1985

Survival function of hypo-exponential distributions

Lotfy, Mamdouh M.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SURVIVAL FUNCTION
OF
HYPO-EXPONENTIAL DISTRIBUTIONS

by

Mamdouh M. Lotfy
and
Ali S. Abdelsamad

March 1985

Thesis Advisor: James D. Esary

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Survival Function of Hypo-exponential Distributions		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis March 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mamdouh M. Lotfy; Ali S. Abdelsamad		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		12. REPORT DATE Marcn, 1985
		13. NUMBER OF PAGES 84
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Exponential Distribution
System Reliability
Convolutions
Hypo-Expontial Distribution

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The reliability of a system is the probability that the system will survive or complete an intended mission of certain duration.

Describing all possible ways that a system can survive a mission in reliability shorthand gives a simple approach to reliability computations. Reliability computation for a system defined by shorthand notation is greatly dependent upon the convolution problem.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Assuming constant component failure rates, this paper presents an analytical approach and a computer program for computing the reliability of any convolution of independent and exponentially distributed random variables.

Survival Function of Hypo-exponential Distributions

by

Mamdouh M. Lotfy
Lt.Col, Egypt.
B.S., Military Technical College, 1972

and

Ali S. Abdelsamad
Col, Egypt.
B.S., Military Technical Colledge, 1970

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1985

ABSTRACT

The reliability of a system is the probability that the system will survive or complete an intended mission of certain duration.

Describing all possible ways that a system can survive a mission in reliability shorthand gives a simple approach to reliability computations. Reliability computation for a system defined by shorthand notation is greatly dependent upon the convolution problem.

Assuming constant component failure rates, this paper presents an analytical approach and a computer program for computing the reliability of any convolution of independent and exponentially distributed random variables.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

In making a mathematical model for a real-life phenomenon it is always necessary to make certain simplifying assumptions so as to render the mathematics tractable. One of the simplifying assumptions that is often made is to assume that certain random variables are exponentially distributed. The reason for this is that the exponential distribution is relatively easy to work with.

The property of the exponential life distribution which makes it easy to analyze is its (memoryless) lack of deterioration with time. By this we mean that if the life time of an item is exponentially distributed, then an item which has been in use for a certain amount of time is as good as a new item in regards to the amount of time remaining until it fails. When the life time of an item is exponentially distributed, the failure rate function for the item is constant.

Under the assumption of constant component failure rates, it is possible to build a reliability shorthand [Ref. 1] for any system. The term system is used to describe a set of components organized to perform some mission.

Any study on system reliability requires a description for the system's life and a derivation of the system's survival function. The reliability shorthand gives a simple and easy way for describing a system's life, but it is difficult to implement computationally since it involves considerable complexity in handling convolutions. Here, the term convolution refers to the summation of independent random variables (lives).

This paper presents an analytical approach for obtaining a general equation for the survival function of any convolution of independent and exponentially distributed random variables.

Section 2 deals with convolutions in detail and gives a mathematical derivation for a general equation for the survival function for any convolution of exponential random variables, using Laplace transforms and Theorem of Residues.

Section 3 gives a mathematical derivation for an alternative formula for the survival function of any convolution of exponential random variables by computing the coefficients of all the polynomials that accompany the exponential terms in the survival function equation.

Appendix A contains a computer program written in Fortran for computing the reliability of any convolution of inependent and exponentially distributed random variables. This program uses the general equation in Section 2.

Appendix B contains another computer program written in Fortran to compute the reliability of any convolution of exponential random variables, using the general formula in Section 3.

II. CONVOLUTIONS

Under the assumption of constant component failure rates, this section presents an analytical approach for finding a general equation for the survival function of any convolution of independent and exponentially distributed random variables.

A. SURVIVAL FUNCTION BY INTEGRATION

Let us start with a simple system; a standby system having one active component and one cold spare component. The life time of the system is

$$T = T_1 + T_2 ,$$

where

$$T_1 \sim EXP(\lambda_1) ,$$

$$T_2 \sim EXP(\lambda_2) ,$$

and

$$T_1, T_2 \text{ are independent.}$$

An active component A is to complete a mission of duration t, while a spare component S replaces the active component when it fails. The life duration of the active

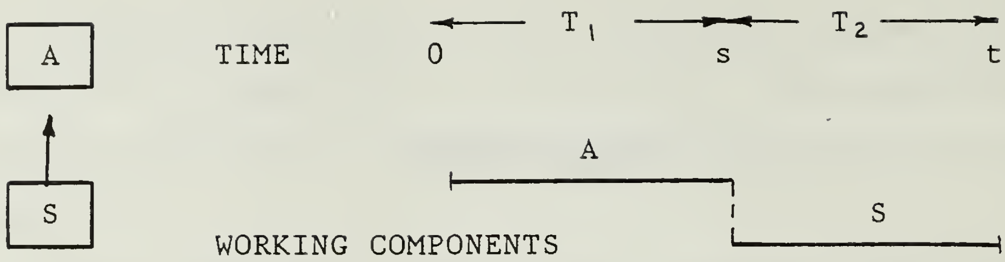

Figure 2.1 A Single Active Component With One Spare

component is T_1 and the life duration for the spare component, if it is used, is T_2 (see Fig 2.1).

The system survival function is given by

$$\bar{F}_T(t) = P\,(T > t)$$

$$= \bar{F}_{T_1}(t) + \int_0^t \bar{F}_{T_2}(t-s) . f_{T_1}(s)\, ds$$

$$= e^{-\lambda_1 t} + \int_0^t e^{-\lambda_2 (t-s)} \lambda_1 e^{-\lambda_1 s}\, ds ,$$

where

$$f_{T_1}(s) = -\, d\bar{F}_{T_1}(s)/ds$$

$$= \lambda_1 e^{-\lambda_1 s} .$$

By integration, the survival function for the system is

$$\bar{F}_T(t) = \frac{\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_1 - \lambda_2} e^{-\lambda_2 t} , \quad t \geqslant 0 .$$

Now, suppose a system consists of one active component with constant failure rate λ_1 and two spares having constant failure rates λ_2 and λ_3 respectively. The life time of the system is

$$T = T_1 + T_2 + T_3 ,$$

where

$$T_1 \sim \text{EXP}(\lambda_1) ,$$

$$T_2 \sim \text{EXP}(\lambda_2) ,$$

$$T_3 \sim \text{EXP}(\lambda_3) ,$$

and

T_1, T_2, T_3 are independent.

The shorthand notation for the system is

$$\text{EXP}(\lambda_1) + \text{EXP}(\lambda_2) + \text{EXP}(\lambda_3) .$$

The survival function of the sustem is given by

$$\bar{F}_T(t) = \bar{F}_{T_1+T_2}(t) + \int_0^t \bar{F}_{T_3}(t-s) . f_{T_1+T_2}(s)\, ds ,$$

where

$$\bar{F}_{T_1+T_2}(t) = \frac{\lambda_2}{\lambda_2-\lambda_1} e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_1-\lambda_2} e^{-\lambda_2 t} ,$$

$$\bar{F}_{T_3}(t-s) = e^{-\lambda_3(t-s)} ,$$

and

$$f_{T_1+T_2}(s) = -d\bar{F}_{T_1+T_2}(s)/ds$$

$$= \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}\left[e^{-\lambda_2 s} - e^{-\lambda_1 s}\right] .$$

By integration, the survival function for the system is

$$\bar{F}_T(t) = \bar{F}_{T_1+T_2+T_3}(t)$$

$$= \frac{\lambda_2\lambda_3 e^{-\lambda_1 t}}{(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)} + \frac{\lambda_1\lambda_3 e^{-\lambda_2 t}}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)} + \frac{\lambda_1\lambda_2 e^{-\lambda_3 t}}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)}, \quad t \geqslant 0.$$

To compute the survival function for a system of n lives,

$$T = T_1 + \ldots + T_n ,$$

we can proceed in a similar manner. However the integrations will become increasingly complex, particularly if the tacit assumption made so far that all component failure rates are different is abandoned. Therefore, we will use another mathematical way to compute the survival function without direct integration. This will be done with Laplace transforms and the Theorem of Residues in the next subsections.

B. CLASSIFICATION OF CONVOLUTIONS

Consider a system which has a life time

$$T = T_1 + \dots + T_n ,$$

where $T_1 , \dots , T_n$ are independent and exponentially distributed random variables. This system can be classified as one of three possible convolution cases :

1. First Case

The system has n dissimilar failure rates,

$$\lambda_i \neq \lambda_j \ , \ i \neq j \ , \ i,j = 1, \ \dots \ , n.$$

2. Second Case

The system has n identical failure rates,

$$\lambda_i = \lambda_j \ , \ i,j = 1, \ \dots \ , n.$$

3. Third Case

The system has a combination of the first case and the second case, some similar and some dissimilar failure rates.

We consider the third case to be of particular interest, since it better represents the general situation in computing the reliability of any convolution of exponentially distributed random variables.

C. CHARACTERISTIC FUNCTIONS AND LAPLACE TRANSFORMS

Consider a random variable T which has a distribution function $F_T(t)$ and a density function $f_T(t)$. The characteristic function of T is defined, for any real number u, [Ref. 2] by

$$\tilde{\phi}_T(u) = E\left[e^{iuT} \right]$$

$$= \int_{-\infty}^{+\infty} e^{iut} \, dF_T(t)$$

$$= \int_{-\infty}^{+\infty} e^{iut} f_T(t) \, dt \; .$$

If T is non negative ($T \geqslant 0$) and exponentially distributed with a failure rate λ, then

$$\tilde{\phi}_T(u) = \int_0^{\infty} e^{iut} . f_T(t) \, dt$$

$$= \int_0^{\infty} e^{iut} \lambda \, e^{-\lambda t} \, dt$$

$$= \frac{\lambda}{(\lambda - iu)} \; .$$

Now, if

$$T = T_1 + \dots + T_n ,$$

where

$$T_i \sim \mathrm{EXP}(\lambda_i), \ i = 1, \dots , n,$$

and $T_1, \dots, T_n$ are independent, then the characteristic function of the sum (convolution) $T_1 + \dots + T_n$ is

$$\tilde{\phi}_T(u) = E\left[e^{iu(T_1 + \cdots + T_n)} \right]$$

$$= E\left[e^{iuT_1} \right] \dots E\left[e^{iuT_n} \right]$$

$$= \tilde{\phi}_{T_1}(u) \quad \dots \quad \tilde{\phi}_{T_n}(u)$$

$$= \prod_{i=1}^{n} \tilde{\phi}_{T_i}(u)$$

$$= \prod_{i=1}^{n} \left(\frac{\lambda_i}{\lambda_i - iu} \right) .$$

This implies that the characteristic function for the sum $T_1 + \dots + T_n$ is just the product of the individual characteristic functions.

When dealing with random variables which only assume non negative values, it is more convenient to use Laplace transforms rather than characteristic functions. The Laplace transform of the random variable having distribution $F_T(t)$ is defined [Ref. 2] by

$$\phi_T(s) = E\left[e^{-st}\right]$$

$$= \int_0^\infty e^{-st}\, dF_T(t)$$

$$= \int_0^\infty e^{-st}\, f_T(t)\, dt,$$

where $f_T(t)$ is the density function of T. This integral exists for a complex variable $s=a+bi$, where $a \geq 0$.

If $T_1, \ldots, T_n$ are independent random variables, then the Laplace transform of the sum (convolution) $T_1 + \ldots + T_n$ is given by

$$\phi(s) = E\left[e^{-s(T_1 + \cdots + T_n)}\right]$$

$$= E\left[e^{-sT_1}\right] \ldots E\left[e^{-sT_n}\right]$$

$$= \phi_{T_1}(s) \ \ldots \ \phi_{T_n}(s)$$

$$= \prod_{i=1}^{n} \phi_{T_i}(s).$$

It is important to note that the Laplace transform uniquely determines the distribution. That is, in fact, there is a one to one correspondence between distribution functions and Laplace transforms.

Now, we will consider the following question. Given a function $\phi_T(s)$, does there exist a function $f_T(t)$ whose Laplace transform is $\phi_T(s)$, and if it exists, how can it be determined ? The answer to the first question is not always positive. In general, the function $\phi_T(s)$ must satisfy some restrictions which will be discussed in the next subsection. If the answer to the first question is positive, then the inversion of the Laplace transform by a complex integral gives the answer to the second question; That is, the density function for the random variable T is given by [Ref. 3]

$$f_T(t) = \mathcal{L}^{-1}\left[\phi_T(s)\right]$$

$$= \frac{1}{2\pi i}\int_{\alpha - i\infty}^{\alpha + i\infty} e^{st} \cdot \phi_T(s)\, ds,$$

where the notation $\mathcal{L}^{-1}[\phi_T(s)]$ refers to the inversion of the Laplace transform and the path of the integral above is the straight line Re $s = \alpha$ ($\alpha > 0$) parallel to the imaginary axis.

D. LAPLACE INVERSION AND THE THEOREM OF RESIDUES

As was stated in last subsection, if we have a Laplace transform $\phi_T(s)$ for a non-negative random variable T, then the Laplace inversion $\mathcal{L}^{-1}[\phi_T(s)]$ gives the density function $f_T(t)$.

When dealing with a convolution of random variables, that is, $T = T_1 + \ldots + T_n$, where $T_1, \ldots, T_n$ are independent, the convolution density function is given by [Ref. 3]

$$f_T(t) = \mathcal{L}^{-1}[\phi_T(s)]$$

$$= \frac{1}{2\pi i} \int_{\alpha - i\infty}^{\alpha + i\infty} e^{st} \cdot \phi_T(s)\, ds,$$

where $\phi_T(s)$ is the Laplace transform of the sum (convolution) $T_1 + \ldots + T_n$.

To obtain the convolution density function, it is clear from the above integral that we have to evaluate a complex integral, but the Theorem of Residues reduces the evaluation of a complex integral along closed contours to passages to the limit and differentiations.

Suppose a function G(s) is holomorphic in a finite domain C, the term holomorphic meaning that the function is analytic and single-valued. Then the integral

$$\frac{1}{2\pi i} \oint G(s)\, ds$$

evaluated along it's boundary, vanishes. If the function G(s) is not holomorphic in C, the integral may not be zero. If there is in C only one singularity at point a, the value of the integral is called the residue of G(s) at the singular point a.

The well known Residues Theorem [Ref. 3] is

> The integral of an analytic function along a closed contour is the sum of the residues at the singular points in the domain enclosed by the contour, multiplied by 2π i.

Now, let us define the analytic function G(s) to be

$$G(s) = e^{st} \cdot \phi_T(s) ,$$

where $\phi_T(s)$ is the Laplace transform of the sum (convolution) $T_1 + \dots + T_n$. Then, according to the Theorem of Residues, the convolution density will be obtained as follows;

$$f_T(t) = \mathcal{L}^{-1}[\phi_T(s)]$$

$$= \frac{1}{2\pi i} \int_{\alpha - i\infty}^{\alpha + i\infty} e^{st} \phi_T(s)\, ds$$

$$= \frac{1}{2\pi i} \int_{\alpha - i\infty}^{\alpha + i\infty} G(s)\, ds$$

$$= \frac{1}{(2\pi i)} \sum_{i=1}^{m} (2\pi i) \; R_i \quad ,$$

where m is number of singularities (poles), and R_i, $i=1,\ldots,m$, is the residue of the analytic function G(s) at the singular point (pole) a_i.

Therefore

$$f_T(t) = \sum_{i=1}^{m} R_i \quad ,$$

that is, the convolution density function is just the sum of the residue of the analytic function G(s) at the singular points.

The result established by the last equation would not be of much value if it were not possible to compute the residue of the analytic function G(s) at the singularities directly and without evaluating any integral. Thus, let us now discuss how the residue at a singular point (pole) a of the holomorphic function G(s), can be evaluated.

First the order of the singular point (pole) must be established by computing the limit, for s tending to a, of (s-a)G(s), $(s-a)^2$ G(s), and so on, until we find a finite limit. The exponent which we find for (s-a) is the order of the singular point a. If the singular point(pole) at a is of order (multiplicity) n, the residue of the function G(s) at a is given by [Ref. 3]

$$R = \lim_{s \to a} \left[\frac{1}{(n-1)!} \frac{d^{n-1}}{ds^{n-1}} \left((s-a)^n G(s) \right) \right] .$$

Therefore, the residue of G(s) at the singular point (pole) a_i is

$$R_i = \lim_{s \to a_i} \left[\frac{1}{(n_i-1)!} \frac{d^{n_i-1}}{ds^{n_i-1}} \left((s-a_i)^{n_i} G(s) \right) \right] ,$$

where n_i is the order (multiplicity) of the singular point (pole) a_i, i = 1,..., m.

By substitution, the convolution density function is given by

$$f_T(t) = \sum_{i=1}^{m} R_i$$

$$= \sum_{i=1}^{m} \lim_{s \to a_i} \left[\frac{1}{(n_i - 1)!} \frac{d^{n_i - 1}}{ds^{n_i - 1}} \left((s - a_i)^{n_i} G(s) \right) \right] .$$

This implies that the convolution density can be obtained by limits and differentiations which are easier operations to deal with than the integrations.

Now, we are ready to deal with the three different convolution cases which have been mentioned before. we will try first to obtain the convolution density for every case. Then, the survival functions will be easily obtained by integration.

E. SURVIVAL FUNCTION OF A CONVOLUTION

In this subsection, we will use the results obtained in Subsection D to find the survival function for all the possible convolution cases which have been mentioned before. This will result in a general equation for the survival function of any convolution of independent and exponentially distributed random variables.

1. A System Having n Dissimilar Failure Rates

Consider a system defined by the shorthand notation

$$\text{EXP}(\lambda_1) + \ldots + \text{EXP}(\lambda_n)$$

where

$$\lambda_i \neq \lambda_j \ , \ i \neq j \ , \ i,j = 1, \ \ldots \ ,n.$$

The life time of the system is

$$T = T_1 + \ldots + T_n \ ,$$

where T_i is an exponentially distributed random variable with a failure rate λ_i, $i=1,\ldots,n$ and T_1, ... , T_n are independent. The Laplace transform of the convolution is given by

$$\phi_T(s) = E\left[e^{-s(T_1 + \cdots + T_n)}\right]$$

$$= \prod_{i=1}^{n} \left(\frac{\lambda_i}{s+\lambda_i}\right)$$

If we define the analytic function

$$G(s) = e^{st} \cdot \phi_T(s),$$

then, according to the Theorem of Residues, the convolution density is given by

$$f_T(t) = \sum_{i=1}^{n} R_i ,$$

where n is the number of singularities (number of distinct failure rates), and R is the residue of the analytic function G(s) at the singular point (pole) $-\lambda_i$, i = 1,..., n.

It is clear from the Laplace transform function $\phi_T(s)$ that there are n singularities, each of which has an order (multiplicity) one; that is, $n_i = 1$ for i = 1,..., n. Therefore, the residue of G(s) at the singular point(pole) $-\lambda_i$ is given by

$$R_i = \lim_{s \to -\lambda_i} \left[\frac{1}{(n_i - 1)!} \frac{d^{n_i-1}}{ds^{n_i-1}} \left((s - (-\lambda_i))^{n_i} G(s) \right) \right]$$

$$= \lim_{s \to -\lambda_i} \left[(s + \lambda_i) \, G(s) \right] .$$

By substitution, the convolution density will become

$$f_T(t) = \sum_{i=1}^{n} R_i$$

$$= \sum_{i=1}^{n} \lim_{s \to -\lambda_i} (s + \lambda_i) \, e^{st} \prod_{i=1}^{n} \left(\frac{\lambda_i}{s + \lambda_i} \right)$$

$$= \sum_{i=1}^{n} \lambda_i \left[\prod_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{\lambda_j}{s + \lambda_j} \right) \right] e^{-\lambda_i t} .$$

The survival function of the system will be obtained as follows ;

$$\bar{F}_T(t) = P (T > t)$$

$$= \int_t^{\infty} f_T(u) \, du$$

$$= \sum_{i=1}^{n} \left[\prod_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{\lambda_j}{s + \lambda_j} \right) \right] \int_t^{\infty} \lambda_i e^{-\lambda_i u} du$$

$$= \sum_{i=1}^{n} \left[\prod_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{\lambda_j}{\lambda_j - \lambda_i} \right) \right] e^{-\lambda_i t}$$

$$= \sum_{i=1}^{n} \frac{\left(\prod_{\substack{j=1 \\ j \neq i}}^{n} \lambda_j \right) e^{-\lambda_i t}}{\prod_{\substack{j=1 \\ j \neq i}}^{n} (\lambda_j - \lambda_i)} \qquad , t \geqslant 0,$$

which is the well known formula for the survival function for a system having n dissimilar failure rates.

2. A System Having n Identical Failure Rates

Consider a system defined by the shorthand notation

$$EXP(\lambda) + \ldots + EXP(\lambda) \quad (\text{ n times })$$

The life time of the system is

$$T = T_1 + \ldots + T_n .$$

where T_i , $i = 1,\ldots,n$, is an exponentially distributed random variable with a constant failure rate λ , and $T_1,\ldots,$ T_n are independent. The Laplace transform of the convolution is

$$\phi_T(s) = \prod_{i=1}^{n} \int_0^\infty e^{-st_i} \cdot f_{T_i}(t) \, dt_i$$

$$= \left(\frac{\lambda}{s+\lambda}\right)^n .$$

Then

$$G(s) = e^{st} \cdot \phi_T(s)$$

$$= e^{st}\left(\frac{\lambda}{s+\lambda}\right)^n .$$

In this case the analytic function G(s) has one sigular point (pole) of order (multiplicity) n at the point $s=-\lambda$.

According to the Theorem of Residues, the residue of the analytic function G(s) is given by

$$R = \lim_{s \to -\lambda} \left[\frac{1}{(n-1)!} \frac{d^{n-1}}{ds^{n-1}} \left((s+\lambda)^n G(s) \right) \right]$$

$$= \lim_{s \to -\lambda} \left[\frac{1}{(n-1)!} \frac{d^{n-1}}{ds^{n-1}} \left((s+\lambda)^n e^{st} \frac{\lambda^n}{(s+\lambda)^n} \right) \right]$$

$$= \lim_{s \to -\lambda} \left[\frac{1}{(n-1)!} \frac{d^{n-1}}{ds^{n-1}} \left(e^{st} \lambda^n \right) \right]$$

$$= \frac{\lambda^n}{(n-1)!} t^{n-1} e^{-\lambda t} \qquad , \; t \geqslant 0.$$

Therefore, the convolution density is given by

$$f_T(t) = R$$

$$= \frac{\lambda^n}{(n-1)!} \cdot t^{n-1} \cdot e^{-\lambda t}$$

$$= \frac{\lambda^n}{\Gamma(n)} t^{n-1} \cdot e^{-\lambda t} \qquad , \; t \geqslant 0,$$

which is the density function of the Erlang (n , λ) distribution.

Now, the survival function is given by

$$\bar{F}_T(t) = P(T > t)$$

$$= \int_t^{\infty} f(u)\, du$$

$$= \int_t^{\infty} \frac{\lambda^n}{(n-1)!} u^{n-1} e^{-\lambda u}\, du$$

Integrating by parts yields

$$\bar{F}_T(t) = \sum_{z=1}^{n} \frac{(\lambda t)^{z-1}}{(z-1)!} e^{-\lambda t} \quad , t \geqslant 0,$$

which is the well known formula for the survival function of the Erlang (n , λ) distribution.

3. <u>A Sytem Having Some Similar and Some Dissimilar Failure Rate</u>

The shorthand notation for the system is

$$\text{EXP}(\lambda_1) + \ldots + \text{EXP}(\lambda_1) \qquad (n_1 \text{ times})$$

$$+\text{EXP}(\lambda_2) + \ldots + \text{EXP}(\lambda_2) \qquad (n_2 \text{ times})$$

$+ \ldots\ldots\ldots\ldots$

$+EXP(\lambda_m) + \ldots + EXP(\lambda_m)$ $\quad$ (n_m times)

where m is the number of distinct failure rates, and n is the multiplicity of the i-th failure rate λ_i, $i = 1,\ldots,m$. The Laplace transform of the convolution is

$$\phi_T(s) = \prod_{i=1}^{m} \left(\frac{\lambda_i}{s+\lambda_i}\right)^{n_i}$$

Then

$$G(s) = e^{st} \cdot \phi_T(s)$$

$$= e^{st} \cdot \prod_{i=1}^{m} \left(\frac{\lambda_i}{s+\lambda_i}\right)^{n_i} .$$

In this case, the system has m distinct failure rates, so that the analytic function G(s) has m singularities. According to the Theorem of Residues the convolution density function is given by

$$f(t) = \sum_{i=1}^{m} R_i ,$$

where R_i is the residue of G(s) at the singular point (pole) $-\lambda_i$, i= 1,..., m.

Since n_i is the multiplicity for the failure rate λ_i, i = 1,...,m, it follows that the the residue of the analytic function G(s) at the singular point (pole) $-\lambda_i$ is given by

$$R_i = \lim_{s \to -\lambda_i} \left[\frac{1}{(n_i-1)!} \frac{d^{n_i-1}}{ds^{n_i-1}} \left((s+\lambda_i)^{n_i} G(s) \right) \right]$$

$$= \lim_{s \to -\lambda_i} \left[\frac{1}{(n_i-1)!} \frac{d^{n_i-1}}{ds^{n_i-1}} \left((s+\lambda_i)^{n_i} e^{st} \prod_{j=1}^{m} \left(\frac{\lambda_j}{s+\lambda_j} \right)^{n_j} \right) \right]$$

$$= \lim_{s \to -\lambda_i} \left[\frac{1}{(n_i-1)!} \frac{d^{n_i-1}}{ds^{n_i-1}} \left(\lambda_i^{n_i} e^{st} \prod_{\substack{j=1 \\ j \neq i}}^{m} \left(\frac{\lambda_j}{s+\lambda_j} \right)^{n_j} \right) \right]$$

$$= \frac{\left(\prod_{i=1}^{m} \lambda_i^{n_i} \right)}{(n_i-1)!} \lim_{s \to -\lambda_i} \left[\frac{d^{n_i-1}}{ds^{n_i-1}} e^{st} \prod_{\substack{j=1 \\ j \neq i}}^{m} (s+\lambda_j)^{-n_j} \right] .$$

Let D = d/ds. Then $D^{n_i-1} = d^{n_i-1} / ds^{n_i-1}$ and

$$R_i = \left(\prod_{i=1}^{m} \lambda_i^{n_i} \right) \frac{1}{(n_i-1)!} \lim_{s \to -\lambda_i} \left[D^{n_i-1} e^{st} \prod_{\substack{j=1 \\ j \neq i}}^{m} (s+\lambda_j)^{-n_j} \right] .$$

Let us define the function

$$H_i(s) = \prod_{\substack{j=1 \\ j \neq i}}^{m} (s + \lambda_j)^{-n_j}.$$

Then

$$D^{n_i-1}\left[e^{st} H_i(s)\right] = e^{st}\left[(D + t)^{n_i-1} H_i(s)\right],$$

and

$$R_i = \frac{\left(\prod_{i=1}^{m} \lambda_i^{n_i}\right)}{(n_i - 1)!} \lim_{s \to -\lambda_i} \left[e^{st} (D + t)^{n_i-1} H_i(s)\right].$$

From the Binomial Theorem the term $(D + t)^{n_i-1}$ can be represented as

$$(D + t)^{n_i-1} = \sum_{K=0}^{n_i-1} \binom{n_i-1}{K} t^{n_i-1-K} D^{K}.$$

Therefore

$$R_i = \frac{\left(\prod_{i=1}^{m} \lambda_i^{n_i}\right)}{(n_i - 1)!} \lim_{s \to -\lambda_i} \left[e^{st} \sum_{K=0}^{(n_i-1)} \binom{n_i-1}{K} t^{n_i-1-K} D^{K} H_i(s)\right]$$

$$= \frac{\left(\prod_{i=1}^{m} \lambda_i^{n_i}\right)}{(n_i - 1)!} \; e^{-\lambda_i t} \left[\sum_{K=0}^{n_i - 1} \binom{n_i - 1}{K} t^{n_i - 1 - K} \frac{d^K}{ds^K} H_i(s) \right]_{s=-\lambda_i} .$$

Let us define the term

$$C(i,k) = \left[\frac{d^K}{ds^K} H_i(s) \right]_{s=-\lambda_i}$$

$$= \left[\frac{d^K}{ds^K} \prod_{\substack{j=1 \\ j \neq i}}^{m} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_i} .$$

Then

$$R_i = \left(\prod_{i=1}^{m} \lambda_i^{n_i} \right) \sum_{K=0}^{n_i - 1} \frac{C(i,K)}{K!\,(n_i - K - 1)!} \; t^{n_i - K - 1} \; e^{-\lambda_i t} \quad .$$

Since

$$f_T(t) = \sum_{i=1}^{m} R_i \quad ,$$

it follows that the convolution density is

$$f_T(t) = \left(\prod_{i=1}^{m} \lambda_i^{n_i} \right) \sum_{i=1}^{m} \sum_{K=0}^{n_i - 1} \frac{C(i,K)}{K!\,(n_i - K - 1)!} \, t^{n_i - K - 1} \, e^{-\lambda_i t} , \quad t \geqslant 0 .$$

This equation is a general equation for the density function of any convolution of exponentially distributed random variables.

Now, the survival function can be obtained as follows;

$$\bar{F}_T(t) = p(\ T > t\ )$$

$$= \int_t^{\infty} f\ (u)\ du$$

$$= \left(\prod_{i=1}^{m} \lambda_i^{n_i}\right) \sum_{i=1}^{m} \sum_{k=0}^{n_i-1} \frac{c(i,k)}{k!\,(n_i-k-1)!} \int_t^{\infty} u^{n_i-k-1} e^{-\lambda_i u} du$$

$$= \left(\prod_{i=1}^{m} \lambda_i^{n_i}\right) \sum_{i=1}^{m} \sum_{k=0}^{n_i-1} \frac{c(i,k)}{k!\ \lambda_i^{n_i-k}} \sum_{z=1}^{n_i-k} \frac{(\lambda_i t)^{z-1}}{(z-1)!}\ e^{\lambda_i t}$$

$$= \left(\prod_{i=1}^{m} \lambda_i^{n_i}\right) \sum_{i=1}^{m} \sum_{k=0}^{n_i-1} \sum_{z=1}^{n_i-k} \frac{c(i,k)}{k!\ \lambda_i^{n_i-k}} \frac{(\lambda_i t)^{z-1}}{(z-1)!} e^{-\lambda_i t}, \quad t \geqslant 0.$$

This is a general equation for the survival function of any convolution of exponentially distributed random variables.

In the next subsection, we will give three examples of the application of the equation. These examples represent all possiple convolution forms.

F. EXAMPLES ON THE GENERAL EQUATION

1. A Convolution of Dissimilar Failure Rates

Using the general equation which was derived in the last subsection, this example illustrates how to compute the survival function for a system defined by the shorthand notation

$$EXP(\lambda_1) + EXP(\lambda_2) + EXP(\lambda_3)$$

where

$$\lambda_1 \neq \lambda_2 , \quad \lambda_1 \neq \lambda_3 , \lambda_2 \neq \lambda_3 .$$

Since all the failure rates are dissimilar, it follows that the total number of distinct failure rates is

$$m = 3.$$

The multiplicity of the first distinct failure rate λ_1 is

$$n_1 = 1.$$

The multiplicity of the second distinct failure rate λ_2 is

$$n_2 = 1.$$

The multiplicity of the third distinct failure rate λ_3 is

$$n_3 = 1.$$

The survival function of the system is given by

$$\bar{F}_T(t) = \left(\prod_{i=1}^{3} \lambda_i^{n_i}\right) \sum_{i=1}^{3} \sum_{K=0}^{n_i-1} \sum_{Z=1}^{n_i-K} \frac{C(i,K)}{K!\, \lambda_i^{n_i-K}} \frac{(\lambda_i t)^{Z-1}}{(Z-1)!} e^{-\lambda_i t}$$

$$= \text{sum1} + \text{sum2} + \text{sum3},$$

where sum1 is the sum at i=1, sum2 is the sum at i=2 and sum3 is the sum at i=3.

At i=1, $n_1 = 1$, k has only the value 0, and z has only the value 1. The value of the derivative term C(1,0) is given by

$$C(1,0) = \left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j \neq 1}}^{3} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \left[\prod_{\substack{j=1 \\ j \neq 1}}^{3} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \left[\frac{1}{(s+\lambda_2)(s+\lambda_3)} \right]_{s=-\lambda_1}$$

$$= \frac{1}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} .$$

Therefore

$$\text{sum1} = \lambda_1 \lambda_2 \lambda_3 \sum_{K=0}^{1-1} \sum_{z=1}^{1-K} \frac{C(1,K)}{K! \; \lambda_1^{1-K}} \frac{(\lambda_1 t)^{z-1}}{(z-1)!} e^{-\lambda_1 t}$$

$$= \lambda_1 \lambda_2 \lambda_3 \quad \frac{C(1,0)}{0! \; \lambda_1} \frac{(\lambda_1 t)^0}{0!} e^{-\lambda_1 t}$$

$$= \lambda_1 \lambda_2 \lambda_3 \frac{1}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)\lambda_1} \quad e^{-\lambda_1 t}$$

$$= \frac{\lambda_2 \lambda_3}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} \quad e^{-\lambda_1 t} .$$

Similarly, at i=2, n_2 = 1, k has only the value 0, and z has only the value 1. The derivative term C(2,0) is

$$C(2,0) = \frac{1}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} \quad ,$$

and

$$\text{sum2} = \frac{\lambda_1 \lambda_3}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} \quad e^{-\lambda_2 t} .$$

Also, at i=3, n_3 = 1, k has only the value 0, and z has only the value 1. The derivative term C(3,0) is given by

$$C(3,0) = \frac{1}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} \quad ,$$

and

$$\text{sum3} = \frac{\lambda_1 \lambda_2}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} e^{-\lambda_3 t}.$$

By adding sum1, sum2 and sum3, the survival function of the system is

$$F(t) = \text{sum1} + \text{sum2} + \text{sum3}$$

$$= \frac{\lambda_2 \lambda_3 e^{-\lambda_1 t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} + \frac{\lambda_1 \lambda_3 e^{-\lambda_2 t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{\lambda_1 \lambda_2 e^{-\lambda_3 t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)}, \quad t \geqslant 0.$$

2. <u>A Convolution of Identical Failure Rates</u>

This example illustrates how to compute the survival function for a system defined by shorthand notation

$$\text{EXP}(\lambda) + \text{EXP}(\lambda) + \text{EXP}(\lambda).$$

This convolution has only one distinct failure rate with multiplicity 3. Therefore m=1 and n=3. The survival function of the system can be obtained as follows;

$$\bar{F}_T(t) = \left(\prod_{i=1}^{1} \lambda_i^{n_i}\right) \sum_{i=1}^{1} \sum_{k=0}^{n_i - 1} \sum_{z=1}^{n_i - k} \frac{c(i,k)}{k!\, \lambda_i^{n_i - k}} \frac{(\lambda_i t)^{z-1}}{(z-1)!} e^{-\lambda_i t}, \quad t \geqslant 0.$$

$$= \lambda^3 \sum_{K=0}^{2} \sum_{z=1}^{3-K} \frac{C(1,K)}{K!\,\lambda^{3-K}} \frac{(\lambda t)^{z-1}}{(z-1)!} \; e^{-\lambda t} .$$

Since $m = 1$, and $n_1 = 3$, it follows that $i = 1$ and $k = 0,1,2$. When $k = 0$ the derivative term $C(1,0)$ is given by

$$C(1,0) = \left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j \neq 1}}^{1} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda}$$

$$= \left[\prod_{\substack{j=1 \\ j \neq 1}}^{1} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda}$$

$$= \left[\frac{1}{(s + 0)^0} \right]_{s=-\lambda}$$

$$= 1 .$$

Since

$$\left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j \neq 1}}^{1} (s + \lambda_j)^{-n_j} \right] = 1,$$

it follows that the k-th derivative

$$\left[d^K / ds^K \prod_{\substack{j=1 \\ j \neq 1}}^{1} (s + \lambda_j)^{-n_j} \right] = 0 , \; k \geq 1.$$

That is, $C(1,k) = 0$, for $k = 1,2$. Now, the survival function for the system will become

$$\bar{F}_T(t) = \lambda^3 \sum_{z=1}^{3} \frac{C(1,0)}{\lambda^3} \frac{(\lambda t)^{z-1}}{(z-1)!} e^{-\lambda t}$$

$$= \sum_{z=1}^{3} \frac{(\lambda t)^{z-1}}{(z-1)!} e^{-\lambda t} \qquad , \; t \geq 0,$$

which is the survival function equation of the Erlang($3,\lambda$) distribution.

3. <u>A Convolution of Similar and Dissimilar Failure Rates</u>

This example illustrates how to compute the survival function for a system having some similar and some dissimilar failure rates, using the general equation.

Suppose we have a system defined by the shorthand notation

$$EXP(\lambda_1) + EXP(\lambda_1) + EXP(\lambda_2)$$

In this case, we have two distinct failure rates λ_1 and λ_2. The total number of the distinct failure rates is

$$m = 2 .$$

The first failure rate λ_1 has the multiplicity

$$n_1 = 2.$$

The second failure rate λ_2 has the multiplicity

$$n_2 = 1 .$$

The survival function equation of the convolution can be given as follows;

$$\bar{F}_T(t) = \left(\prod_{i=1}^{2} \lambda_i^{n_i}\right) \sum_{i=1}^{2} \sum_{k=0}^{n_i-1} \sum_{z=1}^{n_i-k} \frac{C(i,k)}{k!\ \lambda_i^{n_i-k}} \frac{(\lambda_i' t)^{z-1}}{(z-1)!} e^{-\lambda_i t} \quad , \ t \geqslant 0.$$

$$= \text{sum1} + \text{sum2} ,$$

where sum1 is the sum at i=1, and sum2 is the sum at i=2.

At i=1, k takes the values 0 and 1. We will evaluate the two derivatives C(1,0) and C(1,1) as follows;

$$C(1,0) = \left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j\neq 1}}^{2} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \left[\prod_{\substack{j=1 \\ j\neq 1}}^{2} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \frac{1}{(\lambda_2 - \lambda_1)} \quad ,$$

$$C(1,1) = \left[\frac{d^1}{ds^1} \prod_{\substack{j=1 \\ j \neq 1}}^{2} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \frac{-1}{(\lambda_2 - \lambda_1)^2} \quad .$$

Then

$$\text{sum1} = \lambda_1^2 \lambda_2 \sum_{K=0}^{1} \sum_{z=1}^{2-K} \frac{C(1,K)}{K!\ \lambda_1^{2-K}} \frac{(\lambda_1 t)^{z-1}}{(z-1)!} e^{-\lambda_1 t}$$

$$= \lambda_1^2 \lambda_2 \left[\frac{C(1,0)}{\lambda_1^2} e^{-\lambda_1 t} + \frac{C(1,0)\lambda_1 t}{\lambda_1^2} e^{-\lambda_1 t} + \frac{C(1,1)}{\lambda_1} e^{-\lambda_1 t} \right]$$

$$= \left[\frac{\lambda_2}{\lambda_2 - \lambda_1} + \frac{\lambda_1 \lambda_2 t}{\lambda_2 - \lambda_1} - \frac{\lambda_1 \lambda_2}{(\lambda_2 - \lambda_1)^2} \right] e^{-\lambda_1 t} .$$

At i=2, n_2 =1 and k takes only the value 0 . Thus, we have to evaluate only the derivative term C(2,0) as follows;

$$C(2,0) = \left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j \neq 2}}^{2} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_2}$$

$$= \frac{1}{(\lambda_1 - \lambda_2)^2} \quad .$$

Then

$$\text{sum2} = \lambda_1^2 \lambda_2 \quad \frac{C(2,0)}{0!\ \lambda_2} \quad \frac{(\lambda_2 t)^0}{0!} \quad e^{-\lambda_2 t}$$

$$= \lambda_1^2 \lambda_2 \frac{1}{(\lambda_1 - \lambda_2)^2 \lambda_2} \quad e^{-\lambda_2 t}$$

$$= \frac{\lambda_1^2}{(\lambda_2 - \lambda_1)} \quad e^{-\lambda_2 t} \quad .$$

Now, the survival function for the system is given by

$$\bar{F}_T(t) = \text{sum1} + \text{sum2}$$

$$= \left[\frac{\lambda_2}{\lambda_2 - \lambda_1} + \frac{\lambda_1 \lambda_2 t}{\lambda_2 - \lambda_1} - \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} \right] e^{-\lambda_1 t} + \frac{\lambda_1^2 \, e^{-\lambda_2 t}}{(\lambda_2 - \lambda_1)^2} \ , \ t \geqslant 0.$$

The three examples above illustrate how we can compute the reliability of any convolution of exponenial random variables. In practice, a computer program computes the reliability directly. A practitioner of the shorthand methodology can use the computer program in Appendix A to obtain a system's survival function by just inserting the failure rates of the system components.

III. SURVIVAL FUNCTION AND POLYNOMIAL COEFFICIENTS

In section 2, we derived a general equation for the survival function of any convolution of independent and exponentially distributed random variables. This section gives an alternative formula for the survival function, using the idea of computing the coefficients of the polynomials that accompany the exponential terms in the survival function equation. This idea was explored by Sadan Gursel [Ref. 4], adding one distribution at a time. In this section we will derive the same polynomial coefficients from the general equation developed in the previous section.

A. DETERMINATION OF THE POLYNOMIAL COEFFICIENTS

Consider a system defined by the shorthand notation

$$
\begin{aligned}
&\mathrm{EXP}(\lambda_1) + \dots + \mathrm{EXP}(\lambda_1) && (n_1 \text{ times}) \\
&+\mathrm{EXP}(\lambda_2) + \dots + \mathrm{EXP}(\lambda_2) && (n_2 \text{ times}) \\
&+ \dots\dots\dots \\
&+\mathrm{EXP}(\lambda_m) + \dots + \mathrm{EXP}(\lambda_m) && (n_m \text{ times}).
\end{aligned}
$$

This notation represents the convolution of $\sum_{i=1}^{m} n_i$ exponential random variables, where there are n identical exponential random variables having the failure rate

$$\lambda_i , i = 1, \ldots , m .$$

We expect the survival function of the convolution to have the form

$$\bar{F}(t) = A_1(t).e^{-\lambda_1 t} + A_2(t).e^{-\lambda_2 t} + \ldots + A_m(t).e^{-\lambda_m t} , \quad t \geqslant 0,$$

where

$$A_1(t) = a_{10} + a_{11} t + a_{12} t^2 + \ldots + a_{1n_1-1} t^{n_1-1} ,$$

$$A_2(t) = a_{20} + a_{21} t + a_{22} t^2 + \ldots\ldots + a_{2n_2-1} t^{n_2-1} ,$$

$$+ \ldots .$$

$$A_m(t) = a_{m0} + a_{m1} t + a_{m2} t^2 + \ldots + a_{mn_m-1} t^{n_m-1} .$$

The notation $A_i(t)$, i=1,...,m, represents a polynomial of (n_i-1)st degree. Therefore, to obtain the survival function of the convolution, we have to find all the coefficients

$$a_{ij} , \ i=1, \ldots , m, \ j= 0, \ldots , n -1 .$$

For example, the convolution represented by the shorthand notation

$$EXP(\lambda_1) + EXP(\lambda_1) + EXP(\lambda_1) + EXP(\lambda_2) ,$$

where

$$\lambda_1 \neq \lambda_2 ,$$

has a survival function represented by

$$\bar{F}(t) = A_1(t) \cdot e^{-\lambda_1 t} + A_2(t) \cdot e^{-\lambda_2 t} , \quad t \geq 0,$$

where

$$A_1(t) = a_{10} + a_{11} t + a_{12} t^2 ,$$

and

$$A_2(t) = a_{20} .$$

It is important to note that the number of exponential terms is equal to the number of dissimilar failure rates, and each exponential term has a polynomial coefficients of degree one less than the number of identical random variables having the corresponding failure rate.

The problem of computing the coefficients will be solved if we derive a general formula for the constant coefficients of the polynomial coefficients in the survival function equation.

Let us begin with the general equation which is

$$\bar{F}(t) = \left(\prod_{i=1}^{m} \lambda_i^{n_i}\right) \sum_{i=1}^{m} \sum_{K=0}^{n_i-1} \sum_{Z=1}^{n_i-K} \frac{C(i,K)\,(\lambda_i t)^{Z-1}}{K!\;\lambda_i^{n_i-K}(Z-1)!}\, e^{-\lambda_i t}, \quad t \geq 0.$$

The term $\prod_{i=1}^{m} \lambda_i^{n_i}$ can be written as

$$\prod_{i=1}^{m} \lambda_i^{n_i} = \lambda_i^{n_i} \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j} \right) .$$

Substituting the above into the general equation yields

$$\bar{F}(t) = \sum_{i=1}^{m} \lambda_i^{n_i} \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j} \right) \sum_{K=0}^{n_i-1} \sum_{Z=1}^{n_i-K} \frac{C(i,K)\,(\lambda_i t)^{Z-1}}{K!\;\lambda_i^{n_i-K}(Z-1)!}\, e^{-\lambda_i t} .$$

Rearranging the above equation yields

$$\bar{F}(t) = \sum_{i=1}^{m} e^{-\lambda_i t} \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j} \right) \sum_{K=0}^{n_i-1} \sum_{Z=1}^{n_i-K} \frac{C(i,K)}{K!\,(Z-1)!}\, \lambda_i^{K+Z-1}\, t^{Z-1} .$$

Let L = z - 1. Then the survival function will become

$$\bar{F}(t) = \sum_{i=1}^{m} e^{-\lambda_i t} \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j} \right) \sum_{K=0}^{n_i-1} \sum_{L=0}^{n_i-1-K} \frac{C(i,K)}{K!\;L!}\, \lambda_i^{K+L}\, t^{L} .$$

Let us define

$$G_i(k,L) = \frac{C(i,K)}{K!\ L!}\ \lambda_i^{K+L}\ t^{L} .$$

Then,

$$\sum_{K=0}^{n_i-1}\ \sum_{L=0}^{(n_i-1)-K} G_i(k,L) = G_i(0,0)+ \qquad \ldots\ldots. \qquad +G_i(0,n_i-1)$$

$$+G_i(1,0)+ \quad \ldots\ldots \quad +G_i(1,n_i-2)$$

$$+G_i(2,0)+\ldots+G_i(2,n_i-3)$$

$$+ \ \ldots\ldots\ldots\ldots$$

$$+ \ \ldots\ldots.$$

$$+ \ \ldots$$

$$+G_i(n_i-1,0)$$

or, equivalently

$$\sum_{K=0}^{n_i-1}\ \sum_{L=0}^{(n_i-1)-K} G_i(k,L) = G_i(0,0)+ \qquad \ldots\ldots. \qquad +G_i(n_i-1,0)$$

$$+G_i(0,1)+ \quad \ldots\ldots. \quad +G\ (n_i-2,1)$$

$$+G_i(0,2)+\ldots+G_i(n_i-3,2)$$

$$+ \ \ldots\ldots\ldots\ldots$$

$$+ \ \ldots\ldots.$$

$$+ \ \ldots$$

$$+G_i(0,n_i-1)$$

$$= \sum_{L=0}^{n_i-1} \sum_{K=0}^{(n_i-1)-L} G_i(k,L).$$

Therefore

$$\bar{F}(t) = \sum_{i=1}^{m} e^{-\lambda_i t} \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j}\right) \sum_{L=0}^{n_i-1} \sum_{K=0}^{(n_i-1)-L} \frac{C(i,K)}{K!\ L!} \lambda_i^{K+L} t^L$$

$$= \sum_{i=1}^{m} e^{-\lambda_i t} \cdot A_i(t) \ ,$$

where

$$A_i(t) = \sum_{L=0}^{n_i-1} \left(\prod_{\substack{j=1 \\ j \neq i}} \lambda_j^{n_j}\right) \sum_{K=0}^{(n_i-1)-L} \frac{C(i,K)}{K!\ L!} \lambda_i^{K+L} t^L \ .$$

Now, it easily follows that

$$a_{iL} = \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j}\right) \sum_{K=0}^{(n_i-1)-L} \frac{C(i,K)}{K!\ L!} \lambda_i^{K+L}$$

represents the coefficient of t^L for the i-th polynomial in the survival function equation, $L = 0, \ldots, n_i-1$, where

$$C(i,k) = \left[\frac{d^{k}}{ds^{k}} \prod_{\substack{j=1 \\ j \neq i}}^{m} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_i} .$$

This formula is the general formula for the coefficients and it can be considered as an alternative formula for the survival function of any convolution of exponential random variables.

It is clear from the above formula and from the general equation in Section 2 that the most difficult part in the reliability computations is the derivative term C(i,k), particularly when dealing with complex convolutions that have a long stream of failure rates with high corresponding multiplicities. We will give the the practical method for computing the term C(i,k) in Subsection C.

B. EXAMPLE

Consider a system defined by the shorthand notation

$$\mathrm{EXP}(\lambda_1) + \mathrm{EXP}(\lambda_2) + \mathrm{EXP}(\lambda_3) + \mathrm{EXP}(\lambda_3)$$

Then, the survival function for the system is given by

$$\bar{F}(t) = A_1(t).e^{-\lambda_1 t} + A_2(t).e^{-\lambda_2 t} + A_3(t).e^{-\lambda_3 t}.$$

The first distinct failure rate λ_1 has a multiplicity 1, the second distinct failure rate λ_2 has a multiplicity 1, and the third distict failure rate λ_3 has a multiplicity 2. Therefore

$$m = 3 ,$$

$$n_1 = 1 ,$$

$$n_2 = 1 ,$$

$$n_3 = 2 .$$

The polynomial $A_1(t)$ will be obtained from

$$A_1(t) = a_{10} + a_{11} t + \ldots + a_{1 n_1 - 1} t^{n_1 - 1} .$$

Substituting $n_1 = 1$ in the above equation yields

$$A_1(t) = a_{10} .$$

Since

$$a_{iL} = \Big(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j} \Big) \sum_{K=0}^{n_i - 1 - L} \frac{C(i, K)}{K! \; L!} \lambda_i^{K+L} ,$$

it follows that

$$a_{10} = \Big(\prod_{\substack{j=1 \\ j \neq 1}}^{3} \lambda_j^{n_j} \Big) \sum_{K=0}^{1-1-0} \frac{C(1, K)}{K! \; 0!} \lambda_1^{K+0}$$

$$= \lambda_2 \lambda_3^2 \sum_{K=0}^{0} \frac{C(1,K)}{K!} \lambda_1^K$$

$$= \lambda_2 \lambda_3^2 \, C(1,0) ,$$

where

$$C(1,0) = \left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j \neq 1}}^{3} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \frac{1}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)^2} .$$

Thus

$$a_{10} = \frac{\lambda_2 \lambda_3^2}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)^2} .$$

Now, the polynomial $A_1(t)$ is

$$A_1(t) = a_{10}$$

$$= \frac{\lambda_2 \lambda_3^2}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)^2} .$$

Similarly, the polynomial $A_2(t)$ is given by

$$A_2(t) = a_{20}$$

$$= \frac{\lambda_1 \lambda_3^2}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)^2} .$$

The polynomial $A_3(t)$ will be obtained from

$$A_3(t) = a_{30} + a_{31} t + \ldots + a_{3n_3-1} t^{n_3-1} .$$

Substituting $n_3 = 2$ yields

$$A_3(t) = a_{30} + a_{31} t .$$

In this case we have to evaluate the two coefficients a_{30} and a_{31}. Substituting $n_3 = 2$, $m = 3$, $L = 0$ in the general equation of the coefficients yields

$$a_{30} = \Big(\prod_{\substack{j=1\\ j\neq 3}}^{3} \lambda_j^{n_j} \Big) \sum_{K=0}^{2-1-0} \frac{C(3,K)}{K!\ 0!} \lambda_3^{K+0}$$

$$= \lambda_1 \lambda_2 \sum_{K=0}^{1} \frac{C(3,K)}{K!} \lambda_3^{K}$$

$$= \lambda_1 \lambda_2 C(3,0) + \lambda_1 \lambda_2 C(3,1) \lambda_3 ,$$

where

$$C(3,0) = \Big[\frac{d^0}{ds^0} \prod_{\substack{j=1\\ j\neq 3}}^{3} (s+\lambda_j)^{-n_j} \Big]_{s=-\lambda_3}$$

$$= \frac{1}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)} ,$$

and

$$C(3,1) = \left[\frac{d}{ds} \prod_{\substack{j=1 \\ j \neq 3}}^{3} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_3}$$

$$= - \left[\frac{1}{(\lambda_1 - \lambda_3)^2 (\lambda_2 - \lambda_3)} + \frac{1}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)^2} \right] .$$

Thus

$$a_{30} = \frac{\lambda_1 \lambda_2}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} - \frac{\lambda_1 \lambda_2 \lambda_3}{(\lambda_1 - \lambda_3)^2 (\lambda_2 - \lambda_3)} - \frac{\lambda_1 \lambda_2 \lambda_3}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)^2} .$$

Substituting $L = 1$, $n_3 = 2$, and $m = 3$ in the general equation of coefficients yields

$$a_{31} = \left(\prod_{\substack{j=1 \\ j \neq 3}}^{3} \lambda_j^{n_j} \right) \sum_{K=0}^{2-1-1} \frac{C(3,K)}{K!\ 1!} \lambda_3^{K+1}$$

$$= \lambda_1 \lambda_2 \sum_{K=0}^{0} \frac{C(3,K)}{K!} \lambda_3^{K+1}$$

$$= \lambda_1 \lambda_2 C(3,0) \lambda_3 .$$

Since

$$C(3,0) = \left[\frac{d^0}{ds^0} \prod_{\substack{j=1 \\ j \neq 3}}^{3} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_3}$$

$$= \frac{1}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} ,$$

it follows that

$$a_{31} = \lambda_1 \lambda_2 \lambda_3 \frac{1}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} .$$

Finally, the survival function of the convolution is given by

$$\bar{F}_T(t) = A_1(t).e^{-\lambda_1 t} + A_2(t).e^{-\lambda_2 t} + A_3(t)e^{-\lambda_3 t}$$

$$= a_{10}\, e^{-\lambda_1 t} + a_{20}\, e^{-\lambda_2 t} + (a_{30} + a_{31}\, t)e^{-\lambda_3 t}$$

$$= \frac{\lambda_2 \lambda_3^2 e^{-\lambda_1 t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)^2} + \frac{\lambda_1 \lambda_3^2\ e^{-\lambda_2 t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)^2}$$

$$\left[\frac{\lambda_1 \lambda_2}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} - \frac{\lambda_1 \lambda_2 \lambda_3}{(\lambda_1 - \lambda_3)^2(\lambda_2 - \lambda_3)} - \right.$$

$$\left. \frac{\lambda_1 \lambda_2 \lambda_3}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} + \frac{\lambda_1 \lambda_2 \lambda_3\, t}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)}\right] e^{-\lambda_3 t} .$$

C. THE PRACTICAL METHOD FOR COMPUTING THE DERIVATIVE TERM C(I,K)

One of the major difficulties in computing the survival function either by the general equation, or by the general formula of the coefficients, is the derivative term C(i,k),

where

$$C(i,k) = \left[\frac{d^k}{ds^k} \prod_{\substack{j=1 \\ j \neq i}}^{m} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_i} .$$

This term represents the k-th derivative for a product of m-1 terms. The difficulty in comuting C(i,k) increases with k. Since k runs from zero up to n_i -1, i=1,...,m, it follows that the maximum value of k will not exceed the maximum multiplicity of the distinct failure rates minus one. For example, if we have the two distinct failure rates λ_1 and λ_2 where λ_1 has a multiplicity n_1 and λ_2 has a multiplicity n_2 , $n_1 > n_2$, then, the maximum value of k is n_1 - 1.

In practice, due to the high accuracy (precision) required in the computations, we will limit the derivative term C(i,k) to values of k ≤ 9. This implies that the multiplicity of any of the distinct failure rates in any convolution must not exeed 10.

The following is one of the solutions which will be used in the computer program for computing the term C(i,k) (see Appendix. A, Appendix. B).

Suppose we have a convolution which has m distinct failure rates with m corresponding multiplicities . We know that

$$C(i,k) = \left[\frac{d^K}{ds^K} \prod_{\substack{j=1 \\ j \neq i}}^{m} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_i}$$

Suppose for example , we want to compute C(i,k) at i = 1, then k = 0, ... , n_1-1, and

$$C(1,k) = \left[\frac{d^K}{ds^K} \prod_{\substack{j=1 \\ j \neq 1}}^{m} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \left[\frac{d^K}{ds^K} \left(\frac{1}{(s+\lambda_2)^{n_2} \cdots (s+\lambda_m)^{n_m}} \right) \right]_{s=-\lambda_1} .$$

Our problem is how to change a derivative of a product of terms to a derivative of a sum of terms, since the later is easier in computations.

Let us define

$$f = \prod_{\substack{j=1 \\ j \neq i}}^{m} (s + \lambda_j)^{-n_j} .$$

At i=1 we obtain

$$f_1 = \prod_{\substack{j=1 \\ j \neq 1}}^{m} (s + \lambda_j)^{-n_j}$$

$$= \frac{1}{(s+\lambda_2)^{n_2}(s+\lambda_3)^{n_3}\cdots(s+\lambda_m)^{n_m}} .$$

By taking the natural logarithim on both sides, we obtain

$$\ln f_1 = -\left[\ln(s+\lambda_2)^{n_2} + \ldots + \ln(s+\lambda_m)^{n_m}\right] .$$

Now, by taking the first derivative on both sides , we obtain

$$f_1' / f_1 = -\left[\frac{n_2}{s+\lambda_2} + \cdots + \frac{n_m}{s+\lambda_m}\right]$$

or, equivalently

$$f_1' = \left[\frac{n_2}{s+\lambda_2} + \cdots + \frac{n_m}{s+\lambda_m}\right] \cdot f_1 .$$

Let us define the term

$$A_i = \sum_{\substack{j=1 \\ j \neq 1}}^{m} \frac{n_j}{s+\lambda_j} .$$

Then,

$$A_1 = \sum_{\substack{j=1 \\ j \neq 1}}^{m} \frac{n_j}{s + \lambda_j}$$

$$= \left[\frac{n_2}{s+\lambda_2} + \cdots + \frac{n_m}{s+\lambda_m} \right] .$$

Then

$$f_1' = - A_1 \cdot f_1$$

or, equivalently

$$df_1/ds = - A_1 \cdot f_1 .$$

When i=1 and k=1, the term C(i,k) is

$$C(1,1) = \left[\frac{d}{ds} \prod_{\substack{j=1 \\ j \neq 1}}^{m} (s + \lambda_j)^{-n_j} \right]_{s=-\lambda_1}$$

$$= \left[\frac{d}{ds} f_1 \right]_{s=-\lambda_1}$$

But

$$f_1' = - A_1 \cdot f_1 ,$$

so that

$$C(1,1) = \left[- A_1 \cdot f_1 \right]_{s = -\lambda_1} .$$

We can proceed in a similar manner to compute the other derivatives C(1,2),C(1,3),...,C(1,n -1) as follows;

$$f_1'' = df_1' / ds$$

$$= d/ds \left[- A_1 \cdot f_1 \right]$$

$$= - A_1' \cdot f_1 - A_1 \cdot f_1' ,$$

but

$$f_1' = - A_1 \cdot f_1 ,$$

so that

$$f_1'' = - A_1' f_1 + A_1^2 f_1$$

$$= (-1)^2 \left[A_1^2 - A_1' \right] \cdot f_1 ,$$

where

$$A_1' = d/ds \sum_{\substack{j=1 \\ j \neq 1}}^{m} \frac{n_j}{s + \lambda_j}$$

$$= \sum_{\substack{j=1 \\ j \neq 1}}^{m} \frac{d}{ds} \left(\frac{n_j}{s + \lambda_j} \right) .$$

Then, the derivative term C(1,2) is

$$C(1,2) = \left[f''_1 \right]_{s=-\lambda_1}$$

$$= \left(-1\right)^2 \left[\left(A_1^2 - A'_1 \right) f_1 \right]_{s=-\lambda_1} ,$$

and so on until we obtain the derivative term $C(1,n_1-1)$ (see Appendix.A, Appendix.B).

APPENDIX A

This appendix contains a computer program written in Fortran for the survival function of any convolution of independent and exponentially distributed random variables, using the general equation

$$\bar{F}(t) = \left(\prod_{i=1}^{m} \lambda_i^{n_i}\right) \sum_{i=1}^{m} \sum_{K=0}^{n_i-1} \frac{C(i,K)}{K!\,\lambda_i^{n_i-K}} \sum_{z=1}^{n_i-K} \frac{(\lambda_i t)^{z-1}}{(z-1)!}\, e^{\lambda_i t}, \quad t \geqslant 0.$$

A. FUNCTIONS AND SUBROUTINES USED IN THE PROGRAM

1. FUNCTION IFAC

This function computes any factorial required in the program.

2. SUBROUTINE GAMMA

This subprogram computes the incomplete gamma function with the parameters(λ_i, n_i-k) and returns the result in the variable G, where

$$G = \sum_{z=1}^{n_i - k} \frac{(\lambda_i t)^{z-1}}{(z-1)!} e^{-\lambda_i t} .$$

3. SUBROUTINE DERIV

This subprogram computes the k-th derivative C(i,k) for k<9 and returns the result in the variable C.

B. PROGRAM'S LIMITATIONS

(1) The multiplicity of any of the distinct failure rates must not exceed 10.

(2) The computer precision limit.

(3) There is no limitation on the number of distinct failure rates.

section Computer program

```
$JOB
C       LAMDAI....I-th distinct failure rate.
C       LAMDA.....Array of M distinct failure rates.
C       PRODCT....Product of all the failure rates.
C       FBAR......Survival function of the convolution.
C       T.........Time.
C       C.........Value of the derivative term C(i,k).
C       G.........Incomplete gamma function.
C       MULTI.....Multiplicity of the i-th distinct
C                 failure rate.
C       MULTIP....Array of multiplicities of the distinct
C                 failure rates.
C       M.........# of distinct failure rates.
C       I,J,K.....Loop index.
        DOUBLE PRECISION LAMDA(20), LAMDAI,PRODCT,FBAR,
     *T,C,G
        INTEGER I,J,N,M,K,MULTI,MULTIP(20)
        PRINT, 'Please enter the time of duration'
        READ, T
        PRINT, 'Please enter THE # OF dissimilar failure rates'
        READ, M
        PRINT, 'Enter the values of dissimilar failure rates'
        READ, (LAMDA(J) , J=1,M)
        PRINT,'Enter corresponding multiplicities '
        READ, (MULTIP(J) ,J=1,M)
        PRINT,'OUTPUT '
        PRINT,'======'
        WRITE(6,11)T
   11   FORMAT(' TIME = ',F10.5)
        PRINT,'          FAILURE RATE     MULTIPLICITY'
        FBAR = 0.0D0
        PRODCT = 1.0D0
        DO 10 I = 1,M
        WRITE(6,12)LAMDA(I),MULTIP(I)
   12   FORMAT(8X,F10.5,10X,I3)
        MULTI = MULTIP(I)
        PRODCT=PRODCT*LAMDA(I)**MULTI
        DO 20 J=1,MULTI
        K = J - 1
        LAMDAI = LAMDA(I)
        CALL GAMMA(MULTI,K,T,LAMDAI,G)
        CALL DERIV(I,M,LAMDA,MULTIP,K,C)
        FBAR=FBAR+C*G/(IFAC(K)*LAMDAI**(MULTI-K))
   20   CONTINUE
   10   CONTINUE
        FBAR = FBAR * PRODCT
        WRITE(6,13)T,FBAR
   13   FORMAT(1X,'P(T >',F6.2,') = ',F10.5)
        STOP
```

```
      END
      FUNCTION IFAC(N)
      JFAC=1
      IF(N.NE.0) THEN DO
      DO 30 J=1,N
      JFAC=JFAC*J
30    CONTINUE
      END IF
      IFAC=JFAC
      RETURN
      END
      SUBROUTINE GAMMA(MULTI,K,T,LAMDAI,G)
      DOUBLE PRECISION T,G,LAMDAI
      INTEGER L,K,Z,MULTI
      G=.0D0
      L=MULTI-K
      DO 40 Z=1,L
      G=G+(LAMDAI*T)**(Z-1)*DEXP(-LAMDAI*T)/IFAC(Z-1)
40    CONTINUE
      RETURN
      END
      SUBROUTINE DERIV(I,M,LAMDA,MULTIP,K,C)
      DOUBLE PRECISION A,F,C,B(9),LAMDA(10)
      INTEGER I,J,M,L,IK,MULTIP(10)
      A=.0D0
      F=1.D0
      DO 50 J=1,M
      IF(I.NE.J) THEN DO
      F=F*(LAMDA(J)-LAMDA(I))**(-MULTIP(J))
      A=A+(MULTIP(J)/(LAMDA(J)-LAMDA(I)))
      END IF
50    CONTINUE
      IF(K.GT.1) THEN DO
      IK=K-1
      DO 60 L=1,IK
      B(L)=.0D0
      DO 70 J=1,M
      IF(I.NE.J) THEN DO
      B(L)=B(L)+(((-1)**L)*MULTIP(J)*
     *((LAMDA(J)-LAMDA(I))**(-L-1)))*IFAC(L)
      END IF
70    CONTINUE
60    CONTINUE
      END IF
     IF(K.EQ.0)THEN DO
     C=F
     ELSE DO
     IF(K.EQ.1)THEN DO
     C=(-1)*A*F
     ELSE DO
     IF(K.EQ.2)THEN DO
```

```
 C=((A**2)-B(1))*F
 ELSE DO
 IF(K.EQ.3)THEN DO
 C=((-1)*(A**3)+3*A*B(1)-B(2))*F
 ELSE DO
 IF(K.EQ.4)THEN DO
 C=(A**4-6*A**2*B(1)+4*A*B(2)+3*
*B(1)**2-B(3))*F
 ELSE DO
 IF(K.EQ.5)THEN DO
 C=((-1)*(A**5)+10*(A**3)*B(1)-10*(A**2)*
*b(2)-15*A*B(1)**2+5*A*B(3)+10*B(1)*B(2)-B(4))*F
 ELSE DO
 IF(K.EQ.6)THEN DO
 C=(A**6-15*(A**4)*B(1)+20*(A**3)*
*B(2)-15*(A**2)*B(3)+6*A*B(4)-B(5)+45*
*(A**2)*(B(1)**2)-60.DO*A*B(1)*
*B(2)+15*B(1)*B(3)-15*B(1)**3+10*B(2)**2)*F
 ELSE DO
 IF(K.EQ.7)THEN DO
 C=((-1)*(A**7)+21*A**5*B(1)-35*A**4*B(2)+35*A**3*
*b(3)-21*A**2*B(4)+7*A*B(5)-B(6)-105*A**3*
*b(1)**2+210*A**2*B(1)*B(2)-105*A*B(1)*B(3)-105*
*B(1)**2*B(2)-70*a*
*B(2)**2+105*A*B(1)**3+21*B(1)*B(4)+35*B(2)*
*B(3))*F
 ELSE DO
 IF(K.EQ.8)THEN DO
 C=(A**8-28*A**6*B(1)+56*A**5*B(2)-70*A**4*B(3)+56*
*A**3*B(4)-28*A**2*B(5)+8*A*B(6)-B(7)+210*
*A**4*B(1)**2-560*a**3*b(1)*
*B(2)+420*A**2*B(1)*B(3)-168*A*B(1)*B(4)+28*
*B(1)*B(5)-420*A**2*B(1)**3+840*A*B(1)**2*B(2)+280*
*A**2*B(2)**2-210*B(1)**2*B(3)-280*A*B(2)*B(3)-280*B(1)*
*B(2)**2+56*B(2)*B(4)+35*B(3)**2+105*B(1)**4)*F
 ELSE DO
 IF(K.EQ.9)THEN DO
 C=(-A**9+36*A**7*B(1)-84*A**6*B(2)+126*A**5*B(3)-126*
*A**4*B(4)+84*A**3*B(5)-36*A**2*B(6)+9*A*B(7)-B(8)-
*378*A**5*B(1)**2+1230*A**4*B(1)*B(2)-1260*A**3*B(1)*
*b(3)+756*A**2*B(1)*B(4)-252*A*B(1)*B(5)+36*B(1)*B(6))
 C=C+1260*A**3*B(1)**3-3780*A**2*B(1)**2*
*B(2)-840*A**3*b(2)**2+1890*a*
*B(1)**2*B(3)+1260*A**2*B(2)*B(3)-378*B(4)*
*B(1)**2-404*A*B(2)*B(4)+84*B(2)*B(5)-945*a*
*b(1)**4+2520*A*B(1)*B(2)**2+1260*B(1)**3*B(2)-315*A*
*b(3)**2-1260*B(1)*B(2)*B(3)+126*B(3)*B(4)-280*B(2)**3
 C=C*F
  END IF
  END IF
  END IF
```

```
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      RETURN
      END
$ENTRY
```

section Examples subsection All failure rates are dissimilar

Shorthand notation :

```
EXP(.3)+EXP(.32)+EXP(.4)+EXP(.6)+EXP(.62)
```

```
Output
======
TIME =     2.00000

        FAILURE RATE     MULTIPLICITY
          0.30000              1
          0.32000              1
          0.40000              1
          0.60000              1
          0.62000              1

P(T >  2.00) =     0.99817
```

subsection All failure rates are identical

Shorthand notation :

```
EXP(2.1) + ... + EXP(2.1)       (10 times)
```

Output

```
======

TIME =    3.20

        FAILURE RATE     MULTIPLICITY

          2.10000            10

P(T >  3.20) =    0.85772
```

subsection Some similar and some dissimilar failure rates

Shorthand notation :

```
    + EXP(6.5)+ ... + EXP(6.5)     (5 times)
+ EXP(4.5)+ ... + EXP(4.5)     (5 times)
+ EXP(3.5)+ ... + EXP(3.5)     (5 times)
+ EXP(2.5)+ ... + EXP(2.5)     (5 times)
+ EXP(1.5)+ ... + EXP(1.5)     (5 times)
+ EXP(.50)+ ... + EXP(.50)     (5 times)
```

```
 Output
 ======
 TIME =    8.00000

        FAILURE RATE     MULTIPLICITY
            6.50000            5
            4.50000            5
            3.50000            5
            2.50000            5
            1.50000            5
            0.50000            5

 P(T >  8.00) =    0.99894
```

cms cms

APPENDIX B

This appendix contains an alternative computer program written in Fortran for computing the survival function of any convolution of independent and exponentially distributed random variables, using the general equation for the coefficients of the polynomials that accompany the exponential terms in the survival function equation

$$\bar{F}(t) = \sum_{i=1}^{m} e^{-\lambda_i t} \sum_{L=0}^{n_i-1} a_{iL}\, t^{L} \quad , \quad t \geqslant 0 \; ,$$

where

$$a_{iL} = \Big(\prod_{\substack{j=1 \\ j\neq i}}^{m} \lambda_j^{n_j} \Big) \sum_{K=0}^{(n_i-1)-L} \frac{C(i,K)}{K!\; L!} \lambda_i^{K+L} \; .$$

A. FUNCTIONS AND SUBROUTINES USED IN THE PROGRAM

1. Function IFAC

This function computes any factorial required in the program.

2. Subroutine COEFF

This subprogram computes the coeffient of t for the i-th polynomial in the survival function equation and returns the result in the variable CO,

where

$$CO = \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \lambda_j^{n_j}\right) \sum_{K=0}^{n_i - 1 - L} \frac{C(i, K)}{K!\ L!} \lambda_i^{K+L} .$$

3. SUBROUTINE DERIV

This subprogram computes the k-th derivative C(i,k) for k<9 and returns the result in the variable C.

B. PROGRAM'S LIMITATIONS :

(1) The multiplicity for any of the distinct failure rates must not exceed 10.

(2) The computations must be within the computer precision limit.

(3) No limitation on the number of distinct failure rates.

section Computer program

```
$JOB
C      LAMDAI....i-th distinct failure rate.
C      LAMDA.....Array of M distinct failure rates.
C      MULTI.....Multiplicity of the i-th distinct
C                failure rate.
C      MULTIP....Array of multiplicities of the distinct
C                failure rates.
C      PRODCT....Product of the distinct failure rates.
C      FBAR......Survival function of the convolution.
C      T.........Time.
C      C.........Value of the derivative term C(i,k).
C      CO........Coeffetiont of the polynomials.
C      M.........# of distinct failure rates.
C      I,J,K.....Loop index.
C
      DOUBLE PRECISION LAMDA(20), LAMDAI,PRODCT,FBAR,
     *T,C,CO
      INTEGER I,J,N,M,K,MULTI,MULTIP(20)
      PRINT, 'Please enter the time of duration'
      READ, T
      PRINT, 'Please Enter # of dissimilar failure rates'
      READ, M
      PRINT, 'Enter the values of dissimilar failure rates'
      READ, (LAMDA(J) , J=1,M)
      PRINT,'Enter corresponding multiplicities '
      READ, (MULTIP(J) ,J=1,M)
      PRINT,'OUTPUT '
      PRINT,'======'
      WRITE(6,11)T
   11 FORMAT(' TIME = ',F10.5)
      PRINT,'         FAILURE RATE     MULTIPLICITY'
      FBAR = 0.0D0
      DO 10 I = 1,M
   10 WRITE(6,12)LAMDA(I),MULTIP(I)
   12 FORMAT(8X,F10.5,10X,I3)
      DO 20 I = 1,M
      MULTI = MULTIP(I)
      WRITE(6,13)I
   13 FORMAT(1X,'POLYNOMIALS COEFFICIENT FOR LAMDA',I2,' is :')
      DO 30 LL = 1,MULTI
      L = LL - 1
      CALL COEFF(I,L,M,MULTIP,LAMDA,CO)
      WRITE(6,14)I,L,CO
   14 FORMAT(3X,'a',I1,I1,' = ',F25.12)
      FBAR=FBAR+T**L*CO*DEXP(-LAMDA(I)*T)
   30 CONTINUE
   20 CONTINUE
      WRITE(6,15)T,FBAR
   15 FORMAT(/,1X,'P(T >',F6.2,') = ',F20.12)
```

```
      STOP
      END
      FUNCTION IFAC(N)
      JFAC=1
      IF(N.NE.0) THEN DO
      DO 40 J=1,N
      JFAC=JFAC*J
40    CONTINUE
      END IF
      IFAC=JFAC
      RETURN
      END
      SUBROUTINE COEFF(I,L,M,MULTIP,LAMDA,CO)
      DOUBLE PRECISION C,CO,LAMDA(10),PRODCT
      INTEGER I,L,M,K,KK,LL,MULTIP(10)
      PRODCT = 1.0D0
      DO 50 J = 1,M
      IF(I.NE.J) THEN DO
      PRODCT=PRODCT*LAMDA(J)**MULTIP(J)
      END IF
50    CONTINUE
      CO = 0.0D0
      LL = MULTIP(I) - L
      DO 60 KK = 1,LL
      K = KK - 1
      CALL DERIV(I,M,LAMDA,MULTIP,K,C)
      CO = CO + C*PRODCT*LAMDA(I)**(K+L)/IFAC(K)/IFAC(L)
60    CONTINUE
      RETURN
      END
      SUBROUTINE DERIV(I,M,LAMDA,MULTIP,K,C)
      DOUBLE PRECISION A,F,C,B(9),LAMDA(10)
      INTEGER I,J,M,L,IK,MULTIP(10)
      A=.0D0
      F=1.D0
      DO 70 J=1,M
      IF(I.NE.J) THEN DO
      F=F*(LAMDA(J)-LAMDA(I))**(-MULTIP(J))
      A=A+(MULTIP(J)/(LAMDA(J)-LAMDA(I)))
      END IF
70    CONTINUE
      IF(K.GT.1) THEN DO
      IK=K-1
      DO 80 L=1,IK
      B(L)=.0D0
      DO 90 J=1,M
      IF(I.NE.J) THEN DO
      B(L)=B(L)+(((-1)**L)*MULTIP(J)*
     *((LAMDA(J)-LAMDA(I))**(-L-1)))*IFAC(L)
      END IF
90    CONTINUE
```

```
80  CONTINUE
    END IF
    IF(K.EQ.0)THEN DO
    C=F
    ELSE DO
    IF(K.EQ.1)THEN DO
    C=(-1)*A*F
    ELSE DO
    IF(K.EQ.2)THEN DO
    C=((A**2)-B(1))*F
    ELSE DO
    IF(K.EQ.3)THEN DO
    C=((-1)*(A**3)+3*A*B(1)-B(2))*F
    ELSE DO
    IF(K.EQ.4)THEN DO
    C=(A**4-6*A**2*B(1)+4*A*B(2)+3*
   *B(1)**2-B(3))*F
    ELSE DO
    IF(K.EQ.5)THEN DO
    C=((-1)*(A**5)+10*(A**3)*B(1)-10*(A**2)*
   *B(2)-15*A*B(1)**2+5*A*B(3)+10*B(1)*B(2)-B(4))*F
    ELSE DO
    IF(K.EQ.6)THEN DO
    C=(A**6-15*(A**4)*B(1)+20*(A**3)*
   *B(2)-15*(A**2)*B(3)+6*A*B(4)-B(5)+45*
   *(A**2)*(B(1)**2)-60.D0*A*B(1)*
   *B(2)+15*B(1)*B(3)-15*B(1)**3+10*B(2)**2)*F
    ELSE DO
    IF(K.EQ.7)THEN DO
    C=((-1)*(A**7)+21*A**5*B(1)-35*A**4*B(2)+35*A**3*
   *B(3)-21*A**2*B(4)+7*A*B(5)-B(6)-105*A**3*
   *B(1)**2+210*A**2*B(1)*B(2)-105*A*B(1)*B(3)-105*
   *B(1)**2*B(2)-70*A*
   *B(2)**2+105*A*B(1)**3+21*B(1)*B(4)+35*B(2)*
   *B(3))*F
    ELSE DO
    IF(K.EQ.8)THEN DO
    C=(A**8-28*A**6*B(1)+56*A**5*B(2)-70*A**4*B(3)+56*
   *A**3*B(4)-28*A**2*B(5)+8*A*B(6)-B(7)+210*
   *A**4*B(1)**2-560*A**3*B(1)*
   *B(2)+420*A**2*B(1)*B(3)-168*A*B(1)*B(4)+28*
   *B(1)*B(5)-420*A**2*B(1)**3+840*A*B(1)**2*B(2)+280*
   *A**2*B(2)**2-210*B(1)**2*B(3)-280*A*B(2)*B(3)-280*B(1)*
   *B(2)**2+56*B(2)*B(4)+35*B(3)**2+105*B(1)**4)*F
    ELSE DO
    IF(K.EQ.9)THEN DO
    C=(-A**9+36*A**7*B(1)-84*A**6*B(2)+126*A**5*B(3)-126*
   *A**4*B(4)+84*A**3*B(5)-36*A**2*B(6)+9*A*B(7)-B(8)-
   *378*A**5*B(1)**2+1230*A**4*B(1)*B(2)-1260*A**3*B(1)*
   *B(3)+756*A**2*B(1)*B(4)-252*A*B(1)*B(5)+36*B(1)*B(6))
    C=C+1260*A**3*B(1)**3-3780*A**2*B(1)**2*
```

```
     *B(2)-840*A**3*B(2)**2+1890*A*
     *B(1)**2*B(3)+1260*A**2*B(2)*B(3)-378*B(4)*
     *B(1)**2-404*A*B(2)*B(4)+84*B(2)*B(5)-945*A*
     *B(1)**4+2520*A*B(1)*B(2)**2+1260*B(1)**3*B(2)-315*A*
     *B(3)**2-1260*B(1)*B(2)*B(3)+126*B(3)*B(4)-280*B(2)**3
      C=C*F
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      END IF
      RETURN
      END
$ENTRY
```

section Examples subsection All failure rates are dissimilar

Shorthand notation :

```
EXP(.3)+EXP(.32)+EXP(.4)+EXP(.6)+EXP(.62)

Output
======

TIME =     2.00000

        FAILURE RATE     MULTIPLICITY

          0.30000              1

          0.32000              1

          0.40000              1

          0.60000              1

          0.62000              1

POLYNOMIAL COEFFICIENT FOR LAMDA 1 IS :

  a10 =           248.000000000000

POLYNOMIAL COEFFICIENT FOR LAMDA 2 IS :

  a20 =          -332.142857142857

POLYNOMIAL COEFFICIENT FOR LAMDA 3 IS :

  a30 =           101.454545454545

POLYNOMIAL COEFFICIENT FOR LAMDA 4 IS :

  a40 =           -70.857142857143

POLYNOMIAL COEFFICIENT FOR LAMDA 5 IS :

  a50 =            54.545454545454

P(T >  2.00) =       0.998171698511
```

subsection All failure rates are identical :

Shorthand notation :

```
EXP(2.1) + ... + EXP(2.1)       (10 times)
```

Output

======

```
TIME =    3.20000

        FAILURE RATE     MULTIPLICITY

          2.10000            10

POLYNOMIAL COEFFICIENT FOR LAMDA 1 ARE :

  a10 =              1.000000000000

  a11 =              2.100000000000

  a12 =              2.205000000000

  a13 =              1.543500000000

  a14 =              0.810337500000

  a15 =              0.340341750000

  a16 =              0.119119612500

  a17 =              0.035735883750

  a18 =              0.009380669484

  a19 =              0.002188822880

P(T >  3.20) =         0.857717853017
```

subsection Some similar and some dissimilar failure rates

Shorthand notation :

```
+ EXP(6.5)+ ... + EXP(6.5)     (5 times)
+ EXP(4.5)+ ... + EXP(4.5)     (5 times)
+ EXP(3.5)+ ... + EXP(3.5)     (5 times)
+ EXP(2.5)+ ... + EXP(2.5)     (5 times)
+ EXP(1.5)+ ... + EXP(1.5)     (5 times)
+ EXP(.50)+ ... + EXP(.50)     (5 times)
```

```
Output
======
TIME =    8.00000

        FAILURE RATE     MULTIPLICITY
          6.50000             5
          4.50000             5
          3.50000             5
          2.50000             5
          1.50000             5
          0.50000             5

POLYNOMIALS COEFFICIENTS FOR LAMDA 1 ARE :
  a10 =           -0.033655633163
  a11 =           -0.015984638484
  a12 =           -0.002968083834
  a13 =           -0.000255685577
```

a14 = -0.000008633539

POLYNOMIAL COEFFICIENTS FOR LAMDA 2 ARE :

a20 = 78824.407500664440

a21 = 28606.965700048750

a22 = 4242.082627828356

a23 = 308.298280647481

a24 = 9.469913057431

POLYNOMIAL COEFFICIENTS FOR LAMDA 3 ARE :

a30 = -1939257.342698666000

a31 = -833405.255639648600

a32 = -199021.555234616700

a33 = -18294.927183023870

a34 = -1641.852439502142

POLYNOMIAL COEFFICIENTS FOR LAMDA 4 ARE :

a40 = 2591869.114320258000

a41 = -558729.118591437400

a42 = 340359.641443397200

a43 = -14083.275902489360

a44 = 4142.139971320400

POLYNOMIAL COEFFICIENTS FOR LAMDA 5 ARE :

a50 = -736153.955891001200

a51 = 310180.360064797400

a52 = -58447.530039485250

a53 = 5362.158719218832

a54 = -297.897706623269

POLYNOMIAL COEFFICIENTS FOR LAMDA 6 ARE :

a60 = 4718.810424379766

a61 = -1542.762826749752

a62 = 203.749271368373

a63 = -12.977288263887

a64 = 0.350737520646

P(T > 8.00) = 0.998940885595

LIST OF REFERENCES

1. Repicky, John J., Jr., An Introduction to a Reliability Shorthand, M. S. Thesis, Naval Postgraduate School, Monterey, CA., 1981.

2. Ross, S. M., Applied Probability Models With Optimization Applications, Holden-Day, 1970.

3. Gino Moretti, Functions of A Complex Variable, Prentice-Hall, 1968.

4. Sadan Gursel, Some Computer Algorithems to Implement A Reliability Shorthand, M. S. Thesis, Naval Postgraduate School, Monterey, CA., 1982.

5. Esary, J. D., Unpublished course notes and problem sets OA 4302, Naval PostGraduate School, Monterey, CA., 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Ministry of Defence Training Departement Cairo, Egypt.	5
2.	Professor J. D. Esary, Code 55Ey Operations Research Departement Naval PostGraduate School Monterey, CA. 93943	2
3.	Professor P. A. Jacobs, Code 55Jc Operations Research Departement Naval PostGraduate School Monterey, CA. 93943	2
4.	Superintendant Attn : Liberary, Code 0142 Naval PostGraduate School Monterey, CA. 93943	2
5.	Col. Ali S. Abdelsamad 10 EL-Roda Street Appt #1 Heliopolis Cairo, Egypt.	3
6.	Ltcol. Mamdouh M. Lotfy 22C Abdel-Rahman Basha Street Helwaan Cairo, Egypt.	3
7.	Defence Technical Information Center Cameron Station Alexandria, Virginia 22314	2
8.	Col. Abdelazeez Elbasyoni Naval Postgraduate School SMC #1689 Monterey, CA. 93943	1

212112

Thesis
L82572 Lotfy
c.1 Survival function
of hypo-exponential
distributions.