A Non-Parametric Non-Stationary Procedure for Failure Prediction

Jonas D. Pfefferman and Bruno Cernuschi-Frías

***Abstract*—The time between failures is a very useful measurement to analyze reliability models for time-dependent systems. In many cases, the failure-generation process is assumed to be stationary, even though the process changes its statistics as time elapses.**

This paper presents a new estimation procedure for the probabilities of failures; it is based on estimating time-between-failures. The main characteristics of this procedure are that no probability distribution function is assumed for the failure process, and that the failure process is not assumed to be stationary. The model classifies the failures in Q different types, and estimates the probability of each type of failure s-independently from the others.

This method does not use histogram techniques to estimate the probabilities of occurrence of each failure-type; rather it estimates the probabilities directly from the values of the time-instants at which the failures occur. The method assumes quasistationarity only in the interval of time between the last 2 occurrences of the same failure-type.

An inherent characteristic of this method is that it assigns different sizes for the time-windows used to estimate the probabilities of each failure-type. For the failure-types with low probability, the estimator uses wide windows, while for those with high probability the estimator uses narrow windows.

As an example, the model is applied to software reliability data.

***Index Terms*—Predictive validity, software reliability model.**

Acronyms and Abbreviations[1]

CF	cumulative (number of) failures
FT	failure time
pdf	probability density function
r.v.	random variable
ToF	type of failure
ToF_k	ToF #k
P&C-F	method and estimators proposed in this paper.

Notation

n	instance of time
n_{rem}	remaining number of failures
$p_k(n)$	$\Pr\{\text{occurrence of ToF}_k \text{ at } n\}$
$\hat{p}_k(n)$	estimate of $p_k(n)$
t_{rem}	remaining test time
$T_k(n)$	time from the latest occurrence of the ToF_k to the current n
TBF	time between failures
TBF_k	TBF of type k
$\widehat{\text{TBF}}_k(n)$	estimate of the TBF for the ToF_k at n
$\mu(t)$	mean value function
$\tau_k(i)$	time between occurrences i and $(i+1)$ of the ToF_k.

Manuscript received December 31, 1998; revised October 29, 1999, December 26, 2000, and August 16, 2001. This work was supported in part by the University of Buenos Aires, Argentina, Grants TI-09 and I-025, and the *"Consejo Nacional de Investigaciones Científicas y Técnicas"*, (Argentine National Scientific and Technical Research Council), CONICET, Grant PIP-4030. Responsible Editor: P. S. F. Yip.

The authors are with Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires, Argentina (e-mail: JPfeffr@galileo.fi.uba.ar; BCF@ieee.org).

Digital Object Identifier 10.1109/TR.2002.804733

[1]The singular and plural of an acronym are always spelled the same.

Assumptions

1) Each failure-type is considered independently from the others in order to obtain the estimates of the corresponding TBF; then, a normalization factor is introduced to insure (1).
2) The probability distribution for each failure-type does not change within the interval between 2 consecutive occurrences of this failure-type.
3) The underlying pdf is not known.
4) The failure stochastic process is nonstationary.
5) The failure process is locally approximately ergodic.
6) Only 1 failure-type occurs at each discrete n.
7) The $p_k(n)$ do not change within the interval between the 2 latest consecutive occurrences of the failure k.

I. Introduction

TO ANALYZE failures, many models usually assume a particular failure pdf [1]–[3], [5], [9], [13], [15]–[18], [21]–[26]. In many cases, the system under study does not satisfy the hypothesis of s-independence [10] and is nonstationary, but it is assumed to be stationary as a simplifying hypothesis, e.g., when the main interest is prediction of the total number of failures.

Here, assumptions #3 and #4 are used. This model is based on previous work [19] where a nonparametric nonstationary estimation procedure was introduced to improve the compression ratio of some lossless compression methods [8], [14]. This new procedure adapts quickly to statistical changes in time.

The P&C-F estimators use assumption #5, that is, in a sense, the process is locally approximately ergodic over a sliding time window at every n: even though the process might not be stationary or ergodic, it varies slowly enough so that over a sliding time window of appropriate size, the process might be considered stationary and ergodic. Strictly, only averages over the ensemble (expectations) should be considered. But, considering the process locally ergodic, the model can use time averages over sliding windows, to estimate s-expected values from a single realization of the process. Hence, histogram techniques over appropriate window sizes [6] or adaptive

TABLE I
DATA-SET FOR THE SYNTHETIC EXAMPLE: s-INDEPENDENT BERNOULLI r.v. WITH A LINEARLY VARYING PARAMETER

Pr{S}	O	H	W.H. N=6	P.E.	P.M. N=5
1	S	1.00	1.00	1.00	1.00
0	F	0.50	0.50	0.67	0.67
0.30	F	0.33	0.33	0.50	0.50
0.31	F	0.25	0.25	0.50	0.50
0.32	F	0.20	0.20	0.50	0.63
0.33	S	0.33	0.33	0.17	0.47
0.34	F	0.29	0.17	0.29	0.39
0.35	F	0.25	0.17	0.17	0.32
0.36	S	0.33	0.33	0.25	0.27
0.37	S	0.40	0.50	0.50	0.27
0.38	F	0.36	0.50	0.75	0.39
0.39	S	0.42	0.50	0.60	0.45
0.40	F	0.38	0.50	0.50	0.52
0.41	F	0.36	0.50	0.33	0.54
0.42	F	0.33	0.33	0.33	0.50
0.43	S	0.38	0.33	0.20	0.39
0.44	S	0.41	0.50	0.50	0.37
0.45	F	0.39	0.33	0.75	0.42
0.46	F	0.37	0.33	0.50	0.46
0.47	F	0.35	0.33	0.50	0.49
0.48	F	0.33	0.33	0.50	0.55
0.49	F	0.32	0.17	0.50	0.55
0.50	S	0.35	0.17	0.14	0.43
0.51	F	0.33	0.17	0.25	0.38
0.52	F	0.32	0.17	0.14	0.31
0.53	F	0.31	0.17	0.14	0.24
0.54	S	0.33	0.33	0.20	0.18
0.55	S	0.36	0.50	0.50	0.25
0.56	S	0.38	0.50	0.50	0.30
0.57	F	0.37	0.50	0.80	0.43
0.58	S	0.39	0.67	0.67	0.53
0.59	S	0.41	0.83	0.80	0.65
0.60	F	0.39	0.67	0.75	0.70
0.61	S	0.41	0.67	0.60	0.72
0.62	F	0.40	0.50	0.50	0.66
0.63	S	0.42	0.67	0.50	0.63
0.64	F	0.41	0.50	0.50	0.57
0.65	S	0.42	0.50	0.50	0.52
0.66	S	0.44	0.67	0.67	0.53
0.67	S	0.45	0.67	0.67	0.57
0.68	S	0.46	0.83	0.67	0.60
0.69	S	0.48	0.83	0.67	0.63
0.70	S	0.49	1.00	0.67	0.67
0.71	F	0.48	0.83	0.88	0.71
0.72	S	0.49	0.83	0.78	0.73
0.73	S	0.50	0.83	0.88	0.77
0.74	S	0.51	0.83	0.88	0.81
0.75	F	0.50	0.67	0.80	0.84
0.76	S	0.51	0.67	0.67	0.80
0.77	S	0.52	0.83	0.80	0.80
0.78	S	0.53	0.83	0.80	0.79
0.79	S	0.54	0.83	0.80	0.77
0.80	S	0.55	0.83	0.80	0.77
0.81	S	0.56	1.00	0.80	0.80
0.82	S	0.56	1.00	0.80	0.80
0.83	S	0.57	1.00	0.80	0.80
0.84	S	0.58	1.00	0.80	0.80
0.85	S	0.59	1.00	0.80	0.80
0.86	F	0.58	0.83	0.92	0.82
0.87	S	0.58	0.83	0.85	0.83

algorithms [4] can be used. As a simple example, consider the case where the results of a sequential experiment have only 2 possible results, Success and Fail, s-independently but not s-identically distributed in time as Bernoulli r.v. with different parameter at each n; and where the parameter varies slowly. Table I and Fig. 1 give a synthetic run of this model, where the parameter, which corresponds to the probability of Success, varies linearly, though any arbitrary smoothly varying function can be considered. The P&C-F estimation method does not assume a parametric model as a function of time for

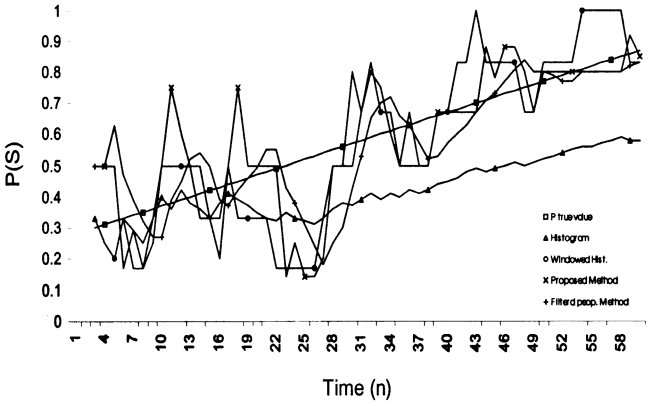

Fig. 1. Typical run of estimators for the synthetic example: s-independent Bernoulli r.v. with a linearly varying parameter.

the probability to be estimated. The main goal is to estimate the probability of Success, at each n, using the current observation of the experiment as well as a few of the previous observations.

In Table 1

• Column 1 (see also Fig. 1) gives the probability of success. Each row corresponds to an n. Each Success event (S) is a Bernoulli r.v., independently drawn at each n, with a probability Pr$\{S\}$.

• Column 2 gives a realization of the process.

• Column 3 corresponds to the estimation of Pr$\{S\}$ at each n, by simply dividing the "number of Success events" by the "total number of events up to the current n" (a histogram). This technique is not acceptable if Pr$\{S\}$ varies with time. If such a variation is slow enough, then a windowed histogram can be considered.

• Column 4 (see also Fig. 1) gives the results for a sliding windowed histogram, corresponding to the "number of Success events over the latest N events" divided by $N = 6$. Defining N is a delicate matter because it depends on the characteristics of the time varying Pr$\{S\}$. The method P&C-F here does not use a histogram approach; instead, the estimate of Pr$\{S\}$ considered here is proportional to $1/T$. Analogously, the estimate for the Fail probability is proportional to $1/R$. Because both probabilities add up to 1, the following estimate for Pr$\{S\}$ is used:

$T \equiv$ time between the latest 2 occurrences of the Success event

$R \equiv$ time between the latest 2 occurrences of the Fail event

$$\hat{p} = \frac{R}{T+R}.$$

• Column 5 (see also Fig. 1) shows this estimate.

• Column 6 is the average of the previous 5 estimates in column 5.

Fig. 1 shows that the averaged method performs much better than the windowed histogram estimator. The remainder of this section improves the rather crude estimate in the first part of this section. This example shows how using the time between the Success events is equivalent to a variable-width window, for which

- events with higher probability use smaller windows,
- events with smaller probability use larger windows.

To include several types of failure-exclusion, proceed as follows. This paper classifies failures in Q types, numbered from 0 to $Q-1$, with type 0 corresponding to the occurrence of 0 failures. To simplify the notation, consider the occurrence of no failure at n as the occurrence of ToF_0 at n. Also, consider that assumption #6 is valid; although this implies that the various failure-types are not s-independent at each n, the purpose is to estimate the probabilities for each failure-type independently from the others, updating the estimated probabilities at every n for all Q failure-types. This is similar to what is done when histograms are used to estimate probabilities.

$\mathbf{S} \equiv \{s_0, s_1, s_2, \ldots, s_{Q-1}\}$: the set of failure-types that can happen to the system under study,
$s_0 \equiv$ no-failure type,
$p_k(n) \equiv \Pr\{\text{failure-type } k \text{ occurred at } n\}$,
$p_0(n) \equiv \Pr\{\text{no-failure at } n\}$.

The corresponding time-dependent probability distribution is:

$$\underset{\mathbf{S}}{\Pr}(n) = (p_0(n), p_1(n), p_2(n), \ldots, p_{Q-1}(n)).$$

Because (assumption #6), at each discrete n only 1 failure-type can occur, the $p_k(n)$ satisfy:

$$\sum_{k=0}^{Q-1} p_k(n) = 1, \qquad \forall n. \tag{1}$$

Because at each n, only one ToF can occur, the Q ToFs are not s-independent. But, if a histogram over a time window of size B is used to estimate the $p_k(n)$, then (1) is automatically verified; also, the estimation of each p_k as the number of ToF_k that occurred over the time window divided by B, is equivalent to considering the estimation of the probabilities of the various ToFs independently from each other. Similarly, estimate the probabilities of each of the Q ToFs independently from each other. As a simplifying hypothesis, use assumption #7 to obtain a first estimate of $p_k(n)$. Because the procedure in this paper does not automatically satisfy (1) as the histogram procedure previously discussed, a normalization factor $\kappa(n)$ is introduced, so that the normalized estimates satisfy (1).

The idea is to consider the TBF for each failure-type, as a nonstationary stochastic process, and to estimate the $p_k(n)$, [7], [24] following a procedure similar to the one presented in [19] as explained in the remainder of this section.

If the $p_k(n)$ do not change between 2 consecutive occurrences of ToF_k, then the TBF for each failure-type follows a geometric distribution at n:

$$\begin{aligned}\Pr\{\text{TBF}_k(n) = m | \text{ToF}_k \text{ at } n\} \\ = p_k(n) \cdot ((1 - p_k(n))^{m-1}, \qquad m = 1, 2, \ldots.\end{aligned} \tag{2}$$

Hence, the mean value of the TBF for each $k = 0, 1, 2, \ldots, Q-1$, is:

$$\mathrm{E}[\text{TBF}_k(n) | \text{ToF}_k \text{ at } n] = \frac{1}{p_k(n)}; \tag{3}$$

thus providing a rationale for estimating the $p_k(n)$ as:

$$\hat{p}_k(n) = \frac{\kappa(n)}{\widehat{\text{TBF}}_k(n)}, \tag{4}$$

$\kappa(n) \equiv$ a normalization factor that insures that (1) is satisfied,
$\widehat{\text{TBF}}_k(n) \equiv$ estimate of the s-expected value of the TBF for failure type k at n.

Estimating $p_k(n)$ for each $k = 0, 1, \ldots, Q-1$, is done simultaneously using Q estimators operating independently, and then using $\kappa(n)$ to insure (1).

Section II presents the P&C-F estimation procedure. Section III applies the model to software failure data.

II. A Model for the Probability Estimation of Failures

The model in this paper is based on estimating the probabilities of having 0, 1, 2, ... failures per a given discrete unit of time, (e.g., days, weeks). To apply the statistic model in Section I, consider that ToF_0 corresponds to the event of having 0 failures in a time-unit, analogously,

$\text{ToF}_i \equiv$ event of having i failures in a time-unit, $i = 1, 2, \ldots, k$.

Hence, the failure process can be viewed as a discrete source of failure-types given by the "number of failures per time-unit," e.g., let k failures occur during time-unit n, then, ToF_k occurs at n. This definition, leads to a probability distribution:

$\Pr_S(n) = \{p_0(n), p_1(n), \ldots, p_{Q-1}(n)\}$ for the set:
$S = \{\text{ToF}_0, \text{ToF}_1, \ldots, \text{ToF}_{Q-1}\}$.

If $\Pr_S(n)$ is known, then prediction of the remaining failures (n_{rem}) is obtained by taking the s-expectation of the "number of failures per time-unit":

$$\hat{\mu}(t_{rem}) = \left(\sum_{i=1}^{Q-1} i \cdot p_i\right) \cdot t_{rem}. \tag{5}$$

In this paper, $\Pr_S(n)$ is not assumed to be known, thus $\mu(t_{rem})$ is estimated as:

$$\hat{\mu}(t_{rem}) = \left(\sum_{i=1}^{Q-1} i \cdot \hat{p}_i\right) \cdot t_{rem}. \tag{6}$$

To motivate the model in this paper, consider software-reliability. Because of its very own nature, the probabilities of failures in software reliability usually change in time (hopefully, decreasing). Hence, it is desirable that $\hat{\mu}(t_{rem})$ should not assume any particular distribution function, nor satisfy any stationarity hypothesis. Also, to be useful, the estimation method should be fast enough to be able to follow the nonstationary characteristics of the model. Thus an adaptive procedure is introduced to estimate these probabilities.

As in [19], estimation of the probabilities of each type of failure are obtained from the times between 2 consecutive occurrences of each ToF_k. Using assumptions #1 and #2, each r.v. τ_k follows a geometric distribution between two consecutive occurrences of ToF_k:

$$\begin{aligned}\Pr\{\tau_k(n) = m | \text{ToF}_k \text{ at } n\} \\ = p_k(n) \cdot (1 - p_k(n))^{m-1}, \qquad m = 1, 2, \ldots,\end{aligned} \tag{7}$$

so that:

$$\mathrm{E}\left[\tau_k(n)|\mathrm{ToF}_k \text{ at } n\right] = \frac{1}{p_k(n)},$$
$$\mathrm{Var}\left[\tau_k(n)|\mathrm{ToF}_k \text{ at } n\right] = \frac{1-p_k(n)}{p_k^2(n)}. \quad (8)$$

To obtain the estimator for $\mathrm{Pr}_S(n)$, several approaches are given.

Approach #1

$\eta_k(n) \equiv$ number of occurrences of failures of type k up to n. Use (9):

$$\widehat{\mathrm{TBF}}_k(n) = \hat{\tau}_k\left(\eta_k(n)\right) = \frac{1}{\eta_k(n)-1} \cdot \sum_{i=1}^{\eta_k(n)-1} \tau_k(i). \quad (9)$$

This estimator is updated only when ToF_k occurs. To estimate $\mathrm{Pr}_S(n)$, (4) and (8) suggest the following estimator for p_k at n:

$$\hat{p}_k(n) = \frac{\sigma'(n)}{\hat{\tau}_k(\eta_k(n))}, \qquad k = 0, 1, 2, \ldots, Q-1; \quad (10)$$

$\sigma'(n)$ is the normalization factor:

$$\sigma'(n) = \left[\sum_{j=0}^{Q-1} \left[\hat{\tau}_j\left(\eta_k(n)\right)\right]^{-1}\right]^{-1}. \quad (11)$$

For the stationary case, as (9) shows, $\hat{\tau}_k$ is an unbiased estimator for $\mathrm{E}[\tau_k]$, and its variance goes to 0 as $n \to \infty$. This method converges, at least weakly, to the probability distribution of the failure-source, S. When the source is not stationary, the main interest is in the possibility of following the changes in p_k. Approach #1 does not follow well the changes of the probability distribution due to the long-term memory of the average over all the past. The estimator (9) can be improved by taking the estimation, not over all the past, but over a sliding window that only considers the latest B occurrences of the ToF_k:

Approach #2

$$\hat{\tau}_k\left(\eta_k(n)\right) = \frac{1}{B} \cdot \sum_{i=\eta_k(n)-B}^{\eta_k(n)-1} \tau_k(i); \quad (12)$$

this estimator has finite memory, but does not adapt as fast as desired.

Approach #3

The idea is to construct an estimator similar to that of (10), but using a different estimator for TBF_k, see (4). An alternative to the sliding window procedure (12), is to introduce a coefficient α, $0 < \alpha < 1$, to produce loss of memory in the form:

$$\widehat{\mathrm{TBF}}_k(n) = \hat{\tau}_k\left(\eta_k(n)\right) = \gamma \cdot \sum_{i=1}^{\eta_k(n)-1} \alpha^{(\eta_k(n)-1)-k} \cdot \tau_k(i); \quad (13)$$

the result is:

$$\widehat{\mathrm{TBF}}_k(n) = \hat{\tau}_k(\eta_k(n)) = \alpha \cdot \hat{\tau}_k(\eta_k(n)-1) + \gamma \cdot \tau_k(\eta_k(n)-1). \quad (14)$$

The estimator for $\mathrm{TBF}_k(n)$ in (12) corresponds to a moving average filter, while (14) corresponds to a 1-pole autoregressive IIR filter. Extending this idea, the desired estimator is:

Approach #4

$$\widehat{\mathrm{TBF}}_k(n) = \hat{\tau}_k(\eta_k(n)) = \alpha \cdot \hat{\tau}_k(\eta_k(n)-1) + \beta \cdot \hat{\tau}_k(\eta_k(n)-2)) + \gamma \cdot \tau_k(\eta_k(n)-1). \quad (15)$$

This estimator (15) corresponds to a 2-pole autoregressive IIR filter. The coefficients should be taken so that $\alpha+\beta+\gamma = 1$ to obtain an unbiased estimator for the stationary case:

$$\mathrm{E}[\hat{\tau}_k(\eta_k(n))] = \mathrm{E}[\tau_k(\eta_k(n))] = \frac{1}{p_k(n)}; \quad (16)$$

the α, β, γ must be selected so that the filter is stable. As in Approach #1 [see (10)], see also (4), the P&C-F estimator for $p_k(n)$ at every n is:

$$\hat{p}_k(n) = \frac{\sigma(n)}{\hat{\tau}_k(\eta_k(n))}, \quad (17)$$

$\widehat{\mathrm{TBF}}_k$ is now given by (15), and $\sigma(n)$ is the normalization factor that insures that the estimated probabilities add up to one. Equation (15) shows that $\widehat{\mathrm{TBF}}_k(n)$ is updated only when ToF_k occurs, and then, only during the period of time while a ToF_k does not appear, the denominator of (17) does not change, but the numerator does change, because some other ToF occurred. Hence, (17) estimates $p_k(n)$ at every instant n.

Setting adequate values for the parameters α, β, γ involves a trade-off between the speed of convergence and the variance of the estimators. As is usually the case with algorithms that must adapt to nonstationary environments, this trade-off depends on the degree-of-stationarity of the source [12]. Other filter-types can be used. The 2-pole IIR filter (15) is an illustrative example.

Another issue that must be addressed is that the estimator for TBF_k is updated only when a ToF_k occurs. If $\tau_k(j)$ is uncorrelated in j for each k, then, solving the Yule–Walker equations [12] for the AR process (15), and using (8), the variance of $\hat{\tau}_k(\eta_k(n))$ is:

$$\mathrm{Var}[\hat{\tau}_k(\eta_k(n))] = \frac{1-\alpha-\beta}{1+\alpha+\beta+\frac{2\alpha\cdot\beta}{1-\beta}} \cdot \mathrm{Var}[\tau_k] = \frac{1-\alpha-\beta}{1+\alpha+\beta+\frac{2\alpha\beta}{1-\beta}} \cdot \frac{1-p_k(n)}{p_k^2(n)}. \quad (18)$$

This result (18) shows that the estimator has large variance for those ToF which have very low probability. Furthermore, the updating rule implies that a ToF that previously had nonzero probability and then has zero probability will never be updated. This is not a difficulty when the source is stationary, but when dealing with nonstationary sources, the situation in which a ToF decreases its probability to low values or zero as time elapses should be considered. To address this problem, (15) is modified.

Approach #5

$T_k(n)$ is a saw tooth like process; it begins counting from 0 at the occurrence of ToF_k, and then increases until ToF_k occurs again, thus resetting to 0, and begins counting again.

With this modification, the final form of $\hat{p}_k(n)$ is:

$$\hat{p}_k(n) = \frac{\sigma(n)}{\alpha \cdot \hat{\tau}_k(\eta_k(n)-1) + \beta \cdot \hat{\tau}_k(\eta_k(n)-2) + \gamma \cdot T_k(n)}; \quad (19)$$

$\sigma(n) \equiv$ the normalization factor, $\hat{\tau}_k(\eta_k(n)-1)$ and $\hat{\tau}_k(\eta_k(n)-2)$ are recursively obtained using (15).

In (19), the estimator for TBF_k is, see (4):

$$\widehat{\mathrm{TBF}}_k(n) = \alpha \cdot \hat{\tau}_k(\eta_k(n)-1) + \beta \cdot \hat{\tau}_k(\eta_k(n)-2) + \gamma \cdot T_k(n). \quad (20)$$

At the n previous to the one in which ToF_k occurs, $T_k(n) = \tau(\eta_k(n))$. The $T_k(n)$ in (19) is introduced to let $\hat{p}_k(n)$ decrease if the ToF_k does not occur; $T_k(n)$ produces jumps in $\hat{p}_k(n)$, and then, this term introduces "noise" in the estimation of those ToF with high probabilities. To deal with this "noisy" estimation of the ToF with high probability, the α, β, γ must be adequately selected, and this can be a delicate matter.

Finally, using (19), the mean value for the "number of failures remaining" at every n can be estimated from (6).

III. An Application to Software Failures Data

Additional Acronyms and Abbreviations

CPGEO	compound Poisson—geometric (model)
CPPTZ	compound Poisson—compounded by a PTZ (model)
CRE	chains of rare-events (model)
G–O	Goel–Okumoto (model)
LS	least squares (method)
M–O	Musa–Okumoto (model)
MLE	maximum likelihood estimation (method)
NHPP	nonhomogeneous Poisson process
PTZ	Poisson truncated at zero.

Additional Notation

t_{tot}	total test time
t_{past}	elapsed test time
n_{tot}	total number of failures
n_{past}	past number of failures.

Software reliability is being increasingly studied [9], [10], [15]–[18], [21]–[23], [25], [26]. Several measurements of software failures production have been reported, and probabilistic models for software failures prediction have been P&C-F. Several models are based on the assumption that the statistics of the failure process are known. Some of these models assume a Poisson process or NHPP.

The estimator in this paper has 2 important characteristics:

1) It adapts faster than other methods;

2) It does not assume *a-priori* probability distribution functions. This section compares the method in this paper with some other models for software reliability.

Some of the best-known models include the G–O [9], and the M–O (also known as logarithmic) [17]; both of these models are based on NHPP. These 2 models can be applied for TBF data, as well as grouped failures in interval-times data. However, the measurement of occurrences of grouped-failures requires less effort. Several software-failures production data are given as grouped failures [15], [22], [25]. To model grouped-failures production, a CPGEO was introduced [21].

Models usually P&C-F for software reliability, predict the number of failures which are produced in a given interval test-time; these are known as growth-models. Some of the best known growth models are NHPP, where the Poisson parameter is a given function of time [15], [17]; for example, in G–O [9]:

$$\mu(t) = a \cdot [1 - \exp(-b \cdot t], \qquad a \geq 0,\, b > 0; \quad (21)$$

in M–O [17]:

$$\mu(t) = \frac{1}{\theta} \cdot \log(\lambda \cdot \theta \cdot t + 1). \quad (22)$$

The CPPTZ [3] is based on an extension [2] of the CRE [5]. As shown in [2], the CRE is equivalent to a CP, with a PTZ as the compounding distribution. The CPPTZ is a 2-parameter model, 1 for the CP, and 1 for the PTZ [2], [3]. The CP parameter can be directly estimated using MLE, or the moments method [2], [3]. For estimating the PTZ parameter, several methods can be considered, e.g., [11]. In [3], to adapt better to data changes, a Mode estimator was introduced for the PTZ. Also in [3], the unbiased Plackett estimator, e.g., [11] is also considered. Results presented here for the CPPTZ model are based on [2], [3].

A characteristic usually applied to compare software reliability models is their predictive validity [17], [26]. Let n_{tot} failures have been produced in time t_{tot}. The failure data produced up to time $t_{\mathrm{past}} \leq t_{\mathrm{tot}}$ are used to estimate the parameters of the mean value function so that

$$\mu(t_{\mathrm{past}}) = n_{\mathrm{past}}.$$

Then, replacing the estimated values of the parameters in $\mu(t)$, an estimate of the number of failures up to t_{tot} is obtained as $\hat{\mu}(t_{\mathrm{tot}})$.

If $\mu(t)$ is proportional to t (as happens in many models) the remaining number of failures can be estimated as:

$$\begin{aligned} n_{rem} &= \hat{\mu}(t_{\mathrm{tot}}) - n_{\mathrm{past}} \\ &= \hat{\mu}(t_{rem} + t_{\mathrm{past}}) - \hat{\mu}(t_{\mathrm{past}}) = \hat{\mu}(t_{rem}). \end{aligned} \quad (23)$$

Now, compare the results obtained using NHPP, CPGEO, CPPTZ models, and the method in Section II, using the following data:

- T5 from [18] grouped by day,
- DS1 from [22],
- J5, Data 7, and Data 8, from [15],

extending results in [20]. For the NHPP model, M–O or G–O are used, as indicated in the figure captions. The results obtained using both models are very similar. The MLE for the NHPP model was used when possible, and LS otherwise. Conditions for the convergence of MLE in NHPP models are in [13].

Software failures data can be classified according to the shape of the CF curve. Generally, it is concave, showing a decreasing failures production rate; or it looks like an S-shaped curve with an inflection point; or it shows a 2-stage system. Next, the predictive validity of the published models (discussed in this paper)

TABLE II
DATA SET FOR DATA 7

CF	Day	CF	Day	CF	Day	CF	Day
4	107	186	84	374	62	494	35
11	105	193	83	379	61	496	34
21	104	200	82	386	60	497	33
34	103	205	81	393	59	508	32
42	102	212	80	407	58	509	31
55	101	218	79	420	57	511	29
59	100	224	78	434	56	513	28
66	99	228	77	445	55	517	27
74	98	240	76	447	54	518	26
75	97	246	75	451	53	522	24
81	96	253	74	455	52	523	23
94	95	261	73	458	51	524	22
101	94	272	72	464	50	526	20
110	93	278	71	470	49	527	17
118	92	287	70	473	48	528	16
123	91	294	69	476	45	529	11
133	90	306	68	480	43	530	9
140	89	318	67	481	41	532	4
151	88	333	66	483	40	533	2
156	87	347	65	484	38	535	0
164	86	354	64	486	37		
177	85	363	63	491	36		

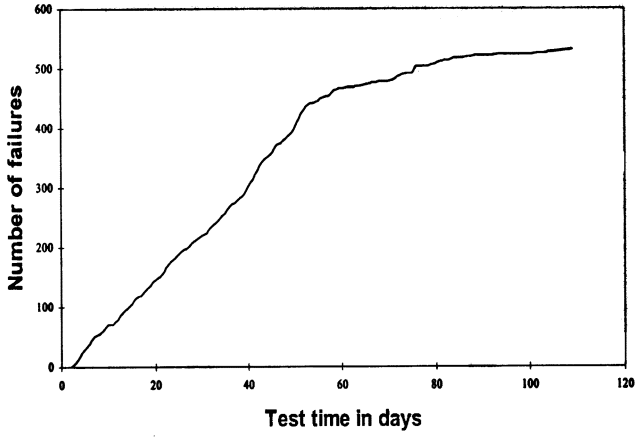

Fig. 2. CF *versus* test-time (days) for Data 7 [15].

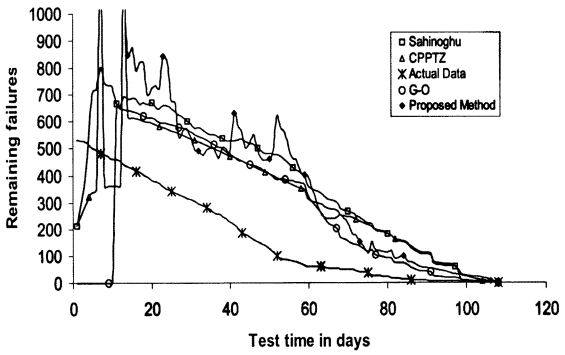

Fig. 3. Actual and predicted remaining failures *versus* time (days) for Data 7.

are evaluated, as well as that of the P&C-F model. Failures predictions are made in the same units of time as they were measured.

Data 7 CF are shown in Table II and Fig. 2. They have an almost constant rate up to day 60; after that, the mean slope decreases abruptly. Predicted values for the 4 models are shown in Fig. 3. Up to day 60, all the models predict similar remaining

TABLE III
DATA SET FOR DATA 8

CF	Day	CF	Day	CF	Day	CF	Day
5	108	211	89	346	69	460	49
10	107	217	88	367	68	463	48
15	106	230	86	375	67	464	46
20	105	234	85	381	66	465	44
26	104	236	84	401	65	466	41
34	103	240	83	411	64	467	40
36	102	243	82	414	63	468	37
43	101	252	81	417	62	469	36
47	100	254	80	425	61	470	32
49	99	259	79	430	60	472	31
80	98	263	78	431	59	473	29
84	97	264	77	433	58	475	22
108	96	268	76	435	57	476	13
157	95	271	75	437	56	477	9
171	94	277	74	444	55	478	6
183	93	290	73	446	54	479	3
191	92	309	72	448	52	480	0
200	91	324	71	451	51		
204	90	331	70	453	50		

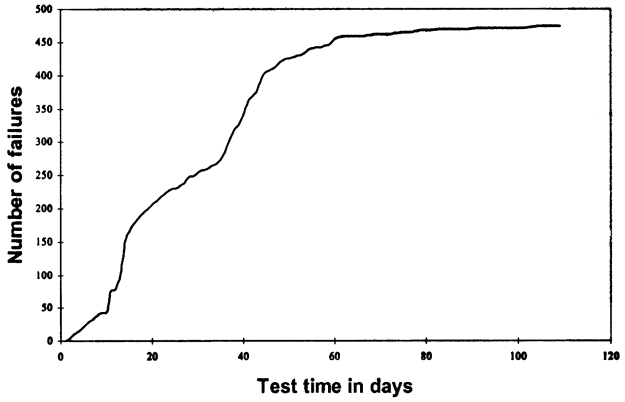

Fig. 4. CF *versus* test time (days) for Data 8 [15].

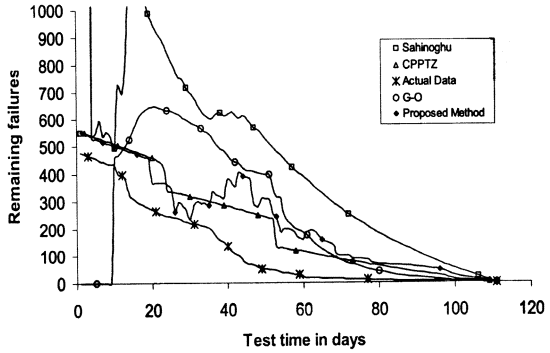

Fig. 5. Actual and predicted remaining failures *versus* time (days) for Data 8.

failures. After that, G–O and the P&C-F model give the best results. The MLE method has been used, except for the G–O model for which the LS method was applied up to day 59.

Data 8 set has the 3 characteristics mentioned in the previous paragraph. It is an S-shaped system (see Fig. 5) with an inflection-point near day 18, joined with a second simple stage beginning around day 40; see Table III and Fig. 4. As shown in Fig. 5, CPPTZ as well a P&CF have the better fit up to day 70. The MLE method was used, except for the G–O model for which the LS method was applied up to day 50. From day 70, the results obtained are similar to those of the G–O model.

TABLE IV
DATA SET FOR DATA DS1

CF	Day	CF	Day	CF	Day	CF	Day
6	163	670	122	1511	84	1957	39
14	161	674	121	1549	83	1968	38
21	160	678	120	1579	82	1970	37
51	159	702	119	1602	81	1975	36
67	158	726	118	1637	80	1988	35
89	157	742	117	1638	76	2000	34
90	156	762	116	1648	75	2010	33
120	155	773	115	1649	74	2021	32
155	154	780	114	1653	73	2025	31
172	153	797	113	1656	71	2027	30
183	152	838	112	1661	69	2028	29
200	151	862	111	1681	68	2037	28
229	150	876	110	1695	67	2042	27
231	149	893	109	1703	66	2045	26
246	147	911	108	1704	65	2048	25
264	146	918	107	1712	63	2060	24
279	145	921	106	1725	62	2064	21
297	144	940	105	1749	61	2081	20
342	143	990	104	1766	60	2088	19
348	142	1041	103	1771	59	2090	18
350	141	1080	102	1773	58	2094	17
361	140	1121	101	1777	57	2096	16
385	139	1141	100	1792	56	2100	15
404	138	1142	99	1799	55	2104	14
426	137	1208	98	1812	54	2108	13
441	136	1247	97	1820	53	2119	12
446	134	1268	96	1826	52	2145	11
464	133	1283	95	1828	51	2163	10
474	132	1318	94	1841	49	2171	9
479	131	1325	93	1858	48	2186	7
503	130	1331	92	1865	47	2196	6
535	129	1362	91	1877	46	2201	5
542	128	1377	90	1881	45	2207	4
549	127	1400	89	1883	44	2212	3
571	126	1419	88	1902	43	2216	2
600	125	1452	87	1910	42	2218	0
628	124	1459	86	1925	41		
646	123	1480	85	1936	40		

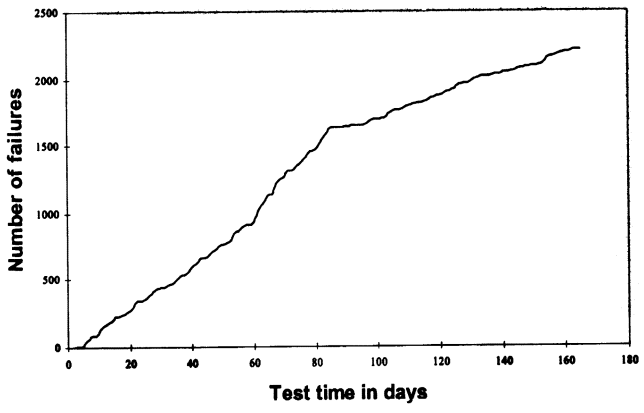

Fig. 6. CF *versus* test time (days) for data DS1 [22].

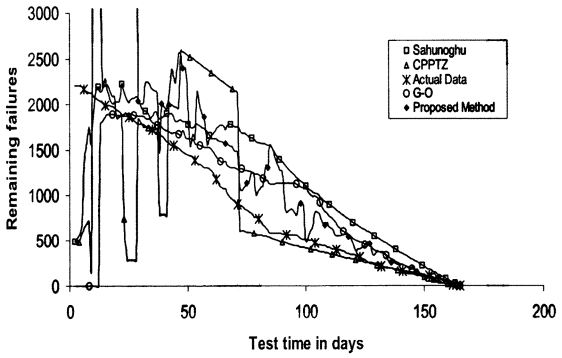

Fig. 7. Actual and predicted remaining failures *versus* time (days) for data DS1.

TABLE V
DATA SET FOR DATA J5

CF	Day	CF	Day	CF	Day	CF	Day
2	72	124	53	199	35	295	17
4	70	126	52	204	34	300	16
7	69	129	51	205	33	303	15
10	68	131	50	209	32	307	14
16	67	134	49	217	31	315	13
24	66	142	48	220	30	318	12
32	65	148	47	222	29	322	11
44	64	155	46	228	28	327	10
54	63	163	45	241	27	333	9
60	62	165	44	250	26	337	7
65	61	168	43	256	25	342	6
69	60	172	42	263	24	346	5
75	59	175	41	266	23	351	4
85	58	178	40	269	22	356	3
91	57	182	39	273	21	361	2
98	56	186	38	278	20	364	1
108	55	191	37	284	19	367	0
118	54	195	36	290	18		

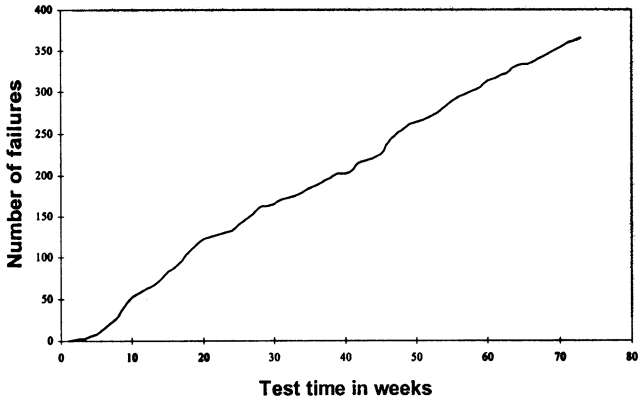

Fig. 8. CF *versus* test time (weeks) for data J5 [15].

Data DS1 [22] are shown in Table IV and Fig. 6. The CF is a concave curve with an inflection point near day 90. The MLE method was used, except for the G–O model for which the LS method was used up to day 100. The only model that follows the inflection point is G–O, as shown in Fig. 7. However, CPPTZ and the P&C-F model are better between day 80 and day 140.

CF from Data J5 are in Table V and Fig. 8. The curve shows an almost constant failures rate. The prediction curves in Fig. 9 show that the closest fit corresponds to the CPGEO and M–O models. They give similar results from day 48. Results obtained using the P&C-F model are similar to these models. The main characteristic for this case is that when the other models prediction is higher than the real data, the P&C-F model crosses the real data-value several times, resulting in a better prediction for some intervals. For this example, the MLE method was used, except for the G–O model for which the LS method was used up to week 31.

Failures data T5 [18] grouped by day are shown in Table VI and Fig. 10. They look like a 2-stage system with an inflection point after day 200 due to design changes, as mentioned in [18]. In this case, according to the theorem in [13], MLE can be applied only between arrival days 59 and 280. Otherwise, it

TABLE VI
DATA SET FOR DATA T5

CF	Day	CF	Day	CF	Day	CF	Day	CF	Day	CF	Day	CF	Day	CF	Day
1	430	123	387	230	332	289	249	389	194	510	153	606	110	750	62
4	429	124	386	233	330	290	247	395	193	511	152	608	109	751	61
8	428	125	385	235	327	293	246	399	192	516	151	609	108	752	59
9	426	127	383	236	323	294	245	406	191	518	150	612	107	753	58
10	425	131	382	237	322	301	244	411	190	521	149	614	106	756	57
12	424	136	381	241	321	303	243	415	189	524	148	615	105	760	56
15	423	139	380	242	320	304	242	420	188	525	146	616	104	761	53
18	422	143	379	245	319	305	241	427	187	532	145	617	102	767	52
23	421	148	378	246	317	306	237	432	186	534	144	618	101	772	51
27	420	150	377	247	315	311	233	434	185	540	143	621	100	773	50
29	419	153	376	250	314	314	231	435	183	543	142	622	98	775	49
31	418	154	373	251	311	317	229	437	182	545	141	626	97	776	48
36	417	155	372	252	310	318	227	439	180	546	140	627	96	779	47
40	416	157	370	253	309	324	226	441	179	549	139	633	95	780	45
42	415	159	369	254	306	329	225	444	178	551	138	637	94	781	42
49	414	160	368	255	303	333	224	449	177	560	137	648	93	784	41
52	413	161	365	256	301	338	223	451	176	562	136	651	92	785	40
59	412	163	364	257	300	340	221	452	175	565	135	658	91	787	39
63	410	166	363	262	298	342	220	459	174	567	133	663	90	788	38
70	409	167	362	263	297	345	219	463	172	568	132	667	89	790	37
71	408	168	360	264	293	346	218	465	171	570	131	675	86	791	33
75	406	170	359	265	292	352	217	468	170	571	130	676	85	792	32
78	403	171	358	267	291	355	215	469	169	573	127	682	84	793	31
79	402	172	357	269	290	359	214	471	167	577	126	687	83	794	30
80	400	174	353	270	287	361	213	474	166	578	125	693	79	795	27
83	399	180	348	271	279	365	210	478	165	581	123	703	78	798	23
86	397	183	344	272	278	366	208	485	164	582	122	712	77	799	20
93	396	190	342	274	276	367	207	488	163	583	121	716	73	800	18
99	395	192	341	275	275	368	204	490	162	586	120	731	71	801	12
103	394	203	340	276	273	369	203	494	160	591	119	736	70	802	11
112	393	207	339	277	269	371	201	496	159	592	118	740	69	811	8
113	392	216	338	280	268	375	200	497	158	593	117	741	68	813	6
114	391	218	337	282	264	378	199	501	157	595	115	742	67	819	5
116	390	219	336	285	263	379	198	504	156	597	113	745	66	824	4
117	389	221	335	286	261	385	197	505	155	600	112	746	65	826	3
119	388	226	333	288	259	386	196	507	154	602	111	748	64	827	2

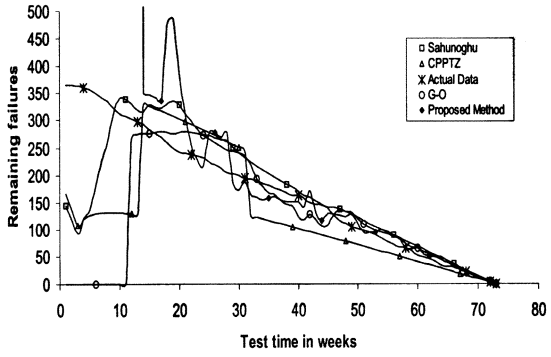

Fig. 9. Actual and predicted remaining failures *versus* time (weeks) for data J5.

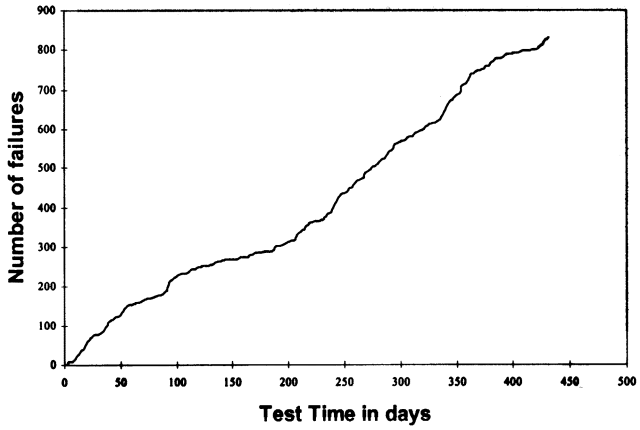

Fig. 10. CF *versus* test time (days) for data T5 [18].

is not possible to find any reasonable good fit using LS because the predicted total number of failures is lower than the number of failures at t_{past} for days beyond 280, resulting in negative

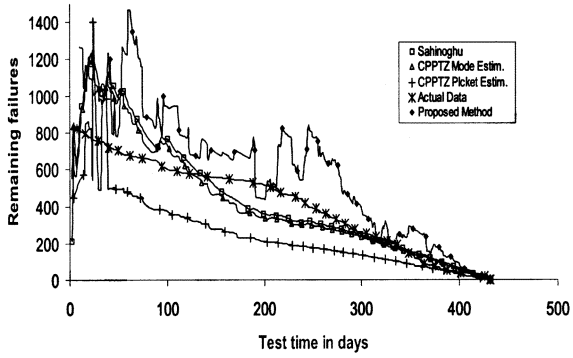

Fig. 11. Actual and remaining failures *versus* time (days) for data T5.

remaining failures. Therefore, this system cannot be treated as having a single stage using either M–O or G–O, and then, NHPP model results are not shown for these data. CPPTZ and CPGEO give the closest results, though the P&C-F model follows the shape of the actual data better as shown in Fig. 11. For these data, there is little grouping of failures, because there is 1 failure per day for the majority of time.

As seen from the examples, the P&C-F estimator is very noisy. Part of this noise is due to the $T_k(n)$ term, which, as described in Section II, introduces jumps in the estimations. Although the prediction can be further filtered to obtain a smoother estimator closer to the real data, here, it is presented in the form (19) to show its main characteristics.

The examples show the advantage of the P&C-F model in the sense that it permits modeling very different situations with a unified framework. To evaluate the performance of the P&C-F model, it is important to consider that there were no assumptions about probability distribution or number of stages.

From the analysis in this paper, the P&C-F model shows good performance compared the other models, even though it might be improved by selecting other filters. The good capability to follow the changes of the real data can be seen from the figures.

ACKNOWLEDGMENT

The authors are pleased to thank 2 anonymous reviewers for their very helpful comments.

REFERENCES

[1] N. R. Barraza, B. Cernuschi-Frías, and F. Cernuschi, "A probabilistic model for grouped events analysis," in *Proc. 1995 IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 4, Oct. 1995, pp. 3386–3390.

[2] ——, "Applications and extensions of the chains-of-rare-events model," *IEEE Trans. Reliability*, vol. 45, pp. 417–421, Sep. 1996.

[3] N. R. Barraza, J. D. Pfefferman, B. Cernuschi-Frías, and F. Cernuschi, "An application of the chains-of-rare-events model to software development failure prediction," in *Proc. 5th Int. Conf. Reliable Software Technologies*. ser. Lecture Notes in Computer Science, H. B. Keller and E. Plödereder, Eds: Springer-Verlag, 2000, vol. 1845, pp. 185–195.

[4] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*: Springer-Verlag, 1990.

[5] F. Cernuschi and L. Castagnetto, "Chains of rare events," *Annals of Mathematical Statistics*, vol. XVII, pp. 53–61, Mar. 1946.

[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*: John Wiley & Sons, 1973.

[7] D. L. Duttweiler and C. Chamzas, "Probability estimation in arithmetic and adaptive-Huffman entropy coders," *IEEE Trans. Image Processing*, vol. 4, no. 3, pp. 237–246, Mar. 1995.

[8] P. Elias, "Universal codeword sets and representation of the integers," *IEEE Trans. Information Theory*, vol. 21, no. 2, pp. 194–203, Mar. 1975.

[9] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Trans. Reliability*, vol. 28, pp. 206–211, Aug. 1979.

[10] K. Goseva-Popstojanova and K. S. Triverdi, "Failure correlation in software reliability models," *IEEE Trans. Reliability*, vol. 49, no. 1, pp. 37–48, Mar. 2000.

[11] F. A. Haight, *Handbook of the Poisson Distribution*: John Wiley & Sons, 1966.

[12] S. Haykin, *Adaptive Filter Theory*: Prentice Hall, 1991.

[13] S. A. Hossain and R. C. Dahiya, "Estimating the parameters of a nonhomogeneous Poisson-process model for software reliability," *IEEE Trans. Reliability*, vol. 42, pp. 604–612, Dec. 1993.

[14] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, Sep. 1952.

[15] M. R. Lyu, Ed., *Handbook of Software Reliability Engineering*: McGraw Hill, 1996.

[16] A. M. B. Miller, "A study of the Musa reliability model," M.Sc. thesis, University of Maryland, Nov. 1980.

[17] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*: McGraw-Hill, 1987.

[18] J. D. Musa, *Software Reliability Data*: Bell Telephone Laboratories, 1980.

[19] J. D. Pfefferman, H. J. González, and B. Cernuschi-Frías, "On the estimation of the probability distribution of a nonstationary source for lossless data compression," in *Proc. IEEE 1997 Int. Conf. Image Processing*, vol. II, ICIP-1997, pp. 270–273.

[20] J. D. Pfefferman and B. Cernuschi-Frías, "A nonstationary model for time-dependent software reliability analysis," in *Proc. 1999 IASTED Int. Conf. Modeling and Simulation*, MS'1999, pp. 427–431.

[21] M. Sahinoglu, "Compound-Poisson software reliability model," *IEEE Trans. Software Engineering*, vol. 18, pp. 624–630, Jul. 1992.

[22] A. N. Sukert, "A software reliability modeling study,", Technical Report RADC-TR-76-247, Rome Air Development Center, 1976.

[23] ——, "Empirical validation of three error prediction models," *IEEE Trans. Reliability*, vol. 28, pp. 199–205, Aug. 1979.

[24] F. Thomson Leighton and R. L. Rivest, "Estimating a probability using finite memory," *IEEE Trans. Information Theory*, vol. 32, no. 6, pp. 733–742, Nov. 1986.

[25] A. Wood, "Software reliability growth models,", Tandem Tech. Report 96.1, Sep. 1996.

[26] ——, "Predicting software reliability," *IEEE Computer*, vol. 29, pp. 69–77, Nov. 1996.

Jonas D. Pfefferman received his degree in 1993 in electrical engineering from the University of Buenos Aires, Argentina. He is working toward his Ph.D. at the University of Buenos Aires. His research interests include statistical modeling, in particular probability models for nonstationary environments, and signal and image processing.

Bruno Cernuschi-Frías received his degree in 1977 in electrical engineering from the University of Buenos Aires, Argentina; and his M.Sc. and Ph.D. in electrical engineering in 1983 and 1984, respectively, from Brown University, USA. He is Full Professor in the Department of Electronics at the Faculty of Engineering, University of Buenos Aires, and Principal Researcher of the Consejo Nacional de Investigaciones Científicas y Técnicas, (CONICET), Argentina. His research interests include statistical modeling, signal and image processing, information theory and statistical thermodynamics.