RELIABILITY INDICES

Flavio Frattini[1], Antonio Bovenzi[1], Javier Alonso[2], Kishor Trivedi[2]

[1]Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Naples, Italy

[2]Department of Electrical and Computer Engineering
Duke University
Durham, NC, USA

Computer and other technical systems are expected to be reliable during operation. However, their growing complexity, which is related to the intricate interdependencies among many heterogeneous components, makes the development of fault-free systems an unaffordable task, in terms of time and cost. On the other hand, quality standards impose strict requirements on the reliability attributes and measures [1] [2]. Fault-tolerance techniques are used to allow systems to continue their operations even in presence of component failures, although in a possibly degraded mode. Hence, the quantification of the system reliability in combination with system performance is necessary.

Reliability can be evaluated using several approaches, generally classified into two categories: measurements-based and model-based [3]. Generally, the former is an attractive way to estimate the reliability of a system since it is based on real operational data. But it is costly and time consuming since it is based on real operational data collected from the system or a prototype. Instead model-based approaches offer an abstraction of the real system avoiding unnecessary details and avoiding system implementation. These models can be used to compute the reliability with an acceptable approximation within affordable cost and time.

State-space models are often used because of their capacity of handling different failure/repair behaviors, such as imperfect coverage, correlated failures, and repair dependencies [3]. Continuous-time Markov Chains (CTMCs) are state-space models commonly used for performance and reliability analysis. Homogeneous Continuous Time Markov Chains (HCTMCs) assume the rates associated with events are time-independent and holding times exponentially distributed. Although HCTMCs are able to cover many cases, time-dependent rates and non-exponential distributions are also present in many real world situations [4, 5]. In such cases, Non Homogeneous Continuous Time Markov Chains (NHCTMCs), Semi-Markov Processes (SMPs), Markov Regenerative Processes (MRGPs), or Phase type approximation can be used.

In order to show how to compute the reliability of a system based on a probabilistic model, consider the example of a *two processors parallel redundant system with imperfect coverage,* adapted from [6]. The system has two processors with the same time-independent failure rate and single repair facility with a certain repair rate. *Imperfect coverage* means that not all individual processor failures are recovered from. If a *covered failure* occurs in one of the two processors, the other one continues working and the system is up, although in a degraded mode. If a *not-covered failure* occurs, the system fails. Such a situation can be easily modeled by means of a HCTMC (see also 2.1.1 *Discrete Time Markov Chains* and 2.1.2 *Continuous Time Markov Chains*). However, if we consider a time-dependent failure rate for

the processors, the model is no longer homogeneous, and the computation of the reliability becomes harder.
In the following, the reliability and related indices are formally defined; subsequently, we show how to compute them for the aforementioned example. In particular, three different techniques are presented: ***Piecewise Constant Approximation***, ***Phase type expansion***, and ***discrete event Simulation***. Finally, results obtained with these techniques are compared.

BASIC DEFINITIONS

Let *X* be the ***Time to Failure*** (TTF) (or *lifetime*) random variable of a system. It can be characterized either by the (cumulative) distribution function (*CDF*), the (probability) density function (*pdf*), or hazard rate function ($h(t)$).

The ***Cumulative Distribution Function*** (*CDF*) of the (non-negative) random variable *X* is simply defined as:

$$F_X(t) = P(X \leq t), \qquad 0 < t < \infty.$$

The ***probability density function*** (*pdf*) of *X* is defined by

$$f_X(t) = \frac{dF_X(t)}{dt}.$$

The ***Hazard Rate***, also called *Instantaneous Failure Rate,* is defined by

$$\hbar(t) = \frac{f(t)}{1 - F_X(t)}.$$

Reliability is defined by the Recommendation E.800 of the International Telecommunications Union (ITU-T) as the "ability of an item to perform a required function under given conditions for a given time interval." Hence, for a time interval $(t_0, t_0 + t]$, reliability $R(t|t_0)$ defines the probability that a system survives in this interval, assuming that the system was working at time t_0.
If $t_0 = 0$, $R(t|0)$ defines the probability that a system is up until time *t*,

$$R(t|0) = R(t) = P(X > t) = 1 - F_X(t).$$

Another reliability index of interest is the ***conditional reliability*** that defines the probability that a system survives in the interval $(t_0,\ t_0 + t]$ of duration t, given that the system survived until time t_0,

$$R_{t_0}(t) = \frac{R(t_0 + t)}{R(t_0)}.$$

Mean Time to Failure (MTTF) defines the expected life of a system,

$$E[X] = \int_0^\infty t f(t) dt = \int_0^\infty R(t) dt.$$

Hence, the MTTF is closely related to the reliability.

Availability is defined by ITU-T Recommendation E.800 as the "ability of an item to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval, assuming that the external resources, if required, are provided."

It is important to note that reliability is the probability of a system being failure-free operation during a time interval, while availability is the probability of a system being failure-free at a given instant of time.
Define the indicator random variable $I(t)$ that is equals to 1 when the system is up, 0 otherwise.

The ***Instantaneous Availability*** (or point availability) is defined as the probability that a system is up at time t,

$$A(t) = P(I(t) = 1).$$

In absence of any component or system repair, $A(t) = R(t)$.
Steady State Availability or Limiting Availability (A) is the limiting value of $A(t)$ when t approaches infinity,

$$A = \lim_{t \to \infty} A(t).$$

Under very general conditions, the Steady State Availability can be shown to be:

$$A = \frac{MTTF}{MTTF + MTTR}.$$

The ***Mean Time to Repair*** (MTTR) includes the time to detect a failure as well as the time to repair it.
The ***Mean Time Between Failures*** (MTBF), instead, is the sum of MTTF and MTTR:

$$MTBF = MTTF + MTTR\ .$$

Interval Availability (or *Average Availability*) is a measure of the proportion of time a system is up within a given interval of time $(0, t]$. It is defined as

$$A_I(t) = \frac{1}{t} \int_0^t A(x)dx.$$

It is easy to demonstrate that (when both limits exist)

$$A = \lim_{t \to \infty} A_I(t) = \lim_{t \to \infty} A(t).$$

For more details about availability indices, modeling, and evaluation, see Section 2.3.7 *Availability*.

Fault-tolerant systems are able to continue providing service even in presence of component failures, although perhaps in a degraded mode. Assuming that at initial state the system is operating at its maximum performance, when a failure occurs, system performance could degrade. This kind of system offers several levels of performance. As a consequence, the idea of combining performance and reliability/availability has been developed under the name of *Performability* [7].
Let S denote all possible configurations in which the system can perform its activity, and let $\{X(t), t \geq 0\}$ on S define a continuous time stochastic process describing the structure of the system at time t. Let $\pi_i(t)$ be the probability that the system is in state $i \in S$ at time t and π_i be the probability that the system is in state $i \in S$ when t approaches infinity. Associate a *reward rate* to every state indicating the performance level offered by the system in that state. r_i represents the reward obtained per unit time spent in state $i \in S$.

Let $Z(t) = r_{X(t)}$ represent the system ***Reward Rate at time t***. It can be shown that [8] $L_i(t) = \int_0^t \pi_i(\tau)d\tau$ is the expected total amount of time spent by the system in state *i* during the interval (0, t].

The ***Expected Instantaneous Reward Rate*** at time t is given by

$$E[Z(t)] = \sum_{i \in S} r_i \pi_i(t).$$

The ***Expected Steady State Reward Rate*** of the system is

$$E[Z] = \lim_{t \to \infty} E[Z(t)] = \sum_{i \in S} r_i \pi_i.$$

The ***Accumulated Reward in*** $(\mathbf{0}, \boldsymbol{t}]$ represents the total amount of work performed by system during the interval $(0, t]$:

$$Y(t) = \int_0^t Z(t) dt.$$

Then, the ***Expected Accumulated Reward in*** $(\mathbf{0}, \boldsymbol{t}]$ is the amount of work done by the system during the interval of time $(0, t]$:

$$E[Y(t)] = E[\int_0^t Z(t) dt] = \int_0^t E[Z(t)] dt = \sum_{i \in S} r_i \int_0^t \pi_i(t) dt = \sum_{i \in S} r_i L_i(t)$$

Figure 1, adapted from [7], presents an example of a Markov Reward Model (MRM) with three states. The reward rate is 2 for *State 1*, 1 for *State 2*, and 0 for *State 3*. *X(t)* represents the possible state variation of the model and *Z(t)* is the corresponding reward rate of the system. *Y(t)* is the accumulated reward over the time interval $(0, t]$.

For a CTMC with one or more absorbing states, we can also compute the ***Expected Accumulated Reward till Absorption***

$$E[Y(\infty)] = \sum_{i \in S} r_i \int_0^\infty \pi_i(x) dx = \sum_{i \in S} r_i L_i(\infty) = \sum_{i \in S} r_i \tau_i$$

where $\tau_i = L_i(\infty) = \lim_{t \to \infty} L_i(t)$ is the expected total time spent in state *i* before absorption [8].

The ***Distribution of the Accumulated Reward till Absorption*** $P(Y(\infty) \leq y)$ can also be computed [9, 10].

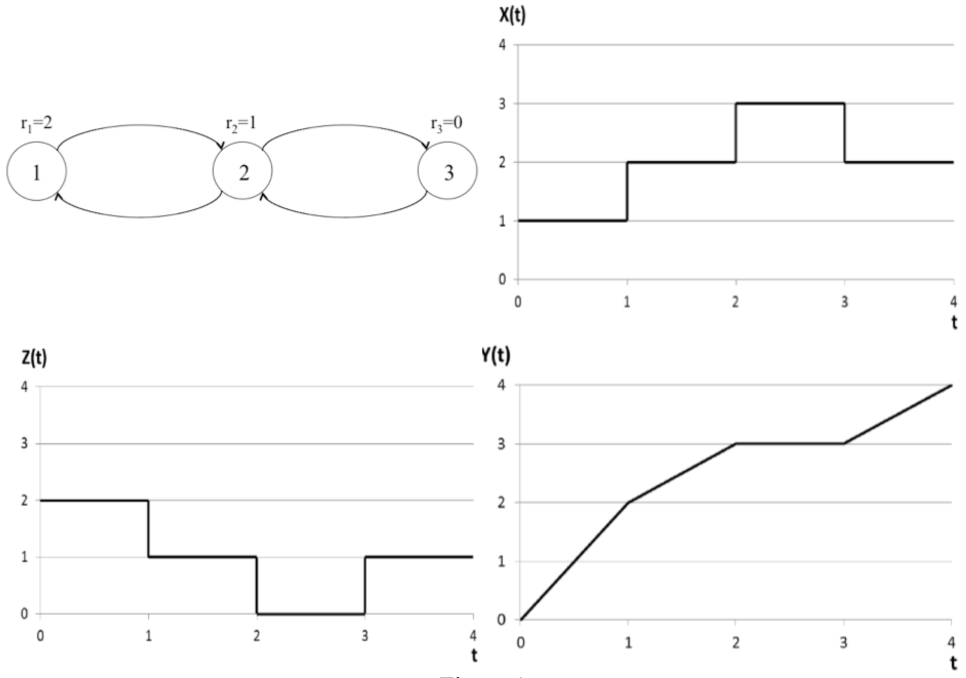


Figure 1

A SIMPLE EXAMPLE

In this section, we present in detail the above-mentioned system to show how to compute the reliability and performability indices described in the previous section.

Consider the *two processors parallel redundant system with imperfect coverage* introduced earlier. Let P_1 and P_2 be the two processors; both of them have the same time-dependent failure rate $\lambda(t)$, which is not dependent on the state or the task of the processor. The coverage factor, denoted by c, represents the proportion of individual processor failures from which the system can automatically be recovered. Once P_1 (or P_2) fails, if the failure is covered by the recovery strategy, the system can properly work, with only one processor, with degraded performance, otherwise it fails. The processor repair facility is characterized by a constant repair rate μ.

The system can be modeled by a Markov chain with three states (see Figure 2):

- *State 2*: both processors are properly working; the failure rate of each processor is $\lambda(t)$, hence the equivalent failure rate of this state is $2\lambda(t)$. If a failure covered by the recovery strategy occurs (rate $2\lambda(t)c$), the system goes to *State 1*, otherwise to *State 0* (rate $2\lambda(t)(1-c)$);
- *State 1*: one processor failed because of a covered failure; the system can continue working in degraded mode with one processor. In this case, two possible scenarios are possible: the failed processor is repaired (with constant repair rate μ) before the working one fails, then the system goes back to *State 2*; or the working processor fails (rate $\lambda(t)$), and thus the system fails (*State 0*);
- *State 0*: the system is down because either both processors failed, or a non-recoverable failure occurred. This is an absorbing state (the system can not be repaired if this state is reached).

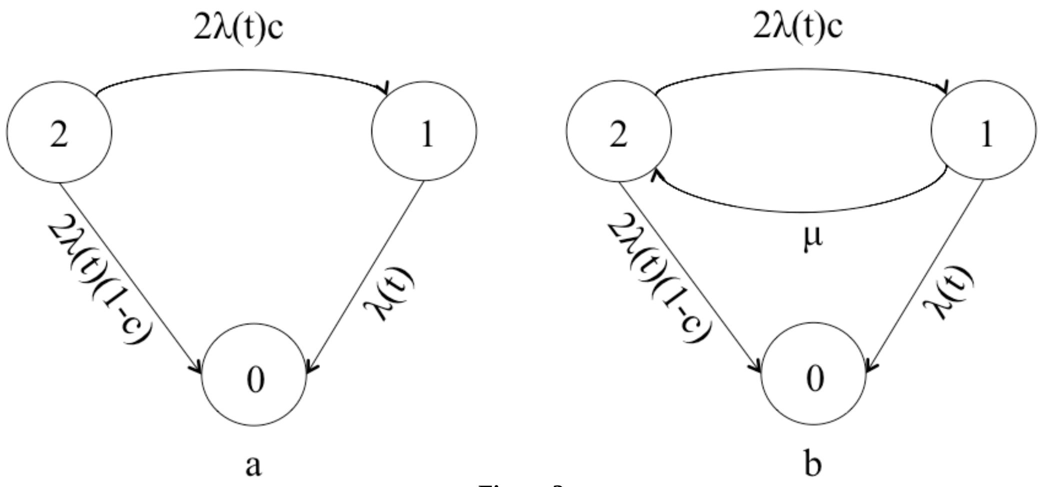


Figure 2

Note that in case there is no repair (Figure 2a) the model is an NHCTMC. Since all rates are dependent upon the same time dependent function $\lambda(t)$, the time-dependent infinitesimal generator matrix can be factored into a time-independent matrix and a scalar function of time. Hence we can get the solution to the NHCTMC by solving a related HCTMC [8].

When repair from *State 1* back to *State 2* is introduced (Figure 2b), we need to make two assumptions for the model to remain an NHCTMC:

1. repair is minimal, i.e., the repaired processor is in a state (age) equal to the one before its failure;
2. repair time is negligible compared to time to failure.

The vector $\pi(t)$, whose *i*-th element $\pi_i(t)$ is the probability that the system is in state *i* at time *t*, is equals to $[0 \quad 0 \quad 1]$ at $t = t_0$, and the infinitesimal generator matrix is (see also 2.1.2.1 *Definitions and Examples of CTMCs*):

$$Q(t) = \begin{bmatrix} 0 & 0 & 0 \\ \lambda(t) & -\mu - \lambda(t) & \mu \\ 2\lambda(t)(1-c) & 2\lambda(t)c & -2\lambda(t) \end{bmatrix}$$

The failure rate $\lambda(t)$ is assumed to be the hazard rate of a two-parameter Weibull distribution:

$$\lambda(t) = \frac{\alpha}{\beta}\left(\frac{t}{\beta}\right)^{\alpha-1}$$

Table 1 shows the values of the parameters of the system. α and β are chosen to obtain a Weibull distribution with an increasing failure rate [11]; the rest of the parameters are typical values used to simplify computations.

Parameter	**Value**
α	2.1
β	1.02
μ	3.33x10^{-1} days
c	0.9
t_0	0

Table 1 – Values of the parameters of the model.

We focus on the computation of the aforementioned *reliability* and *performability* indices for the described Markov model.
The transient behavior of a CTMC is defined by the Kolmogorov Ordinary Differential Equations (ODE) system [12]:

$$\dot{\pi}(t) = \pi(t)Q(t);$$
$$\sum_{i=1}^{m} \pi_i(t) = 1.$$

If $Q(t)$ is integrable, one solution exists and it is of the form $\pi(t) = \pi(t_0)\,\Phi\,(t, t_0)$. $\Phi\,(t, t_0)$ can be, then, evaluated using the Peano-Baker series [13] as:

$$\Phi\,(t, t_0) = I + \int_{t_0}^{t} Q(\tau_1)d\tau_1 + \int_{t_0}^{t} Q(\tau_1)\int_{t_0}^{\tau_1} Q(\tau_2)d\tau_2 d\tau_1$$
$$+ \int_{t_0}^{t} Q(\tau_1)\int_{t_0}^{\tau_1} Q(\tau_2) \dots \int_{t_0}^{\tau_{h-1}} Q(\tau_k)d\tau_k d\tau_{k-1} \dots d\tau_2 d\tau_1 + \cdots$$

However, the computation of $\Phi\,(t, t_0)$ for time-variant systems such as NHCTMCs is rather difficult. As a consequence, several methods have been proposed for the transient analysis of NHCTMCs as an alternative to the ODE approach (see also 2.1.2.2. *Transient Behaviour of CTMCs*), such as Piecewise Constant Approximation (PCA) [14] and Phase type Expansion (PH) [15].
The repair strategy of the system determines the method to use. Two different strategies have been considered [16]:

- *minimal repair* (or *as bad as before*);
- *maximal repair* (or *as good as new*), i.e., the repaired processor behaves as a new one (with its age reset to zero).

In the first case, assuming that $\frac{1}{\mu} \ll \beta\Gamma\left(\frac{1}{\alpha} + 1\right)$ (MTTF of the Weibull distribution), i.e., the recovery strategy is much faster than the inverse of the failure frequency, the model is an NHCTMC. Such a model requires only a *global clock* to describe all time dependent transition rates, since in every state, each component is as old as the system. To compute the reliability/performability of this model PCA can be used.
In the case of maximal repair, instead, the system is non-Markovian, since the transition rates from *State 2* to *State 1*, and from *State 1* to *State 0* depend on how long the system has been in *State 2* or *State 1*, respectively. Furthermore failure transition rate from *State 1* depends on time spent by the non-failed component in *State 2* and in *State 1*. We note that for an SMP all transition rates can depend on local time. In the current model with maximal repair, neither pure global clock nor pure local clock suffices. The model is neither an SMP nor an NHCTMC. However, by using PH approximation we can solve the problem.
For *performability* analysis, rewards are simply attached to the states of the Markov model by counting the number of active processors. In *State 2*, two processors are working, and hence, the reward rate is equal to 2; in *State 1*, where only one processor is working, the reward rate is 1. A reward rate equals to 0 is associated to the absorbing *State 0*. Hence, the reward rates are $r_2 = 2,\ r_1 = 1,\ r_0 = 0$.

PICEWISE CONSTANT APPROXIMATION

Piecewise constant approximation is based on considering a time-variant function $f(t)$ as constant in certain intervals. Given the time interval $[0, t_0]$, it is divided into $n+1$ shorter intervals of length δ: $t \in [0, t_0] \;\longrightarrow\; t \in [i\delta, (i+1)\delta], i = 0, 1, \dots, n$, where the function assumes the constant value $f(i\delta)$. In the case of the failure rate $\lambda(t)$ of the considered Markov model:

$$\lambda(t) = \begin{cases} \lambda(\frac{\delta}{2}) & 0 \le t < \delta \\ \lambda(\frac{\delta}{2} + \delta) & \delta \le t < 2\delta \\ \vdots & \vdots \\ \lambda(\frac{\delta}{2} + i\delta) & i\delta \le t < (i+1)\delta \\ \vdots & \vdots \end{cases} \quad ;\; i = 0, 1, \dots, n.$$

Figure 3 shows both the Weibull failure rate of the considered problem and its piecewise constant approximation with $\delta = 0.5$. With selected parameters of the Weibull function the system has a monotonically increasing failure rate.

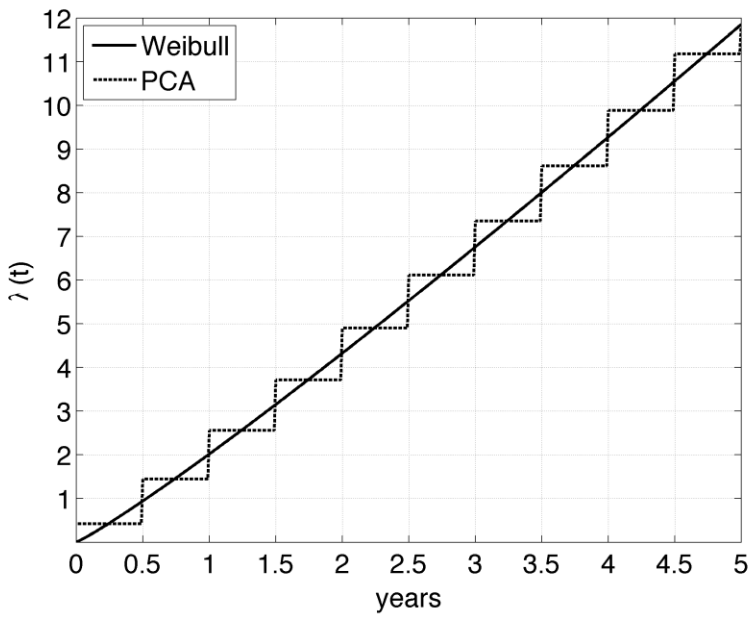


Figure 3

As a consequence of the failure rate approximation, also the infinitesimal generator matrix $Q(t)$ makes discrete changes:

$$Q(t) = \begin{cases} Q(\frac{\delta}{2}) & 0 \le t < \delta \\ Q(\frac{\delta}{2} + \delta) & \delta \le t < 2\delta \\ \vdots & \vdots \\ Q(\frac{\delta}{2} + i\delta) & i\delta \le t < (i+1)\delta \\ \vdots & \vdots \end{cases} \quad ;\ i = 0, 1, \dots, n$$

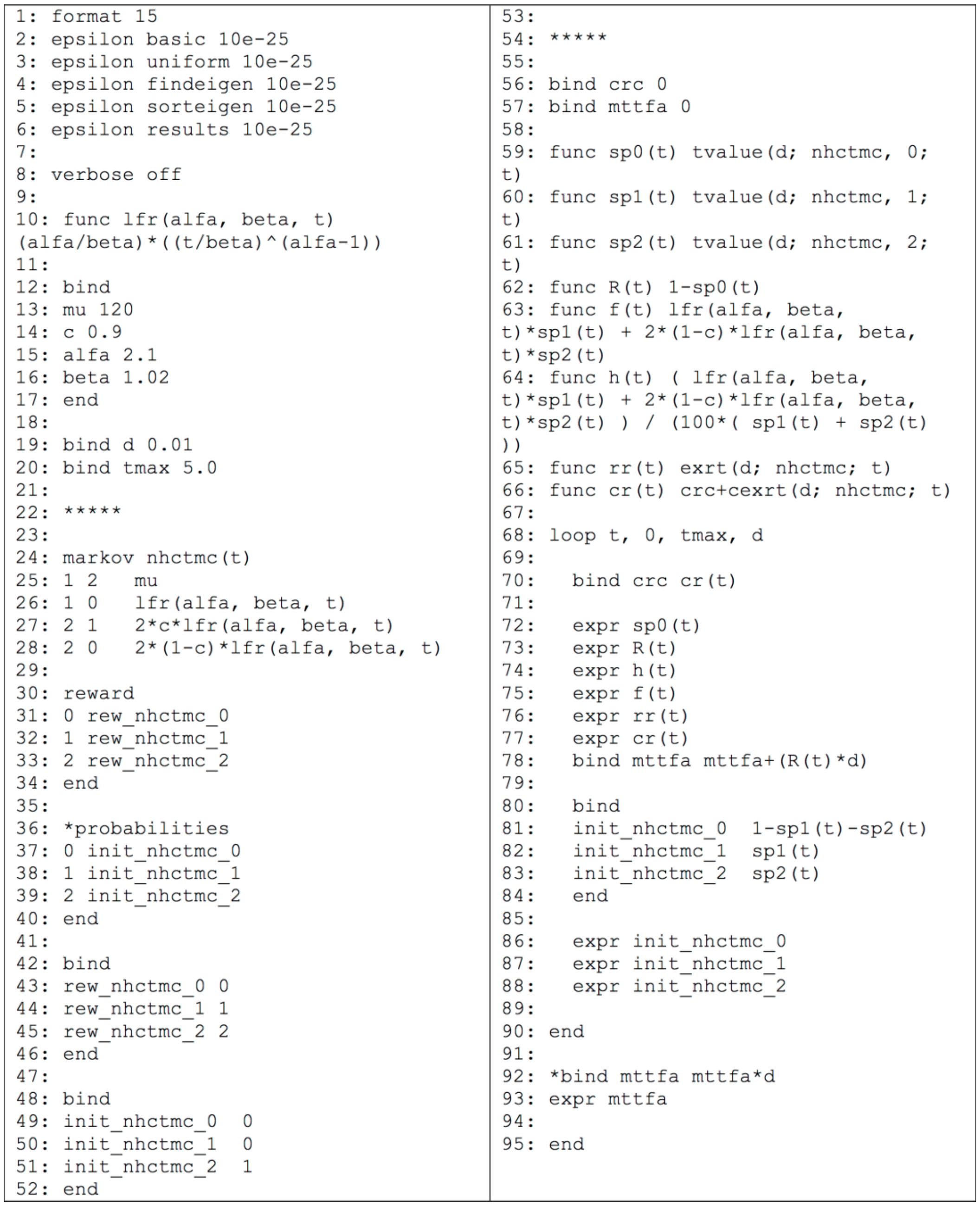
and the system is an HCTMC in each interval $[i\delta, (i+1)\delta], i = 0, 1, \dots, n$. Hence, the transient behavior of the state probabilities can be computed as:

$$\pi(t) = \begin{cases} \pi(0)e^{Q(\frac{\delta}{2})\cdot(t-0)} & 0 \le t < \delta \\ \pi(\delta)e^{Q(\frac{\delta}{2}+\delta)\cdot(t-\delta)} & \delta \le t < 2\delta \\ \vdots & \vdots \\ \pi(i\delta)e^{Q(\frac{\delta}{2}+i\delta)\cdot(t-i\delta)} & i\delta \le t < (i+1)\delta \\ \vdots & \vdots \end{cases} \quad ;\ i = 0, 1, \dots, n$$

where

$$\pi(\delta) = \pi(0)e^{Q\left(\frac{\delta}{2}\right)\cdot(\delta-0)}$$
$$\pi(2\delta) = \pi(\delta)e^{Q\left(\frac{\delta}{2}+\delta\right)\cdot(2\delta-\delta)}$$
$$\vdots$$
$$\pi(i\delta) = \pi((i-1)\delta)e^{Q\left(\frac{\delta}{2}(i-1)\delta\right)\cdot\delta}$$
$$\vdots$$

Reliability indices for the *two processors system*, as well as performability indices can be computed using SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) [17], as shown in Figure 4.



```
format 15
epsilon basic 10e-25
epsilon uniform 10e-25
epsilon findeigen 10e-25
epsilon sorteigen 10e-25
epsilon results 10e-25

verbose off

func lfr(alfa, beta, t)
(alfa/beta)*((t/beta)^(alfa-1))

bind
mu 120
c 0.9
alfa 2.1
beta 1.02
end

bind d 0.01
bind tmax 5.0

*****

markov nhctmc(t)
1 2   mu
1 0   lfr(alfa, beta, t)
2 1   2*c*lfr(alfa, beta, t)
2 0   2*(1-c)*lfr(alfa, beta, t)

reward
0 rew_nhctmc_0
1 rew_nhctmc_1
2 rew_nhctmc_2
end

*probabilities
0 init_nhctmc_0
1 init_nhctmc_1
2 init_nhctmc_2
end

bind
rew_nhctmc_0 0
rew_nhctmc_1 1
rew_nhctmc_2 2
end

bind
init_nhctmc_0  0
init_nhctmc_1  0
init_nhctmc_2  1
end

*****

bind crc 0
bind mttfa 0

func sp0(t) tvalue(d; nhctmc, 0;
t)
func sp1(t) tvalue(d; nhctmc, 1;
t)
func sp2(t) tvalue(d; nhctmc, 2;
t)
func R(t) 1-sp0(t)
func f(t) lfr(alfa, beta,
t)*sp1(t) + 2*(1-c)*lfr(alfa, beta,
t)*sp2(t)
func h(t) ( lfr(alfa, beta,
t)*sp1(t) + 2*(1-c)*lfr(alfa, beta,
t)*sp2(t) ) / (100*( sp1(t) + sp2(t)
))
func rr(t) exrt(d; nhctmc; t)
func cr(t) crc+cexrt(d; nhctmc; t)

loop t, 0, tmax, d

  bind crc cr(t)

  expr sp0(t)
  expr R(t)
  expr h(t)
  expr f(t)
  expr rr(t)
  expr cr(t)
  bind mttfa mttfa+(R(t)*d)

  bind
  init_nhctmc_0  1-sp1(t)-sp2(t)
  init_nhctmc_1  sp1(t)
  init_nhctmc_2  sp2(t)
  end

  expr init_nhctmc_0
  expr init_nhctmc_1
  expr init_nhctmc_2

end

*bind mttfa mttfa*d
expr mttfa

end
```

Figure 4

Since the model presents an absorbing state (*State 0*), i.e. there is no repair strategy for the failure of the whole system, availability is not considered. Indeed, in this case, $A(t) = R(t)$.

Reliability. Since the CTMC has an absorbing state that represents the failed system (*State* 0), $\pi_0(t)$ is the probability that the system has failed at or before time *t*; the reliability can be simply computed as [8]:

$$R(t) = 1 - \pi_0(t).$$

Figure 5 shows the reliability of the considered system for different values of δ. We note that the two approximations are very similar.

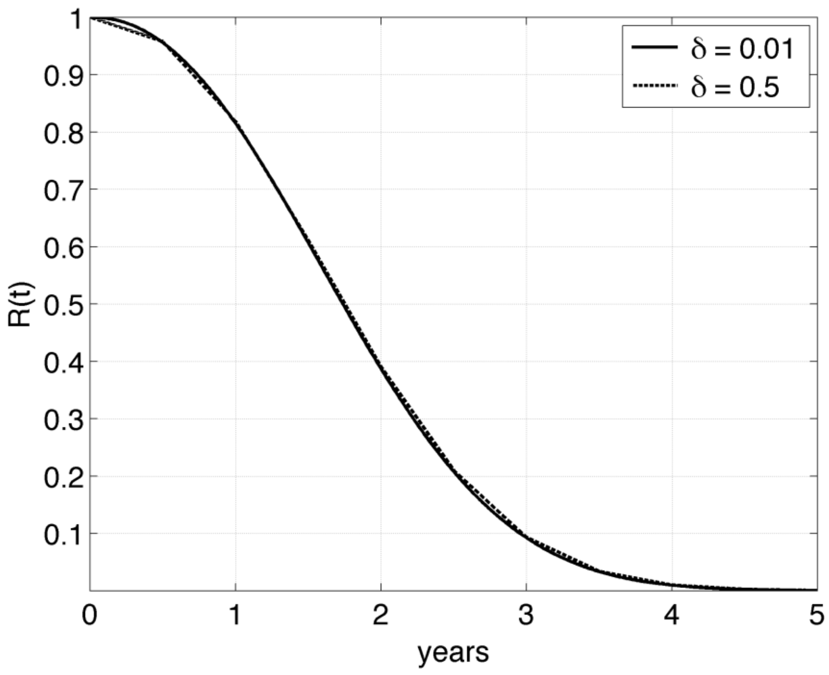


Figure 5

Distribution function. The cumulative distribution function of the time to failure of the system $F(t) = \pi_0(t)$ [8]. Figure 6 shows the $F(t)$ function of the considered system.

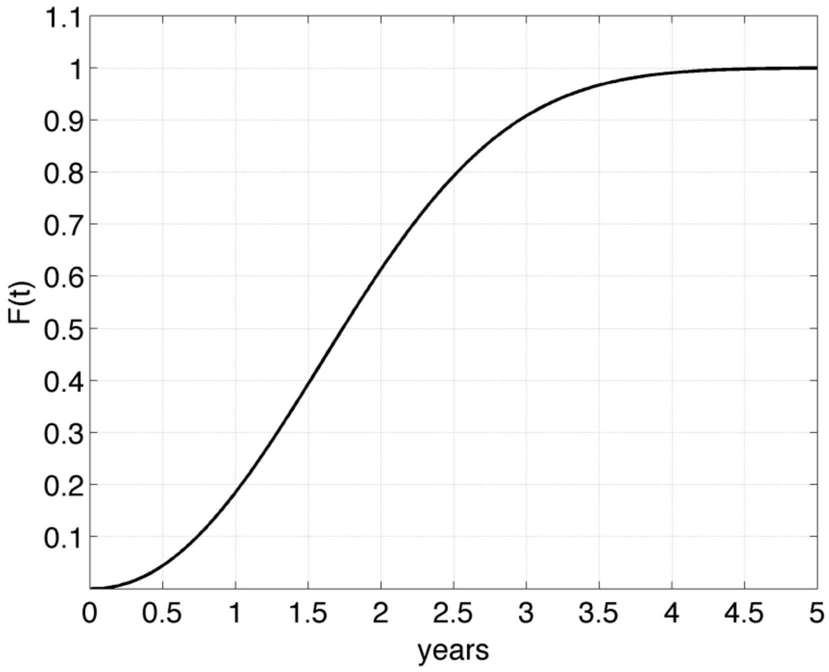


Figure 6

Probability Density function. The probability density function (*pdf*) of the time to failure of the system is presented in Figure 7. It is computed as $f(t) = \frac{d\pi_0(t)}{dt} = \lambda(t)\pi_1(t) + 2\lambda(t)(1-c)\pi_2(t)$ (see line 63 in Figure 4 for the code).

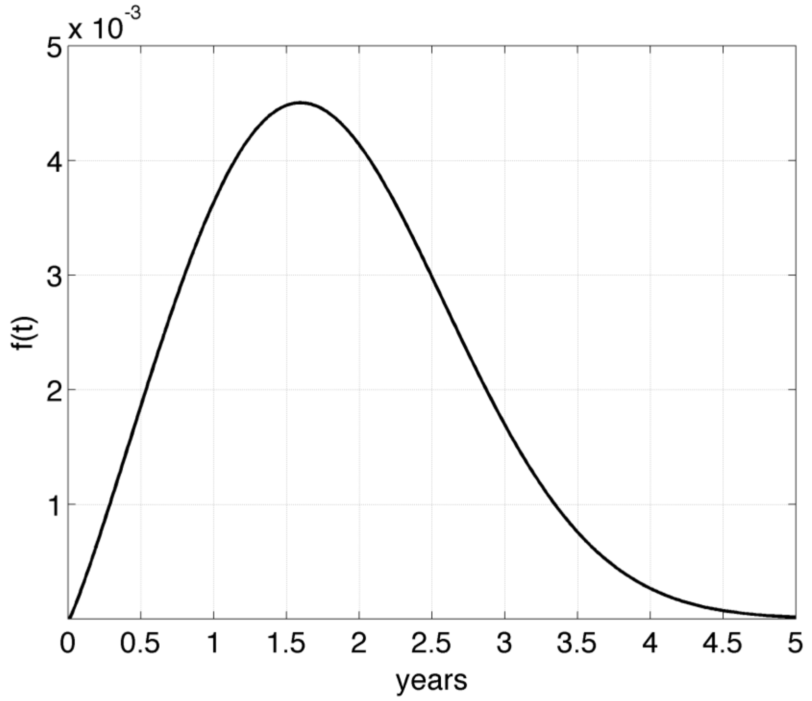


Figure 7

Hazard rate. The $h(t)$ is presented in Figure 8. It is computed considering the state probabilities as described in [18]: $h(t) = \frac{\lambda(t)\pi_1(t)+2\lambda(t)(1-c)\pi_2(t)}{\pi_1(t)+\pi_2(t)}$ (see line 64 in Figure 4). It has an increasing trend, but it is much less than the failure rate of each processor. Hence, the redundancy of processors and the recovery strategy are able to increase the reliability of the whole system.

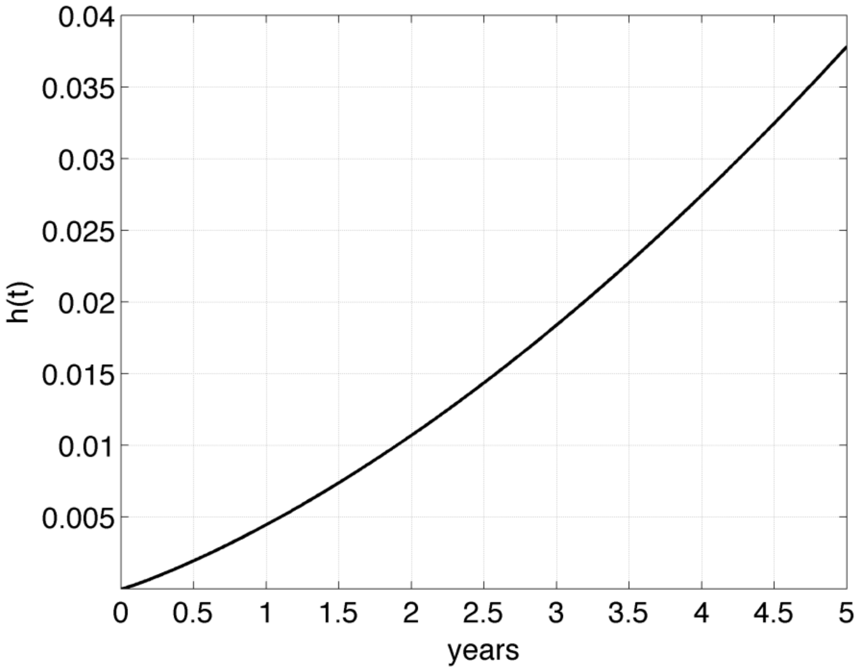


Figure 8

Mean Time To Failure. The MTTF of the considered system is 1.81 years. It is computed considering the approximate numerical solution of the integral $\int_0^\infty R(t)dt$ as the sum of the values of $R(t)$ in each time interval of length δ times δ itself (see lines 57, 80, and 96 in Figure 4).
If we want to increase the average lifetime of the considered system, without changing the number of processors of the system, we should:

- use processors with a smaller failure rate;
- improve the detection strategy to increase the coverage;
- improve the recovery strategy to increase the repair rate.

Expected Instantaneous Reward Rate. Figure 9 shows the expected instantaneous reward rate along time. It is computed in SHARPE with the *exrt()* function (see line 65 in Figure 4). The trend is very similar to the one of the reliability of the system, apart from the amplitude of the curve. Such a measure is very important in order to quantify the performance of the system in presence of failure. For instance, suppose the reward rate being the MIPS of the processors and that it is required for the system to perform at least 0.5 MIPS. The expected instantaneous reward rate shows that after about 2.3 years the system is no longer able to provide the desired service level. Hence, a renewal strategy is necessary.

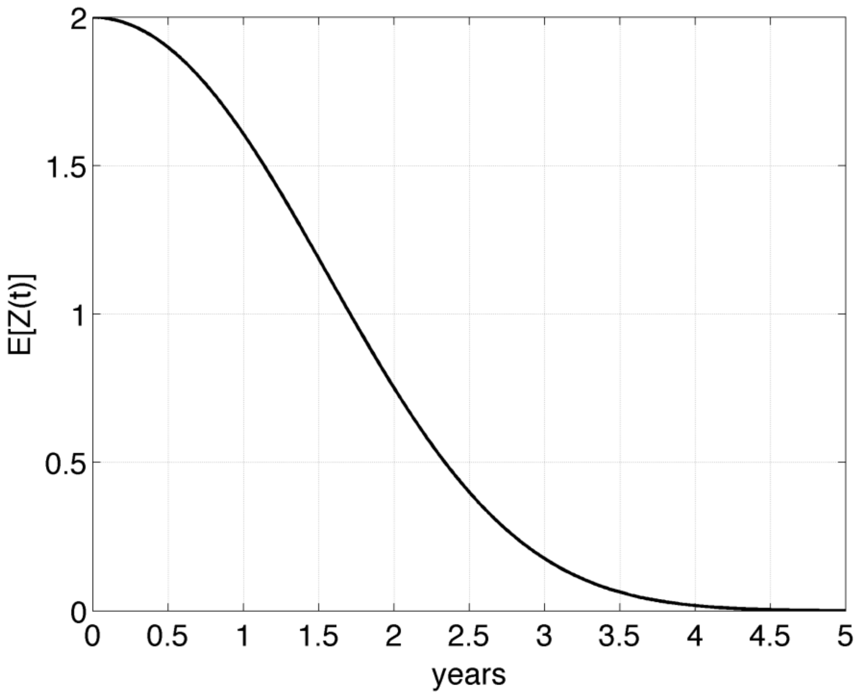


Figure 9

Expected Accumulated Reward. The behavior of such an index is shown in Figure 10 and it is computed in SHARPE by summing the results of the *cexrt()* function in all the time intervals of length δ (see lines 56 and 66 in Figure 4). The maximum value is 3.55, reached after about 4 years. Since the model has an absorbing state, the line *y=3.55* is an asymptote and this value represents the ***Expected Accumulated Reward till Absorption***, i.e., the expected reward accumulated by the system during its lifetime. Hence, if we consider again the reward rate being the MIPS, the maximum number of instructions the system is able to perform during the observation period $(0, 5]$ years is 1.12x10^{14}.

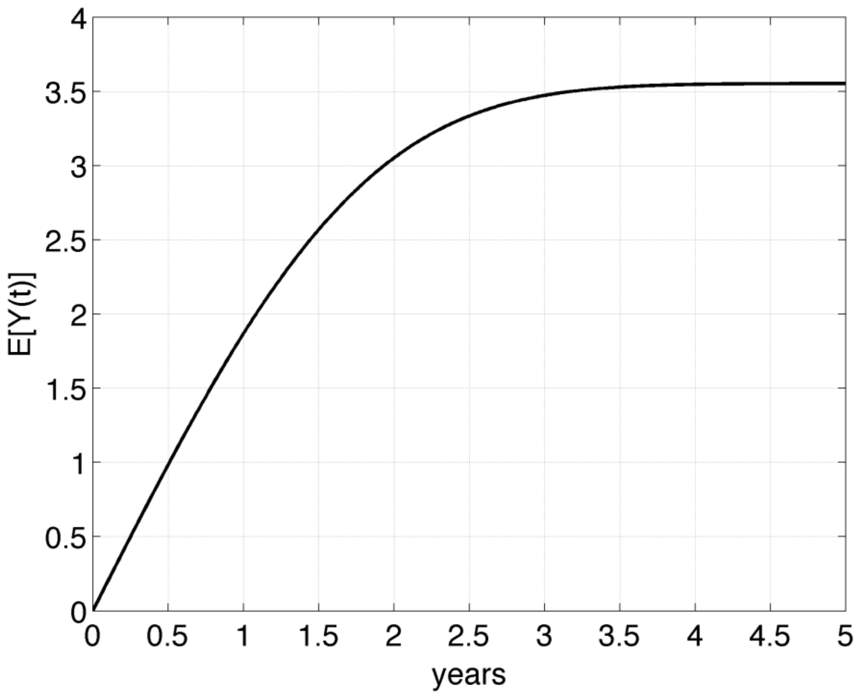


Figure 10

PHASE TYPE EXPANSION

In order to present how the repair policy affects the reliability and performability of the system, now we consider the maximal repair policy. The failure times modeled by Weibull distribution is approximated using the phase type expansion technique [15]. Figure 11 shows an alternative way for the representation of the system, which makes simpler the description of the phase type expansion technique and emphasizes the different clocks of each processor. In this model, the failed component and the respective component failure rate are distinguished. For instance, if the system is in *State (1,1)* both processors are up; in the *States (0,1)* and *(1,0),* one unique processor failed, P_1 and P_2, respectively. Finally the *State (0,0)* is the absorbing state of the model, which represents that both processors failed, and thus the system failed as well.

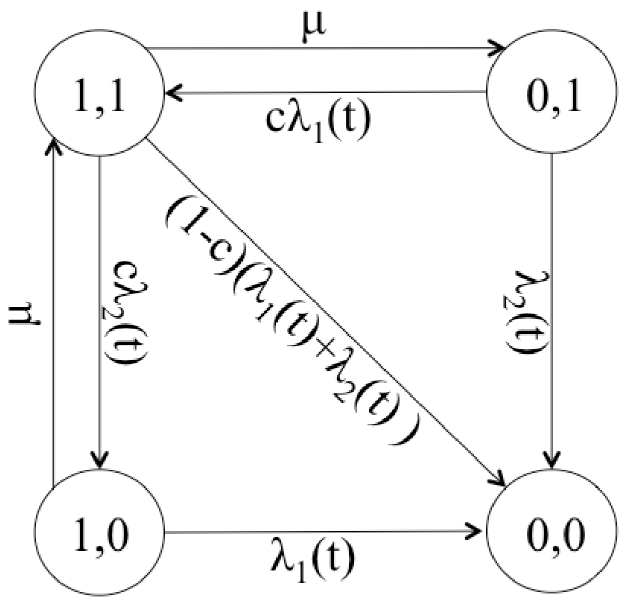


Figure 11

The approximation of the Weibull distribution can be obtained by using Phase type distributions. PH distributions are introduced in [15] and have been widely used in stochastic modeling since they have different practical advantages: the Markovian properties, the closure properties, and the approximating properties. Many studies propose algorithms and numerical techniques for parameter fitting [19, 11], and/or for comparing several approximations of non-exponential distributions, such as lognormal and Weibull. In [15] computation of reliability indices by means of PH distributions is discussed. In [20] it is described the applicability of PH distributions for modeling queuing networks. In [21] the authors compare several PH distributions, both continuous and discrete, and investigate which one best approximates a stochastic model.
A PH distribution is defined as the distribution of time to absorption of a CTMC with *n* transient states and one absorbing state (*State n+1*). The infinitesimal generator matrix Q of the CTMC can be partitioned as follows:

$$\begin{pmatrix} Q' & Q^0 \\ 0 & 0 \end{pmatrix}$$

where Q' is a *(n×n)* matrix that describes the transition rates between transient states of the CTMC and Q^0 is the column vector of the transition rates to the absorbing state *0*. The tuple $({\pi'}^{(0)}, Q')$ completely represents the PH distribution of order *n*, where $\pi'(0)$ is the *(n+1)* initial probability vector. For the formal definition of PH distributions and their properties see Section 2.2.6.2 *Phase-Type Distribution*.

The steps involved in PH expansion are: *i)* the choice of stage combinations, e.g., stages in series or parallels, and *ii)* the derivation of the PH distribution parameters based on parameters of the original distribution. Depending on the approximated stochastic model, several stage combinations can be selected. In [22] some examples are provided. We use the n-stages Erlang distribution. The stages are sequentially traversed and the non-negative continuous random variable represents the sum of the *n* s-independent exponentially distributed random variables with rate γ. Hence, the *pdf* of the resulting random variable *X*, which will be the approximation of the Weibull distribution, is the *pdf* of the n-stages Erlang distribution:

$$f(x) = \frac{\gamma^n x^{n-1}}{(n-1)!} e^{-\gamma x}$$

Depending on the distribution to be approximated, different approaches can be used for evaluating the parameters of PH distribution (i.e., γ and n), like moment matching [22], function fitting and hybrid methods [11]. We will use moment matching approach since it allows us to compute a closed form expression for the failure rate γ and, also, for the number of stages, n:

$$m_1 = \frac{n}{\gamma} \quad m_2 = \frac{n(n+1)}{\gamma^2} \quad => \quad \gamma = \frac{m_1}{m_2 - {m_1}^2}, \qquad n = \frac{{m_1}^2}{m_2 - {m_1}^2}$$

Equalizing m_1 and m_2 to the first two moments of Weibull distribution,

$$m_1 = bG(1+\frac{1}{a}), \quad m_2 = b^2 G(1+\frac{2}{a})$$

By substituting $a = 2.1,\ b = 1.02$ (see Table 1) in the above formulas, we obtain $n=4$ and $\gamma = 4.427$. If the computed value n is non-integer, it has to be rounded to the nearest integer. In this case, γ has to be recalculated accordingly.

Figure 12 shows the comparison of the Weibull *pdf* and the approximation obtained with PH expansion; the maximum error is *0.14*. It is worth noting that, while selecting a proper stage combination, both the *pdf* and the *hazard rate* have to be checked. Figure 13 shows the comparison of the hazard rates. Since the hazard rates look quite different it would be preferable to adopt a different stage combination in order to have a better approximation. However, the identification of the best PH approximation for the considered Weibull distribution is out of the scope of this paper.

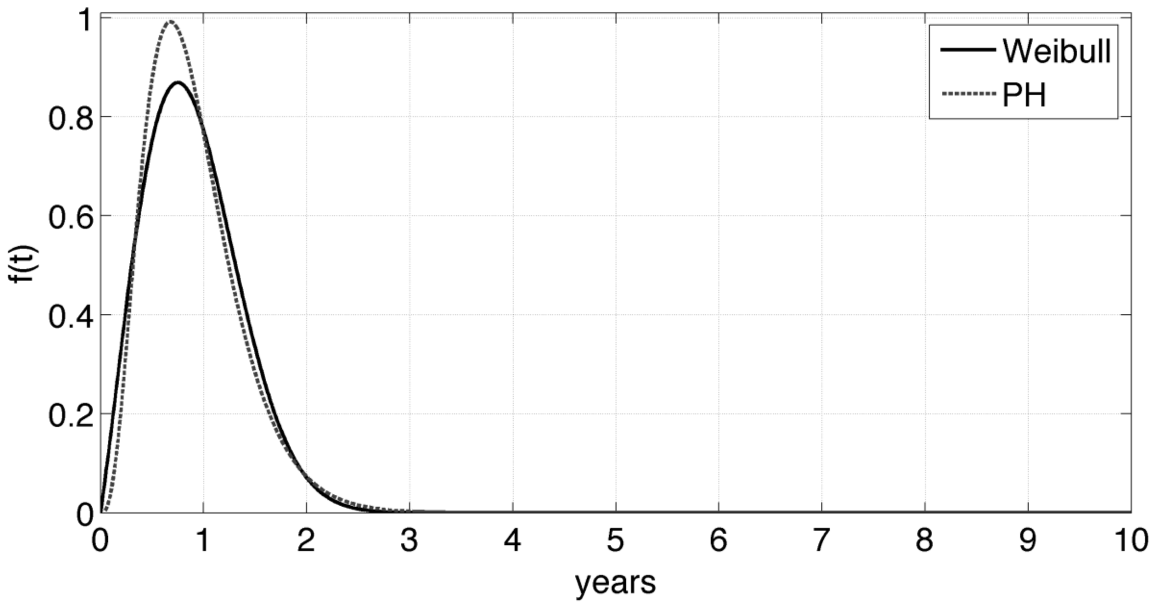


Figure 12

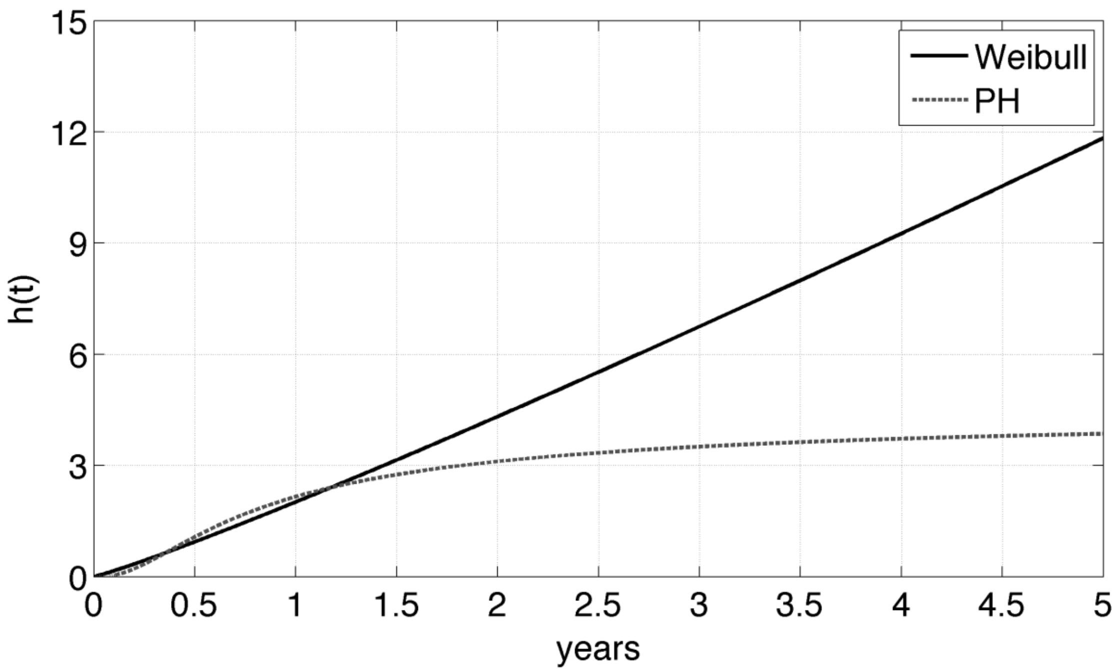


Figure 13

The system model resulting from using PH expansion is presented in Figure 14. System states are represented specifying the number of phases for each processor. Hence, if the system is in *State (4,4)*, both the components are up and they are at the stages 4. The failure of component P_1 is represented by *States (0,j)*, with *j=1,..,4*. The failure of component P_2 is represented by *States (i,0)*, with *i=1,..,4*. *State (0,0)* represents the system failure. A component failure, i.e. the transitions from *States (1,j)* to *(0,j)*, or the transitions from *States (i,1)* to *(i,0)*, can be covered with probability *c*. When a failure is not covered the system fails (transitions from *States (i,1)* and *(1,j)* to *State (0,0)*). When a component fails, it can be repaired or restored, becoming as good as new. This is represented by the transitions from *States (0,j)* to *States (4,j)* and by the transitions from *States (i,0)* to *States (i,4)*.

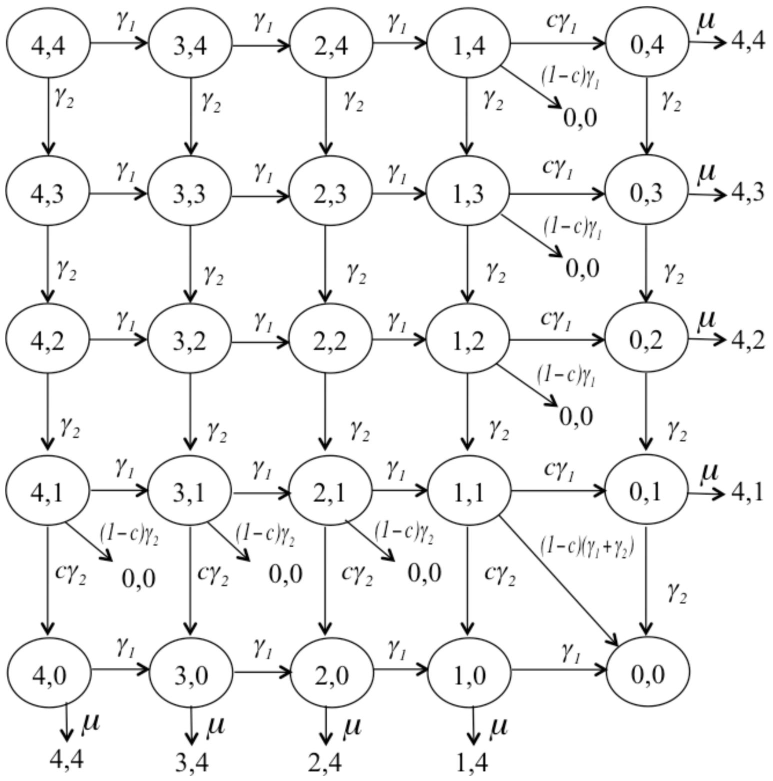


Figure 14

An example of the SHARPE code for the specific example under analysis is shown in Figure 19.

Reliability. The system reliability is computed as $R(t) = 1 - \pi_0(t)$ (see Figure 15). The reliability function is computed using SHARPE (see line 165 in Figure 19 for the code).

Cumulative Distribution function. The *CDF* function is presented in Figure 16. It is computed by means of SHARPE *tvalue()* function (see line 166 in Figure 19).

Probability Density function. Figure 17 presents the *pdf* of the time to failure computed in SHARPE (see line 167 in Figure 19).

Hazard rate. The failure rate is shown in Figure 18. It is computed using the formula defined in the basic definitions section (see line 170 in Figure 19).

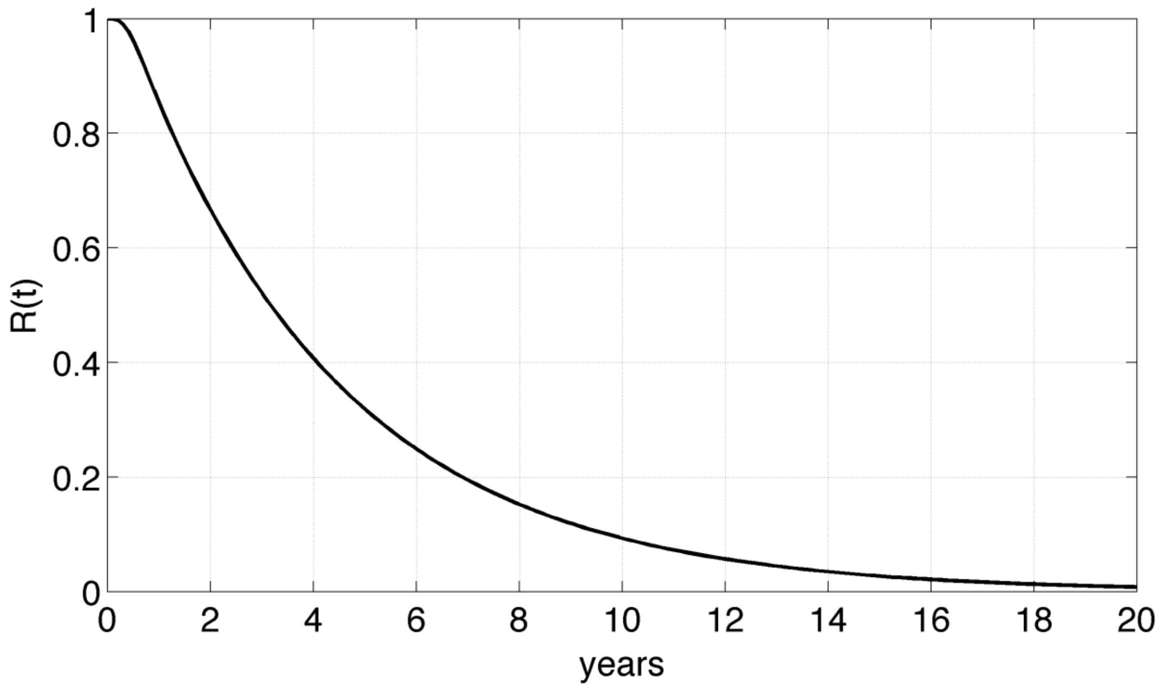


Figure 15

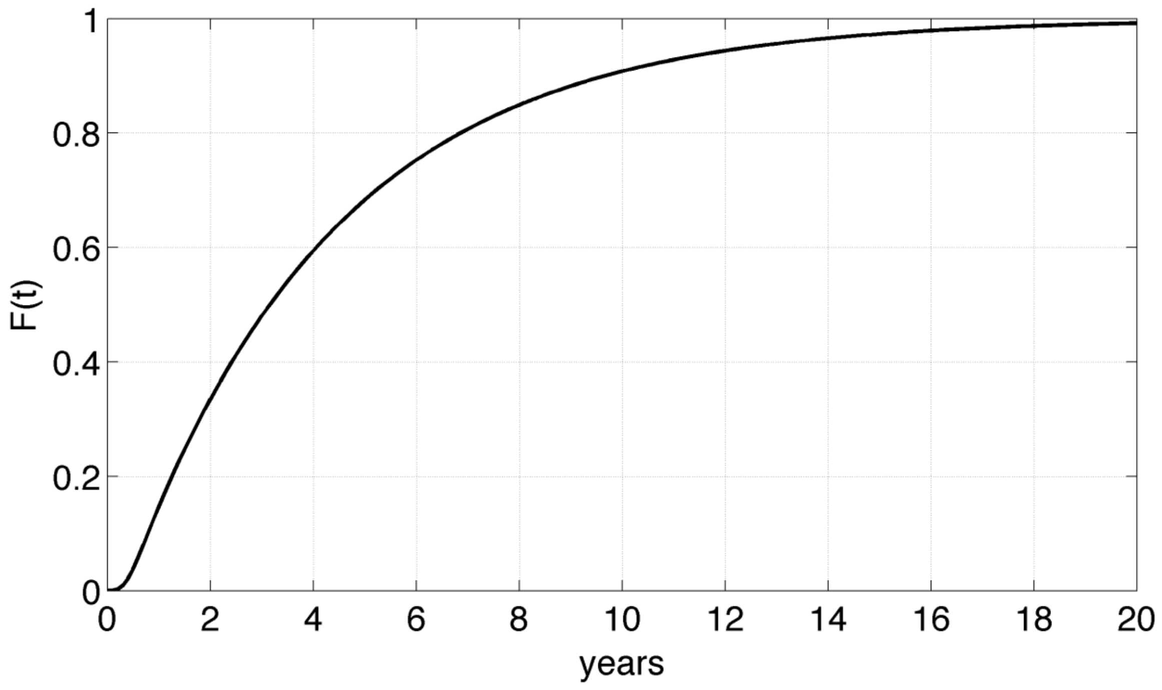


Figure 16

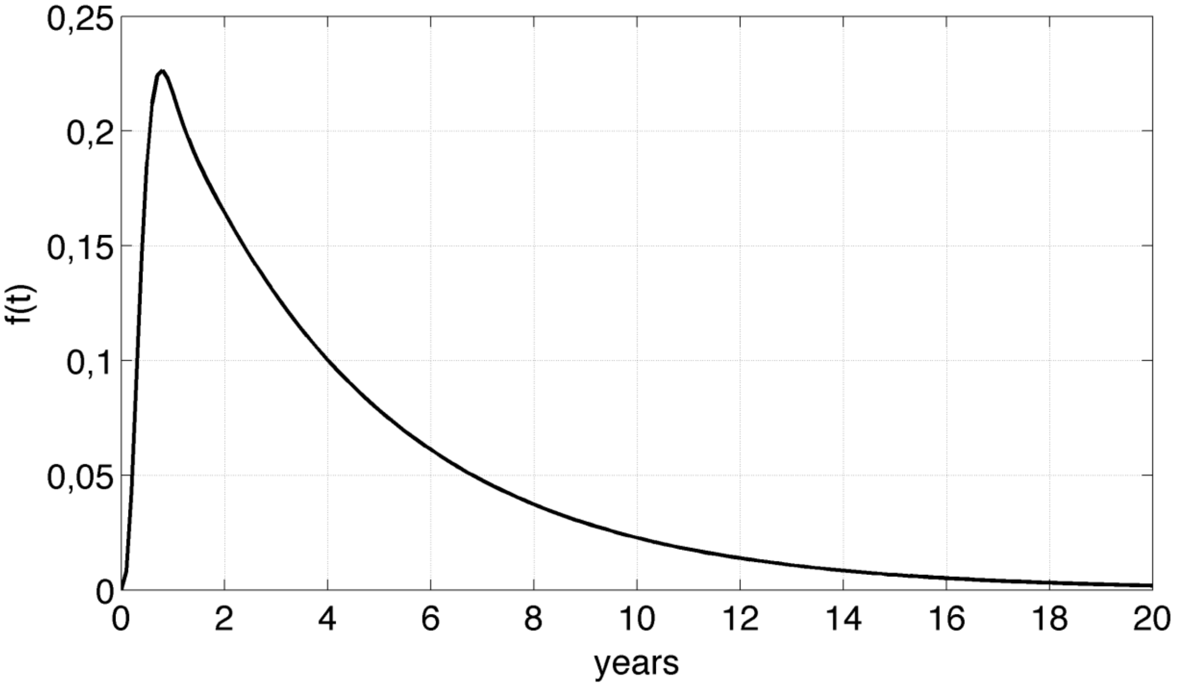


Figure 17

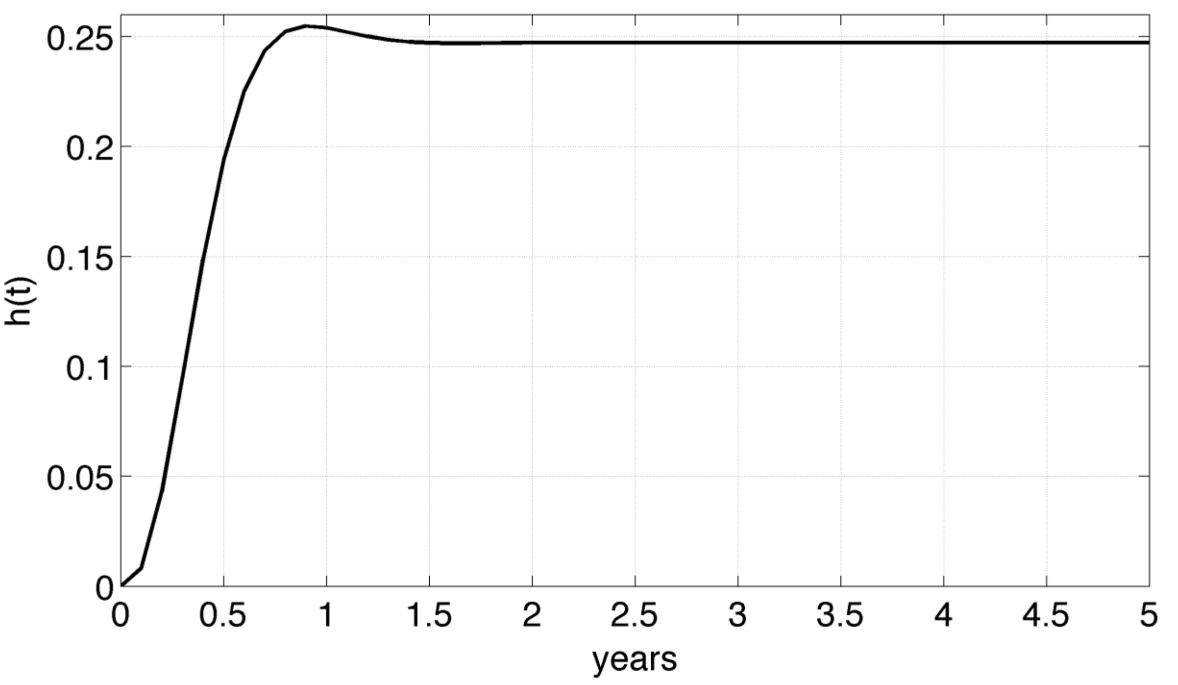


Figure 18

```
format 8
factor on
markov PHModel(lambdax, c, mu, lambday)
1 0  lambdax
1 21   mu
2 1   lambdax
2 22   mu
…
24 23 lambdax
24 19   lambday
*Reward definition
reward
24 r2
23 r2
…
2 r1
1 r1
end
* Initial Probabilities defined
0 init_PHModel_0
1 init_PHModel_1
…
22 init_PHModel_22
23 init_PHModel_23
24 init_PHModel_24
end
* Initial Probabilities assigned:
bind
   init_PHModel_0 0
   init_PHModel_1 0
   init_PHModel_2 0
…
   init_PHModel_23 0
   init_PHModel_24 1
*Rewards
   r2 2
   r1 1
end
echo
echo *** Outputs ***
bind lambdax 4.427
bind c 0.9
bind mu 120
bind lambday 4.427
bind step 0.01
*Compute Reliability Indices*
func Reliability(t) 1-tvalue(t;PHModel; lambdax, c, mu, lambday)
func CDF(t) tvalue(t;PHModel; lambdax, c, mu, lambday)
func pdf(t) (tvalue(t;PHModel,21; lambdax, c, mu, lambday)*lambdax+tvalue(t;PHModel,16; lambdax, c, mu, lambday)*lambdax+tvalue(t;PHModel,11; lambdax, c, mu, lambday)*lambdax+tvalue(t;PHModel,6; lambdax, c, mu, lambday)*(lambdax+lambday)+tvalue(t;PHModel,1; lambdax, c, mu, lambday)*lambdax+tvalue(t;PHModel,7; lambdax, c, mu, lambday)*lambday+tvalue(t;PHModel,8; lambdax, c, mu, lambday)*lambday+tvalue(t;PHModel,9; lambdax, c, mu, lambday)*lambday+tvalue(t;PHModel,5; lambdax, c, mu, lambday)*lambday)
func h(t) (Pdf(t)/(1-tvalue(t;PHModel; lambdax, c, mu, lambday)))
var MTTF mean(PHModel, 0; lambdax, c, mu, lambday)
expr MTTF
loop t,0, 20, step
expr Reliability(t)
expr CDF(t)
expr pdf(t)
expr h(t)
end
*Compute Performability indices*
echo expected reward rate: E[X(t)]
echo expected cumulated reward: E[Y(t)]
echo distribution of cumulated reward:  P[Y(inf)<=r]
loop t,0,20,0.01
 expr exrt(t;PHModel;lambdax, c, mu, lambday)
 expr cexrt(t;PHModel;lambdax, c, mu, lambday)
end
loop r,0,50,0.01
expr rvalue(r;PHModel;lambdax, c, mu, lambday)
end
end
```

Figure 19

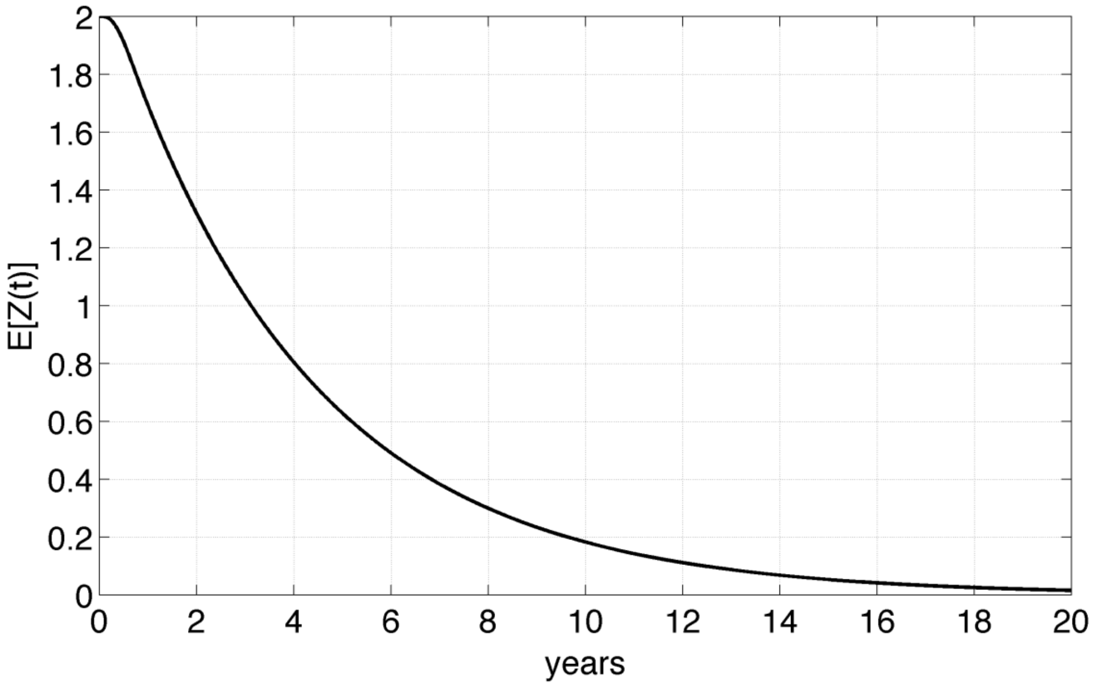


Figure 20

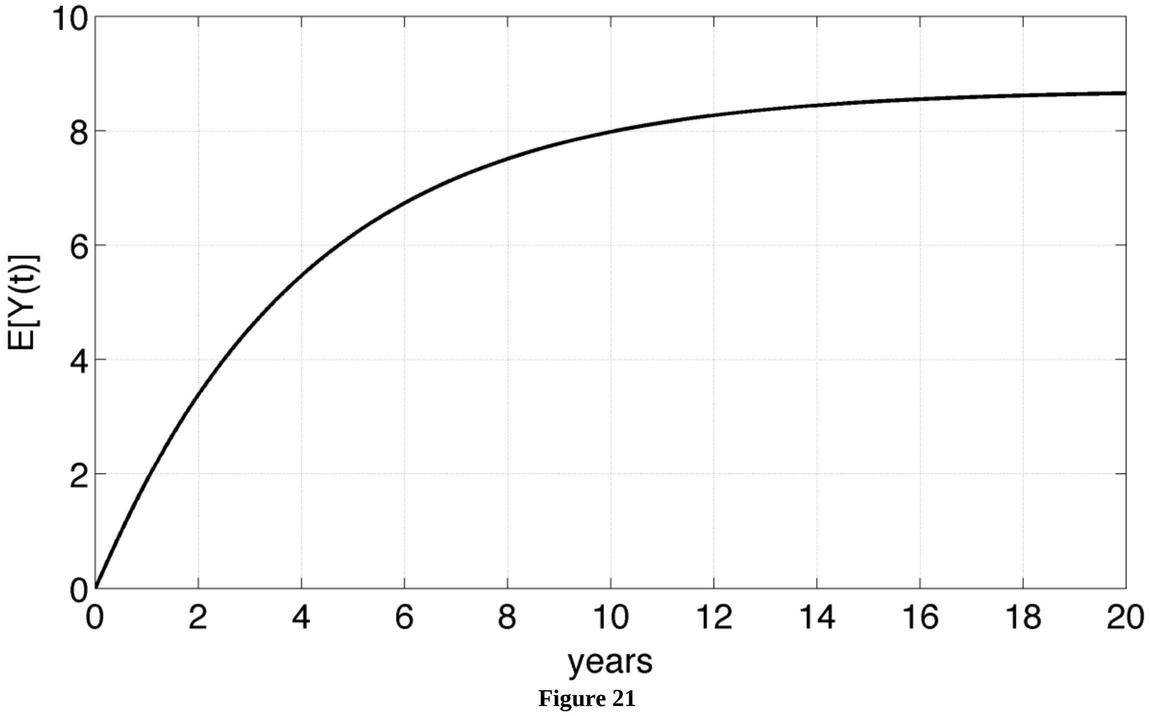


Figure 21

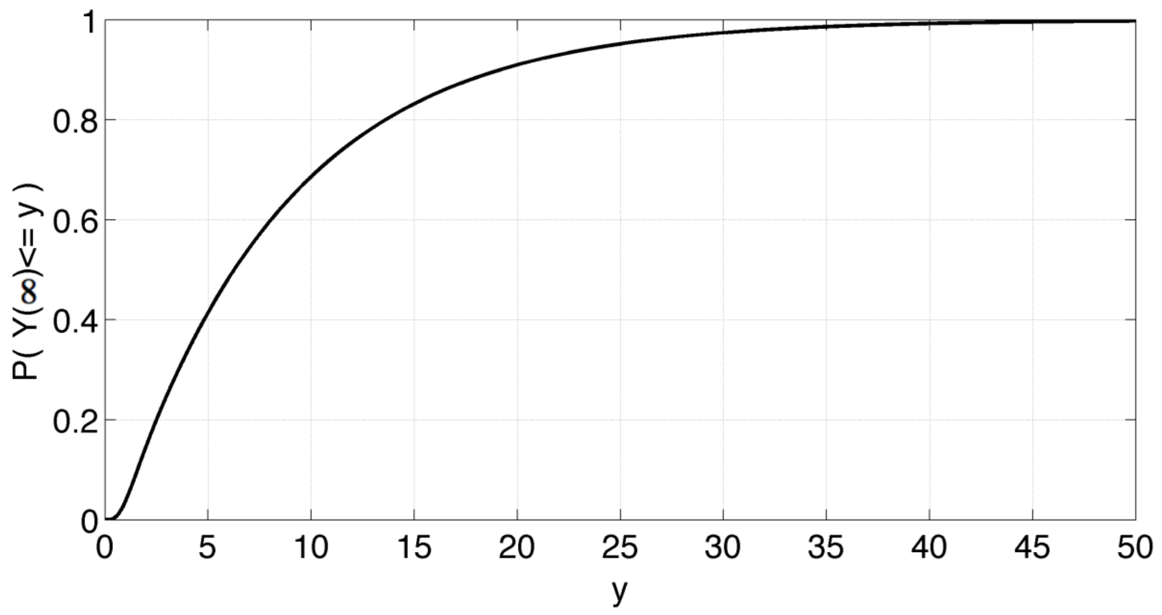


Figure 22

Mean Time To Failure. The MTTF, or Mean Time To Absorption, is using the *mean* function (see line 171 in Figure 19): $MTTF = 4.39$ years.

Expected Instantaneous Reward Rate. This rate is computed by means of the *exrt()* function (see line 184 in Figure 19). Figure 20 shows the trend of the expected reward rate at time *t*.

Expected Accumulated Reward. It is evaluated using the function *cexrt()* (see line 185 in Figure 19) and plotted in Figure 21.

The ***Expected Accumulated Reward till Absorption*** is: $E[Y(\infty)] = 8.66$.

Distribution of the Accumulated Reward till Absorption. The behavior of the distribution is shown in Figure 22. The distribution of the accumulated reward until absorption can be computed numerically using the SHARPE built-in function *rvalue()* (see line 188 in Figure 19).

SIMULATION

The simulation is an alternative to analytic-numeric solution of models, especially for large and complex models [23]. Although, it is possible to simulate the Markov model itself, we propose to compute reliability and performability indices of the considered system using Stochastic Reward Nets (SRN) [24] and to perform simulation by means of SPNP (Stochastic Petri Net Package) [25]. To model the *two processors system* with an SRN, the behavior of each processor is considered separately. Furthermore, the maximal repair policy (*as good as new*) is adopted. Figure 23 presents the SRN proposed to model the above described system. Let us consider the case of processor P_1. At the beginning it is properly working (place *p1w*). A failure can occur (transition *tp1*) leading the processor in a vanishing state (place *p1t*). If the failure is covered (transition *cp1*), the processor is to be repaired (place *p1r*). After repair (transition *rp1*), it works as good as when it was new (place *p1w*). If the failure is not covered (transition *up1*), the processor fails (place *fail*). The behavior of processor P_2 is the same.

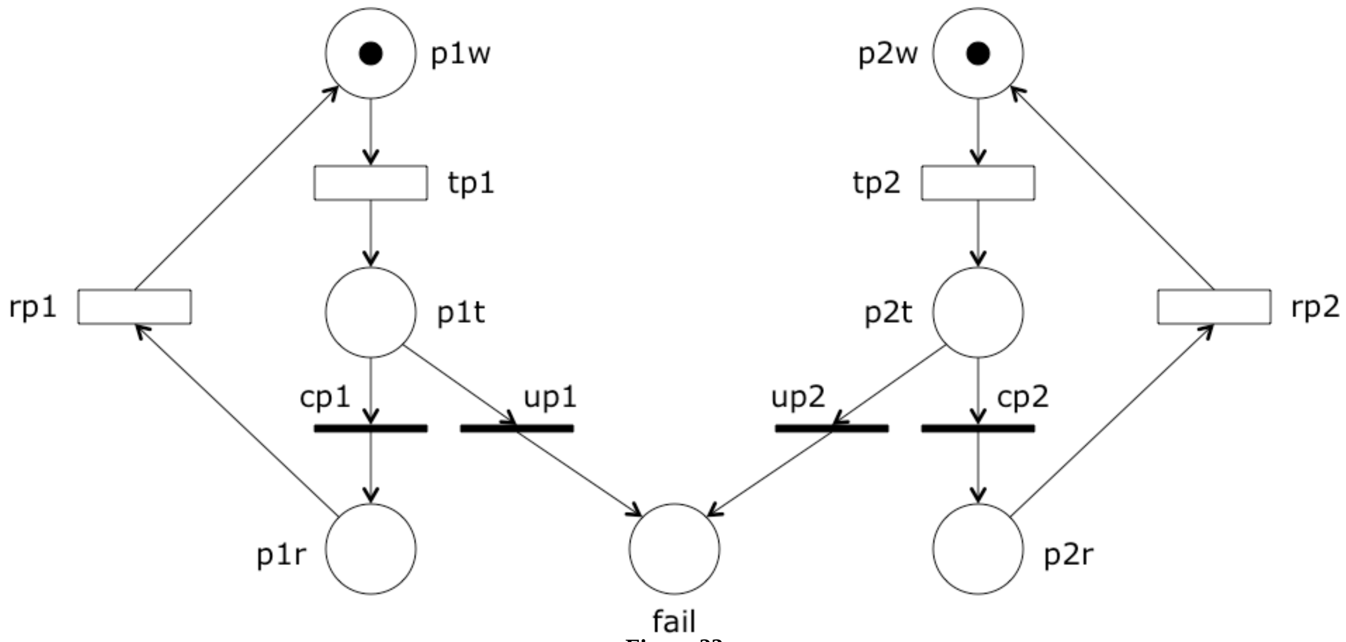


Figure 23

Table 2 shows rates and weights of the transitions.

Transition	Rate/Weight
tp1, tp2	Weibull(α, β)
rp1, rp2	μ
cp1, cp2	c
up1, up2	1-c

Table 2

Figure 24 presents the code to compute reliability and expected instantaneous reward rate in the SPNP software. The maximum number of simulation runs is 100000 and the required confidence interval is 95% (see lines 22 and 24 of the code, respectively). Reward functions *rel()* and *rr()* (lines 111 and 118 in Figure 24, respectively) consider the system up if at least one processor is properly working (places *p1w* and *p2w*) and no processor is failed (place *fail*). The reliability, the expected instantaneous reward rate, and the expected accumulated reward are presented in Figures 25, 26, and 27, respectively.

```
#include <stdio.h>
#include "user.h"

double mu = 120;
double c = 0.9;

double a = 2.1;
double b = 1/1.02;

double rel();
double rr();

/* OPTIONS */

void options() {

  iopt(IOP_SIMULATION,VAL_YES);
  iopt(IOP_SIM_RUNMETHOD,VAL_REPL);
  iopt(IOP_SIM_CUMULATIVE,VAL_NO);
  iopt(IOP_SIM_STD_REPORT,VAL_YES);
  iopt(IOP_SIM_SEED,345983453);
  iopt(IOP_SIM_RUNS,100000);
  fopt(FOP_SIM_ERROR,0.0001);
  fopt(FOP_SIM_CONFIDENCE,.95);
  fopt(FOP_SIM_LENGTH, 20.0);

}

/* DEFINITION OF THE NET */

void net() {

  /*  PLACES  */

  place("p1w");
  place("p1t");
  place("p1r");

  place("p2w");
  place("p2t");
  place("p2r");

  place("fail");

  init("p1w",1);
  init("p2w",1);

  /*  TRANSITIONS  */

  weibval("tp1",b,a);
  imm("cp1");
  imm("up1");
  probval("cp1",c);
  probval("up1",(1-c));
  rateval("rp1",mu);

  weibval("tp2",b,a);
  imm("cp2");
  imm("up2");
  probval("cp2",c);
  probval("up2",(1-c));
  rateval("rp2",mu);

  /*  ARCS */

  /* Input Arcs */
  iarc("tp1","p1w");
  iarc("cp1","p1t");
  iarc("up1","p1t");
  iarc("rp1","p1r");

  iarc("tp2","p2w");
  iarc("cp2","p2t");
  iarc("up2","p2t");
  iarc("rp2","p2r");

  /* Output Arcs */
  oarc("tp1","p1t");
  oarc("cp1","p1r");
  oarc("up1","fail");
  oarc("rp1","p1w");

  oarc("tp2","p2t");
  oarc("cp2","p2r");
  oarc("up2","fail");
  oarc("rp2","p2w");

}


/* FUNCTIONS */

int assert() {}


void ac_init() {
/* Information on the net structure */
pr_net_info();
}


void ac_reach() {}


double rel() {
  if( (mark("p1w")==1 ||
mark("p2w")==1) && (mark("fail")==0) )
    return(1.0);
  else
    return(0.0);
}

double rr() {
  if( (mark("p1w")==1 ||
mark("p2w")==1) && (mark("fail")==0) )
    return(mark("p1w")+mark("p2w"));
  else
    return(0.0);
}


/* OUTPUT */

void ac_final() {
  pr_expected("Reliability", rel);
  pr_expected("Reward", rr);
  pr_cum_expected("Accumulated", rr);
}
```

Figure 24

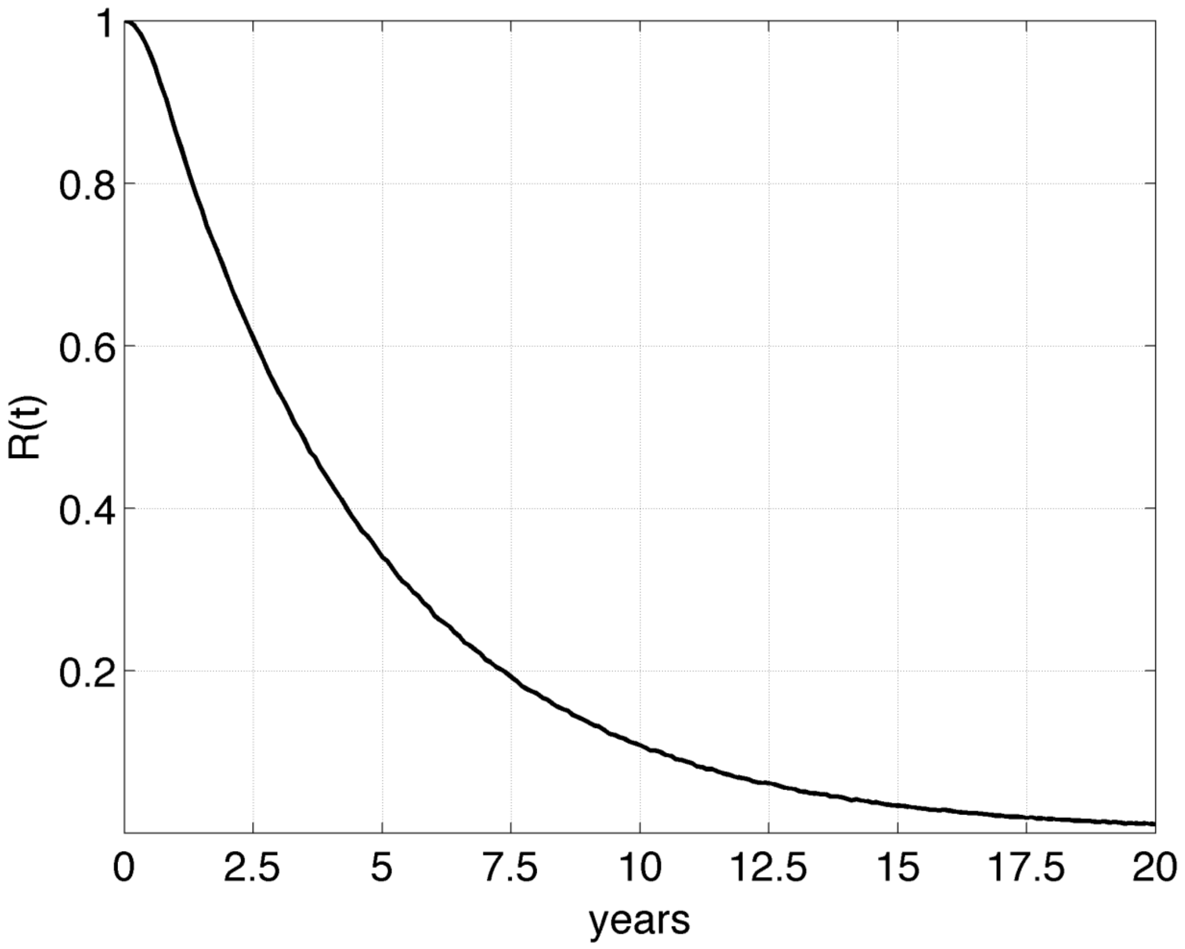


Figure 25

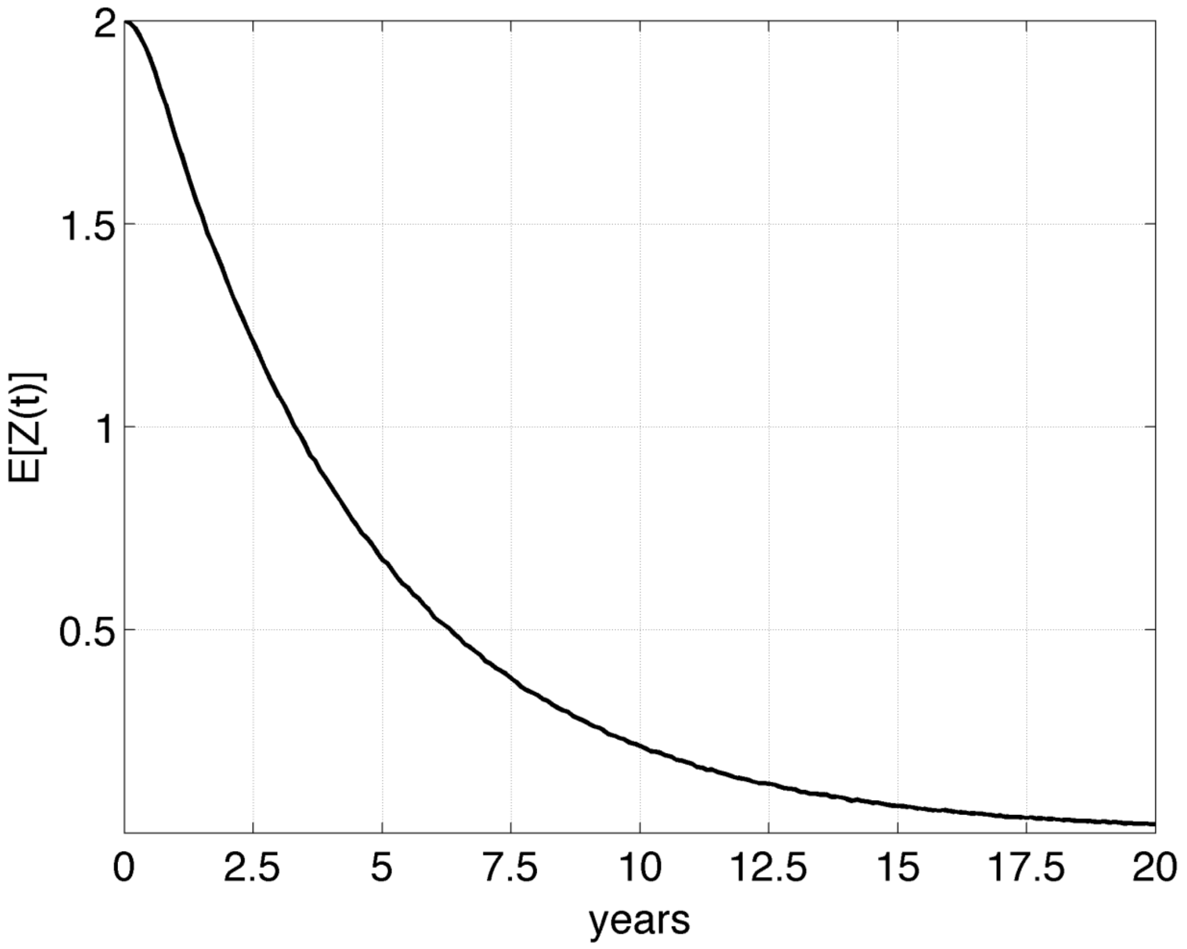


Figure 26

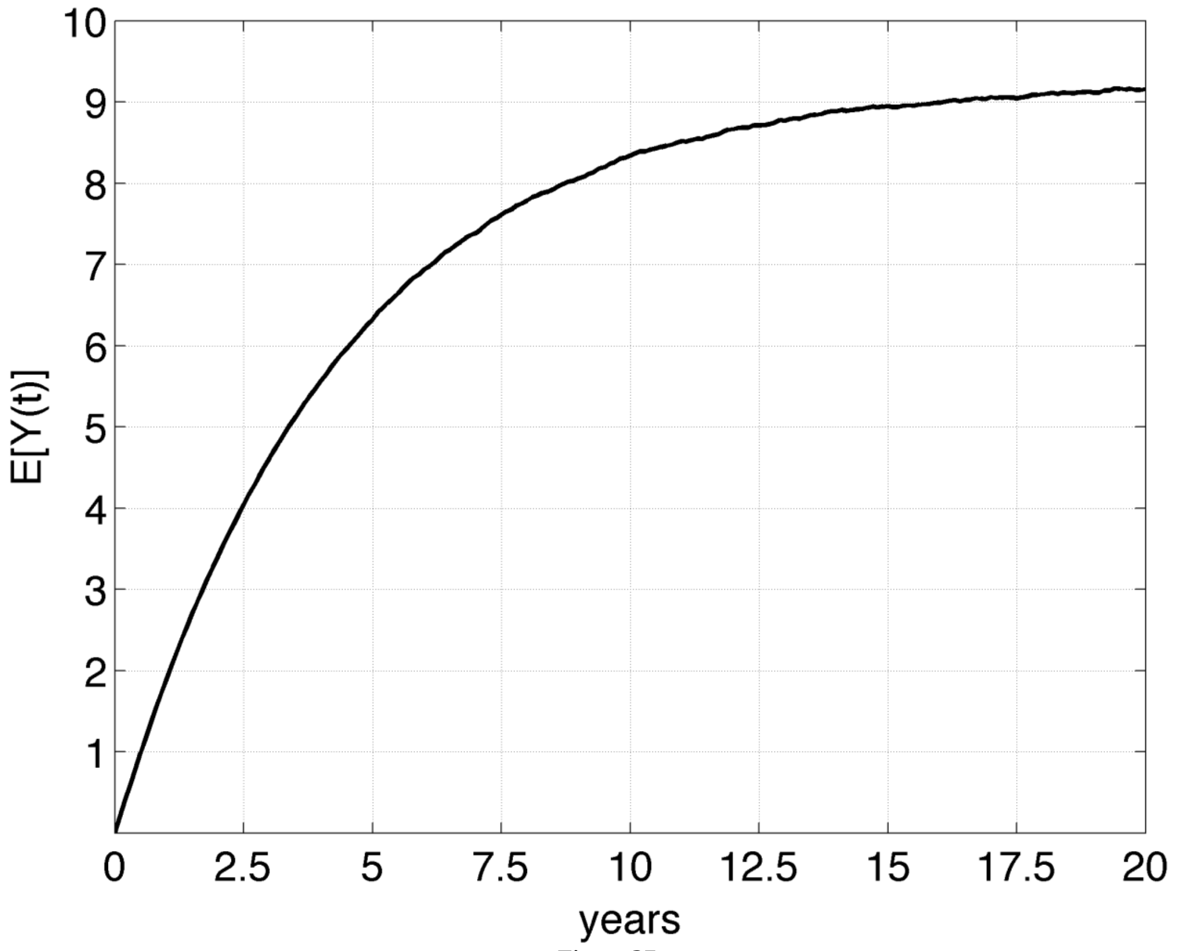


Figure 27

Mean Time To Failure. The MTTF, evaluated as Mean Time to Absorption, is 4.62 years.

The ***Expected Accumulated Reward till Absorption*** is: $E[Y(\infty)] = 9.15$.

COMPARISON

The importance of the recovery strategy is evident considering the reliability computed with PCA and PH. As shown in Figure 28 the reliability of the system with the maximal repair strategy, computed with PH expansion, is much greater than the one with minimal repair strategy, computed with the PCA. With minimal repair the reliability becomes 0 after about 4.5 years while after the same time reliability obtained with maximal repair is greater than 0.35. As a matter of fact, also the MTTF is much different. Maximal repair provides an average lifetime more than two times the one achieved when using an *as bad as before* repair strategy.

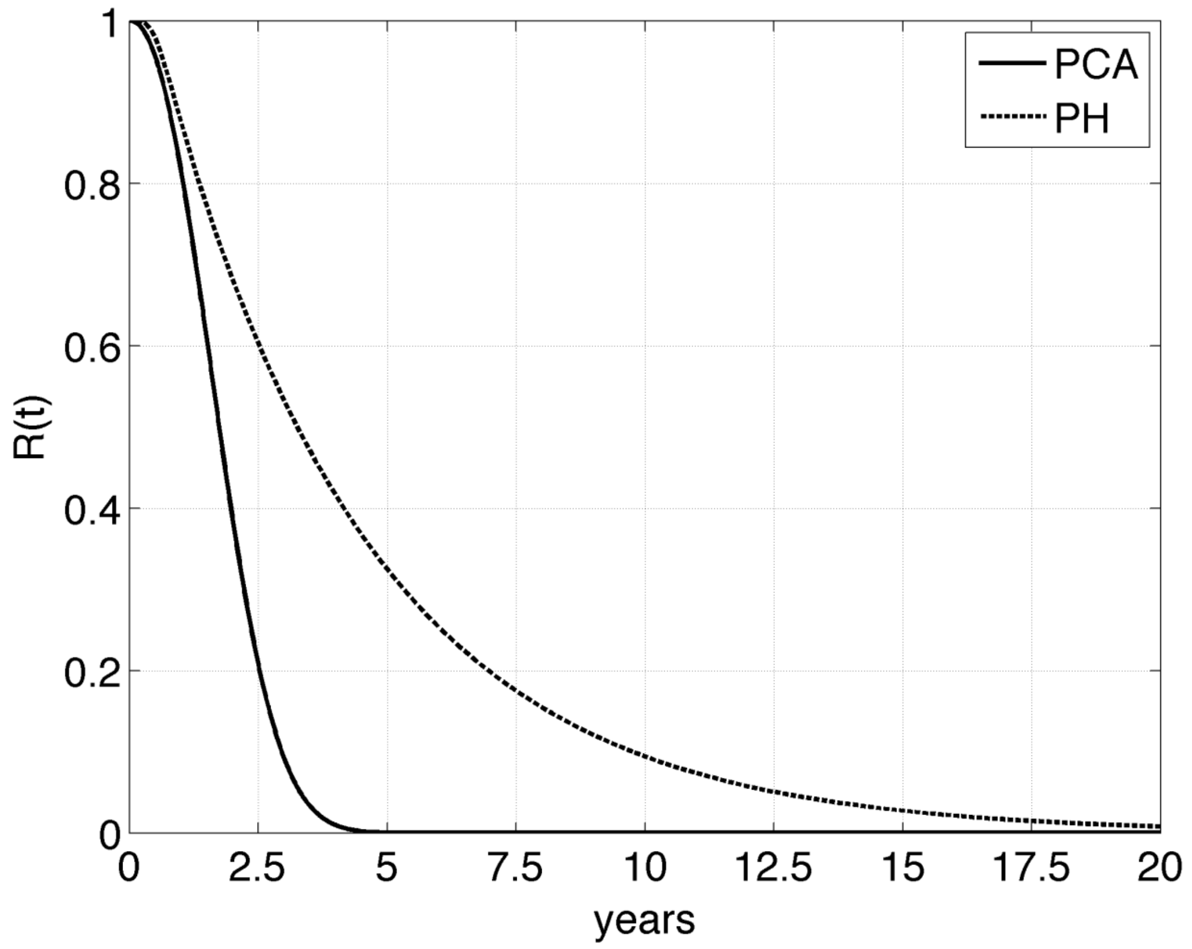


Figure 28

In the cases of PH expansion and simulation the same recovery strategy is adopted. Figure 29 compares the reliability computed with the two techniques. Results obtained with PH expansion fall into the corresponding confidence intervals estimated with simulation when $t \leq 2.5$. Nevertheless, the trends of the reliability are very similar. Such a result is also confirmed by the mean times to failure.

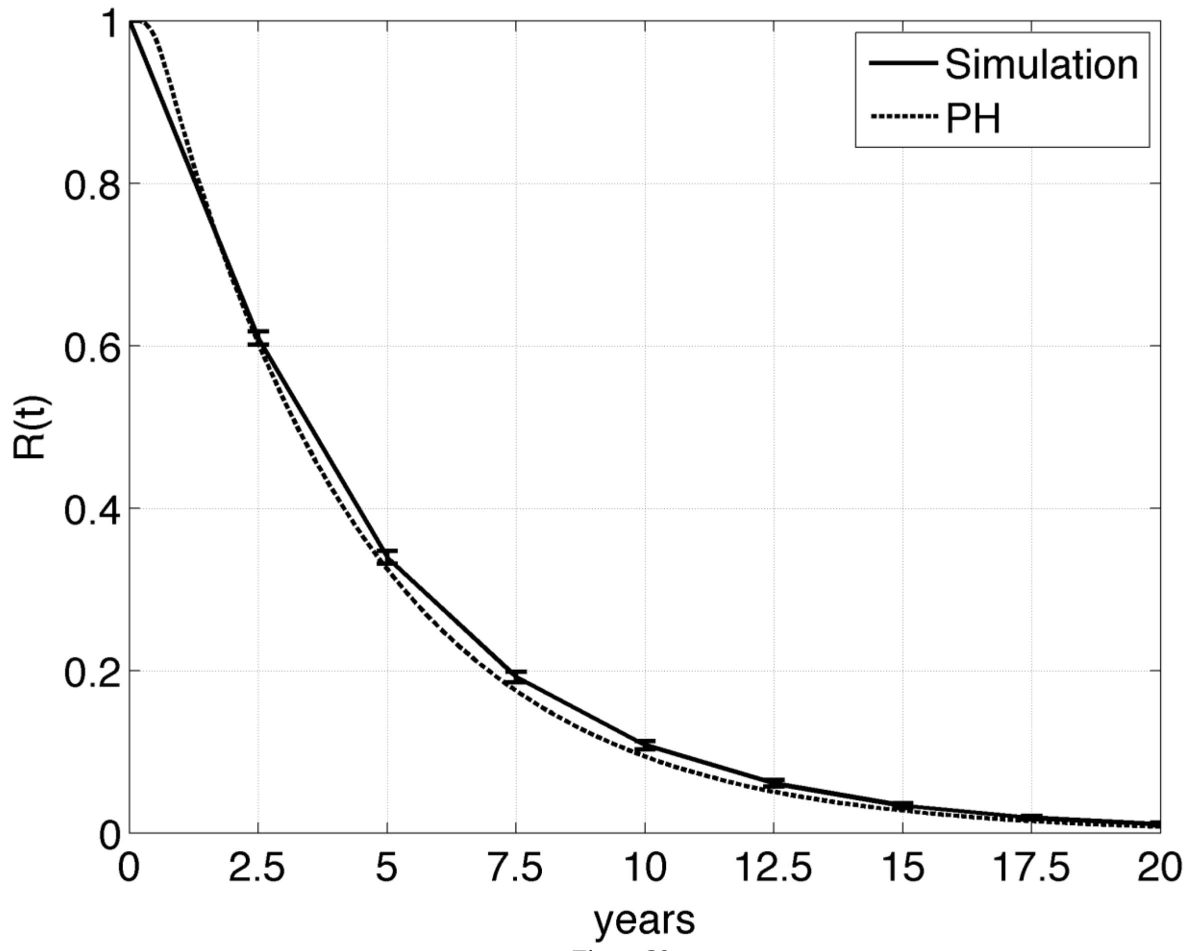


Figure 29

REFERENCES

[1] International Electrotechnical Commission. IEC 61508: Functional safety of electrical/electronic/ programmable electronic safety-related systems. Ed. Peter Kim Wise Balance.

[2] RTCA SC-167, EUROCAE WG-12. DO-178B, Software Considerations in Airborne Systems and Equipment Certification.

[3] Trivedi KS, Andrade, EC, Machida, F. Combining Performance and Availability Analysis in Practice. Advances in Computers. 2012. 84: 1-38.

[4] Heath, T, Martin, RP, Nguyen, TD. Improving cluster availability using workstation validation. SIGMETRICS Performance Evaluation. 2002. 30(1): 217–227.

[5] Sahoo, RK, Sivasubramaniam, A, Squillante, MS, Zhang, Y. Failure data analysis of a large-scale heterogeneous server environment. Proceedings of International Conference on Dependable Systems and Networks. 2004. 772-781.

[6] Arnold, TF. The Concept of Coverage and Its Effect on the Reliability Model of a Repairable System. IEEE Transactions on Computers. 1973. C-22(3): 251- 254.

[7] Sahner, RA, Trivedi, KS, Puliafito, A. Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package. 1996. Kluwer Academic Publishers.

[8] Trivedi, KS. Probability and Statistics with Reliability, Queuing, and Computer Science Applications. 2001. John Wiley and Sons, New York.

[9] Beaudry, MD. Performance-Related Reliability Measures for Computing Systems. IEEE Transactions on Computers. 1978. C-27(6): 540-547.

[10] Ciardo, G, Marie, RA, Sericola, B, Trivedi, KS. Performability analysis using semi-Markov reward processes. IEEE Transactions on Computers. 1990. 39(10): 1251-1264.

[11] Malhorta, M, Reibman, A. Selecting and Implementing Phase Approximations for Semi-Markov Models. Stochastic Models. 1993. 9(4): 473-506.

[12] Rindos, A, Woolet, S, Viniotis, I, Trivedi, KS. Exact Methods for the Transient Analysis of Nonhomogeneous Continuous-Time Markov Chains. 2nd International Workshop on the Numerical Solution of Markov Chains. 1995. W. J. Stewart (ed.), Kluwer Academic Publishers.

[13] Fortmann, TE, Hitz, KL. An Introduction to Linear Control Systems, in Control and systems theory - A Series of Monographs and Textbooks. 1977. 5: 589-599. New York, Marcel Dekker.

[14] Takacs L. Introduction to the theory of Queues. 1962. New York: Oxford University Press.

[15] Neuts, MF, Meier, KS. On the use of phase type distributions in reliability modeling of systems with two components. O.R. Spectrum. 1980. 2(4): 227-234.

[16] Ascher, HE. Evaluation of Repairable System Reliability Using the “Bad-As-Old" Concept. IEEE Transactions on Reliability. 1968. R-17(2): 103-110.

[17] Trivedi, KS, Sahner, R. SHARPE at the Age of Twentytwo. ACM Sigmetrics Performance Evaluation Review. 2009. 36(4): 52–57.

[18] Bao, Y, Sun, X, Trivedi, KS. A workload-based analysis of software aging, and rejuvenation. IEEE Transactions on Reliability. 2005. 54(3): 541- 548.

[19] Johnson, M, Taaffe, M. Matching moments to phase distributions: nonlinear programming approaches. Stochastic Models. 1990. 6: 259–281.

[20] Neuts, MF. Matrix Geometric Solutions in Stochastic Models. 1981. Johns Hopkins University Press, Baltimore, MD.

[21] Bobbio, A, Horvath, A, Telek, M. The scale factor: a new degree of freedom in phase-type approximation. Performance Evaluation. 2004. 56(1-4): 121-144.

[22] Singh, C, Billinton, R, Lee, SY. The Method of Stages for Non-Markov Models. IEEE Transactions on Reliability. 1977. R-26(2): 135-137.

[23] Tuffin, B, Choudhary, PK, Hirel, C, Trivedi, KS. Simulation Versus Analytic-Numeric Methods: Illustrative Examples. Proceedings of the 2nd international conference on Performance evaluation methodologies and tools. 2007. 63.1-63.10.

[24] Ciardo, G, Blakemore, A, Chimento, PF, Muppala, JK, Trivedi, KS. Automated Generation and Analysis of Markov Reward Models Using Stochastic Reward Nets. In Meyer, C, Plemmons, RJ. Linear Algebra, Markov Chains, and Queueing Models - IMA Volumes in Mathematics and its Applications. 1992. Springer-Verlag, Heidelberg, Germany.

[25] Ciardo, G, Muppala, J, Trivedi, KS. SPNP: Stochastic Petri Net Package. Int. Workshop on Petri Nets and Performance Models. Proceedings of the Third International Workshop on Petri Nets and Performance Models (PNPM89). 1989. 96: 142-151.