ELSEVIER

Reliability Engineering and System Safety 79 (2003) 229–238

RELIABILITY ENGINEERING & SYSTEM SAFETY

www.elsevier.com/locate/ress

Variance decomposition-based sensitivity analysis via neural networks

Marzio Marseguerra[a,*], Riccardo Masini[a], Enrico Zio[a], Giacomo Cojazzi[b]

[a]Department of Nuclear Engineering, Polytechnic of Milan, CESNEF, Via Ponzio 34/3, 20133 Milan, Italy
[b]EC, Joint Research Centre, Institute for the Protection and Security of the Citizen, TP 421, I-21020 Ispra (VA), Italy

Abstract

This paper illustrates a method for efficiently performing multiparametric sensitivity analyses of the reliability model of a given system. These analyses are of great importance for the identification of critical components in highly hazardous plants, such as the nuclear or chemical ones, thus providing significant insights for their risk-based design and management. The technique used to quantify the importance of a component parameter with respect to the system model is based on a classical decomposition of the variance. When the model of the system is realistically complicated (e.g. by aging, stand-by, maintenance, etc.), its analytical evaluation soon becomes impractical and one is better off resorting to Monte Carlo simulation techniques which, however, could be computationally burdensome. Therefore, since the variance decomposition method requires a large number of system evaluations, each one to be performed by Monte Carlo, the need arises for possibly substituting the Monte Carlo simulation model with a fast, approximated, algorithm. Here we investigate an approach which makes use of neural networks appropriately trained on the results of a Monte Carlo system reliability/availability evaluation to quickly provide with reasonable approximation, the values of the quantities of interest for the sensitivity analyses. The work was a joint effort between the Department of Nuclear Engineering of the Polytechnic of Milan, Italy, and the Institute for Systems, Informatics and Safety, Nuclear Safety Unit of the Joint Research Centre in Ispra, Italy which sponsored the project.

Keywords: Sensitivity analysis; Neural networks; Monte Carlo

1. Introduction

Sensitivity analysis is a tool which may be used to study the behavior of a model and to ascertain how much the outputs of a given model depend on each or some of its input parameters [1]. In this respect it constitutes an aide for risk-based design and management of hazardous plants, such as the nuclear and chemical ones, in that it can help in retrieving information on components and systems' criticality. Several approaches have been developed for performing sensitivity studies, ranging from differential to Monte Carlo analysis, response surface methodology, and Fourier amplitude sensitivity test (FAST). These approaches are based on Taylor series expansion, random sampling, response surface construction, and Fourier series, respectively [1–4]. Typically, these approaches entail to compute the model output (a reliability or risk measure in our case of interest) several times for different input values sampled from appropriate ranges. Often, the computation times required by the numerical solution of the model render these analyses prohibitively costly, so that one has to resort to simplified but fast models or empirical response surfaces.

The objective of this work is to devise a method for performing a multiparametric uncertainty and sensitivity analysis of the reliability model of a properly selected system. The technique used to quantify the uncertainty contribution of a component is based on the variance decomposition method [5]. It consists in considering several evaluations of the system unreliability/unavailability characteristics in correspondence of different values of the uncertain parameters (e.g. component failure rates) and computing an *index of importance* that measures how much a set of parameters influences the uncertainty in the system unreliability and unavailability.

Abbreviations: ANN, artificial neural network; ASTRA, advanced software tool for reliability analysis; FAST, Fourier amplitude sensitivity test; ISIS, Institute for Systems Informatics and Safety; JRC, Joint Research Centre; MARA, Monte Carlo availability reliability analysis; NEST, neural simulation tool.

* Corresponding author. Fax: +39-2-2399-6309.
E-mail address: enrico.zio@polimi.it (E. Zio).

0951-8320/03/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0951-8320(02)00234-X

Nomenclature

A_x, B_x, C_x	components of the system studied
Q_i	unavailability of component *i*
λ_0, λ^*	failure rate (y^{-1})
μ	repair rate (y^{-1})
η_X^2	index of importance of the parameter *X*
a	aging rate
p	probability of imperfect repairs
π_λ, π_μ	increase of failure rates and reduction of repair rates due to a deteriorating effect for imperfect repairs.
τ	period between maintenances
T_m	mission time (y)

When the model of the system is realistically complicated, its analytical evaluation is at least impractical and one has to resort to Monte Carlo simulation which, however, could be computationally burdensome [6,7]. Therefore, since the variance decomposition method requires a large number of system evaluations, each one to be performed by Monte Carlo simulation, the need arises for substituting the Monte Carlo simulation model with a fast, approximated, algorithm. In our work, we employ an empirical model built by training artificial neural networks (ANN) on the results of the Monte Carlo simulation [8,9]. The type of neural network employed here is the classical multi-layered, feed-forward one trained by the error back-propagation method [10]. The networks used have been generated with a user-friendly software Neural Simulation Tool (NEST) developed at the Department of Nuclear Engineering of the Polytechnic of Milan (http://www.cesnef.polimi.it/ricerca/sicura/pagweb/lasar/nest.htm). The training patterns for the ANNs have been generated using a user-friendly Monte Carlo simulation code, Monte Carlo Availability Reliability Analysis (MARA), also developed at the same Department of the Polytechnic of Milan (http://www.cesnef.polimi.it/ricerca/sicura/pagweb/lasar/mara.htm). The work presented here was supported with JRC contract No. 14546-1998-11 F1 ED ISPIT.

2. The reference system

The system considered here is constituted by three macro-components in series (Fig. 1). Each macro-component is composed of a redundancy configuration (called 'block') of components with constant failure and repair rates (Table 1).

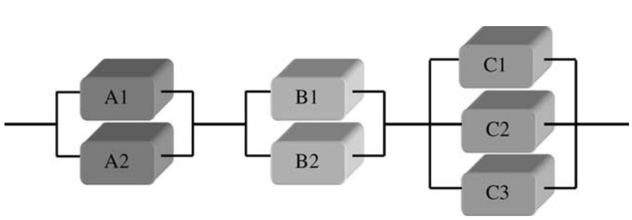

Fig. 1. System layout.

3. The Monte Carlo availability and reliability analysis code

In order to explore the behavior of complex system models, the MARA code was developed at the Department of Nuclear Engineering of the Polytechnic of Milan, in Fortran language and with a Visual Basic graphical interface.

Several procedures have been implemented which allow the definition of different logics of components' operation, e.g. stand-by and load-sharing, the consideration of aging and maintenance, and other realistic aspects of the system behavior. In the following we briefly describe the basics behind the models of interest for our case study.

3.1. The failure and repair process with aging and deterioration

Components are assumed to age in time according to a linear model [11] for the failure rate $\lambda(t)$,

$$\lambda(t) = \lambda_0 + at \tag{1}$$

where λ_0 is a constant term and a is the aging rate. Then, the failure times behave according to the following cumulative distribution function (cdf) and probability density function (pdf):

$$\begin{aligned} F(t) &= 1 - \exp\left(-\int_0^t \lambda(u)\mathrm{d}u\right) \\ &= 1 - \exp\left(-\int_0^t (\lambda_0 + au)\mathrm{d}u\right) \\ &= 1 - \exp\left(-(\lambda_0 t + \tfrac{1}{2}at^2)\right) \end{aligned} \tag{2}$$

Table 1
Components' failure and repair rates

Component	Failure rate λ_i (y^{-1})	Repair rate μ_i (y^{-1})
A_1	6.0×10^{-3}	1.7×10^{-1}
A_2	2.6×10^{-3}	1.0×10^{-1}
B_1	5.3×10^{-3}	3.0×10^{-1}
B_2	3.6×10^{-3}	1.0×10^{-1}
C_1	8.1×10^{-3}	5.0×10^{-1}
C_2	5.3×10^{-3}	3.0×10^{-1}
C_3	7.0×10^{-3}	5.0×10^{-1}