A DEPENDABILITY MEASURE FOR DEGRADABLE COMPUTING SYSTEMS

Erland Jonsson

Department of Computer Engineering, Chalmers University of Technology, Sweden

Søren Asmussen

Institute of Electronic Systems, Aalborg University, Denmark

Abstract

This paper deals with the problem of finding a comprehensive dependability measure or figure of merit for computing systems. Dependability is a term used for a general description of a systems trustworthiness in non-quantitative terms. It is commonly described by a number of aspects, like reliability, availability, safety and security. Quantitative measures are conveniently used for e.g. reliability and availability, but are rare for security.

However, it is felt that a more general measure of a system's dependability would be of great interest and could be used for system evaluations, design trade-offs etc. In order to achieve this, we adopt a generalized view that facilitates a recompilation of the dependability aspects into fewer and more general qualities. Key issues for the generalization are the concepts of *degradability* and *service*. A degraded service is the result of the discontinuation of one or several *subservices*, yielding a system that operates on a reduced *service level*.

A vectorized measure based on Markov processes is suggested, and mathematical definitions are given. The measure describes the expected time a system will be operating at a certain service level, and also the probability that this level be reached. By means of applying the concept of reward rate to each service level, an even more simplified figure of merit can be calculated.

Normally, when making reliability calculations, an assumption of exponential failure rates for system components is made. Sometimes this assumption is not realistic and we outline how phase-type distributions can be used to cope with this situation.

Finally, two different schemes for the calculation of the measure is given. First, a hierarchical procedure feasible for small systems and calculations by hand is presented. Second, a general procedure based on matrix calculus is given. This procedure is suitable for complicated systems. It is also general in the sense that it may be used for measures extended to repairable systems.

CONTENTS

1 Introduction

The ever-increasing complexity and criticality of modern computer systems have led to a growing awareness of the importance that the systems are trustworthy and that the trustworthiness in some way can be demonstrated to the user. The term used for this demonstrated trustworthiness is **dependability** defined as the amount of *reliance* that can be placed on the *service* delivered by the system, a reliance that can also be *justified* to the user [Car79], [Lap85], [Lap90]. This definition is not exact or mathematically well-defined, but rather a heuristic approach intended to give a general feeling for the overall dependability merit of the system.

No general "dependability measure", i.e. a number or a set of numbers, that would reflect the overall degree of demonstrated trustworthiness has been defined. Instead, the dependability concept has tended to mean different things in different types of systems. The numbers used to quantify the merit of the system have reflected the various aspects of dependability, some of which are described in Chapter 2, rather than the overall dependability. See [IEV] and [Hei91].

In many cases, however, it would be valuable for the system designer to be able to make a mathematical calculation of a set of numbers that would describe several dependability aspects of the system he is designing. This would help him to make the correct trade-offs and comparisons between different design alternatives. In this paper a vectorized performance measure based on continuous-time Markov processes is suggested. The numbers in the vector describe the performance of the system when it is operating on different *service levels*.

A non-degradable system exhibits two service levels as perceived by the user. It delivers its service according to the specification, i.e. it is fully operational and working in a non-failed fashion, or the delivered service deviates from that specified, in which case a system failure has occured. In most practical systems, however, the overall service delivered is in fact composed of a number of subservices representing different functions. Each subservice may fail separately, leading to a step-wise reduction of the overall system service. We are then dealing with a degradable system with three or more service levels. It is thus possible to merge aspects such as *reliability* and *performability* into one single quality, provided certain system limitations are accepted. If we introduce two or more service levels for a failed system, the concept of *safety* can also be reflected.

Thus, the proposed measure describes the expected time a system will perform on a certain service level, and also the probability that this level is reached. For simple systems, the probability part of the measure is trivial and can be ignored. If we apply the concept of reward rate to each service level an even more simplified figure of merit can be calculated.

A fact that is not considered is that systems can very often be repaired and upgraded. However, the measure can be applied to repairable systems for each specific operational period.

Another feature of the proposed measure is that it is *application oriented*. The described method provides a general tool or framework, and each user will have to decide exactly how this tool is to be applied to his system. One of the reasons for this is that the service that a system provides is not only a function of the performance of the system, but also depends on how it is used. A degraded service delivered by a system may very well be regarded as a full service in a certain application or by a certain user.

2 Different aspects of dependability

The dependability of non-repairable systems can be described by a number of aspects, like *reliability, performability, safety* and *security.* There exists a generally accepted and well-established definition of reliability. For some of the other aspects, there are a number of different interpretations of the meaning of the terms and their interrelations exist. In the discussion below the most common interpretation for each aspect is presented. The intention of this paper is then to merge those aspects, possibly into a modified form, to form a more general concept on which the proposed dependability measure can be based.

2.1 Reliability

Reliability is a mathematically well-defined time-dependent function $R(t)$. This function describes the probability that the item, whether a component, subsystem or system, will remain operational until some time t. In other words, the reliability function gives, for a non-repairable system, an idea of the expected lifetime, called the MTTF or Mean Time To Failure. The mathematical relation between reliability and Mean Time To Failure is MTTF= $\int_0^\infty R(t)\,dt$.

According to the recommendations in the International Electrotechnical Commission [IEV], reliability is defined as:

- Reliability - the characteristic of an item expressed by the probability that it will perform a required function under stated conditions for a stated period of time.

2.2 Performability

According to [Joh89], the term performability is used for systems that are able to continue to deliver a service after the occurence of a failure. The service will then be delivered at a reduced performance level, specified by a pre-defined subset of the full specification. This property of a system is desirable, since it entails a possible gradual reduction of the system function, at the occurence of some failures.

The performability $P(t, L)$ can be defined as a function of time t and performance level L, denoting the probability that the system is performing at or above the specified performance level at time t [For84].

The ability of a system to make a *graceful degradation* is closely related to performability, meaning that the system can automatically reduce its performance level if a failure occurs.

Other authors have developed mathematical models for performability and suggested various performance measures. See [Mey80], [For84], [Smi87], [Smi88] and [San89]. In all those cases the performability concept has been used to model reduced performance levels of one single service delivered by the computer system.

A different approach to the problem of modelling a gradual reduction of system performance is taken by [Agg89]. He recognizes that the failure of a component in a non fault-tolerant system will not always lead to a system failure. One reason for this is that the notion of system failure is a matter of the expectations of the system user. The same performance reduction of a system may be considered a failure by one user, while another may consider it a non-failure. Furthermore, the judgement may depend on how the system is actually used. In one situation a certain reduction of performance will be perceived as a failure, whereas in other situations it may not.

The author quotes examples like the failure of one loudspeaker in a stereo receiver, which for the classical music lover would certainly be seen as failure, but perhaps not for a person using the equipment only for language courses. Failed headlights on an automobile is not a system failure during day-time but most certainly during the night.

The method proposed to cope with this situation is to attach a measure to each component in the system, a measure that would represent the probability that a component failure would also lead to a system failure. This measure is called *criticality.* The method leads to a probabilistic notion of system failure depending on the criticality of the components in the system, but system failure is still a binary variable giving a simple failure/success view of the system rather than a degraded approach.

2.3 Safety

When considering the safety of a system we have tacitly divided the possible failures of the system into two groups: one group with catastrophic failures and one group with other failures [Joh89, Lap90]. Catastrophic failures are failures that will lead to harm that is orders of magnitude bigger than the cost of the system itself, or that will endanger human lives. The other failures are sometimes called benign. With this definition, safety is a time-dependent function $S(t)$ giving the probability that the system, until some time t, will either be operational or exhibit a benign failure.

2.4 Security

Security is probably the aspect that is least well integrated into the dependability concept. One reason for this is that security is a concept that in itself is composed of several different aspects. According to [Pfl89], computer security consists of maintaining the three characteristics *secrecy, integrity* and *availability.*
Secrecy, or data confidentiality, deals with the unauthorized disclosure of information. Integrity means that information or other assets of the system must only be changed, deleted, created a.s.o. by authorized parties. Availability means that system assets must indeed be available to the proper system user. Sometimes an aspect called *data preservation* is added. It deals with the protection against loss of information or information media. Data preservation is related to availability.

Much of the work in the security area has been related to intentional and malicious system interaction with the purpose of causing damage or gaining personal benefits at the expense of normal system use. However it should be noted that security breaches could just as well occur as a result of accidental handling or normal component failures.

The standpoint taken here is that the availability and data preservation characteristics are applicable to the discussion pursued in this paper, since they deal with aspects that are related to a disrupture of the service to the authorized user of the system, a disrupture that is normally described as a failure or degradation.

3 Considerations on dependability assessment

3.1 Introduction

The methods used for the reliability assessment for computerized and other systems are well-known and straightforward and have been used for many years. These methods are based upon certain assumptions pertaining to the behaviour of the system in its environment, the nature and behaviour of faults and failures, and also to the conception of the service delivered by the system. In many cases these assumptions present unnecessary restrictions on the reliability modelling. Also, the validity of some assumptions has been weakening as systems have become more complicated and critical.

Some considerations that have to be made when attempting to generalize the reliability concept are presented below. A generalized approach would include not only the reliability aspect of dependability, but cover several other dependability aspects. This paper suggests a measure for the aggregation of a number of dependability aspects. We have chosen to use the term *dependability measure* for this, even if the measure does not cover all dependability aspects, but only a subset, and in spite of the fact that the [IEV] recommendations state that dependability is only used for general descriptions in non-quantitative terms.

3.2 Faults

In our attempt towards generalization, the dependability (or rather undependability), "input" to the system has to be considered. The input is manifested as those phenomena that may reduce dependability, i.e. faults. A fault is the phenomenological cause of an error. An error is defined as an undesirable system state. An error may or may not lead to a system failure and thereby reduce the system's dependability [Lap85]. All types of faults that may occur to the system have to be identified, classified and evaluated.

Faults include not only component failures, but also environmental effects, deliberate system interaction with the intention to create system failures, design faults, handling faults and others. Several classifications of faults exist, see e.g. [Lap90] and [Joh89]. Some of these fault types represent pure safety and security aspects of dependability and must therefore be incorporated.

Normally reliability calculations are based on the assumption that the statistical occurrence of faults can be represented by an exponential distribution. This is approximately true for "component-type" faults, but not for many other types like design faults. An improved fault model would require knowledge of the real distribution for all types of faults and a practical method to model the distributions.

3.3 System performance and degradability

An item of particular interest is the modelling of the service delivered by a system. In most cases the service does not have a binary characteristic such as presence or absence, but is rather something that can be reduced gradually or *degraded.* As mentioned in Section 2.2 performability is the concept used to describe systems of this kind. The system has then a quality called *degradability.* The degradation will reduce system performance to different performance levels. It should be noted, though, that performance can fall into two categories. Most of the papers presented on this topic deal with performance analysis of systems that deliver only one single service, like computation capacity. See [For84], [Smi87], [Smi88] and [San89] among others. For those systems a reduction of the performance level is equivalent to a continued delivery of the same service, but at a reduced amount. This is a *quantitative performance reduction.*

However, most systems do not deliver one service only, but a number of different **subservices**. A disruption of one or several subservices will represent a system offering a functionally degraded service. Thus the system has a quality called functional degradability and the service has undergone a *functional performance reduction.*

A performance reduction, be it quantitative or functional, is due to a failure of a system component or subsystem. From the system's point of view, a proper term for this performance reduction is **system degradation**, since the failure of one component and the corresponding discontinuation of a subservice or reduction of service will in general only lead to a degraded system and not a failed system. A *failure* is then a special case of degradation, leading to a failed system.

4 A dependability measure for degradable systems

4.1 Service levels

In the following a quantitative dependability measure is proposed, covering all or most of the dependability aspects described in Chapter 2. The measure is based on a concept called service level.

A **service level** is defined as a group of system states, each with a user-specified degree of performance or functional accomplishment. The service levels are dependent on the design and layout of the system as well as on the application of the system, i.e. how the system is used. Therefore, the service levels are said to be *application-related*.

The highest service level is denoted **service level 0 (SL0)** or **full service level**. This level must include the system state that describes the complete fulfillment of all the requirements in the specification, the fully **operational state**.

In the simplest case there is only one more service level, the **failed service level**, corresponding to the system **failed state**, when no service is delivered or the service delivered is of no use to the user. In this case the system has only one operational state with a specified degree of performance and the sole alternative is the failed state. The expected time the system is in the operational state before making a transition into the failed state is a measure of the reliability of the system.

Most systems, however, do not deliver one service only, but a number of different subservices. In many cases the system is useful even if it does not deliver all subservices. A system that can operate with a reduced number of subservices is called **degradable**. This makes it convenient to define more than one operational service level. The highest service level means full performance or delivery of all subservices. Lower service levels mean reduced performance or delivery of a pre-defined subset of the subservices. Operational service levels with reduced performance are called **degraded service levels**. They provide a means of incorporating the performability aspect of dependability into the concept.

The degraded service levels must be hierarchically ordered below the highest service level. A disruption of one or several subservices leads to a performance reduction that may cause a transition from a higher service level to a lower service level and thereby represent a system offering a further degraded service.

The model does not incorporate transitions from lower levels to higher levels, something that could be used to model maintenance of the system. However, there is no reason why this extension should not be possible to make, something that would pave the way for inclusion of the availability aspect of dependability.

In safety-critical systems some failures are more disastrous than others, implying danger of life or considerable material losses. These failures are called catastrophic or malign, as opposed to the "normal" benign failures. It is convenient to introduce different failed service levels for these two types of failures. This is another type of service level, describing the service, or rather lack of service, when the system is non-operational. The failed service levels are hierarchically ordered below the operational service levels. The service level related to a catastrophic failure is the lowest one and the service level related to a benign failure is the next level up. Above these two is found the lowest operational service level. In this way the safety aspect of dependability is modeled.

Since both failed service levels represent failed states in which the system has ceased to function no transitions between them are possible. Only transitions from some operational service level to either one of the failed service levels is possible.

It is possible to introduce more than two failed service levels, if the user finds that this could be of benefit for the evaluation of the considered system.

4.2 Computational procedure

This section presents the overall procedure for the derivation of the dependability measure.

The first step includes a definition of the service levels that will properly reflect the user's view of the system operation in its environment. A specification review identifying those functions or subservices that belong to each level is necessary. The need to include more than one failed service level should be considered.

As a second step a construction of a dependability block diagram should be made for each operational service level. This implies a decomposition of the system into blocks that are adapted to reliability calculations. The block diagram ought to include hardware as well as software. It should also take into account supporting and auxiliary systems to the extent that they may affect the overall dependability. Conductors of different types, like communication links and power supply lines, are to be regarded as separate blocks.

In the third step failure rates for all of the blocks in the dependability block diagram should be calculated by compiling the influence of the various faults, and then integrating them for each service level. In general the failure rates for each level are not constant but functions of time, which introduces a further complication to the procedure. This problem could be handled by means of applying the type of analysis described in Section 4.6. However, if constant, or approximately constant failure rates can be used the subsequent calculations would be considerably facilitated.

These failure rates will serve as input for the derivation of the dependability measure. The exact definition of the measure will be made in the next section.

4.3 Definition of a dependability measure

This section provides a mathematical definition of the vectorized dependability measure. It also presents the method and equations to be used for the calculations. The method is based on a pre-defined set of service levels and a set of corresponding failure rates which quantify the rate of transitions between levels.

We shall assume that the state of the system can be modelled as a continuous time Markov process $\{X_t\}_{t\geq 0}$ with a finite state space E, in which each service level, SLn, can be identified with a subset of states in E. Thus, E is the disjoint union $SL0 + \cdots + SL\ell$, where ℓ is the number of service levels. Further, we imagine that service levels $0, \ldots, k$ (say) correspond to operational states O, i.e. the states in which the system functions, in the sense that it is delivers a full or degraded service to the user. Service levels $k+1, \ldots, \ell$ correspond to the *failed* states F, i.e. states in which the system is not functioning, meaning that it is not delivering any service of interest to the user. That is[1],

$$\begin{aligned} E &= O + F \qquad \text{where} \\ O &= SL0 + \cdots + SLk, \end{aligned}$$

[1]here as usual "+" means union of disjoint sets

$$F \;=\; SL(k+1)+\cdots+SL\ell.$$

In the simplest case, corresponding to the traditional operational-failed model, O consists of just one single state o and F consists of just one single state f. In more complex situations, the different states in O represent different full or degraded service levels and F represents different types of failed states.

Transitions $i \to j$ have intensity λ_{ij} $(i,j \in E,\ i \neq j)$, and the initial probability $\mathbb{P}(X_0 = i)$ is denoted by π_i. In most situations, the system will always start in a fixed state i_0 so that

$$\pi_j = \begin{cases} 1 & j = i_0 \\ 0 & j \neq i_0 \end{cases} . \tag{1}$$

We shall also assume that the system starts at the highest service level, so that $i_0 \in SL0$. Transitions between operational states represent degradations, and transitions to a failed state represent failures. No transition will ever take place from a failed state, i.e. after entering a failed state, the system stops evolving. Therefore, failed states are absorbing so that $\lambda_{fj} = 0$ for $f \in F$ and all $j \in E$.

For mathematical convenience we shall use the notation that the intensity for leaving state i is $\lambda_i = \sum_{j\neq i} \lambda_{ij}$, and we write $\lambda_{ii} = -\lambda_i$.

Example 1 In many situations, the system state is described by a finite number p of components, each of which may be functioning or failed. Thus a typical state $i \in E$ has the form $i = (b_\alpha), \alpha = 1,\ldots,p$, where the b_α are 0 or 1, $b_\alpha = 1$ indicating that component α is functioning and $b_\alpha = 0$ that it has failed. Note however, that E may be a proper subset of all such 0–1 combinations; for example in a k out of n system, we may collaps all 0–1 combinations with $k+1$ or more zeros into the single state $f = 00\cdots 0$. The Markovian assumption amounts to assuming that each component has an exponential lifetime, with intensity parameter ν_α, say, for the αth, or equivalently a constant failure rate ν_α. (For relaxations of this assumption, see Section 4.6). If the system starts with all components functioning, (1) holds with $i_0 = 11\cdots 1$. A transition from i to j is only possible if j is obtained from i by replacing one of the 1's, say at the αth place by 0, and the intensity is then ν_α. Thus e.g. for a 2 out of 3 system or Triple Modular Redundant (TMR) system,

$$\begin{aligned} \lambda_{111,011} &= \nu_1,\ \ \lambda_{111,101} = \nu_2,\ \ \lambda_{111,110} = \nu_3, \\ \lambda_{011,000} &= \nu_2+\nu_3,\ \ \lambda_{101,000} = \nu_1+\nu_3,\ \ \lambda_{110,000} = \nu_1+\nu_2, \end{aligned}$$

and all other λ_{ij} are zero.

Instead of 0–1 combinations, a common notation is upper– and lower case Roman letters, say A meaning that the first component is functioning and a that it has failed, and we write then ν_A instead of ν_1. Cf. e.g. Example 2 below. □

Assuming that O has n states and F m, we suggest the $2(n+m)$ vector

$$\boldsymbol{v} = \left((u_i)_{i\in O}, (p_i)_{i\in O+F}, (v_i)_{i\in F}\right), \tag{2}$$

as measure of dependability of the system, where u_i is the expected time in state i or Mean Time To Degradation (= MTTD), p_i the probability that the system ever enters i and v_i is the conditional expected time to absorption in $i \in F$, that is, the conditional expected time the system is functioning before being absorbed in i (the MTTF given the system eventually fails in state i). Introducing w_i as the unconditional expected time to absorption in i, we have in formal mathematical terms that

$$\begin{aligned} u_i &= \mathbb{E}\int_0^\infty I(X_t = i)dt, \quad i \in O, &(3)\\ v_i &= \frac{w_i}{p_i}, \quad i \in F, &(4)\\ w_i &= \mathbb{E}\int_0^\infty I(t < \tau_i < \infty)dt, \quad i \in F, &(5)\\ p_i &= \mathbb{P}(\tau_i < \infty), \quad i \in E = O + F, &(6)\end{aligned}$$

where $\tau_i = \inf\{t \geq 0 : X_t = i\}$ ($\tau_i = \infty$ if no t with $X_t = i$ exists) is the hitting time of i, i.e. the time of first entry, and where I is the indicator function (e.g., $I(X_t = i) = 1$ in (3) if $X_t = i$ and $I(X_t = i) = 0$ if $X_t \neq i$). For computational purposes, we note that

$$u_i = \frac{p_i}{\lambda_i}. \tag{7}$$

Also, w_i is usually easier to compute directly than v_i, and thus in the following we often give the formulas for the p_i, w_i only (the values of the u_i, v_i are then trivial to compute from (7), (4)).

4.4 Examples

In this section, we present a few examples in order to illustrate the use of the dependability measure (2). The systems used in the examples are extremely simple, so that the computational procedure shall be as transparent as possible.

Example 2 Consider a washing machine with two functions: wash and spin-dry, A meaning that the machine can wash and a that it cannot, and B meaning that the machine can spin-dry and b that it cannot. Interpreting the failure of the washing function as more serious than that of the spin-drying function, the service levels may be interpreted as

$$\begin{aligned} SL0 &= AB \quad (\text{wash and spin} - \text{dry})\\ SL1 &= Ab \quad (\text{wash only})\\ SL2 &= aB + ab \quad (\text{no wash})\end{aligned}$$

We can identify E by $\{SL0, SL1, SL2\} = \{AB, Ab, aB + ab\}$ where $O = \{SL0, SL1\}$, $F = \{SL2\}$. A state diagram for the system is given in Fig. 1.

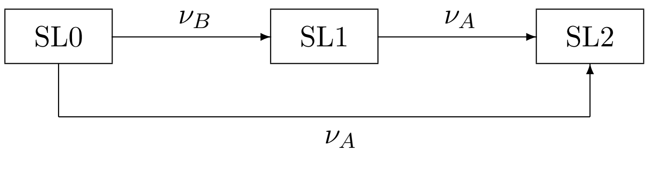


Figure 1

Noting that from $SL0$ we enter $SL1$ or $SL2$ with probabilities $\nu_B/(\nu_A+\nu_B)$, $\nu_A/(\nu_A+\nu_B)$, respectively, we see that

$$\begin{aligned}
p_{SL0} &= 1, \quad u_{SL0} = \frac{1}{\nu_A+\nu_B}, \\
p_{SL1} &= \frac{\nu_B}{\nu_A+\nu_B}, \quad u_{SL1} = p_{SL1}\cdot\frac{1}{\nu_A}, \\
p_{SL2} &= 1, \quad v_{SL2} = w_{SL2} = u_{SL0}+u_{SL1}.
\end{aligned}$$

□

Example 3 Consider a computerized car that has three computers: A, a background functions computer, B, a general purpose computer (ignition etc.) and C, a computer for vehicle dynamics (steering etc.). Interpreting the failure of C as catastrophic but that of A or B (or both) not, the service levels may be interpreted as

$$\begin{aligned}
SL0 &= ABC \quad \text{(full service level)} \\
SL1 &= aBC \quad \text{(degraded service level)} \\
SL2 &= abC + AbC \quad \text{(failed service level)} \\
SL3 &= **c \quad \text{(catastrophically failed service level)}
\end{aligned}$$

We can identify E with $\{SL0, SL1, SL2, SL3\}$ where $O=\{SL0, SL1\}$, $F=\{SL2, SL3\}$. A state diagram for the system is given in Fig. 2.

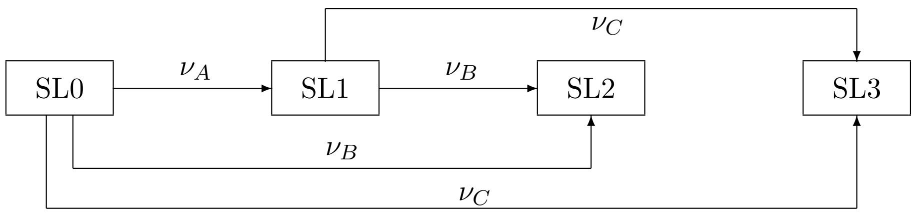


Figure 2

Noting that from SLO we enter $SL1$, $SL2$ or $SL3$ with probabilities

$$\frac{\nu_A}{\nu_A+\nu_B+\nu_C}, \quad \frac{\nu_B}{\nu_A+\nu_B+\nu_C}, \quad \frac{\nu_C}{\nu_A+\nu_B+\nu_C},$$

respectively, and that from $SL1$ we go to $SL2$ and $SL3$ with probabilities $\nu_B/(\nu_B+\nu_C)$, $\nu_C/(\nu_B+\nu_C)$, respectively, we see that

$$\begin{aligned}
u_{SL0} &= \frac{1}{\nu_A+\nu_B+\nu_C}, \\
p_{SL1} &= \frac{\nu_A}{\nu_A+\nu_B+\nu_C}, \quad u_{SL1} = p_{SL1}\cdot\frac{1}{\nu_B+\nu_C}, \\
p_{SL2} &= \frac{\nu_A}{\nu_A+\nu_B+\nu_C}\cdot\frac{\nu_B}{\nu_B+\nu_C} + \frac{\nu_B}{\nu_A+\nu_B+\nu_C} = \frac{\nu_B}{\nu_B+\nu_C}, \\
p_{SL3} &= \frac{\nu_A}{\nu_A+\nu_B+\nu_C}\cdot\frac{\nu_C}{\nu_B+\nu_C} + \frac{\nu_C}{\nu_A+\nu_B+\nu_C} = \frac{\nu_C}{\nu_B+\nu_C}, \\
w_{SL2} &= u_{SL0}\frac{\nu_A+\nu_B}{\nu_A+\nu_B+\nu_C} + \frac{\nu_B}{\nu_B+\nu_C}u_{SL1}, \\
w_{SL3} &= u_{SL0}\frac{\nu_C}{\nu_A+\nu_B+\nu_C} + \frac{\nu_C}{\nu_B+\nu_C}u_{SL1}.
\end{aligned}$$

For example in the expressions for p_{SL2} and w_{SL2}, the first term is the contribution from the event that $SL1$ is entered after $SL0$ and the second term is the contribution from the event that $SL2$ is entered directly after $SL0$. □

4.5 Introducing rewards

The above set–up is also convenient for computing performance measures defined in terms of Markov reward processes, cf. [How71], [Sou89]. That is, we assume that a reward is earned at rate r_i whenever the system is in state $i \in O$. The total reward gained is then

$$R = \mathbb{E} \int_0^\infty r_{X_t}\, dt = \sum_{i \in O} r_i u_i,$$

as is seen by decomposing the integral into the contributions from the sojourns in the individual states $i \in O$.

Example 4 Consider a computer system consisting of n identical processors, working in a parallel configuration, so that performance reductions due to overhead can be ignored. Each processor has a computation rate of β (measured in say million floating operations per second, MFLOPS). Let X_t be the number of processors which are operational at time t. Then $O = \{1, \ldots, n\}$, $F = \{0\}$, and if we define $r_i = \beta i$, then the reward R is the total number of floating point operations performed during the total operational period, whether at full performance or degraded level. □

We could also extend the concept of reward by introducing for $i \in F$ a penalty or cost c_i (typically ≥ 0) imposed if the system ends up in some failed state i. Here, c_i is a non-recurring or time-independent quantity, since the entered state is absorbing and the system will remain there for ever. The total reward gained is then

$$R = \sum_{i \in O} r_i u_i - \sum_{i \in F} c_i p_i$$

Example 5 For the computer system in the preceding example a situation where $n-1$ computers have failed leaving only one computer operational may be considered as a severe degradation, since an additional computer failure would leave no functioning computer, which is regarded as a catastrophic system failure. Therefore, a special automatic shut-down of the system is activated as soon as only one computer is in function. An unforeseen system state with no working computer is then avoided. However, there is a certain probability that the catastrophic state is entered in the short period before the shutdown has taken place. Here $O = \{1, ..., n\}$ and $F = \{b, m\}$, b meaning that the system has been subject to a benign failure (the automatic shut-down did work) and m that the system has been subject to a malign failure (the automatic shut-down did not work or worked too slowly) and we could take $c_b = 0$ and $c_m > 0$.

The corresponding service levels would be:

$$\begin{aligned} SL0 &= \{n, ..., 2\} \quad \text{(operational level)} \\ SL1 &= \{1\} \quad \text{(degraded level)} \\ SL2 &= \{b\} \quad \text{(failed level)} \\ SL3 &= \{m\} \quad \text{(catastrophically failed level)} \end{aligned}$$

□

The reward figure thus calculated may serve as a single figure of merit for systems with a well-defined design or as a means of comparison and trade-off between different design alternatives.

4.6 Non–exponential degradation

From a practical point of view, the assumption of exponential lifetimes may often be quite unrealistic. We shall here explain how phase–type assumptions, see [Neu81], allow to dispense with this, and avoid as much as possible of the general machinery of the area at the cost of restricting ourselves to some specific examples.

The basic examples of phase–type distributions are the following two:

Example 6 *The hyperexponential distribution* H_p with p parallel channels is a mixture of p exponential distributions with rates $\delta_1, \ldots, \delta_p$ so that the density is

$$\sum_{i=1}^{p} \pi_i \delta_i e^{-\delta_i x}, \tag{8}$$

where $\pi_1 + \cdots + \pi_p$=1.

In a reliability context, this could model the lifetime of an item which may be of one of p types, the ith type with exponential lifetime with parameter δ_i. For example if $p = 2$, type 1 could represent 'normal' items and type 2 items which are prone to early failure due to a defect. Thus $\pi_2 = 1 - \pi_1$ is the probability that an item is defective and one would have $\delta_2 \gg \delta_1$. □

Example 7 *The Erlang distribution* E_p with p phases is a Gamma distribution with integer parameter p and density

$$\delta^p \frac{x^{p-1}}{(p-1)!} e^{-\delta x}. \tag{9}$$

This corresponds to a convolution of p exponential densities with the same rate δ. In a reliability context, the Erlang distribution would be appropriate as a model for lifetimes of items which are subject to aging, i.e. have a bell–shaped lifetime density. It can be formally represented using the concept of cold stand–by units; e.g. if $p = 2$, we may think of the lifetime of the component as the lifetime of a system with two units with the same exponential lifetime parameter δ, the second unit being used as a cold stand–by for the first. □

Example 8 Consider a software package consisting of one interactive application program module and one user interface program module. The purpose of the interface module is to protect the application program by means of checking that all data and commands that are entered by the user fulfill certain requirements stated by the application program, e.g. that no illegal data or command is entered. An illegal entry may cause the application program to crash. Assume that the software package has a lifetime distribution B which is exponential with intensity δ_1, say. However, due to a mistake in the program loading sequence, occuring say with probability p, the interface module may be missing or defective so that all user input is simply transferred into the application module. Thus, the software package contains a *deficiency.* See [Jon91]. If we adopt a conservative approach, we assume that the first illegal data/command input will then cause a program crash.

Suppose that the arrival intensity of illegal data/commands is δ_2, which typically is much larger than δ_1. We may take $O = \{1, 2\}$, 2 representing an unprotected application program, 1, a protected one, and F consisting of a single state f. Thus the state diagram is as given on Figure 3, with initial probabilities $1 - p$, p for states 1 and 2, respectively.

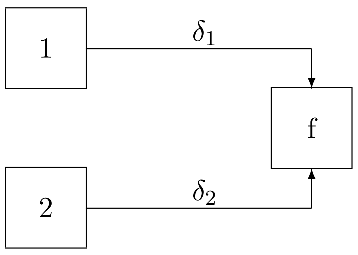


Figure 3

□

Example 9 Consider again the software package of the preceding example, but assume now that the 'typical' lifetime distribution B is much better approximated by an Erlang(2) distribution, with intensity δ_1, than by an exponential distribution. Then we could split state 1 into two states, 0,1, representing the fictitious stages (which of the two exponential units are working) of the lifetime. Thus the state diagram is as given in Figure 4, with initial probabilities $1 - p$, 0, p for states 0,1, 2, respectively

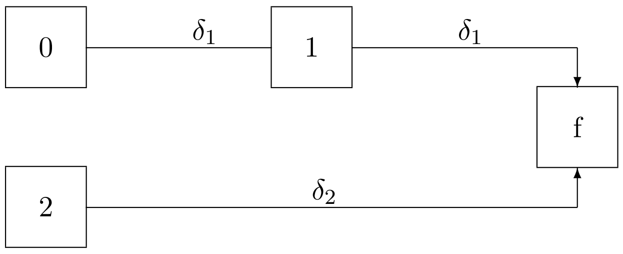


Figure 4

Note that the lifetime of such a system will typically have a failure rate with the commonly encountered 'bath–tub' shape. □

In Examples 8, 9, it is the distribution of the lifetime of the system as a whole that is of phase–type. However, it is equally interesting to use non–exponential models for the lifetimes of individual components. Here are two simple examples:

Example 10 Assume that in Example 2 the washing function has a lifetime distribution of the same form as in Example 9, whereas the spin–dry function has an exponential lifetime as before. Write A_1b for the state corresponding to the washer being in phase 1 of its life and the spin–dry function having failed, and so on. Then the state diagram is as given in Figure 5, with initial probabilities $1 - p$, p for states A_0B and A_2B and 0 for all other states.

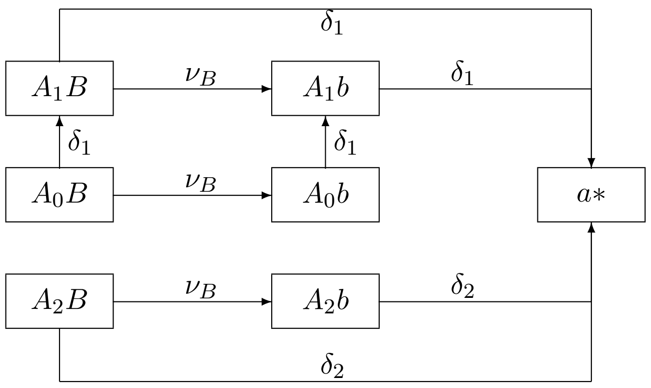


Figure 5

Note that the columns of states correspond to service levels 0, 1, 2, respectively. One case where this could be a reasonable model would be when there is a possibility of the washer having a deficiency which causes early failure, corresponding to states A_2B and A_2b. State A_0B is then interpreted as the washer being 'normal' and in its early stage of life (with the spin–dry function being operative), state A_1b as the washer being 'normal' and in its late stage of life (with the spin–dry function having failed) and so on. □

Example 11 Interchanging the form of the distributions of the lifetimes of the washer and spin–dry function in Example 10, the state diagram becomes instead (using the obvious notation)

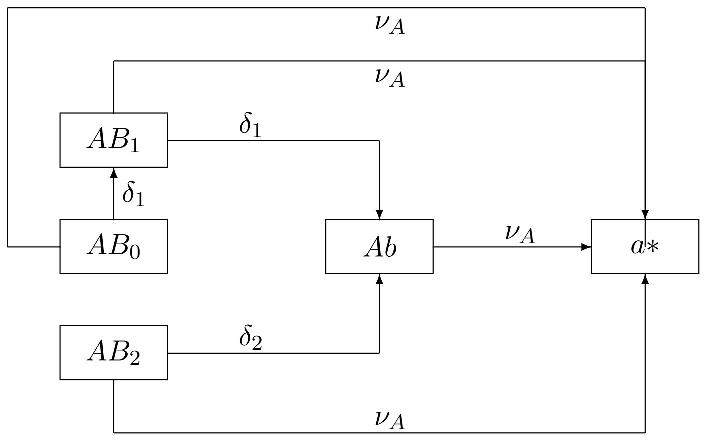


Figure 6

□

In the general formalism [Neu81], phase–type distributions are defined as absorption times of Markov processes with a finite number of states (the 'phases') and one absorbing state Δ. The important points in using them as models for lifetimes are that

- this involves no essential loss of generality since any distribution can be approximated arbitrarily closely by a phase–type distribution. (For the practical implementation see [Asm91] and references there).
- phase–type assumptions allow us to stay within the universe of Markovian modelling by introducing some additional states to the system.

It should be noted that it is not crucial that the phases have a direct physical interpretation as in the preceding examples: phase–type models are most often used in a purely descriptive manner. Thus in the older literature, the hyperexponential distribution is often used as a first approximation to a distribution with a coefficient of variation (c.v.) > 1 (in comparison, the c.v. of the exponential distribution is 1), and the Erlang(p) distribution as a first approximation to a distribution with c.v. < 1.

Note that the lifetime of the system as such (defined as the time $\min_{i \in F} \tau_i$ to absorption in some failed state) is of phase–type, as is immediately seen by collapsing F to a single absorbing state. Thus, the point of Examples 8, 9 is not so much that we get a phase–type lifetime but rather that we can arrive at a specified form of the lifetime distribution.

5 Conclusion

Even though dependability is a concept normally used in general and non-quantitive terms to describe computing systems and other systems, we have taken a step towards a quantitative understanding of it. This is done by merging aspects like reliability, performability, safety and security or parts thereof into a more general quality and defining a measure for it. Some extensions of the quality as a means of modelling dependability has been discussed. A method to dispense with the exponential failure rate assumption for system components using phase–type distributions has been outlined.

The considered system is perceived in terms of the service it delivers to the user and we recognize the fact that this service is normally degradable, i.e. it can be delivered in various amounts or at different levels. A mathematical definition for a vectorized measure based on Markov processes is given. The measure reflects the time the system is operational on a certain service level and the probability that it will reach this level, if ever. The measure is only applied to non-repairable systems, i.e. no feed-back in the Markov process is defined. However, there is no reason why this extension could not be introduced in the future, something that would pave the way for inclusion of the availability aspect into the model.

Two different algoritms for the calculation of the measure are given in the appendix. The first being appropiate for small systems and manual calculations. The second one, which makes use of matrix calculus, gives a general approach and is feasible for big systems and computerbased calculations.

References

[And82] T. Anderson & P.A. Lee (1982) Fault Tolerance terminology proposals. *Proc. 12th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-12**, 29–33. Santa Monica, California, June 1982.

[Agg89] K.K. Aggarwal (1989) A New Concept in Reliability Modelling. *Proceedings of the Annual RELIABILITY AND MAINTAINABILITY Symposium, 1989*, 86-90.

[Asm91] S. Asmussen, O. Nerman (1991) Fitting phase–type distributions via the EM algorithm. In preparation; preliminary version published in *Symposium i Anvendt Statistik, Copenhagen January 21–23, 1991* (K. Vest Nielsen ed.), 335–346. UNI–C, Copenhagen.

[Avi78] A. Avizienis (1978) Fault tolerance, the survival attribute of digital systems. *Proceedings of the IEEE* **66**, 1109-1125.

[Bob90] A. Bobbio & A. Cumani (1990) ML estimation of the parameters of a PH distribution in canonical triangular form. Technical report, Instituto Elettrotecnico Nazionale Galileo Ferraris, Torino.

[BSI] British Standard 5760, *Reliability of Systems, Equipments and Components*

[Car79] W.C. Carter (1979) Fault detection and recovery algorithms for fault-tolerant systems. *Prod. EURO IFIP'79, London, Sept. 1979*, 725-734.

[Chr84] F. Christian (1984) Correct and Robust Programs. *IEEE transactions on Software Engineering* **SE-10**.

[Dob90] J.E. Dobson, J. McDermid & B. Randell (1990), On the trustworthiness of Computing Systems. *Technical Report Series* **306**, University of Newcastle upon Tyne, Computing laboratory.

[For84] J.A.B. Fortes, C.S. Raghavendra (1984), Dynamically Reconfigurable Fault-tolerant Array Processors, pp. 386-392. *Proc. 12th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-12**, Santa Monica California, June 1984.

[Gra81] A. Graham (1981) *Kronecker Products and Matrix Calculus with Applications.* Ellis Horwood, Chichester.

[Hei91] D.I Heimann, N.Mittal, K.S. Trivedi (1991), Dependability Modelling for Computer Systems, *Proceedings of the Annual RELIABILITY AND MAINTAINABILITY Symposium, 1991*, 120-127.

[How71] R.A. Howard (1971) *Dynamic Probabilistic Systems.* New York Wiley 1971, ISBN 99-0002431-1.

[IEV] *International Electrotechnical Vocabulary*, Chapter 191. Reliability, Maintainability and Quality of Service, (IEV191). CCIR/CCITT Joint Study Group on Vocabulary, International Electrotechnical Commission, Geneva, 1987.

[Joh89] B.W. Johnson (1989) *Design and Analysis of Fault-Tolerant Digital Systems.* Addison-Wesley, ISBN 0-201-07570-9.

[Jon91] E. Jonsson (1991) A System-based Model for Dependable Computers, Technical Report No. **116**. Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden.

[Lap82] J.C. Laprie & A. Costes (1982) Dependability: a unifying concept for reliable computing *Proc. 12th IEEE Int. Symp. on Fault-tolerant Computing* **FTCS-12**, 18-21. Santa Monica, California, June 1982.

[Lap85] J.C. Laprie (1985) Dependable computing and fault tolerance: concepts and terminologi. *Proc. 15th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-15**, 2-11. Ann Arbor, Michigan, June 1985.

[Lap90] JC. Laprie (1990) Dependability: A unifying concept for reliable computing and fault tolerance. *Dependability of Resilient Computing Systems* (T. Anderson ed.), 1-28. Blackwell Scientific Publications.

[Lee82] P.A. Lee, D.E. Morgan (eds.) (1989) Fundamental concepts of fault tolerant computing. *Proc. 12th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-12**, 34-38. Santa Monica, California, June 1982.

[Mel77] P.M. Melliar-Smith & B.Randell (1977) Software reliability: the role of programmed exception handling. *SIGPLAN Notices* **12**, 95-100.

[Mey80] J.F. Meyer (1980) On Evaluating the Performability of Degradable Computing Systems, *IEEE Transaction on Computers* **C-29**, 720-731.

[Neu81] M.F. Neuts (1981) *Matrix–Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore.

[Orn86] S.M. Ornstein (1986) Safety issues for computer controlled systems *Proc. 16th IEEE Int. Symp. on Fault-Tolerant Computing* **FTCS-16**.

[Pfl89] C.P. Pfleeger (1989) *Security in Computing.* Prentice-Hall International, ISBN 0-13-799016-2.

[San89] W.H. Sanders, J.F. Meyers (1989) A unified approach for specifying measures of performance, dependability, and performability. *1st IFIP Conferance on Dependable Computers*, Santa Barbara, August 1989.

[Smi87] R.M. Smith, K.S. Trivedi (1987) A performability analysis of two multi-processor systems. *Proc. 17th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-17**, 224-229. Pittsburg, Pennsylvania, July 1987.

[Smi88] R.M. Smith, K.S. Trivedi & A.W. Ramesh (1988) Performability analysis: measures, an algoritm and a case Study. *IEEE Transaction on Computers* **37**, 406-417.

[Sou89] E. de Souza e Silva, H.R. Gail (1989) *Calculating Availability and Performability Measures of Repairable Computer Systems Using Randomization.* Journal of the ACM, Jan. 1989, vol. 36, no. 1.

[Tri82] K.S. Trivedi (1982) *Probability and Statistics with Reliability, Queuing and Computer Science Applications.* Prentice-Hall, Englewood Cliffs.

Appendix: Computing the performance measure

In this Appendix, we present some material that is more mathematically oriented. In particular, we develop two different algorithms for computing the dependability measure (2) and present a few somewhat more complicated examples.

The first algoritm is a hierarchical procedure, which in essence was used in the first few examples, e.g. Examples 2 and 3, even though the formalism was not made evident there. Thus, the hierarchical approach is suitable for calculations by hand for small systems.

The second algoritm is based on a general matrix formalism that may well be used for bigger systems, when using computer-aided tools. The matrix approach can be described by a very compact mathematical form and it is applicable also in the absence of a hierachical structure (i.e. feed–back is possible), see Example A2. Only when systems become so large that the matrix inversion presents a problem, is the hierachical approach again the most feasible. Such examples could occur, e.g. if one relaxes the assumption of exponential lifetimes, cf. Section 4.6.

A: a hierachical procedure

Our use of the word 'hierachical' is essentially a short–hand for no feed–back (the state diagram contains no loops). Under this assumption, it is possible to group the states $i \in E$ into subclasses $E_0, E_1, \ldots, E_L$ such that any state $i \in E_k$ can only be entered from some $j \in \sum_{\ell=0}^{k-1} E_\ell$, and this decomposition then serves to provide a recursive scheme for computing the dependability measure. It should be noted that the decomposition $E = \sum_{\ell=0}^{L} E_\ell$ is a purely technical tool: as will be seen from the examples below, it does not necessarily have any intrinsic interest, and in particular it is not related in a natural way to the different decompositions given by the service levels.

We define E_k as the set of states which can be reached in k or less transitions. That is, $i \in E_k$ if there exists $i_0, \ldots, i_{k-1} \in E$ such that

$$i_0 \neq i_1, i_1 \neq i_2, \ldots, i_{k-1} \neq i, \qquad \pi_{i_0}\lambda_{i_0 i_1}\lambda_{i_1 i_2} \cdots \lambda_{i_{k-1} i} > 0,$$

and k is the maximal number with this property. It is immediately clear that the E_k are disjoint. Furthermore, $\lambda_{ij} = 0$ whenever $i \in E_\ell$, $j \in E_k$ and $\ell \geq k$ (otherwise there is a chain of length $\ell + 1$ leading to j which contradicts k being maximal). The hierachical assumption of no feed–back amounts to

$$E = \sum_{\ell=0}^{L} E_\ell \tag{A.1}$$

for some L. This structure is found in all preceding examples and would appear to cover a broad class of systems without repair or maintenance.

Example A.1 In the setting of Example 1, E_k will consist of those states $i = (b_1, \ldots, b_p)$ with exactly k zeroes, i.e. exactly k failed components.

The following result shows that we can compute the u_i, p_i, w_i recursively. To this end, we let w_i be defined also for operational states $i \in O$.

Theorem 1 *In the presence of the hierachical structure (A.1),*

$$p_i = \pi_i, \quad w_i = \frac{1}{\lambda_i}, \quad i \in E_0, \tag{A.2}$$

$$w_i = \sum_{\ell=0}^{k-1} \sum_{j \in E_\ell : \lambda_{ji} > 0} \frac{\lambda_{ji}}{\lambda_j} \left\{ w_j + \frac{1}{\lambda_i} \right\}, \quad i \in E_k \cap O, \tag{A.3}$$

$$w_i = \sum_{\ell=0}^{k-1} \sum_{j \in E_\ell : \lambda_{ji} > 0} \frac{\lambda_{ji}}{\lambda_j} w_j, \quad i \in E_k \cap F, \tag{A.4}$$

$$p_i = \sum_{\ell=0}^{k-1} \sum_{j \in E_\ell : \lambda_{ji} > 0} p_j \frac{\lambda_{ji}}{\lambda_j}, \quad i \in E_k. \tag{A.5}$$

Proof Formula (A.2) is trivial. For (A.3), note that a $i \in E_k \cap O$ is necessarily entered from some state in E_{k-1}, say j. The probability of going to j from i is λ_{ji}/λ_j; the system has then already spent an average time w_j up to and including the sojourn in j and will spend an average time $1/\lambda_i$ in i. From this, (A.3) follows, and the remaining formulas are similar but easier to work with. □

Example A.2 Assume that in Example 3 we duplicate the computer for vehicle dynamics; states with two VD computers are denoted by $**C_2$, those with one by $**C$ and those with none by $**c$. Then

SL0 $= ABC_2 + ABC$ (full performance level)

SL1 $= aBC_2 + aBC$ (degraded performance level)

SL2 $= abC_2 + abC + AbC_2 + AbC$ (failed)

SL3 $= **c$ (catastrophic failure)

Again, $O = SL0 \cup SL1$, $F = SL2 \cup SL3$, and we get

$$\begin{aligned} E_0 &= \{ABC_2\}, \\ E_1 &= \{aBC_2, AbC_2, ABC\}, \\ E_2 &= \{abC_2, AbC, ABc, aBC\}, \\ E_3 &= \{abC, aBc, Abc\}, \\ E_4 &= \{abc\} \end{aligned}$$

The steps in the calculations of Theorem 1 are as follows. For $i \in E_0$, there is only $i = ABC_2$ to consider, and obviously $p_{ABC_2} = 1$. For $i \in E_1$, we note that a transition from a state $**C_2$ to $**C$ has intensity $2\nu_C$ and get

$$p_{aBC_2} = \frac{\nu_A}{\nu_A + \nu_B + 2\nu_C}, \quad p_{AbC_2} = \frac{\nu_B}{\nu_A + \nu_B + 2\nu_C}, \quad p_{ABC} = \frac{2\nu_C}{\nu_A + \nu_B + 2\nu_C}.$$

For $i \in E_2$,

$$\begin{aligned} p_{aBC} &= p_{ABC} \frac{\nu_A}{\nu_A + \nu_B + \nu_C} + p_{aBC_2} \frac{2\nu_C}{\nu_B + 2\nu_C}, \\ p_{AbC} &= p_{ABC} \frac{\nu_B}{\nu_A + \nu_B + \nu_C} + p_{AbC_2} \frac{2\nu_C}{\nu_A + 2\nu_C}, \\ p_{ABc} &= p_{ABC} \frac{\nu_C}{\nu_A + \nu_B + \nu_C}, \\ p_{abC_2} &= p_{aBC_2} \frac{\nu_B}{\nu_B + 2\nu_C} + p_{AbC_2} \frac{\nu_A}{\nu_A + 2\nu_C}, \end{aligned}$$

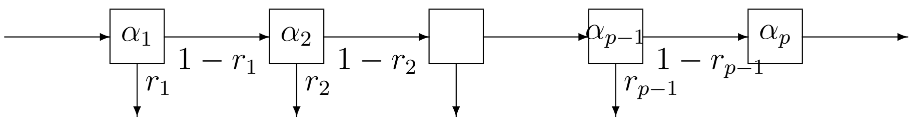
and for $i \in E_3$,

$$\begin{aligned}
p_{abC} &= p_{abC_2} + p_{AbC}\frac{\nu_A}{\nu_A+\nu_C} + p_{aBC}\frac{\nu_B}{\nu_B+\nu_C},\\
p_{aBc} &= p_{aBC}\frac{\nu_C}{\nu_B+\nu_C},\\
p_{Abc} &= p_{AbC}\frac{\nu_C}{\nu_A+\nu_C},
\end{aligned}$$

noting that ABc is absorbing so that, e.g., aBc cannot be entered from there. Similarly, since $aBc, Abc \in E_3$, it follows for E_4 that $p_{abc} = p_{abC}$.

From these formulas we can calculate, e.g., the probability of a catastrophic failure as $p_{ABc} + p_{Abc} + p_{aBc} + p_{abc}$. □

When working with phase–type assumptions on the lifetimes of the individual components, a particular class of distributions that fit nicely into the hierachical set–up are Coxian distributions, defined as lifetimes of systems with a state diagram as follows:



This class of distributions is very popular in much of the reliability literature, see e.g. [Bob90] and references there. Note that the Erlang distribution is a special case of a Coxian distribution.

Example A.3 Assume in Example 2 that the lifetime of the washer and the spin-dryer are both Coxian, with parameters $\alpha_1,\dots,\alpha_p$, $r_1,\dots,r_p$ for the washer and $\beta_1,\dots,\beta_q$, $s_1,\dots,s_q$ for the spin-dryer. Then

$$SL0 = \{ij:\ i=1,\dots,p,\ j=1,\dots,q\}\,,\quad SL1 = \{i:\ j=1,\dots,p\}\,,\quad SL2 = \{f\}\,,$$

where for example state $ij \in SL0$ represents the washer being in stage (phase) i of its life and the dryer in stage j. From $ij \in SL0$ we can go to $(i+1)j$ with intensity $\alpha_i(1-r_i)$ (provided $i<p$), to $i(j+1)$ with intensity $\beta_j(1-s_j)$ (provided $j<q$), to i with intensity β_q when $j=q$ and to f with intensity α_p when $i=p$. It is immediate that the hierachical assumption (A.1) is satisfied (whereas this is not the case for general phase–type distribution with possible feedback between the phases), and if, e.g., $p=3$, $q=2$, we get $L=5$,

$$E_0=\{11\}\,,\quad E_1=\{12,21\}\,,\quad E_2=\{22,31,1\}\,,\quad E_3=\{32,2\}\,,\quad E_4=\{3\}\,,E_5=\{f\}\,.$$

□

This example clearly illustrates that even though phase–type assumptions may blow up the dimension substantially, the hierachical method for performing computations is not all that sensitive to this problem ('the curse of dimensionality').

B: a general matrix formalism

We now develop an alternative to the hierachical procedure, based upon matrix calculus. Let $\boldsymbol{\Lambda}$ be the $O \times O$ matrix with ijth entry λ_{ij}, $i, j \in O$. Note that $\boldsymbol{\Lambda}$ does not involve the λ_{ji}, $i \in F$; these are represented instead in terms of the (column) vectors $\boldsymbol{\ell}^{(i)} = (\lambda_{ji})_{j\in O}$, $i \in F$. We also let $\boldsymbol{e}^{(i)}$ denote the ith (column) unit vector and $\boldsymbol{q}^{(i)}$ the column vector $-\boldsymbol{\Lambda}^{-1}\boldsymbol{\ell}^{(i)}$. Note that the initial vector $\boldsymbol{\pi}$ is written as a row vector. Thus e.g. in (B.1) below, $\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}$ is a row vector and $\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{e}^{(i)}$ a real number (as should be).

The following result shows that once the key step of computing $\boldsymbol{\Lambda}^{-1}$ has been overcome, the performance measure can immediately be calculated:

Theorem 2

$$\begin{aligned} u_i &= -\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{e}^{(i)}, \quad i \in O, && \text{(B.1)} \\ p_i &= -\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{\ell}^{(i)}, \quad i \in F, && \text{(B.2)} \\ w_i &= \boldsymbol{\pi}\boldsymbol{\Lambda}^{-2}\boldsymbol{\ell}^{(i)}, \quad i \in F. && \text{(B.3)} \end{aligned}$$

Proof The formula (B.1) follows by noting that the t–step transition matrix of the Markov process is $e^{\boldsymbol{\Lambda}t}$, hence the vector of state probabilities at time t is $\boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}$ and the probability of being in i is $\boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{e}^{(i)}$, so that by standard formulas for integrating matrix–exponentials (e.g. [Gra81])

$$u_i = \int_0^\infty \boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{e}^{(i)}dt = -\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{e}^{(i)}.$$

Similarly, since the contribution to p_i from the Markov process is in state j in the time interval $[t, t+dt]$ is $\boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{e}^{(j)} \cdot \lambda_{ji}dt$, we have

$$p_i = \sum_{j\in O}\int_0^\infty \boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{e}^{(j)} \cdot \lambda_{ji}dt = \int_0^\infty \boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{\ell}^{(i)} = -\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{\ell}^{(i)}.$$

Finally, for (B.3) we first note that, by a similar argument as for (B.2), the probability that the Markov process is finally absorbed in state i given that it starts in state j is

$$\int_0^\infty \boldsymbol{e}^{(j)'}e^{\boldsymbol{\Lambda}t}\boldsymbol{\ell}^{(i)} = -\boldsymbol{e}^{(j)\prime}\boldsymbol{\Lambda}^{-1}\boldsymbol{\ell}^{(i)} = q_j^{(i)}.$$

Hence

$$\begin{aligned} w_i &= \sum_{j\in O}\mathbb{E}\int_0^\infty I(X_t = j, t < \tau_i < \infty)dt \\ &= \sum_{j\in O}\int_0^\infty \mathbb{P}(X_t = j, t < \tau_i < \infty)dt \\ &= \sum_{j\in O}\int_0^\infty \boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{e}^{(j)} \cdot q_j^{(i)} = \int_0^\infty \boldsymbol{\pi}e^{\boldsymbol{\Lambda}t}\boldsymbol{q}^{(i)} \\ &= -\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{q}^{(i)} = \boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\ell}^{(i)} = \boldsymbol{\pi}\boldsymbol{\Lambda}^{-2}\boldsymbol{\ell}^{(i)}. \end{aligned}$$

□

Example B.1 We shall reinspect the formulas of Example 3 in view of Theorem 2. Recalling that $O = \{SL0, SL1\}$, we have

$$\boldsymbol{\Lambda} = \begin{pmatrix} -\nu_A - \nu_B - \nu_C & \nu_A \\ 0 & -\nu_B - \nu_C \end{pmatrix}, \quad \boldsymbol{\ell}^{(SL2)} = \begin{pmatrix} \nu_B \\ \nu_B \end{pmatrix}, \quad \boldsymbol{\ell}^{(SL3)} = \begin{pmatrix} \nu_C \\ \nu_C \end{pmatrix}.$$

Thus

$$\boldsymbol{\Lambda}^{-1} = \begin{pmatrix} -\frac{1}{\nu_A+\nu_B+\nu_C} & -\frac{\nu_A}{(\nu_A+\nu_B+\nu_C)(\nu_B+\nu_C)} \\ 0 & -\frac{1}{\nu_B+\nu_C} \end{pmatrix}.$$

Since $\boldsymbol{\pi} = (1\ 0)$, $\boldsymbol{\pi}\boldsymbol{\Lambda}^{-1}$ is simply the first row of $\boldsymbol{\Lambda}^{-1}$, which according to (B.1) is $(-u_{SL0}\ \ -u_{SL1})$ — in accordance with the formulas of Example 3. The remaining formulas are easily checked in the same way, using

$$\boldsymbol{\Lambda}^{-2} = \begin{pmatrix} \frac{1}{(\nu_A+\nu_B+\nu_C)^2} & \frac{\nu_A(\nu_A+2\nu_B+2\nu_C)}{(\nu_A+\nu_B+\nu_C)^2(\nu_B+\nu_C)^2} \\ 0 & \frac{1}{(\nu_B+\nu_C)^2} \end{pmatrix}.$$

E.g.

$$p_{SL2} \;=\; \begin{pmatrix} -\frac{1}{\nu_A+\nu_B+\nu_C} & -\frac{\nu_A}{(\nu_A+\nu_B+\nu_C)(\nu_B+\nu_C)} \end{pmatrix} \begin{pmatrix} \nu_B \\ \nu_B \end{pmatrix} = \frac{\nu_B}{\nu_B+\nu_C}.$$

□

The point that hierachical structure is not needed for Theorem 2 is illustrated by the following example:

Example B.2 Consider the washing machine in Example 2, and assume that the spin-drying function is maintainable, with a maintenance time which we assume for simplicity to be exponential, with rate say μ_B. This means that in the state diagram on Fig. 1, we have to add an arrow from SL1 to SL0 with mark μ_B. Thus

$$\begin{aligned} \boldsymbol{\Lambda} &= \begin{pmatrix} -\nu_A - \nu_B & \nu_B \\ \mu_B & -\nu_A - \mu_B \end{pmatrix}, \\ \boldsymbol{\Lambda}^{-1} &= \begin{pmatrix} \frac{-\nu_A-\mu_B}{\nu_A^2+\nu_A\nu_B+\nu_A\mu_B} & -\frac{\nu_B}{\nu_A^2+\nu_A\nu_B+\nu_A\mu_B} \\ -\frac{\mu_B}{\nu_A^2+\nu_A\nu_B+\nu_A\mu_B} & \frac{-\nu_A-\nu_B}{\nu_A^2+\nu_A\nu_B+\nu_A\mu_B} \end{pmatrix}, \end{aligned}$$

and from this the dependability measure (2) can be computed in a straightforward manner. □

It should be noted that the form of the formulas of Theorem 2 is close to formulas for moments and other functionals of phase–type distributions, see [Neu81]. This is in fact not surprising, since in our model with $E = O \cup F$ one may see O as corresponding to the set of phases (transient states) and F to the absorbing state $\{\Delta\}$. Thus, if F has $m > 1$ elements, we are in a set–up generalizing the phase–type setting to more than one absorbing state.

The following examples show how to define the parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\ell}^{(i)}$ in some of our earlier examples involving phase–type modelling:

Example B.3 For the software package in Example 8, we have in the case of a hyperexponential lifetime that

$$\boldsymbol{\pi} = \ (1-p \ \ p), \quad \boldsymbol{\Lambda} = \begin{pmatrix} -\delta_1 & 0 \\ 0 & -\delta_2 \end{pmatrix}, \quad \boldsymbol{\ell}^{(f)} = \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix}.$$

□

Example B.4 If instead the lifetime has the more complicated form in Example 9, we get

$$\boldsymbol{\pi} = \ (1-p \ \ 0 \ \ p), \quad \boldsymbol{\Lambda} = \begin{pmatrix} -\delta_1 & \delta_1 & 0 \\ 0 & -\delta_1 & 0 \\ 0 & 0 & -\delta_2 \end{pmatrix}, \quad \boldsymbol{\ell}^{(f)} = \begin{pmatrix} 0 \\ \delta_1 \\ \delta_2 \end{pmatrix}.$$

Example B.5 As a more complicated example, consider again the washing machine of Example 2 and assume that both the washing and the spin–dry functions have lifetime distributions of the form in Examples 9, B3, with parameters $p_A, \delta_1^{(A)}, \delta_2^{(A)}$ for the washer and $p_B, \delta_1^{(B)}, \delta_2^{(B)}$ for the dryer. Then SL0 splits into 9 states ij, i indicating the phase of the washer and j the phase of the dryer, and SL1 into three states corresponding to the phase of the washer. Ordering the states in SL0 lexicographically, $(00, 01, 02, 10, \cdots)$, separating $SL0$ and $SL1$ by double lines and letting $\cdot$ indicate a zero entry, the matrix $\boldsymbol{\Lambda}$ is then as given on p. 26. □

$$\left(\begin{array}{ccc|ccc|cc}
-\delta_1^{(A)} - \delta_1^{(B)} & \delta_1^{(B)} & \cdot & \delta_1^{(A)} & \cdot & \cdot & \cdot & \cdot \\
\cdot & -\delta_1^{(A)} - \delta_1^{(B)} & \cdot & \cdot & \delta_1^{(A)} & \cdot & \cdot & \cdot \\
\cdot & \cdot & -\delta_1^{(A)} - \delta_2^{(B)} & \cdot & \cdot & \delta_1^{(A)} & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & -\delta_1^{(A)} - \delta_1^{(B)} & \delta_1^{(B)} & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & -\delta_1^{(A)} - \delta_1^{(B)} & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & -\delta_1^{(A)} - \delta_2^{(B)} & \cdot & \cdot \\
\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -\delta_2^{(A)} - \delta_1^{(B)} & \delta_1^{(B} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -\delta_2^{(A)} - \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\hline\hline
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}$$