PARAMETER ESTIMATIONS FOR AVAILABILITY GROWTH

Z. Bluvband & S. Porotsky

•

A.L.D. , Tel - Aviv, Israel

e-mail: zigmund@ald.co.il, sergey@ald.co.il

ABSTRACT

The reliability growth process applied to a complex system undergoing development and field test involves surfacing failure modes, analyzing the modes, and, in addition to repair, in some cases implementing corrective actions to the surfaced modes. In such a manner, the system configuration is matured with respect to reliability. The conventional procedure of reliability growth implies evaluation of two principal parameters of the NHPP process only for failure rate. Since standard NHPP does not take into account parameters of repairs, it is necessary to develop expanded procedure as the basis for the Availability Growth. It implies evaluation of both: a) the parameters of failure rate and, b) the parameters of repair rate. Authors suggest a model and numerical method to search these parameters.

1. INTRODUCTION

Accurate prediction and control of reliability plays an important role in the profitability and competitive advantage of a product. Service costs for products within the warranty period or under a service contract are a major expense and a significant pricing factor. Proper spare part stocking and support personnel hiring and training also depend upon good reliability fallout predictions. On the other hand, missing reliability targets may invoke contractual penalties and cost future business.

Telecommunication networks, oil platforms, chemical plants and airplanes consist of a great number of subsystems and components that are all subject to failures. Reliability theory studies the failure behavior of such systems in relation to the failure behavior of their components, which often isn't easier to analyze. There are multiple failure analysis methods in the design and development phase, like FMECA (Failure Mode Effect and Criticality Analysis), FTA (Fault Tree Analysis), ETA (Event Tree Analysis), BFA (Bouncing Failure Analysis), Markov chains, etc. The Analysis of failures, faults and errors from the field (Manufacturing, Test, Operation and Support) is usually performed by FRACAS (Failure Reporting, Analysis and Corrective Action System) using the investigation of the Physical nature of failures and studying the possible causes and roots of the Failures.

Typical task of Reliability Analysis is Reliability Growth Analysis, which deals with failures in the repairable systems. A repairable system is one which can be restored to satisfactory operation by any action, including parts replacements or changes to adjustable settings. When discussing the rate at which failures occur during system operation time (and are then repaired) we will define a Rate Of Occurrence Of Failure (ROCF) or "repair rate".

For systems with repairable failures the standard model is NHPP – Non-Homogeneous Random Poison Process. According this model Amount of Failures into small interval

[T; T + t] is equaled for Rate(T)t. For NHPP Power Law (Crow model, AMSAA model) it is assumed, that

$$Rate(T) = \lambda\beta T^{(\beta-1)}$$

i.e. first failure is according Weibull Distribution, λ and β are Power Law parameters.

For any NHPP process with intensity function Rate(T), the distribution function (CDF) for the inter-arrival time t to the next failure, given a failure just occurred at time T, is given by

$$F(t) = 1 - \exp\left(- \int_0^t R(T+t)dt \right)$$

In particular, for the Power Law the waiting time to the next failure, given a failure at time T, has probability density function (PDF)

$$f(t) = \lambda\beta (T+t)^{\beta-1} \exp\left(- \lambda\left((T+t)^{\beta} - T^{\beta} \right) \right)$$

This NHPP Power Law model really is same as Duane model, for which is assumed, that

$$MTBF_{cumue} = \gamma (t - \delta)^{\alpha}$$

where γ and α are Duane model parameters.

Following expressions are right:

$$\gamma = \frac{1}{\lambda} \quad , \quad \alpha = 1 - \beta$$

Below all models are for NHPP Power Law parameters search, parameters of corresponding Duane model are recalculated according above expressions.

During analysis systems with repairable failures, two main problems are solved:

- Definition of NHPP distribution parameters by means of statistics of failures
- Forecasting of some output criteria (Amount of failures on some period, MTBF, etc.) based on obtained parameters.

This classical task of Reliability Growth Analysis physically may be extended for the Availability Growth Analysis, which assumes, that repairable failures and its restoration are performed due to two factors – failure rate and repair rate [1]. For this task we have to define parameters of "mixed" flows – failures and repairs – instead of single ("continuous") flow for standard NHPP task.

The rest of the article is organized as follows. Availability Growth model as extension of Reliability Growth model is introduced in Chapter 2. First we consider simplest case – single system. Various techniques to solve this model are considered in Chapter 3. In Chapter 4 we present how the Cross-Entropy method can be applied to search parameters of proposed model. The more challenging tasks of Availability Growth are tackled in Chapter 5. In Chapter 6, we show how to get some output estimations of Availability Growth.

2. DEFINITION OF DISTRIBUTION PARAMETERS FOR SINGLE SYSTEM

First consider case of single system. Input statistics of failures and repairs is following: TF[1], TR[1],…, TF[i], TR[i],…, TF[n], TR[n], where

- n is amount of failures
- TF[i] is time of failure number i (failure arrival time – FAT)
- TR[i] is time of finishing of repair number i, i = 1…n

We assume, that both flow of failure and flow of repairs are NHPP processes. So,

$$MTBF(t) = \frac{t^{(1-\beta_f)}}{\lambda_f}$$ – for failure flow

$$MTTR(t) = \frac{t^{(1-\beta_r)}}{\lambda_r}$$ – for repair flow

We have to define parameters λ_f, β_f, λ_r, β_r and for this purpose we will use MLE (Maximum Likelihood Estimations) approach.

Comment. Generally speaking, we can describe failure and/or repair flows by means of some other NHPP Law (e.g. Exponential Law of ROCOF), but usually NHPP Power Law is used.

To define these parameters for flow of failures, we have to consider two different cases:

- Rate of Failures doesn't change during repair.

In this case the deterioration (or reliability growth) of the system during repair is absent (i.e. during repair the failure rate of tire isn't increased, because really it isn't according time, rather according miles). For this case the classical exact Crow formulas [2] are applicable:

$$\beta_f = \frac{n}{\left(n\log(Z[n]) - \sum_{i=1}^{n}\log(Z[i])\right)} \quad , \quad \lambda_f = \frac{n}{\left(Z[n]^{\beta_f}\right)}$$

where Z[i] are "shifted" failure arrivals times and last measurement time (without influence of repair time):

Z[1] = TF[1], Z[i + 1] = Z[i] + (TF[i + 1] – TR[i])

- Rate of Failures changes during repair as usually.

In this case the deterioration of the system during repair is normal (i.e. during repair the failure rate of car is increased according time). For this case the classical Crow formulas are not applicable. Conditional PDF, that i-th failure will be at moment TF[i] in condition, that (i-1)-th repair has finished at moment TR[i-1], is

$$P_f[i] = \lambda_f \beta_f \left(TF[i]^{(\beta_f - 1)}\right)\exp\left(-\lambda_f\left(TF[i]^{\beta_f} - TR[i-1]^{\beta_f}\right)\right) \quad (1)$$

Comment. In this expression for i = 1 we use TR[0] = 0.

$$Negative\ \ Logarifm\ \ Liklihood_{\ f} = -\sum_{i=1}^{n} \log(P_f[i]) \tag{2}$$

Our goal is to search values of λ_f and β_f such, that *Negative Logarifm Likelihood*$_f$ will be minimum.

To define required parameters for flow of repairs, we have to consider only one case – Rate of Repairs changes during repair and non-repair without differences. Formulas will be same, as above. Conditional PDF., that i-th repair will finish at moment TR[i] in condition, that i-th failure was at moment TF[i], is

$$P_r[i] = \lambda_r \beta_r \left(TF[i]^{(\beta_r - 1)}\right) \exp\left(-\lambda_r \left(TR[i]^{\beta_r} - TF[i]^{\beta_r}\right)\right) \tag{3}$$

$$Negative\ \ Logarifm\ \ Liklihood_{\ r} = -\sum_{i=1}^{n} \log(P_r[i]) \tag{4}$$

Our goal is to search values of λ_r and β_r such, that *Negative Logarifm Likelihood*$_r$ will be minimum.

3. COMPARISON OF DIFFERENT GLOBAL OPTIMIZATION APPROACHES

Global Optimization of non-linear function is a common task of a lot of practical problems (supply optimization, text categorization, distribution parameters estimation, etc. and etc.). For example, concerning problem of Parameters Estimation, a Linear Regression model can support only a few cases. It couldn't be used for interval and multiplied censored data, for 3 parameter Weibull estimation, Duane model with multiple systems, Gompertz model, etc. and etc. For this numerous cases we have to search distribution parameters by means of non-linear and non-convex, global optimization – both for MLE and non-linear regression using.

Our task is to search value of **Z**, which provides min G(**Z**)
under constraints $Low_j <= z[j] <= High_j$, j = 1…K, where:

- Z = {z[1],…,z[j],…z[K]} is a set (vector) of parameters
- K is amount of parameters
- Low_j is Low Boundary of Parameter j value (j = 1…K)
- $High_j$ is High Boundary of Parameter j value (j = 1…K)
- G is some Goal Function (analytical-form or, perhaps, table or even algorithm-calculated-form), dependent of vector Z.

To solve this task, two different approaches can be used:

- To write and transform derivatives of Goal Function (e.g., Logarifm-Likelihood for MLE method, Sum of Leased Squares for Non-Linear Regression method, etc.) for each single task, to solve system of non-linear equations, corresponding these situations, to support Global Minimum finding (instead of possible local minimum finding) by means of convex/concave check, etc.
- To use "direct search methods", provided universal search of Global Minimum (without analytical definition of derivatives).

For first approach using we have to define complex analytical expressions for derivatives for each single task. Early usually this approach was used and for each single task it required additional resources both for algorithm developing and software implementation. For example, Quasi-Newton method minimizes the Negative Logarifm Likelihood Function in order to bring partial derivatives to zero. Perhaps, it isn't very hard for simple cases, but for more complex models this approach requires essential additional time.

We propose to use second (universal) approach, which will allow us to search optimal solution not only for single task, but rather for all same situations (LogNormal, Gamma and other distributions, MLE for repairable failures, Non Linear Regression for Gompertz model, etc.), and, generally speaking – for all complex non-convex, multi-extremal optimization tasks. In differ of "derivative" oriented algorithms, the proposed approach will require only one implementation.

For second approach there are developed a lot of methods, based on gradient (or, if a goal function hasn't gradient – on pseudo-gradient) calculation and analysis. But for many real tasks the Goal Function isn't convex, it has many Local Minimums. In these cases such approaches require to know initial point of search, which has to be not far from optimal solution. In such optimization algorithms the initial guesses for the parameters are very crucial. But really we often don't know some information to define this initial point. So, it is impossible to use regular methods (gradients-based).

For Global Optimization Task we propose to use one of the RANDOM SEARCH oriented methods – Cross-Entropy Optimization [3]. It is relatively new random-search oriented approach (for comparison with Genetic Algorithm, implemented as Toolbox on Matlab, or Simulated Annealing Algorithm), but it has provided very good results for several analogous tasks.

4. SHORT DESCRIPTION OF CROSS-ENTROPY ALGORITHM

The method derives its name from the cross-entropy (or Kullback-Leibler) distance - a well known measure of "information", which has been successfully employed in diverse fields of engineering and science, and in particular in neural computation, for about half a century. Initially the Cross-Entropy method was developed for discrete optimization [3], but later was successfully extended for continuous optimization [4]. The Cross-Entropy method is an iterative method, which involves the following two phases:

– Generation of a sample of random data. Size of this data is 500…5000 random vectors of each algorithm steps, amount of steps is 50…100. Generation is performed according to a specified random mechanism.

– Updating the parameters of the random mechanism, on the basis of the data, in order to produce a 'better" sample in the next iteration. Choice of these parameters is performed by means of maximization of Cross-Entropy function. This optimization is performed on the each algorithm step, but in differ on global optimization usually this optimization is performed VERY EASY and FAST, because Cross-Entropy function is convex.

On the first phase we generate sample $Z_1 \ldots Z_V \ldots Z_N$, which has size of N different parameter sets. This generation is performed according common Probability Density Function F(Z) for parameter vector Z, which was calculated on the previously step of the algorithm.

For each v from N (v = 1…N) generated parameter vectors the value of Goal Function is calculated. Then best N_{EL} (N_{EL} = 10…50) parameter vectors Z from all N generated are selected – it is named ELITE part from full sample. This selection is performed according Goal Function values, i.e. parameter vector with number 1 will have minimum value of Goal Function, parameter vector

with number 2 will have second value of Goal Function, parameter vector with number N_{EL} will have N_{EL} ordered value of Goal Function.

After this the algorithm calculates new values of the Probability Density Function F(Z) – it is second phase of each algorithm step.

The aim of the new function F(Z) is to maximize Cross-Entropy Function. On the general case the Cross-Entropy Function is following:

$$\sum_{v=1}^{N_{EL}} \ln\{F(Z_V)\}$$

which is Kullback-Leibler probability measure of distance between different Probability Density Functions. In this formula Z_V – value of generated parameter vector on the v-th set of Elite part of current sample.

So, first we have to choice type of PDF to generate random parameter vectors Z. For continuous optimization we can use following types of PDF:

- Beta PDF.
- Normal PDF.
- Double-Exponential PDF
- Etc.

Usage of the Normal PDF F(Z) is advantageous, since in contrast to Beta and Double-Exponential PDFs the Normal PDF allows analytical solution of above task. Other types of PDF involve numerical solution. It is known following analytical solution for Normal PDF parameters (with respect to Mean and Covariance Matrix) of function F(Z):

$$\text{Mean}[j] = \sum_{v=1}^{N_{EL}} \frac{Z_V[j]}{N_{EL}}$$

$$\text{Covariance}[i,j] = \sum_{v=1}^{N_{EL}} (Z_V[i] - \text{Mean}[i]) \frac{(Z_V[j] - \text{Mean}[j])}{N_{EL}} \ldots i,j = 1 \ldots K$$

We have to prevent the too earliest occurrences of the PDF parameter, because in this case optimization is stopped non-correct (PDF will be simply Dirak function!). For this aim instead of simple choice by means of independent current step result analysis we will use smoothed updating procedure:

- $\text{Mean}[j](t) = \alpha\, \text{Mean}_{\text{prel}}\ [j](t) + (1 - \alpha)\text{Mean}[j](t-1)$

where:

$\text{Mean}_{\text{prel}}\ [j](t)$ – preliminary value of Mean[j], which we had got on current step t, i.e. before smoothed updating,

Mean[j](t) – final value of Mean[j], which we had got on current step t, i.e. after smoothed updating,

Mean[j] (t-1) – final value of Mean[j], which we had got on previously step (t-1),

α – smoothing parameter for Mean updating,

t – step number

- $\text{Cov}[i, j](t) = \zeta(t)\text{Cov}_{\text{prel}}\ [i, j](t) + (1 - \zeta(t))\text{Cov}[i, j](t - 1)$,
- $\zeta(t) = \zeta - \zeta(\ (1 - 1/t)^{\gamma}$,

where:

Cov_{prel} [i, j](t) – preliminary value of Covariance[i, j], which we had got on current step t, i.e. before smoothed updating,

Cov[i, j](t) – final value of Covariance[i, j], which we had got on current step t, i.e. after smoothed updating,

Cov[i, j](t-1) – final value of Covariance[i, j], which we had got on previously step (t-1),

ζ and γ – smoothing parameters for Covariance updating.

As seen, for PDF parameter Mean we use fixed smoothing parameter α and for PDF parameter Covariance we use dynamic (dependent of step number) smoothing parameter $\zeta(t)$.

5. SOME EXTENSIONS

5.1 Multiple Systems

In this case the input statistics of failures and repairs will be following: TF[j, 1], TR[j, 1],…, TF[j, i], TR[j, i],…, TF[j, n], TR[j, n], where

- k is amount of systems
- n(j) is amount of failures/repairs on system j
- TF[j, i] is time of failure number i on system number j
- TR[j, i] is time of finishing of repair number i on system number j, i = 1…n(j), j = 1…k.

For definition of λ_f and β_f we have to minimize following Goal Function:

$$Negative\ Logarifm\ Likelihood_f = -\sum_{j=1}^{k}\sum_{i=1}^{n(j)} \log(P_f[j,i]) \tag{5}$$

where P_f [j, i] - Conditional PDF, that i-th failure will be at moment TF[j, i] in condition, that (i -1)-th repair has finished at moment TR[j, (i -1)]. For these conditional PDF-s the expression (1) is applicable without some modifications, we only have to use TF[j,i] instead of TF[i] and TR[j, i] instead of TR[i]. Cross-Entropy Optimization algorithm to search parameters λ_f and β_f also will be exactly same, as for case of single system.

For definition of λ_r and β_r all expressions will be analogous.

5.2 How to take into account End Time and Start Time

Formula (1) assumes, that system starts to operate at time 0, and last measurement corresponds for last failure.

If for some single system j we use non-zero start time TS[j], we have to modify expression for P_f [j, i] for i = 1 – to use TR[j, 0] = TS[j] instead of 0 (see comment under formula (1)).

If for some single system j we use additional end (censored) time TE[j], we have to use additional expression P_f [j, i] for i = n(j) + 1 :

$$P_f[j,n(j)+1] = \exp\left(-\lambda_f\left(TE[j]^{\beta_f} - TR[j,n(j)]^{\beta_f}\right)\right)$$

and for this j to use additional component P_f [j, n(j)+1] on expression (5).

5.3 Definition of un-known parameters δ_f and δ_r

Sometimes initial moments (initializations) of failure rate and repair rate are not zeros (don't confuse with start times of single systems !). Suppose, they are δ_f for failure rate and δ_r for repair

rate. In this case instead of t we have to use $(t - \delta_r)$ and $(t - \delta_f)$ in all formulas of NHPP process. We also have to modify expression (1) – instead of TF[i] and TR[i] to use $(TF[i] - \delta_f)$ and $(TR[i] - \delta_f)$, to modify expression (3) – instead of TF[i] and TR[i] to use $(TF[i] - \delta_r)$ and $(TR[i] - \delta_r)$.

If values of parameters δ_f and/or δ_r are unknown, we have to search its by means of minimization of *Negative_Logarifm_Likelihood* – not only for parameters β and λ, but also for parameter δ. To search value of parameter δ, we can use Cross-Entropy Optimization algorithm for modified expressions (2) and (4) (for single system) or expression (4) (for multiple systems).

We also have to note, that MLE approach gets us solution for three parameter optimization only for case $\beta>1$ (it is widely known fact for Weibull three parameter search). So, for these situations we have to use some other methods, e.g.:

– To use some non-parametric estimation method (for example, well known MCF approach of Nelson [5]) and based of received results to use Least Squares optimization for thee parameters (β, λ, δ). Least Squares non-linear optimization will be performed by means of Cross-Entropy method.

– Based on defined value of parameter δ to correct values of β and λ by means of MLE optimization using expressions (2) or (4).

6. OUTPUT ESTIMATIONS

Based on obtained parameters we can get some estimations and perform numerical analysis. For instantaneous values of MTBF and MTTR the following formulas are proved:

– $$MTBF_i(t) = \frac{(t-\delta_f)^{(1-\beta_f)}}{\beta_f \lambda_f}$$

– $$MTTR_i(t) = \frac{(t-\delta_r)^{(1-\beta r)}}{\beta_r \lambda_r}$$

For cumulative values of MTBF and MTTR the following formulas are proved:

– $$MTBF_C(t) = \frac{(t-\delta_f)^{(1-\beta_f)}}{\lambda_f}$$

– $$MTTR_C(t) = \frac{(t-\delta_r)^{(1-\beta r)}}{\lambda_r}$$

It is impossible to obtain analytically the exact expression for instantaneous value of Availability depending of time, but approximately we can assume, that

$$Availabili\,ty_i(t) \approx \frac{MTBF_i(t)}{MTBF_i(t) + MTTR_i(t)}$$

If $\delta_f = \delta_r = \delta$ (for default $\delta_f = \delta_r = 0$) we can simplify last expression:

$$Availability_i(t) = \frac{1}{1 + \left(\frac{\beta_f \lambda_f}{\beta_r \lambda_r}\right)(t-\delta)^{(\beta_f - \beta_r)}}$$

For cumulative (or mean) value of Availability we use formula

$$Availability_i(t) = \frac{\int_0^t Availability_i(x)dx}{t}$$

It is impossible to obtain analytically the exact expression for cumulative value of Availability depending of time, but approximately we can assume, that

$$Availabili\,ty_c(t) \approx \frac{MTBF_c(t)}{MTBF_c(t) + MTTR_c(t)}$$

If $\delta_f = \delta_r = \delta$ we can simplify last expression:

$$Availability_c(t) = \frac{1}{1 + \left(\frac{\lambda_f}{\lambda_r}\right)(t-\delta)^{(\beta_f - \beta_r)}}$$

It is evident, that if $\beta_f < \beta_r$, the Instantaneous and Cumulative values of Availability increase depending on time (i.e. we see Availability Growth), although $MTBF_i(t)$ and $MTBF_c(t)$ can be reduced. Otherwise, if $\beta_f > \beta_r$, the Instantaneous and Cumulative values of Availability decrease depending on time (i.e. we see Availability Aging), although $MTBF_i(t)$ and $MTBF_c(t)$ can be increased.

7. CONCLUSION

It is important to recognize, that the Availability parameter should be integrated into general process of a system improvement. But currently the technique of the Reliability Growth doesn't take into account the factor of Availability.

The above described procedure was developed in order to calculate and track the Availability (Dependability) measures based on repair rates', as well as failure rates', modification.

The procedure is based on Cross-Entropy Global Optimization algorithm, which is used to optimize MLE function.

8. REFERENCES

1. Bluvband, Z., 1990, Availability-Growth Aspects of Reliability Growth. *Proceedings Annual Reliability and Maintability Symposium*, p. 522-526
2. Crow L., 1990, Evaluating the Reliability of Repairable Systems. *Proceedings Annual Reliability and Maintability Symposium*, p. 275- 279
3. Rubinstein R.Y. and Kroese D.P., 2004, The Cross-Entropy Method: A unified approach to Combinatorial Optimization, *Monte Carlo Simulation and Machine Learning*. Springer-Verlag, New York
4. Kroese D.P., Porotsky S., Rubinstein R.Y. 2006, The Cross-Entropy Method for Continuous Multi-Extremal Optimization. Methodology and Computing in Applied Probability, Vol. 8, No. 3, pp.383–407.
5. Nelson W., 2003, Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and other Applications. ASA-SIAM.