steps: firstly, the vulnerability of subsystems to external disruptive events is defined using probabilities; secondly, the responses of the reactor system are simulated to reduce the impact of events on the system, and the physical state of the reactor is assessed; lastly, the process of maintenance and restoration of the system is modeled. Reliability studies of photovoltaic systems have been carried out by [5] using Petri nets and considering several failure modes for each repairable component. In addition, a sensitivity study of the various factors has been conducted to highlight the influence of input data on the lifetime and reliability of a photovoltaic module. Given that a key element for a system to achieve its design reliability is adequate logistical support, the author proposes in [6] a methodology for coupling a system-oriented approach to spare management, defining analytically which spare parts combination is required to keep the system at a given operational availability level for a particular scenario, with reliability and maintainability simulation using Monte Carlo and stochastic Petri nets. The author in [7] presents a method for modeling a reactor protection system during maintenance by employing Petri nets to provide a reliability calculation model. The same Petri net model is simulated to study the influence of maintenance status and maintenance compliance distribution probability on the system's availability. The author in [8] suggests an approach to evaluate the reliability and availability of some typical subsea safety systems. This approach works by first identifying the stakeholders' requirements, and then generating various design alternatives. The method Design Failure Mode, Effects, and Criticality Analysis DFMECA was used for dysfunctional analysis. To

estimate the failure rates, the author took the original data from an existing recognized database for subsea applications together with the judgment from designers. To model and estimate reliability and availability, the stochastic Petri Net was chosen. The author in [9] proposes a methodology, based on stochastic and deterministic Petri nets, to evaluate the reliability, availability, and global production time of an industrial renewable energy system. It is primarily based on a qualitative analysis, including both functional and dysfunctional analysis, allowing for the identification of failure modes, this is followed by a quantitative analysis to evaluate the performances. Ref. [10] suggests a hybrid graphical formalism that enhances the capabilities of stochastic Petri nets with those of Bayesian networks. It enables the analysis of continuous input data without the necessity of a time-slicing discretization process. The model is used to assess reliability and dynamic safety. Ref. [11] provides a generalized approach to dynamic and dependent tree theory that is intended to overcome the restrictions of an ordinary fault tree and simplify the modeling of subsystem dependencies avoiding the representation of their individual components by leveraging the features of stochastic Petri net, Markov chains, and binary decision diagrams. The suggested approach comprises seven steps and is essentially founded on the identification of train components covered by the dependency relationship and the extraction of relative sub-trees in the fault tree. The work of [12] is built around the application of a systematic approach to evaluate the reliability of mechatronic systems in the design phase. It commences with a qualitative functional and dysfunctional analysis, then proceeds to a dynamic modeling of the mechatronic system and its various functional and dysfunctional transitions using stochastic Petri nets and different probability distributions. Ref. [13] presents a framework for modeling the reliability of mechatronic systems throughout their life cycle. Based on the data records, and both functional and dysfunctional analysis of the system, it estimates the predictive reliability using determinist stochastic Petri nets. Then, based on data from reliability tests with the precedent predictive reliability, the experimental reliability is estimated using the Bayesian restoration maximization method. Operational reliability is also obtained with the BRM method, using previously defined experimental reliability and data extracted from experience feedback. Ref. [14] develops a methodology for modeling the dependability of two-feedwater nuclear turbo-pumps using hierarchical, timed, colored Petri nets and probabilistic distributions. A physical model, a logic model, and a top model were constructed, and Monte Carlo simulations were performed to evaluate different dependability mean times. Ref. [15] gives an insight into probabilistic models for assessing the dependability of industrial systems. They first introduce several static Boolean approaches, such as fault tree analysis, reliability block diagram, and event trees, then emphasize dynamic models featuring examples of Markovian approaches such as Markov graphs as well as behavioral approaches using stochastic Petri nets and Monte Carlo simulations.

The above-mentioned works are founded on the use of Petri nets in the context of assessing the dependability of industrial systems. The authors suggest different methods for assessing dependability, although, in most of the cited cases, they focus on specific dependability attributes while neglecting others. For instance, Petri net models are developed by [4] to evaluate losses incurred by extreme events, mitigation processes, the health status of the systems involved, and recovery and maintenance processes. However, other dependability attributes, such as reliability and availability, have not been assessed. From the same perspective, ref. [5] presents a methodology for assessing the reliability of a photovoltaic module that is subjected to a stochastic condition using Petri nets. Yet, the author has not addressed other dependability attributes such as availability and maintainability. Ref. [7] provides a reliability and availability calculation model during the maintenance of a reactor protection system. Nevertheless, he did not consider the maintainability attribute. Similarly, ref. [8] suggests a reliability and availability estimation based on DFMECA and stochastic Petri nets, and the author did not evaluate maintainability and other attributes. Ref. [9] proposes a methodology to assess reliability, availability, and global production time using stochastic and deterministic Petri nets without considering other dependability attributes, like maintainability. Ref. [10] suggests a model allowing for the assessment of

reliability and dynamic safety for a system without taking into consideration availability and maintainability. Based on fault Petri nets and Markov chains, ref. [11] provides a generalization of the ordinary fault tree to assess reliability including dependencies between the basic events, variable failure, and repair rates, but maintainability and availability have not been addressed.

The reliability assessment of mechatronic systems, while considering their multi-technological characteristics, has been addressed in the works of [12,13]. In fact, ref. [12] proposes a global methodology to assess the predictive reliability of mechatronic systems considering their dynamic aspect in the design stage, while [13] proposes an overall methodology to assess the reliability of mechatronic systems throughout their life cycle. Both suggested methodologies focused on reliability attributes only.

Refs. [6,14] estimate the dependability attributes: reliability, availability, and maintainability. Ref. [6] evaluates reliability availability and maintainability including spares management of system of systems using stochastic Petri nets and Monte Carlos simulations. Ref. [14] suggests an overall approach to assessing the dependability of feedwater pumps of a nuclear power plant using a reliability block diagram and stochastic and deterministic colored Petri nets. However, both methodologies were not applied to mechatronic systems and, thus, the specificities of such systems have not been referred to and accounted for, as the existing synergies come from different technologies: mechanical, electronic, and software.

Unlike ordinary Petri nets, where tokens are trivialized, they constitute distinct entities in colored Petri nets. They permit to [16] perform the following:

- Track their movement through the network;
- Assign them properties that may vary during the Monte Carlo simulation;
- Validate the same transition from several different tokens.

Since mechatronic systems are complex systems involving different components and various technologies, from the point of view of their modeling the use of colored Petri nets is very appealing, as it reduces the size of models and allows for Monte Carlo simulations.

Within the scope of this work, some dependability attributes will be evaluated. Particularly, a predictive reliability, availability, and maintainability assessment of an advanced mechatronic system based on timed colored Petri nets in an earlier design stage will be presented. Initially, it is pivotal to explore Petri nets and timed colored Petri net definitions together with key related concepts. Then, the different steps of the approach adopted will be described. Next, a mechatronic system will be investigated under the umbrella of the proposed methodology. Finally, discussions, conclusions, and an outlook on the further research needed to complete this work will be drawn.

2. Materials and Methods

2.1. Timed Colored Petri Net (Timed CPN)

Colored Petri net is a language for modeling discrete events. It merges the advantages of Petri nets with the features of a high-level programming language. In fact, Petri nets constitute the essentials of graphical presentation and provide the key features for modeling communication, synchronization, and concurrency, while CPN ML, which relies on the Standard ML functional programming language, offers precepts for identifying data types, outlining data processing, and building compact configurable models [17].

2.1.1. Timed Colored Petri Net, Formal Definitions:

Timed Multiset

Considering a non-empty set S and a set of time values $\mathbb{T}$. The function $tm : S \times \mathbb{T} \to \mathbb{N}$ assigning a non-negative integer $tm(s,t) \in \mathbb{N}$ to each element $(s,t) \in S \times \mathbb{T}$ is a timed multiset. This sum should be is finite for all $s \in S$. The non-negative integer $tm(s)$ is the number of appearances (or coefficient) of s in tm [18].

$$tm(s) = \sum_{t \in \mathbb{T}} tm(s,t) \quad (1)$$

Next, we refer to EXPR as the set of expressions provided by the language employed in inscription (e.g., CPN ML in the case of CPN Tools), and to Type[e] as the type of an expression e ∈ EXPR, i.e., the type of values obtained when e is evaluated. The set of free variables of an expression e is named Var[e], and the type of a variable v is Type[v]. The variables of a transition t are denoted Var(t).

Timed Colored Petri Net

A timed colored Petri net is defined as follows [19]:

$$CPN = (P, T, A, \mathrm{S}, V, C, G, E, I) \tag{2}$$

where:

- P: is a finite state of places;
- T: is a finite state of transitions, and $\mathrm{P} \cap \mathrm{T} = \emptyset$;
- $\mathrm{A} \subseteq \mathrm{P} \times \mathrm{T} \cup \mathrm{T} \times \mathrm{P}$: is a set of directed arcs;
- S: Non-empty color sets finite set;
- V: Typed variables finite set where Type[v] ∈ S for v ∈ V;
- C: $\mathrm{P} \rightarrow \mathrm{S}$: Function of color set attributing for each place a color set;
- G: $\mathrm{T} \rightarrow \mathrm{EXPR_v}$: Guard function providing for each transition t a Boolean guard;
- The set of free variables appearing in a guard is required to form a subset of V, it is required that $\mathrm{G(t)} \in \mathrm{EXPR_v}$;
- $\mathrm{E} : \mathrm{A} \rightarrow \mathrm{EXPR_v}$: Function of arc expression defining for each arc a, a corresponding arc expression E(a), and $\mathrm{E(a)} \in \mathrm{EXPR_v}$ is required. For an arc (p,t) ∈ A connecting a place p ∈ P to a transition t ∈ T, it is required that the type of arc expression is the multiset type over the color set C(p) of the place p. $\mathrm{Type[E(a)]} = \mathrm{C(p)_{MS}}$ if p is untimed, and $\mathrm{Type[E(a)]} = \mathrm{C(p)_{TMS}}$ if p is timed.
  where p is the place linked to the arc a. MS and TMS, multisets and timed multisets, respectively;
- $\mathrm{I} : \mathrm{P} \rightarrow \mathrm{EXRP}_{\emptyset}$: initialization function, attributing an initialization expression I(p) to each place p which is required to evaluate to a multiset over the color set of the place p, where $\mathrm{Type[I(p)]} = \mathrm{C(p)_{MS}}$ if p is untimed and $\mathrm{Type[I(p)]} = \mathrm{C(p)_{TMS}}$ if p is timed.

Enabling and Occurrence of Timed CPN Steps

In this section, we will see how timed CPN steps are enabled and occur. The concepts of bindings, binding elements, steps, and marking will be defined.

A marking is a function M that maps each place p ∈ P into a multiset M(p) of tokens such that the following is true:

- $\mathrm{M(p)} \in \mathrm{C(p)_{MS}}$ if p is untimed.
- $\mathrm{M(p)} \in \mathrm{C(p)_{TMS}}$ if p is timed.

A timed marking is a pair (M, t^*) where M is a marking and $t^* \in \mathbb{T} \in$ is the value of the global clock.

A binding of a transition t is a function b that maps each variable v ∈ Var(t) into a value b(v) ∈ Type[v]. The set of all bindings for a transition t is denoted B(t).

A binding element is a pair (t, b) such that t ∈ T and b ∈ B(t). The set of all binding elements BE(t) for a transition t is defined by BE(t) = {(t, b) | b ∈ B(t)}. The set of all binding elements in a CPN model is denoted BE.

A step $Y \in BE_{MS}$ is a non-empty, finite multiset of binding elements.

2.2. Predictive Reliability Availability and Maintainability Assessment Using Timed Colored Petri Nets

To assess RAM indicators, we need to have access to the different mean times. They are quantified by the mean time elapsed in the place(s) corresponding to the relevant indicator, we will therefore distinguish, in the subsequent section, between the two methods

employed in this work to evaluate reliability and maintainability as well as the availability indicators.

2.2.1. CPNTools Software (4.0.1) and Its Monitors

CPN Tools is a tool designed to edit, simulate, and analyze hierarchical, timed, and untimed colored Petri nets (CPN or CP-nets). A monitor is a CPN Tools mechanism employed to visualize, test, inspect, or edit a simulation of a CP net. Multiple monitors can be assigned to a particular net. Monitors can check the place markings and the binding elements occurring within a simulation and can subsequently take suitable actions based on practical findings [20].

2.2.2. Reliability and Maintainability Using Timed CPN

We can extract reliability and maintainability data using mean times collection:

Let us take a place P_i of the CPN describing the indicator I to be evaluated. P_I is the subset of all these places. $K^*(P_i)$ the average time elapsed at P_i. It is obtained by the Little formula [15], constituting the ratio of the place P_i average marking, $M^*(P_i)$, and the sum of the input arcs coming from the transition T_j to P_i: $w(T_j, P_i)$, multiplied by the average frequency F_a of input transition T_j of P_i. This sum includes all input transitions of the place P_i, denoted $\circ P_i$.

$$K^*(P_i) = \frac{\mathcal{M}^*(P_i)}{\sum_{T_i \in \circ P_i} w(T_j, P_i) F_a * (T_j)} \tag{3}$$

Let L be a list where we can store $K_I = \sum_{P_i \in P_I} K^*(P_i)$, the different average values of the time elapsed in the place(s) characterizing the indicator I:

The mean time between failures indicator MTBF, KPI_{MTBF} is given by the following:

$$KPI_{MTBF} = hd\ L(K_{MTBF}) - L_{prec}(K_{MTBF}) \tag{4}$$

where:

- hd L(K) is the head of the list where the last value of failure time is stored;
- L_{prec}(K) is the precedent value of the failure time;
  The mean time to the first failure is the value of KPI_{MTBF} where the value of $L_{prec}(K_{MTBF})$ is equal to 0.

The mean time to repair is given by the average repair time values collected in the panne state:

$$KPI_{MTTR} = \sum_{P_i \in P_{Panne}} K^*(P_i) \tag{5}$$

where:

- P_{panne} is the panne places subset.

2.2.3. Availability Using Timed CPN

We obtained unavailability data based on marking invariants records.

Two kinds of invariants (called semi-flows) can be found in a colored Petri net: marking invariants (also known as P-invariants) and firing invariants (also known as T-invariants) [21]. Marking invariants demonstrate the token conservation in a subset of CPN places. Vector $\mathcal{K}$, of dimension equal to the number of places in the CPN, is a P-invariant if, and only if, it fulfils the following equation:

$$\mathcal{K}^t \times \mathcal{C} = \vec{0}, \qquad \mathcal{K} \neq \vec{0} \tag{6}$$

where $\mathcal{C}$ is the incidence matrix of the CPN.

If $\mathcal{K}$ is a marking invariant, then for a given marking denoted $\mathcal{M}_i$ derived from an initial marking $\mathcal{M}_0$, we have:

$$\mathcal{K}^t \times \mathcal{M}_i = \mathcal{K}^t \times \mathcal{M}_0 = n \quad , \qquad n \in \mathbb{N}^* \tag{7}$$

This equation represents a marking invariant, meaning that if $\mathcal{K}$ is a P-invariant of the CPN then the transpose of the vector $\mathcal{K}$ weighted by the marking vector $\mathcal{M}_i$ of the CPN is a constant integer whatever the reachable marking $\mathcal{M}_i$ from the initial marking $\mathcal{M}_0$ [21]. Transitions invariants are the dual of place invariants, they characterize a set of occurrence sequences that have no total impact, i.e., they have the same start and end marking.

CPN Tools does not support the calculation of marking invariants. Therefore, the indicator could be estimated on the basis of collecting the average marking of the place(s) reflecting the condition(s) typifying the indicator in question, divided by the total average marking of all places associated with the marking invariant, i.e., the number of tokens in the subset of places [14]:

$$RAM_{KPI-a} = \mathrm{P}(State_{KPI-a}) = \frac{\mathcal{M}^*(State_{KPI-a})}{\sum_{P_m \in P_{Subset_{KPI_a}}} (\mathcal{M}^*(P_m))} \tag{8}$$

where:

$State_{KPI-a}$ is the state describing the indicator KPI-a.

$\mathcal{M}^*(State_{KPI-a})$ is the corresponding average marking and $P_{Subset_{KPI_a}}$ is the subset of places belonging to the marking invariant.

For instance, unavailability could be obtained using the following equation:

$$RAM_{\overline{A}} = \mathrm{P}(State_{\overline{A}}) = \frac{\mathcal{M}^*(State_{\overline{A}})}{\sum_{P_m \in P_{Subset_{\overline{A}}}} (\mathcal{M}^*(P_m))} \tag{9}$$

2.3. Predictive RAM Assessment Methodology

After conducting qualitative analysis, including functional analysis using the Structured Analysis and Design Technique (SADT), dysfunctional analysis by means of both Failure Mode and Effect Analysis (FMEA) for mechanical and electronic components, and Software Error Effects Analysis (SEEA) for software components [22], this paper presents the quantitative analysis of a mechatronic system using timed colored Petri nets together with Monte Carlo simulations. The steps of our methodology are described in Figure 1 below:

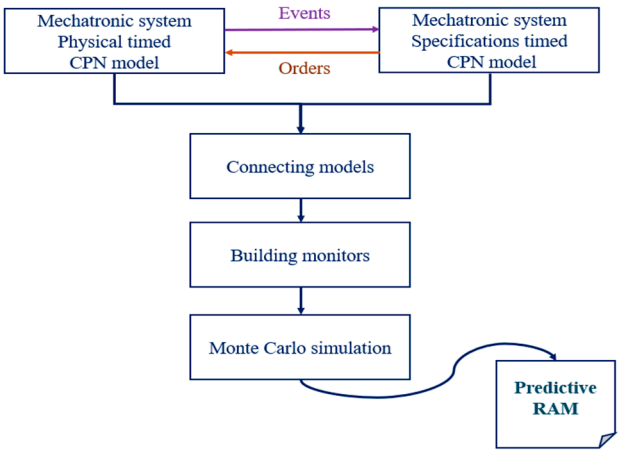


Figure 1. Predictive RAM methodology for mechatronic systems: dynamic modeling phase.

The first step consists of building the two physical and specifications-timed CPN models of the mechatronic system, including a physical and a specification model [14].

The physical model is temporal and stochastic, describing the system's failures and repairs. Whereas, the specification model is logical and deterministic, outlining the control underlying the system's overall behavior.

Second, we need to ensure a connection between both models. Next, monitoring is defined for the whole system and for each component separately in the timed CPN

specifications model. At last, we run Monte Carlo simulations, analyze, and plot data needed to predict the reliability, availability, and maintainability of the mechatronic system.

3. Application

In regenerative braking, kinetic energy from the drive wheels is converted into electrical energy by the electric motor (generator). As a result, part of the energy that is usually wasted as heat under braking is transferred to the battery as electrical energy and then reused [23,24].

The regenerative braking system (RBS) of Figure 2, is the mechatronic system selected to implement our approach. It includes a set of mechanical, electronic, and software components.

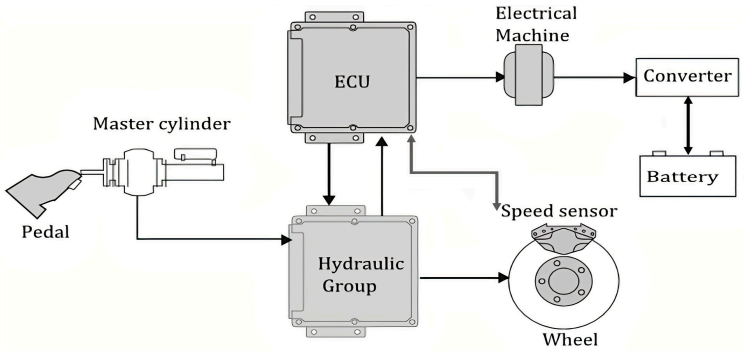


Figure 2. Regenerative braking system.

The RBS components considered in the present work are listed in Table 1.

Table 1. RBS components.

Component	Type	Abbreviation in the Model
Pedal	Mechanical	Ped
Master cylinder	Mechanical	MC
Hydraulic group	Mechanical	HG
Reservoir	Mechanical	Res
Electrical machine	Mechanical	EM
Disc	Mechanical	Disc
Caliper	Mechanical	Cal
Brake Pad	Mechanical	BP
Tyre	Mechanical	Tyre
Speed sensor	Electronic	SS
DC-DC converter	Electronic	DC
Battery	Electronic	Bat
ECU	Electronic	ECU
Data analysis software module	Software	AData
Input Output software module	Software	IO
OS software module	Software	OS

3.1. Timed CP-Net Models of the RBS

The first step of our approach is the construction of the timed CPN physical model of the RBS.

3.1.1. RBS Timed Physical CPN Model

Figure 3 below displays the timed physical CPN model.

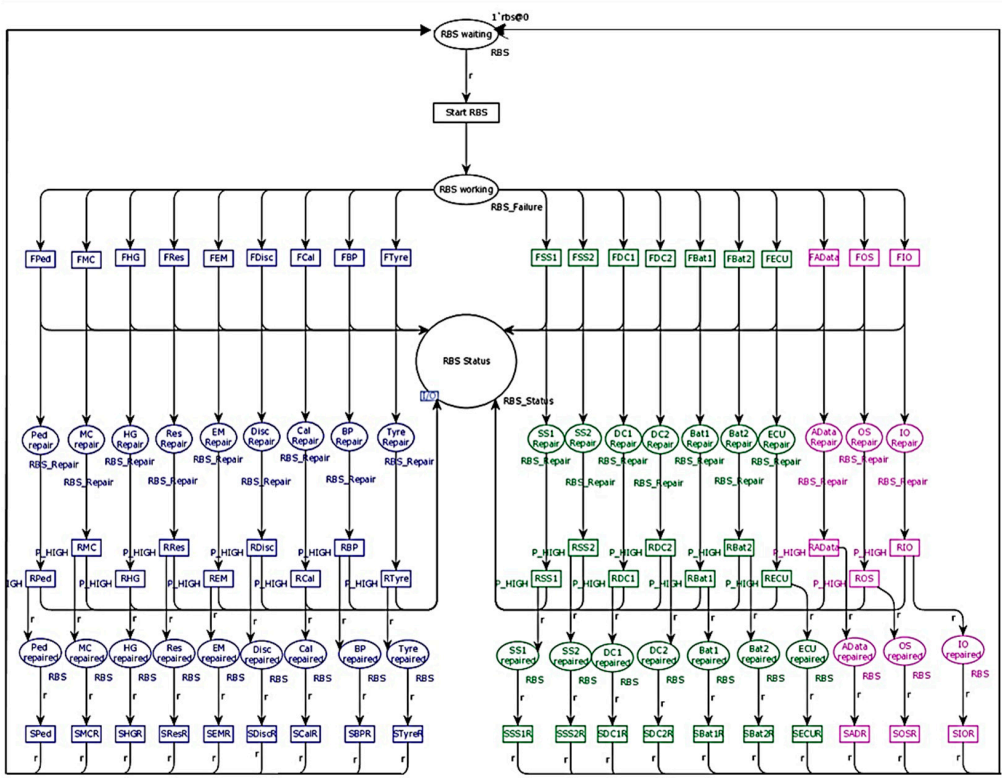


Figure 3. Regenerative braking system: timed CPN physical model.

The places are defined to model the state of the RBS while the transitions represent the events that may occur. The blue, green, and pink colors are used to represent mechanical, electronic, and software components, respectively.

The **RBS waiting** place is initially marked with an **rbs** token of type **RBS** indicating that we have one regenerative braking system at the initial time. Every other place in the physical model is originally unmarked.

We assume that the system is operational at time 0. By removing the **rbs** token from the **RBS waiting** place and placing it in the **RBS working** place, the variable **r** enables the **Start RBS** transition. In the same transition, a transfer function is defined to execute random actions based on Weibull, Exponential, and Exponential-Musa distributions, explained in Appendix A, respectively, for the mechanical, electronic, and software components' times to failure. The minimal time to failure among all components' failure times is obtained using a defined minimum function intended to represent the moment when each RBS component is most likely to fail based on the components' probability distributions and comparing the obtained components' failure time values.

The output token of this transition contains the variable r, components failure times (**Ped_MFT**,**MC_MFT**,**Res_MFT**,**EM_MFT**,**Disc_MFT**,**Cal_MFT**,**BP_MFT**,**Tyre_MFT**,**SS_EFT**, **DC_EFT**,**Bat_EFT**,**ECU_EFT**,**ADat_SFT**,**OS_SFT**,**IO_SFT**), **wait**, which is equal to the minimal time to failure among all failure times, and **wait**, the RBS global simulation time. Indeed, the RBS operates, and following, a delay, **wait**, the failure occurs and a failure transition is enabled.

RBS working place is connected to the component failure transitions (the name of these transitions is preceded by the letter **F**). To select which failure transition will be enabled, a guard condition is defined, specifying that the failure time should be equal to **wait** the minimal failure time value.

Let us take the example of the master cylinder of Figure 4, the transition **FMC** is enabled if the time to failure **MC_MFT** is equal to **wait**.

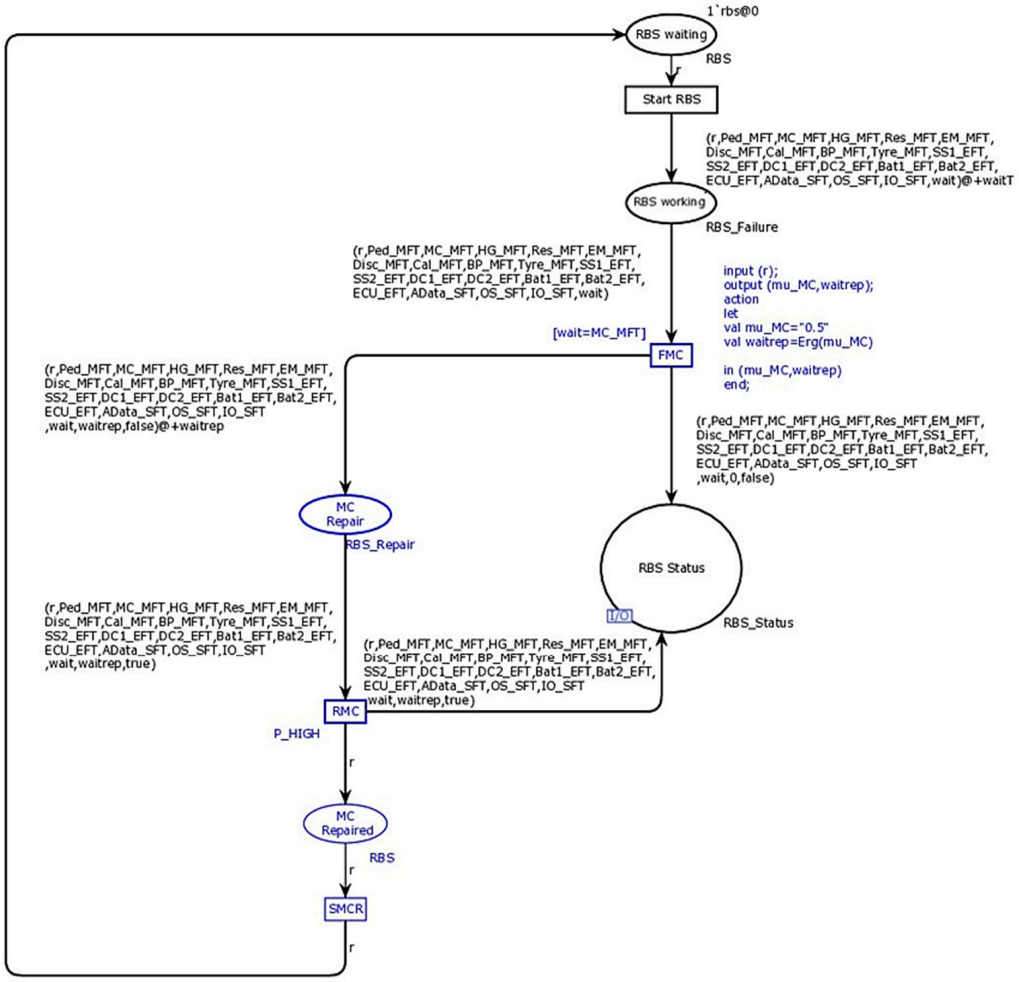


Figure 4. Master cylinder: timed CPN physical model.

Two arcs appear at the output of failure transitions, one is connected to **the RBS Status** place, and the other to the components' repair places.

RBS Status is an important place, the output arc of a failure transition contains the variable **r**, the different component failure times **Ped_MFT**,**MC_MFT**,**Res_MFT**,**EM_MFT**, **Disc_MFT**,**Cal_MFT**,**BP_MFT**,**Tyre_MFT**,**SS_EFT**,**DC_EFT**,**Bat_EFT**,**ECU_EFT**,**AData_SFT**, **OS_SFT**,**IO_SFT**, **wait**, a time **waitrep** (repair time, which is equal to 0 before the repair), and a Boolean false, informing that a failure occurs, this is used to inform the specifications model that a component is down.

The arcs linking the failure transitions and the repair places contain as well the variable **r**, the various components 'failure times, **Ped_MFT**,**MC_MFT**,**Res_MFT**,**EM_MFT**, **Disc_MFT**,**Cal_MFT**,**BP_MFT**,**Tyre_MFT**,**SS_EFT**,**DC_EFT**,**Bat_EFT**,**ECU_EFT**,**AData_SFT**, **OS_SFT**,**IO_SF**, **wait**, **waitrep**, and a Boolean equal to **false.** A transfer function is defined at failure transitions to calculate the repair time needed for each component using Erlang distribution.

If a repair operation occurs in **component name repair** place of type **RBS_Repair** (MC repair place in the case of the master cylinder), and then a repair transition is enabled, after the repair time **@waitrep**, the output arc of the repair transition communicates the information to the place **RBS status**, mentioning that the component is repaired and returning a **true** Boolean value.

The output of the repair places contains the variable **r**, the different failure times, **wait**, **waitrep**, and the repair time **@waitrep**.

The restart of the RBS is ensured via the restart transitions which are executed instantaneously; we assume that the system restarts instantly after a repair.

The restart transitions are identified as **SComponentNameR**, and, in the case of the master cylinder, is **SMCR**.

3.1.2. Timed CPN Specifications Model

The timed CPN specifications model of the RBS is illustrated in Figure 5 (only software components are presented, the modeling methodology remains the same for mechanical and electronic components). It permits the recording of all data needed to carry out the performance analysis. Once a component is broken, the place **RBS Status** receives a token containing the variable **r**, the different failure times **Ped_MFT**,**MC_MFT**,**Res_MFT**,**EM_MFT**, **Disc_MFT**,**Cal_MFT**,**BP_MFT**,**Tyre_MFT**,**SS_EFT**,**DC_EFT**,**Bat_EFT**,**ECU_EFT**,**AData_SFT**, **OS_SFT**,**IO_SF**, **wait**, repair time **waitrep** (which is equal to zero initially), and the Boolean **false**. This information is shared with the component's corresponding failure transition instantaneously (transition name preceded by F, **FMC** in the case of Figure 6). In this context, once the instantaneous failure transition is enabled, the marking size of the panne place, preceded by the P letter (**PMC** in the case of the master cylinder of type **RBS_Panne**), will increase infinitely, which is known as the combinatorial explosion phenomenon [25]. In fact, CPN Tools executes instantaneous transitions before timed ones, then the failure transitions will always be enabled by the place **RBS Status**. We define a complementary place to solve this issue (anti-place in CPN Tools) [26]. The anti-place is looped between failure and repair transitions. They cannot have more tokens than the ones in the anti-place (in our case one token). The failure transitions will be enabled only once, the token from the anti-place will be removed via the variable **SystPanne**. Failure transitions will not be enabled again unless the token is removed from failure transitions and moved to repair transitions. Thus, the place Component name panne has, at maximum, one token.

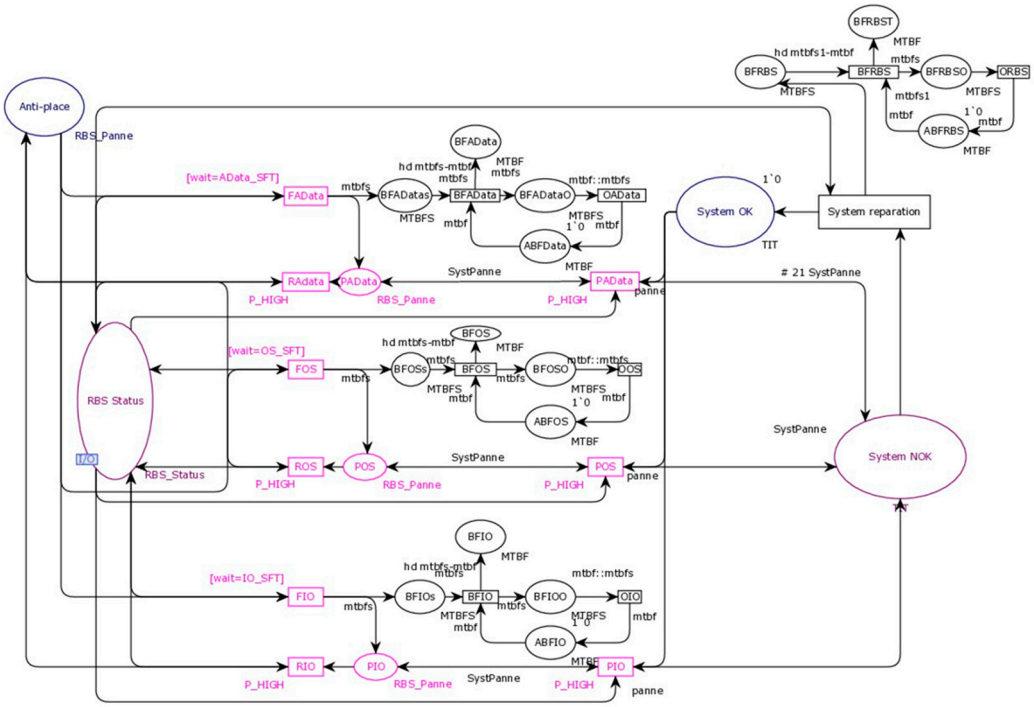


Figure 5. Regenerative braking system software components: timed CPN specifications model.

The output of panne transitions (PMC considering Figure 6) is the **wait** value. Access to the wait value is achieved using **#21 SystPanne**. Then, a token is placed on the place **System KO**, informing the system that a component is broken. Once a repair occurs, the bidirectional arc linking **RBS_Status** and repair transitions (starting by R letter, like **RMC** transition in Figure 6) contains the information about repair time.

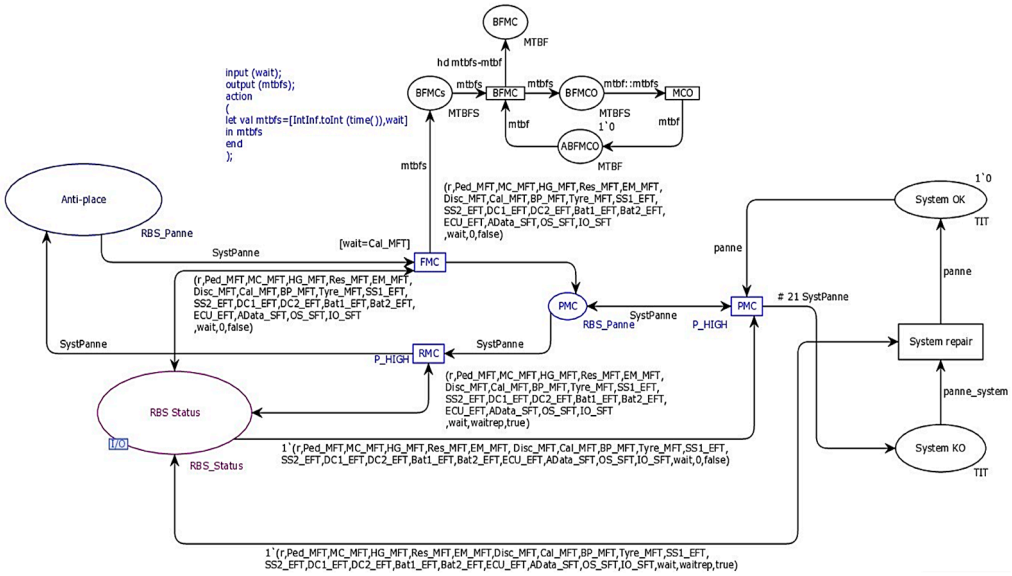


Figure 6. Master cylinder: timed CPN specifications model.

In the same direction, considering that a system fails if one of its components fails, the bidirectional arc between **RBS_Status** place and **RBS Repair** transition contains the same information.

We suppose that repair starts immediately after the occurrence of a failure.

When we have a token in the place **System KO**, and the component is repaired, the **System Repair** transition is enabled. It is about an instantaneous transition. In this situation, we may encounter a concurrency between the **System repair** and repair transitions [27]. Since priority transitions are supported in CPN TOOLs, we can assign priorities to each transition: P_LOW, P_NORMAL,P_HIGH [28], we assign P_HIGH proprieties to our model for panne and repair to avoid this concurrency. Indeed, if a component is broken and then repaired, then the whole system is repaired.

Lists are defined to collect data about components' mean time between failures and the mean time to the first failure. For instance, in Figure 6, the place **BFMCs** of type list **MTBFS** permits the collection of both the first and the current time to failure **wait** and the performance of subtractions to obtain the time between failures. Similarly, a list is defined to collect the mean times to failure for the whole system.

3.2. Connecting Models

Now that we have constructed the two models of our RBS, it is time to connect them. For this purpose, we use a fusion place. The places belonging to a fusion set are termed fusion places and refer to unique compound places. In other words, all instances of fusion set places will automatically be marked in the same way, and, therefore, will have identical color sets and initial markings [19]. In our case, the fusion place of the two timed CPN models is **RBS Status** and the instances of the fusion place **RBS Status** have the same marking, thanks to the fusion set.

3.3. Build Monitoring

3.3.1. Data Collector Monitors

Data collector monitors are destined to obtain numerical data from a model and accomplish statistical calculations for the retrieved data, i.e., to construct one or more performance metrics [20]. We define data collector monitors to obtain the different mean times, the mean time to first failure: MTTFF, the mean time between failures: MTBF, and the mean time to repair: MTTR. For instance, MTTFF monitors shall record the component's first failure time. They are applied to the different failure transitions for the components.

3.3.2. Marking Size Monitors

Marking size monitors are used to count the quantity of tokens at a specific place. In our case, we use this kind of monitors to assess the unavailability and the availability, the times when each component or the RBS is in a broken or working state.

3.3.3. Breakpoint Monitor

The system is intended for infinite simulation. A breakpoint monitor is therefore required to halt simulations. The breakpoint monitor stops the simulation after two years (17,520 h) in our case to not expand the run time during the Monte Carlo Simulation.

4. Simulations and Results

The last step of our methodology consists of running simulations of our timed CPN model. Since we are using the Erlang distribution, the Markovian hypothesis is not satisfied for our model, and we we choose the Monte Carlo simulation to assess predictive RAM indicators. Using Equation (4), we evaluate reliability and maintainability while we assess availability using Equation (8).

The **CPN'Replications.nreplications** function is deployed to automatically run simulations. The Fenton and Griffiths equation is employed to determine the minimum number of simulations, n, needed to achieve a given estimation error with a confidence level (1-α). It is expressed as follows [29]:

$$n = \left(\frac{z_{\alpha/2}\sigma}{e}\right)^2 \quad (10)$$

where:

- σ: Standard deviation;
- e: Standard error.

If we admit a standard error equal to 0,02 σ and a confidence interval of 95% (α = 0.05), the $z_{\alpha/2}$ value for a 95% confidence level is 1.96. We obtain a number of needed iterations equal to 9604, and, thus, we choose 10,000 iterations.

Table 2 below shows the different dysfunctional distributions defined for each type of component, we use the same parameter values mentioned in the work of [12,13] extracted from available databases. We specify that the data are provided as an example, without being truly faithful to reality, the purpose of this section is to apply the methodology for evaluating predictive dependability attributes and assessing the RBS performance.

Table 2. Average parameter values for reliability distributions.

Component	Distribution	Average Parameters Values
Pedal	Weibull	η = 1000, β = 1.5
Master cylinder	Weibull	η = 1000, β = 1.5
Hydraulic group	Weibull	η = 1000, β = 1.5
Reservoir	Weibull	η = 1000, β = 1.5
Electrical machine	Weibull	η = 1000, β = 1.5
Disc	Weibull	η = 1900, β = 1.5
Caliper	Weibull	η = 1900, β = 1.5
Brake pad	Weibull	η = 1900, β = 1.5
Tyre	Weibull	η = 1000, β = 1.5
Speed sensor	Exponential	$\lambda = 3 \times 10^{-4}$
DC-DC converter	Exponential	$\lambda = 7.5 \times 10^{-3}$
Battery	Exponential	$\lambda = 5 \times 10^{-4}$
ECU	Exponential	$\lambda = 50 \times 3 \times 10^{-6}$
AData	Exponential-Musa	$\lambda = 5.04 \times 10^{-4}$, N0 = 10
IO	Exponential-Musa	$\lambda = 5.04 \times 10^{-4}$, N0 = 10
OS	Exponential-Musa	$\lambda = 5.04 \times 10^{-4}$, N0 = 10

We highlight Erlang distributions for repair transitions in the Table 3 below:

Table 3. Average parameter values for Erlang repair distributions.

Component	Average Parameters Values
Pedal	1/2
Master cylinder	1/3
Hydraulic group	1/3
Reservoir	1/2,5
Electrical machine	1/3
Disc	1/1.2
Caliper	1/1.2
Brake pad	1/1.2
Tyre	1/1.5
Speed sensor	1/5
DC-DC converter	1/2
Battery	1/3
ECU	1/5
AData:	1/1.08
IO	1/2.17
OS	1/0.5

Information provided by selected CPN Tools monitors are recorded in separate text files and in a single folder for each simulation. CPN Tools software provides a report file in HTML with different statistical and performance analysis. This report offers an overview of the statistics calculated over the course of a simulation.

We list, in Table 4, the statistical results; 95% CI is intended for the half length of the 95% confidence interval.

Table 4. RBS and component mean times (10,000 replications).

Component	MTTFF	MTBF	MTTR	Unavailability
Ped: Average	1119.90	512.19	2.43	0.000050
95% CI	41.95	26.49	0.11	0.0000016
MC: Average	3901.44	1960.25	4.58	0.002019
95% CI	76.30	66.60	0.16	0.000491
HG: Average	1107.46	513.64	2.42	0.000100
95% CI	41.39	27.82	0.11	0.000036
Res: Average	1105.04	511.70	2.33	0.000023
95% CI	40.98	35.55	0.11	0.000008
EM: Average	1107.54	515.99	2.68	0.000466
95% CI	41.63	27.06	0.14	0.00001591
Disc: Average	853.80	415.99	0.07	0.000021
95% CI	44.19	30.49	0.09	0.0000015
Cal: Average	867.45	404.53	1.28	0.00008
95% CI	43.89	29.55	0.09	0.000004
BP: Average	896.36	424.92	1.25	0.000013
95% CI	45.70	30.98	0.08	0.000007
Tyre: Average	854.08	413.09	1.28	0.000017
95% CI	43.90	30.71	0.09	0.0000011
SS: Average	1129.17	500.65	2.49	0.000123
95% CI	41.74	41.74	0.11	0.000055
DC: Average	1289.13	605.21	4.15	0.000168
95% CI	27.01	8.96	0.0554	0.000063
Bat: Average	1049.05	462.07	3.01	0.000130
95% CI	35.01	20.74	0.11	0.000045
ECU: Average	930.41	445.05	1.59	0.000045
95% CI	43.03	29.54	0.08	0.0000019
AData: Average	1036.06	466.28	2.93	0.000037
95% CI	35.18	21.16	0.11	0.000008
IO: Average	1083.23	494.09	2.42	0.000032
95% CI	31.79	19.29	0.09	0.000009
OS: Average	1044.99	473.97	3.02	0.000107
95% CI	35.22	21.41	0.10	0.000033
RBS: Average	268.79	262.70	4.70	0.006238
95% CI	6.02	1.30	0.07	0.000560

The data obtained are also employed to assess the empirical probability density function (PDF) and the cumulative distribution function (CDF) of the different mean times, and to draw the reliability of each component and of the whole system.

We gathered all the data provided by CPN Tools (separated log files for each mean time and each simulation) in one file for each mean time and each component; for this reason, we used the command prompt and the loop.

Let us take the example of the master cylinder MTBF. The master cylinder MTBF monitor records data in a file named MTBF_MC.log for each individual simulation (same name for different simulations folders), and we need to record all the data for all the simulations in one file named MTBF_MC_Global.log, in this case, we enter the command below:

for /R %f in (MTBF_MC.log) do type "%f" MTBF_MC_Global.log

In the same way, we collect all the different mean times and put them in one Excel file named MTBF_components to serve as input to a MATLAB script.

Following this, we analyze the obtained Excel file using a MATLAB script, as shown in Figure 7:

```
% Read the table of MTBF monitor' results

MTBF_components=readtable(file,'ReadVariableNames', true);
MC=MTBF_components.MC;
figure
subplot(3,1,1);

% Probability Density Function : Master cylinder

h_MC=histogram(MC,'Normalization','pdf','EdgeColor','#FF00FF','FaceColor','#FF00FF');
xlabel('time(h)');
ylabel('Frequency');
title('MC- Probability density function ', ' FontSize', 9);

% Cumulative Distribution Function : Master cylinder

subplot(3,1,2);
cdf_MC=cdfplot(MC);
xlabel('time(h)');
ylabel('cdf');
title('MC-Cumulative Distribution function', 'FontSize', 9);
```

Figure 7. Matlab script example for PDF and CDF: MTBF MC.

4.1. Mean Time to First Failure (MTTFF)

Mean Time to First Failure is the anticipated time amount that operations will last before the first failure. The empirical MTTFF PDF (exponentially distributed) and CDF of the RBS are depicted in Figure 8.

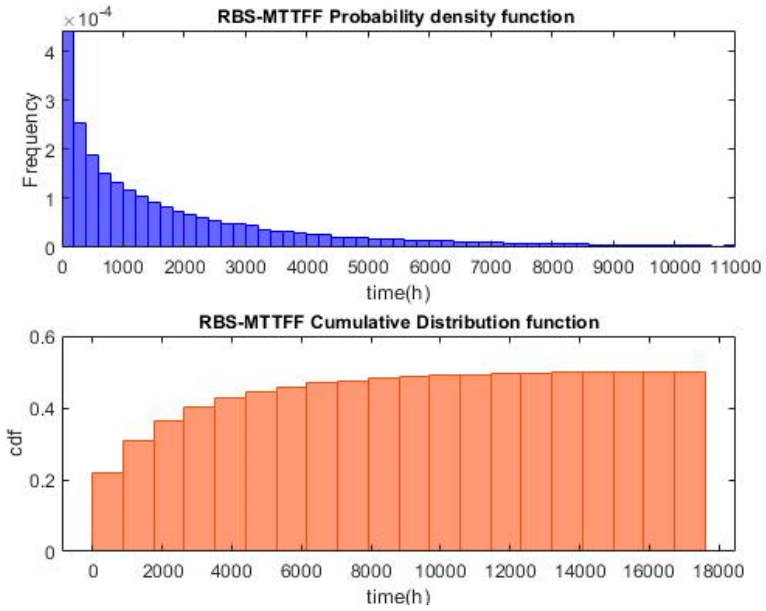


Figure 8. RBS mean time–to–first–failure, empirical PDF, and CDF.

Data collector monitors are applied in the different transitions preceded by **BF** for each component (BFMC in the example of Figure 6 and **BFRBS** for the whole system, Figure 5). This monitor will record only the first time-to-failure equal to the first **wait** value.

4.2. Reliability and Mean Time between Failures (MTBF)

Reliability is the ability of the entity to carry out a specified feature under fixed circumstances at a given time interval [30]. Mean Time Between Failures is the anticipated interval of time between two failures of a repairable entity.

In Figure 9, we include the empirical MTBF PDF and CDF distributions, and the resulting reliability for the mechanical component: the master cylinder. Similarly, we can obtain the corresponding information for all the components of our system.

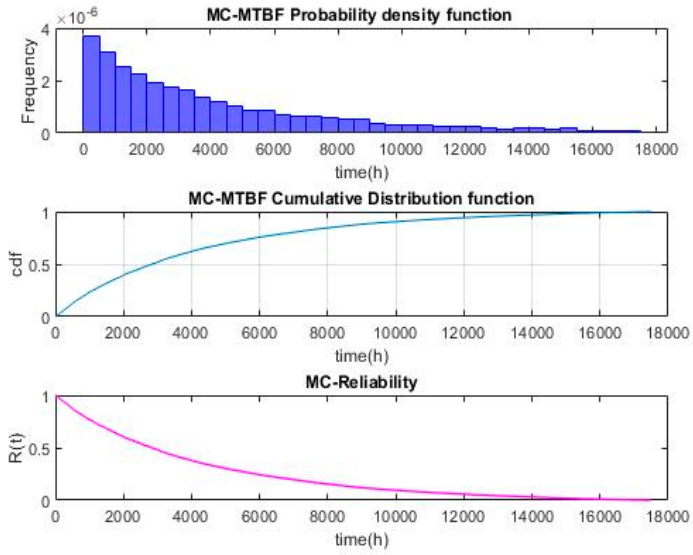


Figure 9. Master cylinder MTBF empirical PDF, CDF, and reliability.

Data collector monitors are defined for the same transition as MTTFF. As explained before, lists are defined to obtain the MTBF of different components and the whole system. This monitor will record the subtracted time resulting from the two input arcs when the

transition is enabled, i.e., only if two input arc conditions are satisfied: if we have the new time-to-failure, which is equal to the head of the defined list **hd MTBFS**, and the precedent time-to-failure **MTBF**.

In Figures 10 and 11, we have plotted the reliability of the RBS and its various components.

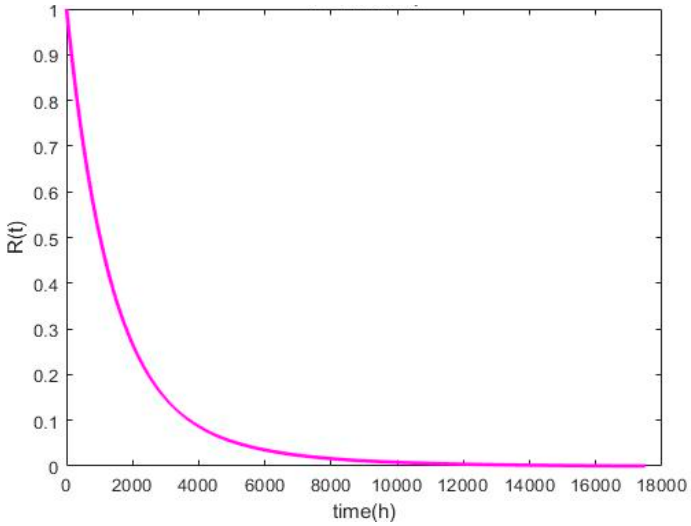


Figure 10. Regenerative braking system's reliability.

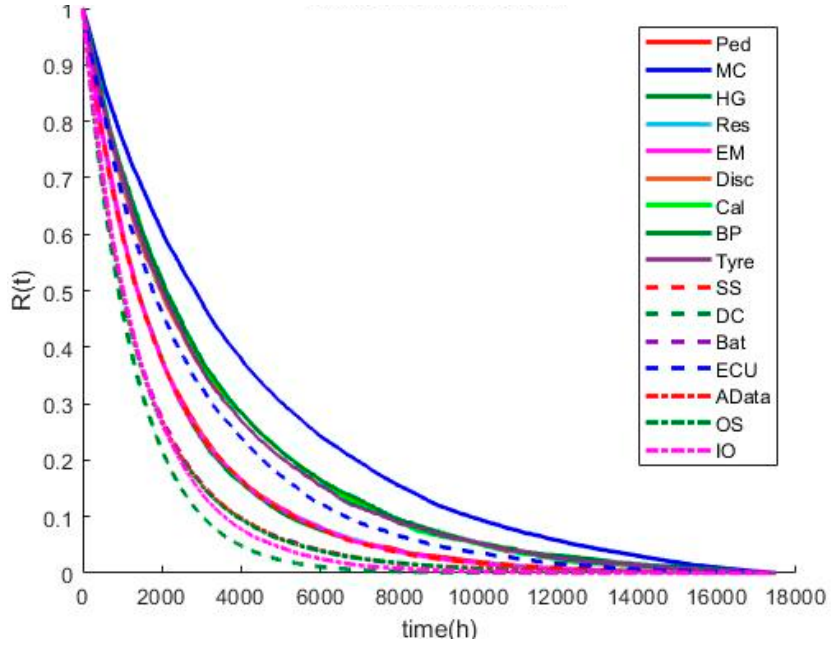


Figure 11. Reliability of the different components of the RBS.

The predictive reliability curve for the entire system is illustrated in Figure 10; the reliability value at 2000 h is approximately 0.3.

We can observe, on the curves of Figure 11, that the reliability of the master cylinder is the highest among all the RBS components, and the DC converter has the lowest reliability value.

4.3. Availability: Mean Up Time (MUT) and Mean Down Time (MDT)

Availability is the ability of an item to carry out a particular task under specific circumstances and at a specific time [30]. Availability predictions are generally difficult since both human and logistical factors must be considered in conjunction with reliability

and maintainability [31]. For simplicity's sake, we assume that human and logistical conditions are ideal in the present work.

Marking size monitors are defined to evaluate the availability. They record the duration of the state when each component is broken. They are applied on **panne** places, **PMC** for the example of the master cylinder, and **System_NOK** place for the whole system.

From the data obtained using this monitor, we can conclude the mean up times and the mean down times.

In fact, if the component is not broken (available), this monitor records the value 0 (no token is available on the panne place) and then we can extract its associate MUT. Otherwise, the monitor returns the value 1, and then the MDT is extracted.

MDT is the expected time of an entity's failure, including detection, intervention, reparation, and function restoration times. Therefore, MDT is the system's anticipation of downtime. If the system has only one failure state, then the following is true [32]:

$$\text{MDT} = \text{MTTR} \tag{11}$$

MUT and MDT average values of each component and of the whole RBS are listed in Table 5.

Table 5. MUT and MDT results.

Component	MUT	MDT
Ped	509.57	2.43
95% CI	26.18	0.11
MC	1955.67	4.58
95% CI	65.80	0.16
HG	511.22	2.42
95% CI	40.88	0.11
Res	509.37	2.33
95% CI	27.32	0.11
EM	513.31	2.68
95% CI	26.73	0.14
Disc	415.92	0.07
95% CI	43.65	0.09
Cal	403.25	1.28
95% CI	57.21	0.09
BP	423.67	1.25
95% CI	30.60	0.08
Tyre	411.81	1.28
95% CI	30.33	0.09
SS	498.16	2.49
95% CI	26.16	0.11
DC	601.06	4.15
95% CI	95.19	0.0554
Bat	459.06	3.01
95% CI	20.49	0.11
ECU	443.46	1.59
95% CI	29.18	0.08
AData	463.35	2.93
95% CI	40.97	0.11
IO	491.67	2.42
95% CI	19.06	0.09
OS	470.95	3.02
95% CI	21.15	0.10
RBS	258	4.70
95% CI	31.03	0.07

From Tables 4 and 5, we note that the MTTR and MDT values are identical because we have presumed that the repair is carried out right after the failure occurs.

4.4. Maintainability and Mean Time to Repair (MTTR)

The capacity of an entity to continue being in or return to a state where it can carry out a required function under set conditions and utilizing the advised processes and resources is known as maintainability [30].

The MTTR is the average time spent eliminating a fault and restoring the drive to service (time for all activities between fault clearing and restart). This time depends on proper maintainability, like an adequate diagnostic system, available spare parts, and the competence level of the personnel involved [33]. We suppose that, in our system, a repair starts immediately after the occurrence of a failure and that the RBS restarts instantaneously after a repair.

We apply data collector monitors in each repair transition (**RMC** for the master cylinder and **System_reparation** for the whole system) to obtain the MTTR. If the transitions are enabled then the monitor should collect the repair time **waitrep**.

The RBS MTTR empirical PDF distribution, depicted in Figure 12, is close to an exponential distribution.

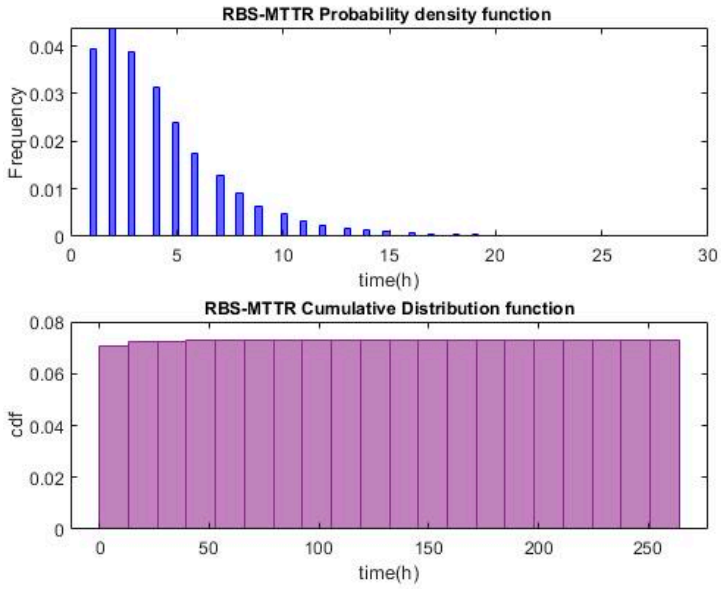


Figure 12. Regenerative braking system: MTTR empirical PDF and CDF.

5. Discussions

The rise of mechatronic systems is a major breakthrough for the industrial world. It is increasingly impacting the transport field, and, especially, the automotive industry. They are complex systems including different technologies. Against the backdrop of Industry 4.0, where new and intensified requirements for more and more reliable and safe systems exist, it is imperative to check system performance as early as possible.

The proposed approach is applied to mechatronic systems under the example of a regenerative braking system. It is structured around timed colored Petri nets, which are used to model physical behavior and system specifications for the system under study, with the view to assess predictive reliability, availability, and maintainability mean times in the earlier design stage. Data recorded from data collection or expert opinion are injected into the timed CPN physical model to obtain the stochastic model. The use of monitors in a timed CPN specifications model allows data collection for performance analysis after running the Monte Carlo simulation.

The different mean times for each component separately and for the whole system are obtained, and their empirical distributions are outlined. In the same context, the impact of each component on the whole RBS reliability has been investigated, offering valuable decision-making support. Indeed, in the event of reliability non-compliance,

design modifications are undertaken on unreliable components. The approach is useful for validating the design and the choice of components to meet predictive reliability, availability, and maintainability objectives.

During the present study, some simplifications were applied to the model, such as considering that the repair begins immediately after a failure, which is not the real case during the serial life of the mechatronic system. In fact, several constraints could be included, like the availability of spare parts, human resources, logistics, or others.

To implement the suggested methodology, we rely on database collection. Data collection is a key and indispensable tool. Data collections are validated and/or developed after a long process of expertise and processing, tied to a field of knowledge, and organized to be available for consultation by users.

A distinctive feature of mechatronic systems is the interaction between the various technologies composing them. In fact, analyzing the reliability, availability, and maintainability of each individual component is not enough. It is essential to study the system, its components, and the interactions between the various components, and then feed these aspects into our approach.

Although we have focused on the dependability attributes, reliability, availability, and maintainability, there are others that can be integrated into the drive, such as safety. In addition, to ease the future use of our methodology in an industrial context, we need to design an automatic calculation tool.

6. Conclusions

In the context of Industry 4.0, and the derived increasing dependence on mechatronics with ever-rising complexity, more customer quality requirements are established to further implement tailored solutions needed for the market demand. The evaluation of reliability, availability, and maintainability of mechatronic products in the earlier design stages enables entirely new opportunities for industries.

This paper outlines a methodology for the predictive assessment of reliability, availability, and maintainability when applied to mechatronic systems, starting with an introduction of the formal definitions of Petri net and time-colored Petri net, as well as bringing to light a couple of useful related concepts.

Thereafter, successive steps in the methodology were mapped out, covering the first stage of setting up both timed CPN physical and specifications models, ensuring their connection, and applying the system monitoring to collect data for performance analysis purposes in view of evaluating the various mean-time indicators and running Monte Carlo simulations. The methodology was illustrated in the case of a regenerative braking system.

Dependability encompasses other attributes, like safety, which could be evaluated using the same model. We assumed, during the present work, that there were no interactions between the different components. As a mechatronic system is an integrated, multi-technology system, the study of its reliability must also consider the multi-domain interactions between the components. The suggested model deals with dependability in the design phase, works are in progress to extend the same model including multi-domain interactions and evaluating experimental and operational dependability as well.

Author Contributions: Conceptualization, I.M. and E.M.B.; methodology, I.M.; software, I.M.; validation, E.M.B. and M.A.M.; formal analysis, I.M.; investigation, I.M.; resources, I.M.; data curation, I.M. and E.M.B.; writing—original draft preparation, I.M.; writing—review and editing, I.M. and M.A.M.; visualization, E.M.B. and M.A.M.; supervision, E.M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

To assess RAM indicators, we refer to the following distributions [34]:

Appendix A.1. Weibull Distribution

The Weibull distribution is used to evaluate the product's behavior in its three life phases depending on the shape parameter β: infant mortality period ($\beta < 1$), useful life period ($\beta = 1$), and wear out period ($\beta > 1$). It is commonly employed in mechanics.

The Weibull distribution is characterized by two parameters η and β. It is marked by the following:

The probability density function:

$$f(t) = \frac{\beta}{\eta} \cdot \left(\frac{t}{\eta}\right)^{\beta-1} e^{-\left(\frac{t}{\eta}\right)^{\beta}} \qquad \text{with t} \geq 0 \tag{A1}$$

The reliability:

$$R(t) = e^{-\left(\frac{t}{\eta}\right)^{\beta}} \qquad \text{with t} \geq 0 \tag{A2}$$

The failure rate:

$$\lambda(t) = \frac{\beta}{\eta} \left(\frac{t}{\eta}\right)^{\beta-1} \qquad \text{with t} \geq 0 \tag{A3}$$

Appendix A.2. Exponential Distribution

This distribution applies in a wide range of disciplines. It covers the lifespan of equipment subject to random failure. The exponential distribution is frequently encountered in electronic reliability to outline the time frame when the equipment failure rate is assumed constant. It is specified by a unique parameter, λ, and is distinguished by the following:

The probability density function:

$$f(t) = \lambda e^{-\lambda t} \qquad \text{with t} \geq 0 \tag{A4}$$

The reliability:

$$R(t) = e^{-\lambda t} \qquad \text{with t} \geq 0 \tag{A5}$$

The failure rate:

$$\lambda(t) = \lambda \qquad \text{with t} \geq 0 \tag{A6}$$

Appendix A.3. Musa Execution Time Model

The Musa runtime model is a software reliability model. It estimates the initial failure rate λ_0 of the software from the exponential distribution at the beginning of the tests. Where t is the execution time of the software [12]:

The failure rate:

$$\lambda = kpN_0 \tag{A7}$$

where:

k: constant in relation to the program's dynamics and the machine where it is installed ($k = 4.2 \times 10^{-7}$);

p: number of executions by time unit (p = r/SLOC/ER), where: r: instruction execution rate (constant), SLOC: code source lines (excluding reused code), ER: expansion ratio, a constant on the programming language (ER = 1 for Assembly; ER = 1, 5 for Macro Assembly; ER = 2, 5 for C; ER = 3 for COBOL, FORTRAN; ER = 4, 5 for Ada);

N_0: Initial number of faults in the program.

Appendix A.4. Erlang Distribution

The Erlang distribution was pioneered by A.K. Erlang as a way of investigating the possible number of simultaneous calls to the switchboard station operator. This distribution is typically adopted to qualify waiting times in queuing operations. While the exponential random variable outlines the time between adjacent events, the Erlang random variable considers the time interval between any event and the kth following event. The following parameters characterize this distribution [35]:

The probability density function:

$$f(t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!} \qquad \text{with t} \geq 0 \tag{A8}$$

The reliability:

$$R(t) = \sum_{n=0}^{k-1} \frac{1}{n!} e^{-\lambda t} \lambda t^n \qquad \text{with t} \geq 0 \tag{A9}$$

k is the shape parameter. When k = 1 we obtain the exponential distribution of parameter λ.

References

1. Guerineau, B.; Bricogne, M.; Durupt, A.; Rivest, L. Mechatronics vs. cyber physical systems: Towards a conceptual framework for a suitable design methodology. In Proceedings of the 2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS)/17th International Conference on Research and Education in (REM), Compiegne, France, 15–17 June 2016.
2. Ghobakhloo, M. Industry 4.0, digitization, and opportunities for sustainability. *J. Clean. Prod.* **2020**, *252*, 119869. [CrossRef]
3. Kuru, K.; Yetgin, H. Transformation to Advanced Mechatronics Systems within New Industrial Revolution: A Novel Framework in Automation of Everything (AoE). *IEEE Access* **2019**, *7*, 41395–41415. [CrossRef]
4. Yan, R.; Dunnet, S.; Andrews, J. A Petri net model-based resilience analysis of nuclear power plants under the threat of natural hazards. *Reliab. Eng. Syst. Saf.* **2023**, *230*, 2469. [CrossRef]
5. Laronde, R.; Charki, A.; Bigaud, D. Reliability of photovoltaic modules based on climatic measurement data. *Int. J. Metrol. Qual. Eng.* **2010**, *1*, 45–49. [CrossRef]
6. Volovoi, V.; Peterson, D.K. Coupling reliability and logistical considerations for complex system of systems using Stochastic Petri Nets. In Proceedings of the 2011 Winter Simulation Conference (WSC), Phoenix, AZ, USA, 11–14 December 2011.
7. Zhang, D.; Zhang, K.; Wang, L.; Hong, Q. Reliability modeling and analysis of reactor protect system based on Petri net. *J. Phys. Conf. Ser.* **2021**, *1754*, 012059. [CrossRef]
8. Zhang, W. Performance Prediction for Subsea Structures during Lowering Operations. Ph.D. Thesis, University of Tasmania, Tasmania, Australia, 2018.
9. Hellel, E.K.; Hamaci, S.; Ziani, R. Modelling and reliability analysis of multi-source renewable energy systems using deterministic and stochastic Petri net. *Open Autom. Control. Syst. J.* **2018**, *10*, 25–40. [CrossRef]
10. Taleb-Berrouane, M.; Khan, F.; Amyotte, P. Bayesian Stochastic Petri Nets (BSPN)—A new modelling tool for dynamic safety and reliability analysis. *Reliab. Eng. Syst. Saf.* **2020**, *193*, 106587. [CrossRef]
11. Andrews, J.; Tolo, S. Dynamic and dependent tree theory (D2T2): A framework for the analysis of fault trees with dependent basic events. *Reliab. Eng. Syst. Saf.* **2023**, *230*, 108959. [CrossRef]
12. Demri, A.; Charki, A.; Guerin, F.; Christofol, H. Functional and dysfunctional analysis of a mechatronic system. In Proceedings of the 2008 Annual Reliability and Maintainability Symposium, Las Vegas, NV, USA, 28–31 January 2008; pp. 114–119.
13. Mihalache, A.; Guerin, F.; Barreau, M.; Todoskoff, A.; Dumon, B. Reliability assessment of mechatronic systems: Operating field data analysis. In Proceedings of the 2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT'04, Hammamet, Tunisia, 8–10 December 2004.
14. Pinna, B.; Babykina, G.; Brinzei, N.; Pétin, J.F. Using coloured petri nets for integrated reliability and safety evaluations. *IFAC Proc.* **2013**, *46*, 19–24. [CrossRef]
15. Aubry, J.F.; Brinzei, N.; Mazouni, M.H. *Systems Dependability Assessment: Benefits of Petri Net Models*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
16. Signoret, J.P.; Leroy, A. *Reliability Assessment of Safety and Production Systems: Analysis, Modelling, Calculations and Case Studies*; Springer Nature: Berlin/Heidelberg, Germany, 2021.
17. Ratzer, A.V.; Wells, L.; Lassen, H.M.; Laursen, M.; Qvortrup, J.F.; Stissing, M.S.; Jensen, K. CPN tools for editing, simulating, and analyzing coloured Petri nets. In Proceedings of the International Conference on Application and Theory of Petri Nets, Eindhoven, The Netherlands, 23–27 June 2003; pp. 450–462.
18. Jensen, K.; Kristensen, L.M. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*; Springer: Aarhus, Denmark, 2009.
19. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [CrossRef]

20. Wells, L. Performance analysis using CPN tools. In Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools, Pisa, Italy, 11–13 October 2006.
21. David, R.; Alla, H. *Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems*; Prentice-Hall, Inc.: Saddle River, NJ, USA, 1992.
22. Mehdi, I.; Boudi, E.M. Qualitative Functional and Dysfunctional Analysis and Physical Modeling of an Eco-Designed Mechatronics System Using Coloured Petri-nets: Application on a Regenerative Braking System. In Proceedings of the International Conference on Advanced Technologies for Humanity, Rabat, Morocco, 26–27 November 2021.
23. Mehdi, I.; Boudi, E.M. Towards a sustainable conceptual design of mechatronic systems application to a regenerative braking system. In Proceedings of the Materials Today: Proceedings, Virtual, 20–22 May 2021.
24. Bildstein, M.; Mann, K.; Richter, B. *Regenerative Braking System. Fundamentals of Automotive and Engine Technology: Standard Drives, Hybrid Drives, Brakes, Safety Systems*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2017.
25. Valmari, A. The state explosion problem. In *Advanced Course on Petri Nets*; Springer: Berlin/Heidelberg, Germany, 1996.
26. Liu, L. Verification of the SIP transaction using coloured petri nets. In Proceedings of the Thirty-Second Australasian Conference on Computer Science, Wellington, New Zealand, 1 January 2009.
27. Staines, A.S. Concurrency and Petri Net Models. *Int. J. Circuits Syst. Signal Process.* **2022**, *16*, 852–858. [CrossRef]
28. Westergaard, M.; Verbeek, H.M.W. Efficient implementation of prioritized transitions for high-level Petri nets. In Proceedings of the International Workshop on Petri Nets and Software Engineering, Newcastle upon Tyne, UK, 20–21 June 2011.
29. Fenton, G.A.; Griffiths, D.V. *Risk Assessment in Geotechnical Engineering*; John Wiley & Sons: New York, NY, USA, 2008.
30. Frederick, S.R. *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*; Springer: Berlin/Heidelberg, Germany, 2009.
31. Pham, H. *Handbook of Reliability Engineering*; Springer: London, UK, 2003.
32. Ding, L.; Wang, H.; Kang, K.; Wang, K. A novel method for SIL verification based on system degradation using reliability block diagram. *Reliab. Eng. Syst. Saf.* **2014**, *132*, 36–45. [CrossRef]
33. Xu, H.; Yu, W.; Griffith, D.; Golmie, N. A survey on industrial Internet of Things: A cyber-physical systems perspective. *IEEE Access* **2018**, *6*, 78238–78259. [CrossRef]
34. Birolini, A. *Reliability Engineering: Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2017.
35. Coen, A.; Gutiérrez, L.; Mena, R.H. Modelling failures times with dependent renewal type models via exchangeability. *Statistics* **2019**, *53*, 1112–1130. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.