Availability Growth Modeling and Assessment

Zigmund Bluvband, PhD, ALD ltd.

Sergey Porotsky, PhD, ALD Software ltd.

Key Words: Availability Growth, Reliability Growth, Global Optimization, NHPP, parameter estimation, cross-entropy

SUMMARY

The reliability growth process, applied to a complex system under development, involves surfacing failure modes, analyzing the modes and their causes, and implementing corrective actions (fixes) to detected problems. In such a manner, the system reliability is grows and its configuration is going to be mature with respect to reliability. The conventional procedure of the Reliability growth implies evaluation of two principal parameters of the Non-Homogeneous Poison Process (NHPP) related to the failure rate only. In addition to the Reliability aspect, the Availability factor, and, as the result, the Availability growth (not only Reliability growth) is extremely important for the repairable Systems. Yet because the standard NHPP does not take into account the repair rate parameters, the practitioners are awaiting for a long time for an expanded procedure for the Availability Growth tracking. This paper suggests a model and a numerical method to evaluate these parameters, establishing consequently the Inherent Availability Growth model, i.e. considering only corrective maintenance times due to failures. The model can be further generalized for Operational and Achieved Availability by taking in account the preventive maintenance, administrative and logistics times as appropriate.

1 INTRODUCTION

Accurate reliability prediction and control play an important role in the product cost-effectiveness and profitability. Costs of a product service within the warranty period or under the service contract are the major expense and significant pricing factor. Proper spare part stocking and support personnel hiring and training also depend upon a good reliability predictions. On the other hand, missing reliability targets may invoke contractual penalties and even loss of business.

Telecommunication networks, oil platforms, chemical plants and airplanes consist of a great number of subsystems and components that are all subject to failure. Reliability theory studies system failure behavior in relation to their component failure behavior, which often isn't easier to analyze.

Typical task in Reliability Management is the Reliability Growth Analysis, which deals with Test-Analyze-and-Fix process. A repairable system is the system which can be restored to satisfactory operation by any action other than the replacement of the entire system: i.e. from parts replacements to adjustments and settings. When discussing the rate at which failures occur during system operation time (and are then repaired), one can define a Rate of Occurrence of Failure (ROCOF) or "failure rate".

For systems with repairable failures the standard model is NHPP. According to 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

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

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

For any NHPP process with intensity function R(T), the distribution function for the inter-arrival time t to the next failure, given that the previous failure just occurred at time T, is known as

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

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

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

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

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

where γ and α are the Duane model parameters,

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

During repairable systems analysis, two main problems are solved:

- Definition of NHPP distribution parameters by means of statistics of failures
- Forecasting of some output criteria, e.g. Amount of failures at some period, Mean Time to Failure (MTTF), Mean Time to Repair (MTTR), etc., based on obtained parameters.

This classical task of Reliability Growth Analysis is not enough for repairable systems with essential Availability needs. In this case the process can be extended for the Availability Growth Analysis, which assumes, that repairable systems' restorations are performed due to two factors –

failure rate and repair rate [1]. For this task one has to define parameters of "mixed" flows – failures and repairs – instead of single ("continuous") flow of failures for the standard NHPP task.

The rest of the article is organized as follows. Chapter 2 introduces Availability Growth model as extension of Reliability Growth model. First the simplest case – single system is considered. Various techniques to solve this model are presented in Chapter 3. Chapter 4 presents the Cross-Entropy method, as applicable to search for/estimate the proposed model parameters. The more challenging tasks of Availability Growth are tackled in Chapter 5. Chapter 6 shows how to get some output estimations of the Availability Growth.

2 DEFINITION OF DISTRIBUTION PARAMETERS FOR SINGLE SYSTEM

First consider case of single system.

Input statistics of failures and repairs is as 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.
Assume that both failure flow and repair flow are NHPP processes. So,

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

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

One has to define parameters λ_f, β_f, λ_r, β_r and for this purpose MLE (Maximum Likelihood Estimations) approach can be used.

Comment. Generally speaking, failure and/or repair flows can be described 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, two different cases should be considered:

I. Failure Rate doesn't change during repair.

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

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

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])

II. Failure Rate changes during repair as usually.

In this case the deterioration of the system during repair is normal (for example, during repair the car failure rate is increased according to the time increase). For this case the classical Crow formulas are not applicable. Conditional PDF, that i^{th} failure will be at moment TF[i] on condition, that $(i-1)^{th}$ repair has been 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 TR[0] = 0 is used.

The goal is to search for the values of λ_f and β_f such, that *Negative LogLikelihood* will be minimum:

$$Negative\ LogLikelihood_f = -\sum_{i=1}^{n} \log(P_f[i]) \quad (2)$$

To define required parameters for repair flow, only one case has to be considered – changes of Repair Rate during repair don't differ from these during normal operation. 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

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

A goal is to search for values of λ_r and β_r such, that *Negative LogLikelihood* will be minimum:

$$Negative\ LogLiklihood_r = -\sum_{i=1}^{n} \log(P_r[i]) \quad (4)$$

3 COMPARISON OF DIFFERENT GLOBAL OPTIMIZATION APPROACHES

Global Optimization of a non-linear function is a common approach for solving multiple practical problems (supply optimization, text categorization, distribution parameters estimation, etc.). That's because the Linear Regression model can not be used for interval and multiplied censored data: for three parameter Weibull estimation, for Duane model with multiple systems, for Gompertz model, etc. In this numerous cases one should search for distribution parameters by means of non-linear and non-convex, global optimization – both for MLE and non-linear regression.

The task is to search for a 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 table, or even algorithm-calculated-form), dependent of vector Z.

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

I. To write and transform derivatives of Goal Function (e.g., LogLikelihood for MLE method, Sum of Least 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.

II. To use "direct search methods", provided universal search of Global Minimum (without analytical

definition of derivatives).

For the first approach the complex analytical expressions should be defined for derivatives for each single task. Formerly this approach was commonly used, requiring additional resources for both algorithm development and software implementation for each single task. For example, Quasi-Newton method minimizes the Negative LogLikelihood Function in order to bring partial derivatives to zero. Perhaps, it is not very tough job for simple cases, but for more complex models this approach requires essential additional efforts.

The second (universal) approach allows to search for the optimal solution not only for a single task, but rather for all similar situations (LogNormal, Gamma and other distributions, MLE for repairable failures, Non Linear Regression for Gompertz model, etc.), i.e. in general - for all complex non-convex, multi-extremal optimization tasks. Contrasting to "derivative"-oriented algorithms, the proposed will require only one implementation.

For the last approach a lot of methods based on gradient (or, if goals function has not a gradient, on pseudo-gradient) calculation and analysis, are developed. Nevertheless for many real tasks the Goal Function isn't convex, but 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. Moreover, not enough, if any, information is often available to define this initial point and then it is impossible to apply regular (gradients-based) methods.

The Cross-Entropy Optimization [3], one of RANDOM SEARCH-oriented methods, is proposed in this paper for the Global Optimization Task. It is relatively new random search-oriented approach (in comparison with Genetic Algorithm, implemented as Toolbox in 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's name derives from the cross-entropy (or Kullback-Leibler), 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 two following phases [3]:

1. 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 the specified random mechanism.

2. 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 unlike global optimization, this optimization is performed VERY EASY and FAST, because of convexity of Cross-Entropy function.

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

The value of Goal Function is calculated for each v from N (v = 1…N) generated parameter vectors. 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 to the 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, and parameter vector with number N_{EL} will have N_{EL} ordered value of Goal Function.

Next 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 the Kullback-Leibler probability measure of distance between different Probability Density Functions (PDF), where Z_V is the value of the generated parameter vector on the v-th set of Elite part of current sample.

So, first a type of PDF has to be selected to generate random parameter vectors Z. For continuous optimization the following types of PDF can be used:
Beta PDF,
Normal PDF,
Double-Exponential PDF,
etc.

Using of 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):

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

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

The too earliest occurrences of the PDF parameter should be prevented, 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, the smoothed updating procedure can be used:

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

where:

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

Mean[j](t) – final value of Mean[j], obtained on current step t, i.e. after smoothed updating,
Mean[j] (t-1) – final value of Mean[j], obtained on previously step (t-1),
α – smoothing parameter for Mean updating,
t – step number
$Cov[i, j](t) = \zeta(t)Cov_{prel}[i, j](t) + (1 – \zeta(t))Cov[i, j](t - 1)$,
$\zeta(t) = \zeta – \zeta((1 – 1/t)^{\gamma}$,
where:
Cov_{prel} [i, j](t) – preliminary value of Covariance[i, j], obtained on current step t, i.e. before smoothed updating,
Cov[i, j](t) – final value of Covariance[i, j], obtained on current step t, i.e. after smoothed updating,
Cov[i, j](t-1) – final value of Covariance[i, j], obtained on previously step (t-1),
ζ and γ – smoothing parameters for Covariance updating.
As one can see, for the PDF parameter Mean the fixed smoothing parameter α is used, and for the PDF parameter Covariance the dynamic (dependent of step number) smoothing parameter $\zeta(t)$ is used [4].

5 SOME EXTENSIONS

5.1 Multiple Systems

In this case the input statistics of failures and repairs will be as 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.

To define λ_f and β_f, the following Goal Function should be minimized:

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

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

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 operating at time 0, and last measurement corresponds for last failure.

If for some single system j the non-zero start time TS[j] is used, the expression for P_f [j, i] has to be modified for i = 1: use TR[j, 0] = TS[j] instead of 0 (see comment under formula (1)).

If for some single system j additional end (censored) time TE[j] is used, the additional expression P_f [j, i] should be used 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 $(t – \delta_r)$ and $(t – \delta_f)$ have to be used in all formulas of NHPP process instead of t. Expression (1) should be modified too: 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, they can be found by means of minimization of *Negative LogLikelihood* – not only for parameters β and λ, but also for parameter δ. To search for value of parameter δ, Cross-Entropy Optimization algorithm can be applied to modified expressions (2) and (4) (for single system) or expression (5) (for multiple systems).

It is necessary to note, that MLE approach gets solution for optimization of three parameters only for case $\beta>1$ (widely known fact for Weibull three parameter estimation). So, for situation when $\beta<1$ some other method should be used, 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 three parameters (β, λ, δ). Least Squares non-linear optimization will be performed by means of Cross-Entropy method.

- Based on above estimated value of parameter δ it is possible to improve the estimation of β and λ by means of MLE optimization using expressions (2) or (4).

6 OUTPUT ESTIMATIONS

Based on obtained parameters one can get some estimation and perform numerical analysis.

For instantaneous values of MTTF and MTTR the following expressions are used:

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

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

For cumulative values of MTBF and MTTR one can use the following:

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

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

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

$$Availability_i(t) \approx \frac{MTTF_i(t)}{MTTF_i(t) + MTTR_i(t)}$$

If $\delta_f = \delta_r = \delta$ (for default $\delta_f = \delta_r = 0$) one 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 the formula can be applied

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

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

$$Availability_c(t) \approx \frac{MTTF_c(t)}{MTTF_c(t) + MTTR_c(t)}$$

If $\delta_f = \delta_r = \delta$, than last expression can be simplified :

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

It is evident, that if $\beta_f < \beta_r$, the Instantaneous and Cumulative values of Availability are increased in time (i.e. see Availability Growth), although $MTTF_i(t)$ and $MTTF_c(t)$ will decrease. Otherwise, if $\beta_f > \beta_r$, the Instantaneous and Cumulative values of Availability are decreased depending on time (i.e. see Availability Aging), although $MTTF_i(t)$ and $MTTF_c(t)$ can increased.

CONCLUSION

It is important to recognize, that the Availability aspect should be integrated into general process of a repairable System improvement and maturity progress evaluation, consecutively complementing and expanding the commonly used Reliability Growth procedure. The above described methodology has been developed to introduce and put into practice the Inherent Availability Growth process, derived from the failure and repair rate estimations, based on Entropy Global Optimization algorithm applied to the MLE function.

REFERENCES

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

BIOGRAPHIES

Zigmund Bluvband, Ph.D.
A.L.D. Ltd.
52 Menachem Begin Road
Tel-Aviv, Israel 67137

e-mail: zigmund@ald.co.il

Zigmund Bluvband is the President of Advanced Logistics Developments Ltd. His PhD (1974) is in Technical Sciences. He is a Fellow of ASQ and ASQ–Certified Reliability Engineer, Quality Engineer, Quality Manager and Certified Six Sigma Black Belt. Z. Bluvband has accrued more than 30 years of industrial and academic experience. Z. Bluvband was the President of the Israel Society for Quality (1989 – 1994). He has published more than 70 papers and tutorials, ten patents and four books. In 2006 Dr. Zigmund Bluvband has been honored with the IEEE Reliability Society Lifetime Achievement Award. In 2009 Dr. Z. Bluvband was elected as an IAQ (International Academy for Quality) Academician.

Sergey Porotsky, Ph.D
ALD Software Ltd.
52 Manachem Begin Road,
Tel-Aviv, Israel 67137.

e-mail: sergey.porotsky@ald.co.il

Sergey Porotsky is a Chief Scientist of ALD Software Ltd. He is an expert in analytical and numerical methods of Applied Mathematics. Dr. Porotsky is involved in design and optimization algorithms for reliability analysis of complex systems - Monte Carlo, Markov Chains, FTA methods. While with Moscow Scientific and Research Center of Computers, he developed analytical methods for Computer Systems and Networks Evaluation. His PhD is in Computer Science (1985) and his Dr. Sc. degree is in Operation Research (1992).