# A NOVEL APPROACH TO REALISTIC MODELING AND SIMULATION OF STATE-VARYING FAILURE RATES

**Sanja Lazarova-Molnar**

Faculty of Information Technology, United Arab Emirates University, United Arab Emirates

sanja@uaeu.ac.ae

**ABSTRACT**
There has been a lot of research on time-varying failure rates, which deems constant failure rates as inadequate to model failures accurately. However, besides time, failure rates can also be affected by the state of the system (or its history, in terms of sequences of states and events that it has been through). In our paper we define several classes of state-varying failure rates and extend the formalism of Petri nets to model them. We further use the flexibility of the proxel-based method to accurately analyze behavior of systems that incorporate these kinds of failures. To illustrate our approach and study the effect of such dependencies, we compare simulation results for two models: one that exhibits state-varying failure rates, and another that only contains predefined failure rate functions.

Keywords: state-varying failure rates, reliability, proxels, Petri nets

## 1. INTRODUCTION

There has been an intensive research on time-varying failure rates, including their significant impact on reliability (Hassett, Dietrich et al. 1995; Retterath, Venkata et al. 2005; Zhang, Cutright et al. 2010), which have been defined as such almost two decades ago (Billinton and Allan 1992). Recently, Xie developed an analytical model of unavailability due to aging failures too (Xie and Li 2009). Since long time ago it has been shown that constant failure rates are inadequate for describing systems' failures (Proschan 1963). Nevertheless, they are still widely used due to the fact that the methodology for their analysis is less complex and more accurate. The popular MTTF (meantime to failure) measure is still a widely used one (Coskun, Strong et al. 2009; Sharma, Kahlon et al. 2010), even though it has been deemed many times as inadequate (Schroeder and Gibson 2007). We go one step further as to claim that even time-varying failure rates are not sufficient, as in many cases the rates completely change their functions based on the occurrence of some events or based on the complete state of the system (e.g. a part has been replaced by a new one that is based on a new technology, or if a mechanical part has been physically broken, then it is logical that the failure rate would increase with each time it breaks). This is what we term as a *state-varying* failure.

According to a study of medical equipment (Baker 2001) it was shown that there was a decreasing hazard of (first) failure after repair for some types of equipment. The interpretation was that it is a consequence of imperfect or hazardous repair, and also, because of differing failure rates among a population of machines.

Likewise, in (Liberopoulos and Tsarouhas 2005) a pizza production line is studied and it was found that most of the failures have a decreasing failure rate because proactive maintenance improves the operating conditions at different parts in the line, and a few failures have an almost constant failure rate. It was also concluded that the longer the time between two failures, the more problems accumulate, and therefore, it takes longer time to fix the latter failure. It also suggests that the more time the technicians spend fixing a failure, the more careful job they do, and therefore, the time period until the next failure is longer. This is a very interesting observation that calls for state-varying failure rates and it can be addressed using our approach.

These are some examples that show that failures need to be described more realistically to obtain accurate and useful simulation results. Unfortunately, this has very rarely been the case.

Our goal is to provide a deterministic approach to analyze systems that exhibit not only time-, but more importantly, state-varying failure rates. For this we use method of proxel-based simulation, which based on our previous experience, is highly adjustable to treat these complex activities. In (Lazarova-Molnar 2008) we have analyzed and described state-dependent transitions and used proxel-based simulation for their analysis. These are the types of transitions that correspond and can be used to describe state-varying failure rates. Thus, in addition to the simulation approach, this paper provides a concept of how to model this type of failure rates and what changes need to be undertaken in the standard stochastic Petri net (SPN) models to introduce them.

The paper is organized as follows. In the subsequent section we describe the state-varying failure rates, along with an introduction to the proxel-based simulation method. Further, we provide a concept for modeling state-varying failures using SPN. Next, we present an example model which we use to demonstrate our approach and we run experiments based on it. Finally, we present the results of the experiments with a discussion and conclusions.

## 2. PRELIMINARIES

### 2.1. State-varying Failures

It is a common observation that a failure rate cannot simply be described by one function during its entire lifetime. Even more, failure rates in reality can change not solely based on time (Retterath, Venkata et al. 2005), but also based on the occurrence of certain events in the system (e.g. replacing the service person by another one which fixes them in a different manner, i.e. more thoroughly would influence the failure rate function). We refer to these types of failures as *state-varying failure rates*.

Description of failure rate functions of state-varying failure rates is a complex process and would require an algorithmic description to supplement the graphical model. To illustrate it, one such description may be:

If machine is repaired by repairman A
    Then the failure rate function ~ Normal(a, b)
Else if machine is repaired by repairman B
    Then the failure rate function ~ Normal(c, d)

If we add another factor to this, i.e. the age of the machine, and then the description would change to:

If machine is repaired by repairman A
    Then the failure rate function ~ Normal(f(t), b)
Else if machine is repaired by repairman B
    Then the failure rate function ~ Normal(g(t), d)

where $t$ is the age of the machine (which can easily be exchanged to represent the number of failures or any other relevant quantity). This observation is more general than the one that uses fixed failure rate functions, and as such, more realistically models the phenomenon of a machine that exhibits failures.

Obviously, these models would need more advanced (or extended) modeling formalisms to be described. Thus, we extend stochastic Petri nets to account for the state-varying rates.

Finally, to show the difference and compare the effects of such (even very small) dependencies, we compare the simulation results for two models: one that exhibits state-varying failure rates, and another, similar and over-simplified one, that only contains predefined failure rate functions with fixed parameter values.

### 2.2. Proxel-based Simulation

The proxel-based method (Horton 2002; Lazarova-Molnar 2005) is a relatively novel simulation method, whose underlying stochastic process is a discrete-time Markov chain (Stewart 1994)and implements the method of supplementary variables (Cox 1955). The method, however, is not limited to Markovian models. On the opposite, it allows for a general class of stochastic models to be analyzed regardless of the involved probability distribution functions. In other words, the proxel-based method combines the accuracy of numerical methods with the modeling power of discrete-event simulation.

The proxel-based method is based on expanding the definition of a state by including additional parameters which trace the relevant quantities in one model through a previously chosen time step. Typically this includes, but is not limited to, age intensities of the relevant transitions. The expansion implies that all parameters pertinent for calculating probabilities for the future development of a model are identified and included in the state definition of the model.

Proxels (stands for probability elements), as basic computational units of the algorithm, follow dynamically all possible expansions of one model. The state-space of the model is built on-the-fly, as illustrated in Figure 1, by observing every possible transiting state and assigning a probability value to it (*Pr* in the figure stands for the probability value of the proxel). Basically, the state space is built by observing all possible options of what can happen at the next time step. The first option is for the model to transit to another discrete state in the next time step, according to the associated transitions. The second option is that the model stays in the same discrete state, which results in a new proxel too. Zero-probability states are not stored and, as a result, no further investigated. This implies that only the truly reachable (i.e. tangible) states of the model are stored and consequently expanded. At the end of a proxel-based simulation run, a transient solution is obtained which outlines the probability of every state at every point in time, as discretized through the chosen size of the time step. It is important to notice that one source of error of the proxel-based method comes from the assumption that the model makes at most one state change within one time step. This error is elaborated in (Lazarova-Molnar 2005).

Each proxel carries the probability of the state that it describes. Probabilities are calculated using the instantaneous rate function (IRF), also known as hazard rate function. The IRF approximates the probability that an event will happen within a predetermined elementary time step, given that it has been pending for a certain amount of time $\tau$ (indicated as 'age intensity'). It is calculated from the probability density function ($f$) and the cumulative distribution function ($F$) using the following formula:

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)} \tag{1}$$

As all state-space based methods, this method also suffers from the state-space explosion problem (Lin, Chu et al. 1987), but it can be predicted and controlled by calculating the lifetimes of discrete states in the model. In addition, its efficiency and accuracy can be further improved by employing discrete phases and extrapolation of solutions (Isensee and Horton 2005). More on the proxel-based method can be found in (Lazarova-Molnar 2005).

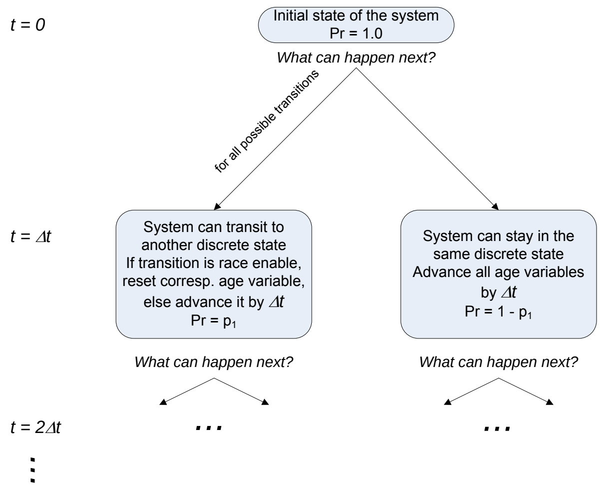


Figure 1: Illustration of the development of the proxel-based simulation algorithm

## 3. MODELING STATE-VARYING FAILURES

According to the performed observation, studies and research, we identify several classes of state-varying failure rates, i.e. failure rates that depend on:

a) the number of failure occurrences up to the observed point in time,
b) the age of the machine up to the observed point in time,
c) the duration of the last repair,
d) the time between the last two failures,
e) the properties of the repair facilities, introduced as additional parameters, and
f) the types of failures that have occurred.

We allow a combination of a number of these factors to occur in our sample model to illustrate their effects through the proxel-based simulation analysis. Proxel-based simulation can easily be applied to analyze a model that exhibits any combination of them, as well as other types of dependencies on quantities that are part of the model. In the following, we will provide the details of the formal classification of the state-varying failures and our simulation approach. This will be further demonstrated using an example model.

### 3.1. Formal Model of State-Varying Failure Rates

The underlying discrete stochastic model that exhibits state-varying failure rates is described using a stochastic Petri net (SPN) (Bause and Kritzinger 2002). Nevertheless, we further extend the basic description of SPN to allow the tracking of the *relevant rewards*. Those are the quantities that are in fact parameters of the distribution functions of the timed transitions, besides the age intensities of the relevant transitions. Typically, they are introduced by extending the basic SPN with additional places and transitions that enable the tracking, as shown in Figure 2. However, to record quantities, such as the *duration of the last repair* (type (c)), we introduce a novel element which we term as *tracking variable* (TV), and it is represented by a hexagon in the SPN graphical model. TVs are connected by diamond-shape-ended arrows to the transitions for which they record the last firing time.

To summarize, the extension is at both the level of the SPN formalism, and at the Petri net model itself, which is enriched by a number of extra places and transitions to ensure the tracking of relevant rewards. As for the SPN formalism: we extend it by the new element TV, and, in order to account for the state-varying transitions, we allow distributions to have discrete states, i.e. markings, as parameters of the distribution functions that control firing of transitions. In the following, we show by example how a SPN can be extended to allow the tracking of the various relevant quantities.

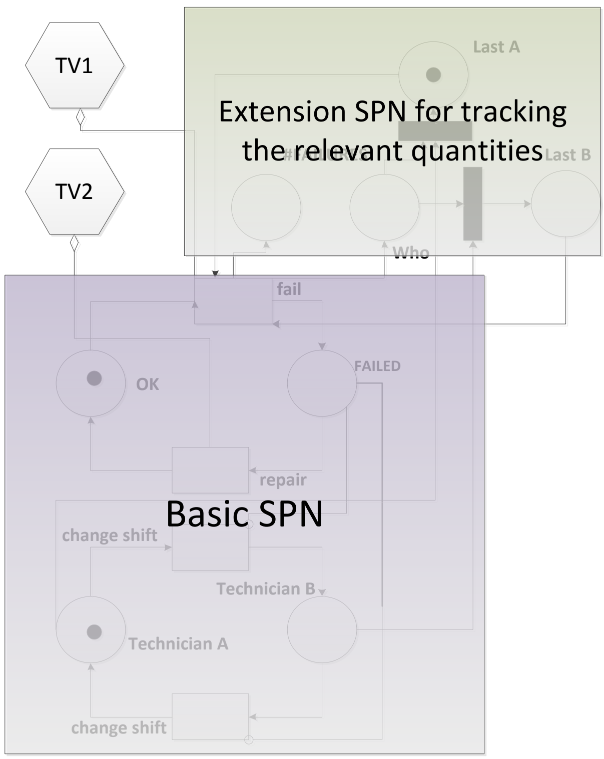


Figure 2: Illustration of the extended SPN model

### 3.2. Petri Net Specifications

In the following we provide the formal definition of the extension of the SPN to account for the state-varying failure rates. Each Petri net *SPN* is defined as:

$$SPN = (P, T, A, G, TV, m_0)$$

where:

- $P = \{P_1, P_2, \ldots, P_n\}$, the set of places, drawn as circles
- $T = \{ T_1, T_2, \ldots, T_m \}$, the set of transitions along with their distribution functions or probability values, drawn as bars
- $A = A^I \cup A^O \cup A^H \cup A^T$, the set of arcs, where $A^O$ is the set of output arcs, $A^I$ is the set of input arcs, $A^H$ is the set of inhibitor arcs, and $A^T$ is the set of tracking arcs (connect transition to a tracking variable and are ended by a diamond-shape at the tracking variable end); each arc has a multiplicity assigned to it,
- $G = \{g_1, g_2, \ldots, g_r\}$, the set of guard functions which are associated with different transitions,
- $TV = \{ TV_1, TV_2, \ldots, TV_m \}$, the set of tracking variables that store the last duration of the enabling time of a transition (drawn as hexagons),
- $m_0$ – the initial marking of the Petri net.

Each transition is defined as $T_i = (F, type)$, where $type \in \{enabling, age, immediate\}$ is the type of memory policy if it is a timed transition or "*immediate*" if the corresponding transition is an immediate one. $F$ is a cumulative distribution function if the corresponding transition is a timed one. Immediate transitions have a constant value instead of a distribution function assigned to them, which is used for computing the probability of firing of an immediate transition if more than one are enabled at once. The sets of arcs are defined such that

$$A^O = \{a^o_1, a^o_2, \ldots, a^o_k\},\ A^I = \{a^i_1, a^i_2, \ldots, a^i_j\}, A^H = \{a^h_1, a^h_2, \ldots, a^h_i\}, \text{ and } A^T = \{a^t_1, a^t_2, \ldots, a^t_l\},$$

where

$A^H, A^I \subseteq P \times T \times \mathbb{N}, A^O \subseteq T \times P \times \mathbb{N}, A^T \subseteq T \times P \times \mathbb{R}$.

The multiplicity of the tracking arcs can be a real number, unlike the others, where it is a non-negative integer number. We denote by $M = \{m_0, m_1, m_2, \ldots \}$ the set of all reachable markings of the Petri net. Each marking is a vector made up of the number of tokens in each place in the Petri net along with the values of the tracking variables, $m_i = (\#P_1, \#P_2, \ldots, \#P_n, val(TV_1), val(TV_2), \ldots, val(TV_m))$. The set of all reachable markings is the discrete state space of the Petri net. The changes from one marking to another are consequences of the firing of enabled transitions which move (destroy and create) tokens; thus creating the dynamics in the Petri net. This makes the firing of a transition analogous to an event in a discrete-event system. The markings of a Petri net, viewed as nodes, and the possibilities of movement from one to another, viewed as arcs, form the reachability graph of the Petri net.

### 3.3. Adaptation of the Proxel-based method to Accommodate State-varying Failure Rates

The main adjustment of the proxel-based method to accommodate state-varying failure rates is the extension of the definition of the proxel to introduce the notion of relevant rewards. They incorporate in the state definition all quantities, which in addition to age intensities; can be parameters of probability distribution functions of events in the model. This yields the following definition of a state:

State = (Discrete state, Relevant Rewards, State Relevant Age Intensities)

Thus, all parameters required for computing transition distribution functions are contained in the state vector (making the model implicitly a non-homogeneous Markov chain). The discrete state typically corresponds to a marking in the SPN model. The relevant rewards are determined by the nature of the events in the model, i.e. what kind of dependencies the failure rates in the model exhibit. In Table 1 we provide the relevant rewards for the six classes of state-varying failures that we have identified.

Table 1: Relevant rewards for the six state-varying failure classes

| State-varying failure class | | Relevant reward |
|---|---|---|
| a) | the number of failure occurrences up to the observed point in time, | Number of failures |
| b) | the age of the machine up to the observed point in time, | Age of machine |
| c) | the duration of the last repair, | Duration of last repair |
| d) | the time between the last two failures, | Duration of operation of machine between two consecutive failures |
| e) | the properties of the repair facilities, introduced as additional parameters, | Parameters of repair facilities (e.g. quantification of experience of repairman) |
| f) | the types of failures that have occurred. | Types of failures that have occurred so far |

## 4. EXPERIMENTS AND RESULTS

In this section we present an example model which we will use to illustrate the simulation of state-varying failure rates. We will describe the proxel-based simulation of this model and run it using various time steps.

### 4.1. The Model

The model that we use to demonstrate our approach is a simple model that describes a machine that incorporates both time- and state- varying failure rates, similar to the scenarios described in Section 2.1.

Using a Petri net, the model can be described as shown in Figure 2. It represents a machine that exhibits one of two possible states: OK and FAILED. When the machine has failed, one of the two repairmen arrives and repairs it, after what the machine's state becomes OK. Changing shifts during repair is not allowed. The two repairmen have different lengths of working experience. Thus, when the machine is fixed by the Repairman A, the time to the next failure is on average longer, than when it is repaired by Repairman B. This implies that the proxels will need to record the information of who performed the last repair as well.

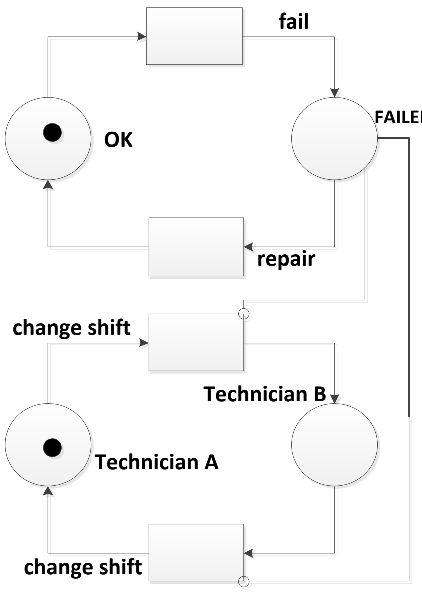


Figure 3: Basic Petri net model of the example

In addition to the afore-described scenario, the distribution of the time to next failure of the machine is also a function of the number of failures and the age of the machine. For instance, in our example model we use the following formula to describe the distribution of the repair time:

$$f_{repair}(R, age, n_f) \sim$$

$$\begin{cases} N(10 + age * 0.01 + n_f * 0.1, age * n_f + 1.0), R = A \\ N(12 + age * 0.015 + n_f * 0.12, age * n_f + 1.5), R = B \end{cases} \quad (2)$$

where:

- *age* is the age of the machine, i.e. the global simulation time,
- $n_f$ is the number of failures that have occurred, i.e. *#FAILURES* in the SPN,
- $R$ is the repairman that did the serviced the last failure, i.e. can be obtained from the SPN by checking the token is in place *Last A* or *Last B*, and
- $N$ stands for the normal distribution parameters, with the standard parameters: mean and variance, correspondingly.

In other words, repairs are normally distributed, where the mean and the variance are functions of the Repairman that completed the last repair, the total number of failures, and the age of the machine. This implies that we observe the dependences (a), (b), and (e), as pointed out in Section 3. This directly implies that, as described in Table 1, we need to add the following relevant rewards:

- Number of failures,
- Age of machine, and
- Parameters of repair facilities.

In order to illustrate the enhancements that the state-varying failures would require, we include the required information in the basic Petri net model, whose extended version is shown in Figure 4.

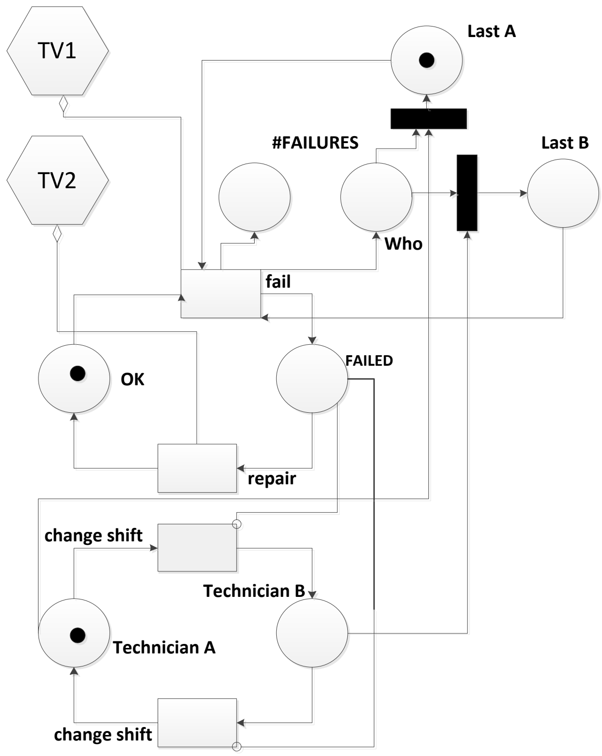


Figure 4: Extended Petri net model of the example

In Figure 5, the state-transition diagram of the Petri net model from Figure 4, is shown. Note that the model is an unbounded Petri net, i.e. it is practically impossible to accurately analyze it using numerical approaches. This, however, is not a limitation of the proxel-based method, as it dynamically builds the state space on-the-fly.

Besides the repair duration probability distribution function, as shown by the Equation (2), the remaining distribution functions that we used in our experiments are as follows:

- $f_{fail}(age, n_f) \sim E(150 + age * 0.05 + n_f * 0.1)$
- $f_{change\ shift}() \sim D(12)$

where *E*() stands for the exponential distribution function, with the mean as its only parameter, and D() is the deterministic probability distribution function.

As described in the following subsection, for this example we slightly modify the discrete state description to better explain our approach. In general, the proxel-based simulation can be directly performed on the enhanced Petri net model.

### 4.2. Insight in the Proxel-based Simulation

In the following we provide insight in the proxel-based simulation for the example model. The goal is show what exactly happens at lower level when simulating the state-varying failures. We begin by defining the state of the systems in the concrete example as:

$$((Machine\ State, Repairman), (\#Failures, Last\ Repairman), Age\ Intensity\ Vector),$$

which implies that the discrete state of the system is described by the state of the machine ($Machine\ State$) and the repairman on shift ($Repairman$). The relevant rewards are the number of failures of the machine ($\#Failures$) and the repairman that completed the last repair of the machine ($Last\ Repairman$). Finally, the last element of the state of the system is the $Age\ Intensity\ Vector$ that keeps track of the time that the machine has spent in the specified state, as well as the time during which the repairman has been on shift. This yields the initial proxel as:

$$((OK, A), (0, A), (0,0)).$$

which shows that initially the machine is in state OK, and Repairman A is on shift. The number of failures up to simulation time $t = 0$ is zero, and the age intensities of both machine state *OK* and duration of Repairman A on shift are zero as well. Note that initially we assume that the last repair was completed by the more experienced repairman, i.e. Repairman A. The subsequent proxels which originate from the initial one at time $t = \Delta t$, along with the three potential events, are the following:

a) Machine fails - $((F, A), (1, A), (0, \Delta t))$,
b) Repairman shift change -$((F, B), (0, A), (\Delta t, 0))$,
c) No events - $((OK, A), (0, A), (\Delta t, \Delta t))$.

where *F* stands for the machine's state *FAILED*. The age of the machine is implicitly recorded by the global simulation time variable. Note that we assume that the repairman on shift that has started to work on the repair also has to complete it, and thus, extend his shift. Further, for illustration purposes, we will develop the proxel for the case (a), i.e. when the machine has failed, which yields the following subsequent events and proxels:

a-1) Machine is repaired - $((OK, A), (1, A), (0, 2\Delta t))$,
a-2) No events - $((F, A), (1, A), (\Delta t, \Delta t))$.

The model description yields that the "change shift" transition is of *race age* policy, i.e. it needs to record the time spent on shift and not be restarted it when a failure occurs. During this processing, the required statistics that yield the simulation results are collected.

### 4.3. Experiments and Results

In the following we present the results of our simulation experiments, i.e. the statistics that were collected during the proxel-based simulation of the example model. The qustions that our simulation model provides answers to are the following:

a) What is the probability of having the machine running?, and
b) What is the probability that the machine has 5 or more failures?

The question (a) is a classical reliability analysis case, and the most typical question for a model like this one. In Figure 6 and Figure 7 we present the answers to the questions (a) and (b), correspondingly. The simulation parameters that we used were: a maximum simulation time $t$ = 300, and a time step $\Delta t$ = 0.5. Apparently, in Figure 6, we can observe that the model has not reached a steady-state during the simulation time of 300 time units, thus it needs to be simulated for a longer period of time. We did this, i.e. we ran the simulation up to time $t$ =10000 using the same time step, and the obtained solution for the steady state reliability of the system is periodically oscillating, as shown in Figure 8.

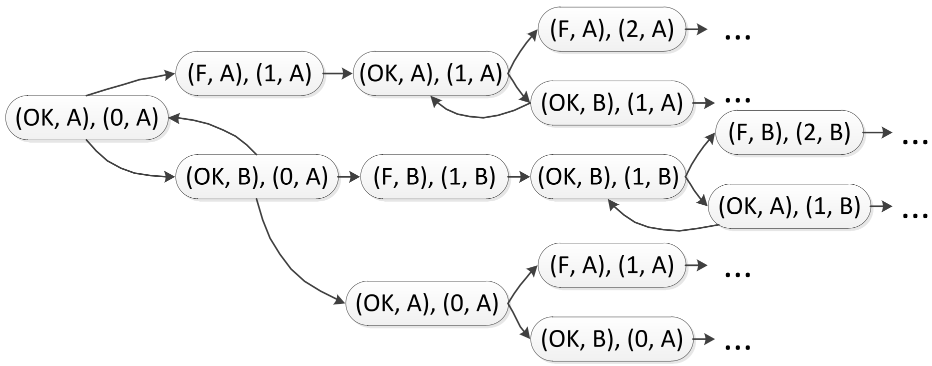


Figure 5: State-transition diagram of the unbounded Petri net model from Figure 4

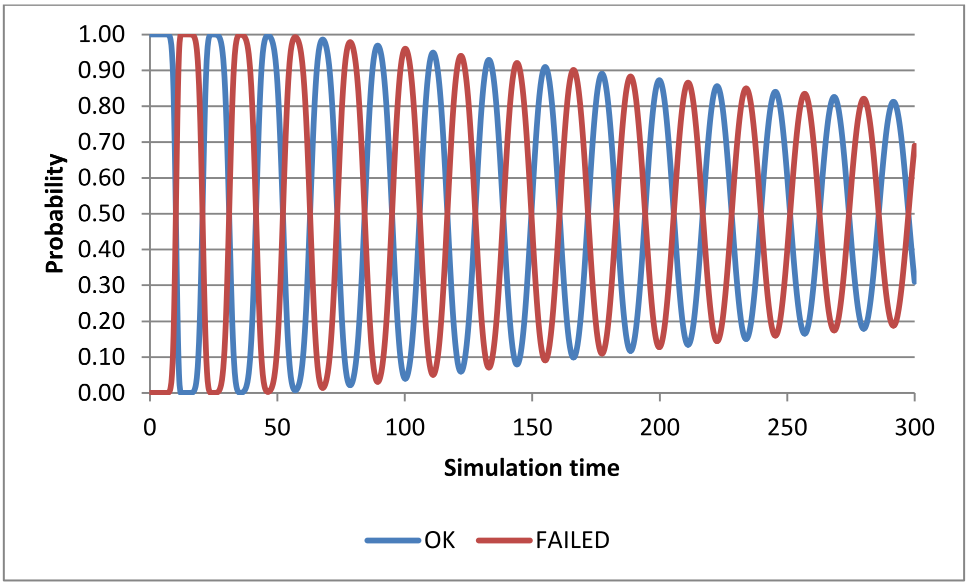


Figure 6: Transient solution for the 2 discrete states,  neglecting the fact of what repairman is on shift

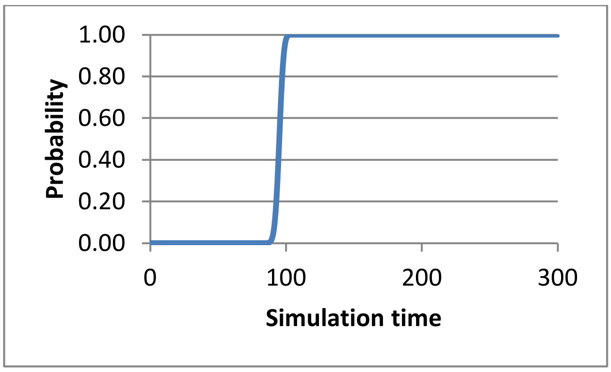


Figure 7: The probability of the machine having 5 or more failures

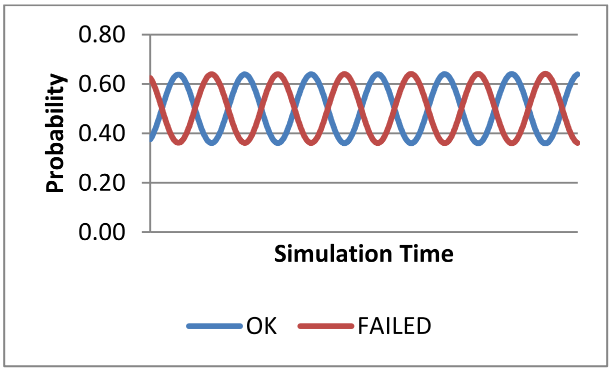


Figure 8: Steady-state solution of the discrete stochastic model

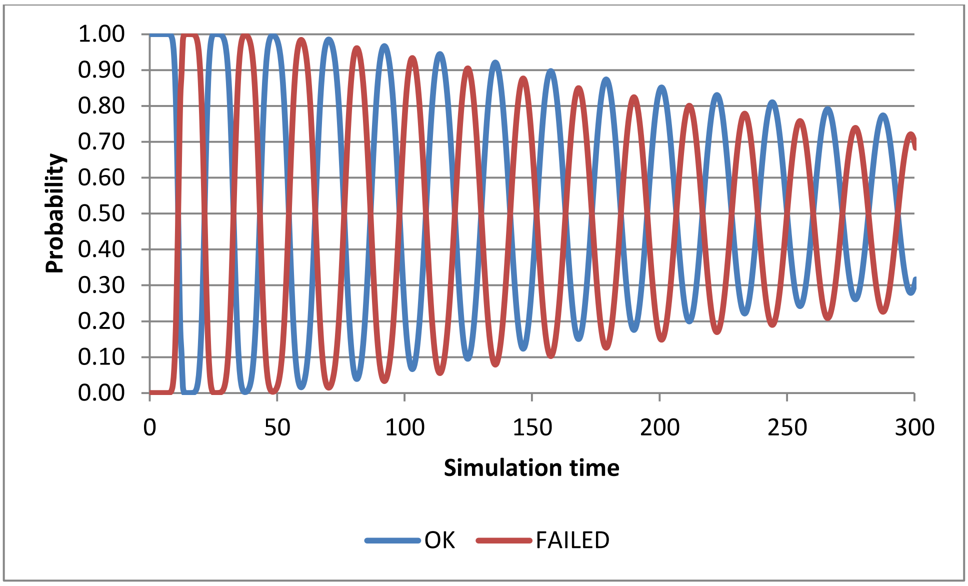


Figure 9: Transient solution for the 2 discrete states for the simplified model,  neglecting the fact of what repairman is on shift

The simulation results were obtained in 0.5 seconds on an Intel Core i5 2.53GHz workstation with a 4GB of RAM. The extended computation for the steady-state solution took longer, i.e. 3.5 minutes, which is still an acceptable running time.

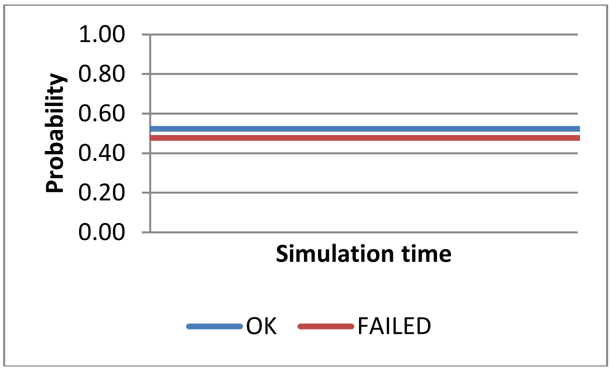


Figure 10: Steady-state solution of the simplified discrete stochastic model

For comparison, in Figure 9 and Figure 10, we provide the transient and steady state solutions of the same model, where the state-varying distributions are substituted with state-independent ones, i.e. with distributions with fixed parameters. Apparently, the results and their nature are quite different as the oscillating steady-state pattern is not present in the simplified model.

More specifically, the distribution functions that we used for the state-independent failure rates were the following:

- $f_{repair}(\quad)\sim N(11.0, 1.0)$,
- $f_{fail}(\quad)\sim E(150)$

## 5. SUMMARY AND OUTLOOK

We presented an approach to more realistically model and simulate failures that exhibit a wide range of dependencies which are typically neglected. Their neglecting, however, can provide highly misleading results, and thus, it is imperative to avoid their oversimplification. The proxel-based method has shown to be very accurate and highly flexible in describing the complex types of dependencies that typically occur in stochastic models. We anticipate extending of the presented work to provide a tool that would facilitate reliability modeling considering state-varying failure rate functions.

## REFERENCES

Baker, R. D. (2001). "Data-based modeling of the failure rate of repairable equipment." Lifetime Data Analysis 7(1): 65-83.

Bause, F. and P. S. Kritzinger (2002). Stochastic Petri Nets, Vieweg.

Billinton, R. and R. N. Allan (1992). Reliability evaluation of engineering systems, Plenum Press New York.

Coskun, A. K., R. Strong, et al. (2009). Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors, ACM.

Cox, D. R. (1955). "The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables." Proceedings of the Cambridge Philosophical Society 51(3): 433-4

Hassett, T. F., D. L. Dietrich, et al. (1995). "Time-varying failure rates in the availability and

reliability analysis of repairable systems." Reliability, IEEE Transactions on 44(1): 155-160.
Horton, G. (2002). "A new paradigm for the numerical simulation of stochastic Petri nets with general firing times." Proceedings of the European Simulation Symposium.
Isensee, C. and G. Horton (2005). "Approximation of Discrete Phase-Type Distributions." Proceedings of the 38th annual Symposium on Simulation: 99-106.
Lazarova-Molnar, S. (2005). The Proxel-Based Method: Formalisation, Analysis and Applications. Faculty of Informatics. Magdeburg, University of Magdeburg. Ph.D.
Lazarova-Molnar, S. (2008). State-Dependent Transitions in Discrete Stochastic Models: Deterministic Simulation Approach. Summer Computer Simulation Conference 2008.
Liberopoulos, G. and P. Tsarouhas (2005). "Reliability analysis of an automated pizza production line." Journal of Food Engineering 69(1): 79-96.
Lin, F. J., P. M. Chu, et al. (1987). "Protocol verification using reachability analysis: the state space explosion problem and relief strategies." ACM SIGCOMM Computer Communication Review 17(5): 126-135.
Proschan, F. (1963). "Theoretical explanation of observed decreasing failure rate." Technometrics 5(3): 375-383.
Retterath, B., S. S. Venkata, et al. (2005). "Impact of time-varying failure rates on distribution reliability." International Journal of Electrical Power and Energy Systems 27(9-10): 682-688.
Schroeder, B. and G. A. Gibson (2007). Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?, USENIX Association.
Sharma, S., K. S. Kahlon, et al. (2010). "Reliability and path length analysis of irregular fault tolerant multistage interconnection network." ACM SIGARCH Computer Architecture News 37(5): 16-23.
Stewart, W. J. (1994). Introduction to the Numerical Solution of Markov Chains., Princeton University Press.
Xie, K. and W. Li (2009). "Analytical model for unavailability due to aging failures in power systems." International Journal of Electrical Power and Energy Systems 31(7-8): 345-350.
Zhang, H., E. Cutright, et al. (2010). Time-varying failure rate for system reliability analysis in large-scale railway risk assessment simulation. Safety and Security in Railway Engineering: 29.

## AUTHORS BIOGRAPHY

**SANJA LAZAROVA-MOLNAR** is an Assistant Professor in Computer Science at United Arab Emirates University. She received her Diploma in Computer Science from the “Sts. Cyril and Methodius” University in Macedonia and her M.Sc. in Computational Visualistics and Ph.D. in Computer Science from “Otto-von-Guericke” University of Magdeburg in Germany. Her main research interests are simulation and modeling, with the main focus on their methodology for achievement of more accurate and valid results. Her e-mail is sanja@uaeu.ac.ae.