Abstract: The usage of software is getting increased day to day and it is very much essential to maintain the reliability of software. Various software reliability growth models have been proposed for estimating the reliability of software and referred to as software reliability growth models. It is difficult to estimate the parameters as most of the functions are non-linear. Several parameter estimation methods are suggested to estimate the unknown parameters. An estimation methods based on Least Square and Maximum Likelihood are discussed in this paper. The main objective of this paper is to compare the parameter estimation techniques considering the numerical example of IBM failure data set for time domain data in assessing the reliability of software. The paper documents the Statistical Process Control mechanism to monitor when the failure occurs using control charts and the results are exhibited for three different growth models.

Keywords: Software Reliability Growth Model, control charts, Statistical Process Control (SPC), Least Square Estimation (LSE), Maximum Likelihood Estimation (MLE)

1. INTRODUCTION

Various software reliability growth models have been proposed during the last three decades to assess the reliability of the software. An optimized estimation of parameters of software reliability growth models is the matter of concern as the accurate prediction of reliability depends on these parameters. Society is becoming ever more dependent on software and software-controlled systems and there is always a need of high quality software. Software reliability is the probability of failure free operation of software in a specified environment during specified duration [9]. Reliability of software is not deterministic, as a faulty program can still give correct output sometimes. Hence reliability is best measured probabilistically [3]. Unlike hardware, software does not age, wear out or rust. Unreliability of software is mainly due to bugs or design faults in the software [12].

Software reliability is dynamic & stochastic. The exact value of product reliability is never precisely known at any point in its lifetime. The study of software reliability can be categorized into three parts: Modelling, Measurement & improvement. Many Models exist, but no single model can capture a necessary amount of software characteristics. It cannot be directly measured, so other related factors are measured to estimate software reliability. Software reliability improvement is necessary & hard to achieve [1],[7].

One of the common practices in manufacturing industry is Statistical Process Control (SPC). The investigation on quantitative mechanisms as an aid to control process variation led to the application of the SPC since 1930’s. The idea of applying SPC to software development however, is exemplified mainly by Capability Maturity Model (CMM) in mid-90’s. Statistical Process Control (SPC) is an analytical decision making tool for monitoring and controlling manufacturing processes. SPC determines when a statistically significant change has taken place in the process or when a seemingly significant change is just due to
chance causes. Control charts are an essential tool of continuous quality control. Control charts are also used to determine the capability of the process. They can help identify special or assignable causes for factors that impede peak performance [13], [6].

2. RELATED WORK

Research is going on rigorously on software reliability engineering over the past several years and numerous statistical models had come into existence with which the reliability of software can be estimated. Almost all the models that were present for assessing the software reliability depend on the observations of the failures of software product. With the help of these failure data that can time domain or interval domain, the reliability can be estimated[4],[5].

A number of analytical models have been proposed to address the problem of software reliability measurement. [8],[9]. These approaches are based mainly on the failure history of software and can be classified into Time Between Failure Models, Failure Count Models, Fault Seeding Models and Input Domain Based Models.

2.1. Maximum Likelihood Estimation (MLE)

In statistics, Maximum Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model given data. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model’s parameters. The parameters are estimated using MLE by considering the mean and variance as parameters and finding particular parametric values that make the observed results most probable. The Maximum Likelihood Estimates (MLE’s) of the parameters are computed using Newton Raphson iterative method for the given time domain cumulative data. Most probably MLE selects the set of values that maximizes the likelihood function as parameters. The mean value \( m(t) \) and the successive differences of the \( m(t) \)'s can be computed once the unknown parameters are estimated.

2.2. Least Square Estimation

Least Square Estimation (LSE) is a standard approach in regression analysis and widely used in many fields for function fit and parameter estimation. The LSE may be simple but very useful in estimating model parameters when there are more equations than unknowns. Least squares mean that the overall solution minimizes the sum of the squares of the errors made in the results of every single equation. Hence, the LSE methods should be comprehensively studied in the context of software reliability engineering. It finds values of the parameters such that the sum of the squares of the difference between the fitting function and the experimental data is minimized. Mathematically, the LSE method concerns in determining the value of the unknown parameters that minimizes the quantity.

A problem of curve fitting, which is unrelated to normal regression theory and MLE estimates of coefficients but uses identical formulas, is called the method of least squares. This method is based on minimizing the sum of the squared distance from the best fit line and the actual data points. It just so happens that finding the MLE’s for the coefficients of the regression line also involves these sums of squared distances [15].

3. PROPOSED WORK–MATHEMATICAL DERIVATION

Parameter estimation plays a significant role in software reliability prediction. In this section, it is explained how the unknown parameters are estimated using MLE and LSE technique for the specified reliability growth models like Pareto Type II and Dunane Model. The parameters are estimated taking into consideration the IBM time domain failure data set. Statistical Process Control mechanism is implemented to estimate
the reliability. Control charts are developed for both MLE and LSE parameters applying all the three growth models on IBM live time domain failure data set.

3.1. Maximum Likelihood Estimation (MLE)

3.1.1. Parameter Estimation–Mathematical Derivation for Pareto Type II model

The mean value of Pareto Type II Software Reliability Growth Model based on NHPP is given by

\[ m(t) = a \left[ 1 - \frac{c^b}{(t + c)^b} \right], \]

where \( m(t)/a \) is the cumulative distribution function of Pareto Type II distribution.

Substituting the expressions for \( m(t) \) in the above equations, taking logarithms, differentiating the equations with respect to \( 'a' \), \( 'b' \), \( 'c' \) and equating to zero and simplifying the equations we get [14].

\[
a = \frac{n(t + c)^b}{(t + c)^b - c^b} \quad (1)
\]

\[
g(b) = \frac{\partial \log L}{\partial b} = \frac{n \log \left( \frac{1}{(t + 1)^b - 1} \right)}{b} + n \sum_{i=1}^{n} \log (t_i + 1) \quad (2)
\]

\[
g'(b) = \frac{\partial^2 \log L}{\partial b^2} = -n \log \left( \frac{1}{(t + 1)^b - 1} \right) \left[ (t + 1)^b \log (t + 1) \right] - \frac{n}{b^2} \quad (3)
\]

\[
g(c) = \frac{\partial \log L}{\partial c} = \frac{n}{(t + c)} + \frac{n}{c} \sum_{i=1}^{n} \frac{2}{t_i + c} \quad (4)
\]

\[
g'(c) = \frac{\partial^2 \log L}{\partial c^2} = \frac{-n}{(t + c)^2} - \frac{n}{c^2} + \frac{2}{\sum_{i=1}^{n} (t_i + c)^2} \quad (5)
\]

The values of \( 'b' \) and \( 'c' \) in the above specified equations can be obtained using Newton Raphson Method. Solving the above equations simultaneously, yields the point estimates of the parameters \( a \), \( b \) and \( c \) [14]. These equations are to be solved iteratively and their solutions in turn when substituted in the log likelihood equation of \( 'a' \) would give analytical solution for the MLE of \( 'a' \). The values of \( b \) and \( c \) are obtained by applying numerical methods.

3.1.2. Parameter Estimation – Mathematical Derivation for M-O model

The mean value function of M-O model is given by

\[ m(t) = a \log (1 + bt) \quad (6) \]

In order to assess the software reliability, the unknown parameters \( a \), \( b \) are to be computed and necessary equations are to be found. An experimental data that is obtained with \( 'n' \) independent observations, \( t_1, t_2, \ldots, t_n \) for IBM data set is taken into consideration. The Log likelihood function (LLF) for time domain data is given by,

\[ L = e^{-m(t)} \prod_{i=1}^{n} m'(t_i) \quad (7) \]
After calculating the natural log values and performing partial differentiation the equations for \( a \) is obtained and the equations of \( g'(b) \) is derived. Applying the newtonraphson method on a data set the values of \( 'a' \) and \( 'b' \) can be obtained.

\[
a = 1/n \log(1 + bt) \quad (8)
\]

\[
g(b) = \frac{\partial \log l}{\partial b} = -t \frac{\log(1 + bt)}{n} + \frac{n}{b} - \sum_{i=1}^{n} t_i \quad (9)
\]

\[
g'(b) = \frac{\partial^2 \log l}{\partial b^2} = -t \left[ \frac{b - t \log(1 + bt)}{(1 + bt)^2} \right] + \frac{n}{b^2} + \sum_{i=1}^{n} \frac{t_i^2}{(1 + bt_i)^2} \quad (10)
\]

### 3.1.3. Parameter Estimation–Mathematical Derivation for Dunane Model

In order to assess the software reliability \( a, b \) are to be known or they are to be estimated from software failure data. We conduct an experiment and obtain ‘\( n \)’ independent observations, \( t_1, t_2, \ldots, t_n \). The likelihood function (LLF) for time domain data is given by,

\[
L = e^{-m(t)} \prod_{i=1}^{n} m(t_i) \quad (11)
\]

Applying the natural logarithm on both the sides, the value of \( a \) can be,

\[
a = - \frac{n}{t^b} \quad (12)
\]

The same procedure is followed for calculating the value of \( b \) which is followed above, hence

\[
g(b) = \frac{\partial (\log l)}{\partial b} = abt^{b-1} + \frac{n}{b} + \sum_{i=1}^{n} \log t_i \quad (13)
\]

\[
\frac{\partial^2}{\partial b^2} (\log l) = -n \frac{n}{t^b} \quad (14)
\]

### 3.2. Least Square Estimation

#### 3.2.1. Parameter Estimation–Mathematical Derivation for Pareto Type II model

Given data \{ \( x_1, y_1 \), \ldots, \( x_n, y_n \) \}, we may define the error associated to saying

\[
y = y = ax^2 + bx + c \quad (15)
\]

\[
E(a, b) = \sum_{i=1}^{n} (y_i - (ax_i^2 + bx_i + c))^2 \quad (16)
\]

This is just \( N \) times the variance of the data set \[ \{ y_1 - (ax_1^2 + bx_1 + c), \ldots, y_n - (ax_n^2 + bx_n + c) \} \]. It makes no difference whether or not we study the variance or \( N \) times the variation as our error, and note that the error is a function of two variables.

The goal is to find the values of \( a \) and \( b \) that minimize the error. In multivariable calculus we learn that this requires us to find the values of \( a \) and \( b \) such that

\[
\partial E / \partial a = 0, \partial E / \partial b = 0
\]

To obtain the values of \( a \) and \( b \), we need to find out the values of \( \text{Det} \, M \),
Similarly, in order to calculate the values of $\det M_1$, $\det M_2$ and $\det M_3$, we have to replace the rows of the matrix with the equations which are at the right hand side,

In order to get the values of the parameters of $a$, $b$

$$a = \frac{\det M_1}{\det M}, \quad b = \frac{\det M_2}{\det M}, \quad c = \frac{\det M_3}{\det M}$$

Hence the values of $a$, $b$ and $c$ are calculated by using the Cramer’s rule for least Square technique.

### 3.2.2 Parameter Estimation–Mathematical Derivation for MO Model model

Given data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we may define the error associated to saying $y = ax + b$ By,

$$E(a, b) = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

This is just $N$ times the variance of the data set $\{y_1 - (ax_1 + b), \ldots, y_n - (ax_n + b)\}$.

The goal is to find the values of $a$ and $b$ that minimize the error In multivariable calculus we learn that this requires us to find the values of $(a, b)$ such that

$$\frac{\partial E}{\partial a} = 0, \quad \frac{\partial E}{\partial b} = 0$$

Differentiating $E(a, b)$ we get

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} 2(y_i - (ax_i + b))(-x_i)$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{n} 2(y_i - (ax_i + b))(-x_i)$$

Setting $\frac{\partial E}{\partial a} = 0, \frac{\partial E}{\partial b} = 0$ and dividing by 2, we get

$$\sum_{i=1}^{n} (y_i - (ax_i + b))(-x_i) = 0$$

$$\sum_{i=1}^{n} (y_i - (ax_i + b)) = 0$$

The values of $a$ and $b$ which minimize the error are obtained.

To obtain the values of $a$ and $b$, we need to find out the values of $\det M$ for $M1$ & $M2$

$$\det M = \left(\sum_{n=1}^{N} x_n^2 \sum_{n=1}^{N} x_n \right) - \left(\sum_{n=1}^{N} x_n^2 \sum_{n=1}^{N} x_n \right)$$

$$a = \frac{\det M_1}{\det M}, \quad b = \frac{\det M_2}{\det M}$$

Hence the values of $a$ and $b$ are calculated by using the Cramer’s rule for least square technique.
4. PARAMETER ESTIMATION

The parameters $a$, $b$ and $c$ are computed for the data set specified in Table 1 using well known Cramer’s method LSE and MLE method. The values of the unknown parameters are obtained by solving the equations. The Pareto Type II, MO and Dunane model are applied on IBM live data set and the values of unknown parameters are computed and the results are exhibited in the Table 2 for the growth models.

Testing is performed on an on-line data entry software package developed by IBM which is reported by Ohba(1984) and the inter failure times are recorded[10]. The table given below exemplifies both the observation time and the cumulative number of errors.

Table 1
IBM online Data Entry Software Testing (Pham Hoang, 2005)[11]

<table>
<thead>
<tr>
<th>No. of Error</th>
<th>Inter-failure time</th>
<th>Cumulative failure time</th>
<th>No. of Error</th>
<th>Inter-failure time</th>
<th>Cumulative failure time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>22</td>
<td>125</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>19</td>
<td>10</td>
<td>25</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>32</td>
<td>11</td>
<td>19</td>
<td>169</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>43</td>
<td>12</td>
<td>30</td>
<td>199</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>58</td>
<td>13</td>
<td>32</td>
<td>231</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>70</td>
<td>14</td>
<td>25</td>
<td>256</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>88</td>
<td>15</td>
<td>40</td>
<td>296</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>103</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2
Parameter Estimation using MLE and LSE for Pareto Type II, MO Model and Dunane Model

<table>
<thead>
<tr>
<th>Software Reliability Growth Models</th>
<th>MLE Technique For IBM Data set</th>
<th>Least Square Technique For IBM Data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto Type-II Model</td>
<td>$a = 36.9118$</td>
<td>$a = 0.960488$</td>
</tr>
<tr>
<td></td>
<td>$b = 0.999637$</td>
<td>$b = 4.535762$</td>
</tr>
<tr>
<td></td>
<td>$c = 432.19166$</td>
<td>$c = 7.5802197$</td>
</tr>
<tr>
<td>M-O Model</td>
<td>$a = 0.362708$</td>
<td>$a = 19.903571$</td>
</tr>
<tr>
<td></td>
<td>$b = 0.775626$</td>
<td>$b = 35.961904$</td>
</tr>
<tr>
<td></td>
<td>$c = 0.0525382$</td>
<td>$c = 15.325712$</td>
</tr>
<tr>
<td>Dunane Model</td>
<td>$a = 0.9936567$</td>
<td>$b = 0.21576$</td>
</tr>
<tr>
<td></td>
<td>$b = 0.999637$</td>
<td>$b = 4.535762$</td>
</tr>
</tbody>
</table>

5. COMPUTATION OF CONTROL LIMITS FOR PARETO TYPE-II, MO MODEL & DUNANE MODEL

The software failure process is demonstrated through Mean Value Control charts for the time domain failure data set specified in Table 1. For monitoring the reliability of software process, cumulative failures have been taken into consideration. The MLE and LS estimates of unknown parameters of Pareto Type II, MO Model &Dunane model are computed for the cumulative failure data set. The mean value for each failure can be computed by substituting the values of ‘$a$’, ‘$b$’ and ‘$c$’.

The mean value function of the Pareto Type II, MO &Dunane models are given by

\[ m(t) = a \left[ 1 - \frac{c^b}{(t+c)^b} \right] \]

Mean Value of MO Model is given by

\[ m(t) = a \log(1 + bt) \]

Mean Value of Dunane Model is given by

\[ m(t) = at^b \]
Control limits for Pareto Type II Model  |  Control limits for MO Model  |  Control limits for Dunane Model
--- | --- | ---
\[ t = e^{Log_{-1} \frac{1}{b} \log 0.00135} - c = T_U \] | \[ t = (e^{0.99865} - 1)/b = T_U \] | \[ t = e^{1/b(\log 0.99865)} = T_U \]
\[ t = e^{Log_{-1} \frac{1}{b} \log 0.00135} - c = T_L \] | \[ t = (e^{0.00135} - 1)/b = T_L \] | \[ t = e^{1/b(\log 0.00135)} = T_L \]
\[ t = e^{Log_{-1} \frac{1}{b} \log 0.5} - c = T_C \] | \[ t = (e^{0.5} - 1)/b = T_C \] | \[ t = e^{1/b(\log 0.5)} = T_C \]

Procedure for computing the control limits:

Step 1: Remove the term ‘\(a\)’ from the mean value function
Step 2: Equate the remaining function successively to 0.99865, 0.00135 & 0.5 and solve for ‘\(t\)’, to get the 3 sigma corresponding control limits, upper control limit(UCL), Lower Control Limit(LCL) and Central limit(CL)

The control limits in the mean value control chart indicates that the point above the \(m(t_u)\) (UCL) is an alarm signal. A point below the \(m(t_L)\) (LCL) is an indication of better quality of software. The point within the control limits indicates that the process is stable.

### Table 3

<table>
<thead>
<tr>
<th>Estimation method</th>
<th>No of Samples</th>
<th>Growth Model</th>
<th>Estimated Parameters</th>
<th>Control Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(A)</td>
<td>(b)</td>
</tr>
<tr>
<td>MLE</td>
<td>15</td>
<td>Pareto Type II</td>
<td>30.91483</td>
<td>0.99953</td>
</tr>
<tr>
<td>MO Model</td>
<td>19.9</td>
<td>Pareto Type II</td>
<td>0.362708</td>
<td>0.775626</td>
</tr>
<tr>
<td>Dunane Model</td>
<td>0.0525382</td>
<td>0.9936567</td>
<td>0.052467</td>
<td>0.497792</td>
</tr>
<tr>
<td>LSE</td>
<td>15</td>
<td>Pareto Type II</td>
<td>0.9604</td>
<td>4.5357</td>
</tr>
<tr>
<td>MO Model</td>
<td>19.9</td>
<td>Pareto Type II</td>
<td>19.903571</td>
<td>35.961904</td>
</tr>
<tr>
<td>Dunane Model</td>
<td>15.3</td>
<td>0.21576</td>
<td>15.325712</td>
<td>0.21576</td>
</tr>
</tbody>
</table>

### 6. RESULTS & OBSERVATIONS

The basic idea of software reliability modeling is to predict the reliability of software with its failure data set. The parameters are estimated with MLE and LSE method for Pareto Type II, MO and Dunane Reliability Growth models and failure control charts are developed considering failure live data set of IBM and the results are compared.

#### 6.1. Developing Failure Control Charts using Maximum Likelihood Parameter Estimation

The control charts are generated applying Statistical Process Control mechanism on the time domain IBM data by computing the \(m(t)\)’s and successive differences of \(m(t)\)’s. The control charts are generated for both the parameter estimation techniques and observed when the failures are detected. Given ‘\(n\)’ inter-failure data, the values of \(m(t)\) are computed for each failure. The successive differences of the \(m(t)\)’s are taken which gives \(n-1\) values. The control graph is generated by taking the inter-failure times on X-axis and the
Figure 1: Control Chart for Pareto Type II model using MLE

Figure 2: Control Chart for MO model using MLE

Figure 3: Control Chart for Dunane model using MLE
n-1 values of $m(t)$'s on Y-axis and three control lines parallel to X-axis at $m(TL)$, $m(TC)$, $m(TU)$ respectively. To assess the reliability of software, the control charts are generated for the given inter-failure time data sets.

6.2. Developing Failure Control Charts using Least Square Parameter Estimation

![Figure 4: Control Chart for Pareto Type II using LSE](image)

![Figure 5: Control Chart for M-O model using Least square](image)

![Figure 6: Control Chart for Dunanemodel using Least square](image)
7. COMPARATIVE ANALYSIS

The values of the unknown parameters obtained through two parameter estimation techniques are tabulated in the table 8. The results obtained through control charts for the Pareto Type II, MO Model and Dunane growth models applying MLE and LSE parameter estimation techniques applying on IBM failure data set are tabulated in the table 4. It is observed that better results are obtained when the parameters are estimated with Least Square Estimation.

<table>
<thead>
<tr>
<th>Software Reliability Growth Model</th>
<th>MLE Technique</th>
<th>Least Square Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto Type-II</td>
<td>No Error Detected</td>
<td>2nd point</td>
</tr>
<tr>
<td>M-O Model</td>
<td>Continuous</td>
<td>2nd point</td>
</tr>
<tr>
<td>Dunane Model</td>
<td>Alarm signal</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

Parameter Estimation of software reliability growth model is one of the important aspect reliability engineering. As most of the growth models are non-homogeneous it is difficult to estimate. The parameters are assessed based on MLE and LSE and the control charts are developed with which the software reliability can be assessed. An experiment with two typical estimation techniques on three different models demonstrates that failures have been detected at early stages and hence shows that it has good applicability.

References