Inflectoin s-shaped model: Order Statistics

Dr. R. Satya Prasad[1], K. Prasada Rao[2] and G. Krishna Mohan[3]

Associate Professor, Dept. of Computer Science & Engg., Acharya Nagrjuna University
Nagarjuna Nagar, Guntur, Andhrapradesh, India.
profrsp@gmail.com

Professor, Dept. of MCA, CMRIT, Kundalahalli, Bangalore.
prasadarao.k@cmrit.ac.in

Reader, Dept. of Computer Science, P.B.Siddhartha college
Vijayawada, Andhrapradesh, India.
km_mm_2000@yahoo.com

ABSTRACT-Software Reliability is defined as the probability that the software will work without failure for a specified period of time. Software Reliability Growth Model (SRGM) is one of the most well-known theoretical models for estimating and predicting software reliability in development and maintenance. In SRGM, software reliability growth is defined by the mathematical relationship between the time span of program testing and the cumulative number of detected faults. SRGMs can estimate the total number of initial faults in the target software by applying the well-known SRGM described by non-homogeneous Poisson processes (NHPPs) to the bug data. In this paper we proposed a control mechanism based on order statistics of the cumulative quantity between observations of time domain failure data using mean value function of inflection S-shaped model, which is Non Homogenous Poisson Process (NHPP) based. The Maximum Likelihood Estimation (MLE) method is used to derive the point estimators of a two-parameter distribution.

Index terms: SRGM, NHPP, MLE, order statistics, time domain, grouping, inflection S-shaped.

1 Introduction

To improve and to understand the logic behind process control methods, it is necessary to give some thought to the behavior of sampling and of averages. If the length of a single failure interval is measured, it is clear that occasionally a length will be found which is towards one end of the tails of the process's normal distribution. This occurrence, if taken on its own, may lead to the wrong conclusion that the process requires adjustment. If, on the other hand, a sample of four or five is taken, it is extremely unlikely that all four or five failure interval lengths will lie towards one extreme end of the distribution. If, therefore, we take the average or length of four or five failure intervals, we shall have a much more reliable indicator of the state of the process. Sample failure interval length or means will vary with each sample taken, but the variation will not be as great as that for single failure.

In the distribution of mean lengths from samples of four failures, the standard deviation of the means, called the standard error of means, and denoted by the symbol SE, is half the standard deviation of the individual Time between failures taken from the process. When n=4, half the spread of the parent distribution of individual TBF. The smaller spread of the distribution of sample averages provides the basis for a useful means of detecting changes in processes. Any change in the process mean, unless it is extremely large, will be difficult to detect from individual results alone. A large number of individual readings would, therefore, be necessary before such a change was confirmed.

Noise is inherent in the software failure data. A transformation of data is needed to smooth out the noise. Malaiya et al., (1990) tried to smooth out the noise by data grouping. They noticed that the smoothing improves quality initially as the size increases and becomes worse as the group size is large. Order statistics deals with properties and applications of ordered random variables and functions of these variables. The use of order statistics is significant when failures are frequent or inter failure time is less.

1.1 Order Statistics

Order statistics deals with properties and applications of ordered random variables and functions of these variables. The use of order statistics is significant when inter failure time is less or failures are frequent. Let A denote a continuous random variable with probability density function(pdf), f(a) and cumulative distribution function(cdf), F(a), and let (A1 , A2 , …, Ak) denote a random sample of size k drawn on A. The original sample observations may be unordered with respect to magnitude. A transformation is required to produce a corresponding ordered sample. Let (A(1) , A(2) , …, A(k)) denote the ordered random sample such that A(1) < A(2) < … < A(k); then (A(1), A(2), …, A(k)) are collectively known as the order statistics derived from the parent A. The various

ISSN 2229-5518

distributional characteristics can be known from Balakrishnan and Cohen(1991).

1.2 SPC

Statistical Process Control (SPC) is about using control charts to manage software development efforts, in order to effect software process improvement. The practitioner of SPC tracks the variability of the process to be controlled. The early detection of software failures will improve the software reliability. The selection of proper SPC charts is essential to effective statistical process control implementation and use. The SPC chart selection is based on data, situation and need (MacGregor, 1995). Many factors influence the process, resulting in variability. The causes of process variability can be broadly classified into two categories, viz., assignable causes and chance causes.

The control limits can then be utilized to monitor the failure times of components. After each failure, the time can be plotted on the chart. If the plotted point falls between the calculated control limits, it indicates that the process is in the state of statistical control and no action is warranted. If the point falls above the UCL, it indicates that the process average, or the failure occurrence rate, may have decreased which results in an increase in the item between failures. This is an important indication of possible process improvement. If this happens, the management should look for possible causes for this improvement and if the causes are discovered then action should be taken to maintain them. If the plotted point falls below the LCL, It indicates that the process average, or the failure occurrence rate, may have increased which results in a decrease in the failure time. This means that process may have deteriorated and thus actions should be taken to identify and causes may be removed. It can be noted here that the whole process involves the mathematical model of the mean value function and knowledge about its parameters. If the parameters are known they can be taken as they are for the further analysis is. If the parameters are not known, they have to be estimated using a simple data by any admissible, efficient method of distribution. This is essential because the control limits depend on mean value function which intern depend on the parameters.

The control limits for the chart are defined in such a manner that the process is considered to be out of control when the time to observe exactly one failure is less than LCL or greater than UCL. Our aim is to monitor the failure process and detect any change of the intensity parameter. When the process is in control, there is a chance for this to happen and it is commonly known as false alarm. The traditional false alarm probability is to set to be 0.27% although any other false alarm probability can be used. The actual acceptable false alarm probability should in fact depend on the actual product or process (Swapna et al., 1998).

Rational sub grouping of data

We have seen that a subgroup or a sample is a small set of observations on a process parameter or its output, taken together in time. The two major problems with regard to choosing a subgroup relate to its size and the frequency of sampling. The smaller the subgroup, the less opportunity there is for variation within it, but the larger the sample size the narrower the distribution of the means, and the more sensitive they become to detecting change (Oakland, 2008).

A rational subgroup is a sample of items or measurements selected in a way that minimizes variation among the items or results in the sample, and maximizes the opportunity for detecting variation between the samples. With a rational subgroup, assignable or special causes of variation are not likely to be present, but all of the effects of the random or common causes are likely to be shown. Generally, subgroups should be selected to keep the chance for differences within the group to a minimum, and yet maximize the chance for the subgroups to differ from one another. At this stage it is clear that, in any type of process control charting system, nothing is more important than the careful selection of subgroups. The software failure data is in the form of <failure number, failure time>. By grouping a fixed number of data into one, the noise values may compensate each other for that period and thus the noise inherent in the failure data is reduced significantly (Malaiya et al., 1990).

2 NHPP MODEL

The Non-Homogenous Poisson Process (NHPP) based software reliability growth models (SRGMs) are proven to be quite successful in practical software reliability engineering (Musa et al., 1987). The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time point. Model parameters can be estimated by using Maximum Likelihood Estimate (MLE).

Let $\{N(t), t \geq 0\}$ denote a counting process representing the cumulative number of faults detected by the time 't'. An SRGM based on an NHPP with the mean value function (MVF) $m(t)$ is the mean value function, representing the expected number of software failures by time 't' can be formulated as (Lyu, 1996).

$$P\{N(t)=n\} = \frac{m(t)^n}{n!} e^{-m(t)}, \quad n = 0,1,2,\ldots$$

$\lambda(t)$ is the failure intensity function, which is proportional to the residual fault content. In a more general NHPP SRGM $\lambda(t)$ can be expressed as

$\lambda(t)=\frac{dm(t)}{dt}=b(t)\left[a(t)-m(t)\right]$. Where, $a(t)$ is the time-dependent fault content function which includes the initial and introduced faults in the program and $b(t)$ is the time-dependent fault detection rate. In software reliability, the initial number of faults and the fault detection rate are always unknown. The maximum likelihood technique can be used to evaluate the unknown parameters.

Inflection S-shaped model

Software reliability growth models (SRGM's) are useful to assess the reliability for quality management and testing-progress control of software development. They have been grouped into two classes of models concave and S-shaped. The most important thing about both models is that they have the same asymptotic behavior, i.e., the defect detection rate decreases as the number of defects detected (and repaired) increases, and the total number of defects detected asymptotically approaches a finite value. The inflection S-shaped model was proposed by Ohba in 1984. This model assumes that the fault detection rate increases throughout a test period. The model has a parameter, called the inflection rate, which indicates the ratio of detectable faults to the total number of faults in the target software. True, sustained exponential growth cannot exist in the real world. Eventually all exponential, amplifying processes will uncover underlying stabilizing processes that act as limits to growth. The shift from exponential to asymptotic growth is known as sigmoidal, or S-shaped, growth.

Ohba models the dependency of faults by postulating the following assumptions:

- Some of the faults are not detectable before some other faults are removed.
- The detection rate is proportional to the number of detectable faults in the program.
- Failure rate of each detectable fault is constant and identical.
- All faults can be removed.

Assuming [Ohba 1984b]: $b(t)=\frac{b}{1+\beta e^{-bt}}$

This model is characterized by the following mean value function:

$$m(t)=\frac{a}{1+\beta e^{-bt}}\left(1-e^{-bt}\right)$$

where 'b' is the failure detection rate, and 'β' is the inflection factor. The failure intensity function is given as:

$$\lambda(t)=\frac{abe^{-bt}\left(1+\beta\right)}{\left(1+\beta e^{-bt}\right)^2}.$$

3 Illustrating the MLE method.

Based on the inter failure data given in Data Set#1 & Set#2, we demonstrate the software failures process through failure control chart. We used cumulative time between failures data for software reliability monitoring. The use of cumulative quality is a different and new approach, which is of particular advantage in reliability. '$\hat{a}$' and '$\hat{b}$' are Maximum Likely hood Estimates (MLEs) of parameters 'a' and 'b' and the values can be computed using iterative method for the given cumulative time between failures data.

The probability density function of a two-parameter inflection S-shaped model has the form:

$$f(t)=\frac{be^{-bt}\left(1+\beta\right)}{\left(1+\beta e^{-bt}\right)^2}.$$

The corresponding cumulative distribution function is:

$$F(t)=\frac{\left(1-e^{-bt}\right)}{1+\beta e^{-bt}}.$$

Mean Value Function of the model is $m(t)=\frac{a\left(1-e^{-bt}\right)}{1+\beta e^{-bt}}$.

For rth order statistics, the mean value function is expressed as $m^r(t)=\left(\frac{a\left(1-e^{-bt}\right)}{1+\beta e^{-bt}}\right)^r$.

The failure intensity function of rth order is given as: $\lambda^r(t)=\left[m^r(t)\right]'$.

To estimate 'a' and 'b' , for a sample of n units, first obtain the likelihood function:

The Likelihood function $L=e^{-m^r(t_n)}\prod_{i=1}^{n}\lambda^r(t_i)$.

Take the natural logarithm on both sides, The Log Likelihood function is given as (Pham, 2006):

$$\log L=\sum_{i=1}^{n}\log\left[\lambda^r(t_i)\right]-m^r(t_n)$$

$$=\sum_{i=1}^{n}\log\left(\frac{a^r rbe^{-(bt_i)^{r-1}}\left(1-e^{-bt_i}\right)^{r-1}\left(1+\beta\right)}{\left(1+\beta e^{-bt_i}\right)^{r+1}}\right)-\left(\frac{a^r\left[1-e^{-(bt_n)}\right]^r}{\left[1+\beta e^{-(bt_n)}\right]^r}\right) \quad (3.1)$$

The parameter 'a' is estimated by taking the partial derivative w.r.t 'a' and equating to '0'.

$$a^r = n\left(\frac{1+\beta e^{-bt_n}}{1-e^{-(bt_n)}}\right)^r \qquad (3.2)$$

The parameter 'b' is estimated by iterative Newton Raphson Method using $b_{n+1} = b_n - \frac{g(b_n)}{g'(b_n)}$, which is substituted in finding 'a'. Where $g(b)$ & $g'(b)$ are expressed as follows.

Taking the Partial derivative w.r.t 'b' and equating to '0'.

$$g(b) = \frac{n}{b} + \sum_{i=1}^{n}\left(-t_i + \frac{t_i e^{-(bt_i)}(r-1)}{1-e^{-(bt_i)}} + \frac{t_i e^{-(bt_i)}(r+1)\beta}{1+\beta e^{-(bt_i)}}\right) - \frac{nrt_n e^{-bt_n}(1+\beta)}{\left(1-e^{-bt_n}\right)\left(1+\beta e^{-bt_n}\right)} \qquad (3.3)$$

Again partially differentiating w.r.t 'b' and equating to 0.

$$g'(b) = -\frac{n}{b^2} + \sum_{i=1}^{n}\left(-t_i^2 e^{-(bt_i)}\right)\left[\frac{r-1}{\left(1-e^{-(bt_i)}\right)^2} + \frac{(r+1)\beta}{\left(1+\beta e^{-bt_i}\right)^2}\right] + nrt_n^2(1+\beta)\left[\frac{e^{-(bt_n)}\left(\left(1-e^{-(bt_n)}\right)+e^{-(bt_n)}\left(1+\beta e^{-(bt_n)}\right)\right)}{\left(1-e^{-(bt_n)}\right)^2\left(1+\beta e^{-bt_n}\right)^2}\right] \qquad (3.4)$$

4 Time Domain Data Sets for Ordered Statistics

Data Set #1, #2: The Real-time Control System Data

The data sets were listed in "DATA" directory Containing 45 industry project failure data sets in the Handbook of Software Reliability Engineering (Lyu, 1996).

Table 4.1: Data Set #1 [SYS 1]

FNo	TBF	FNo	TBF	FNo	TBF	FNo	TBF
1	3	35	227	69	529	103	108
2	30	36	65	70	379	104	0
3	113	37	176	71	44	105	3110
4	81	38	58	72	129	106	1247
5	115	39	457	73	810	107	943
6	9	40	300	74	290	108	700
7	2	41	97	75	300	109	875
8	91	42	263	76	529	110	245
9	112	43	452	77	281	111	729
10	15	44	255	78	160	112	1897
11	138	45	197	79	828	113	447
12	50	46	193	80	1011	114	386
13	77	47	6	81	445	115	446
14	24	48	79	82	296	116	122
15	108	49	816	83	1755	117	990
16	88	50	1351	84	1064	118	948
17	670	51	148	85	1783	119	1082
18	120	52	21	86	860	120	22
19	26	53	233	87	983	121	75
20	114	54	134	88	707	122	482
21	325	55	357	89	33	123	5509
22	55	56	193	90	868	124	100
23	242	57	236	91	724	125	10
24	68	58	31	92	2323	126	1071
25	422	59	369	93	2930	127	371
26	180	60	748	94	1461	128	790
27	10	61	0	95	843	129	6150
28	1146	62	232	96	12	130	3321
29	600	63	330	97	261	131	1045
30	15	64	365	98	1800	132	648
31	36	65	1222	99	865	133	5485
32	4	66	543	100	1435	134	1160
33	0	67	10	101	30	135	1864
34	8	68	16	102	143	136	4116

Table 4.2: Data Set #2 [SYS 2]

FNo	TBF	FNo	TBF	FNo	TBF	FNo	TBF
1	760	33	87	65	276	97	15
2	758	34	19	66	1	98	1960
3	33	35	29	67	999	99	60
4	6	36	0	68	30	100	19
5	22	37	5	69	495	101	20
6	14	38	360	70	472	102	79
7	42	39	10	71	344	103	24
8	4	40	11	72	550	104	1737
9	84	41	100	73	131	105	7984
10	15	42	252	74	47	106	10
11	221	43	460	75	92	107	20
12	14	44	179	76	863	108	338
13	15	45	3	77	991	109	250
14	41	46	24	78	35	110	1682
15	1	47	253	79	9549	111	212
16	153	48	163	80	249	112	287
17	409	49	54	81	607	113	56
18	54	50	137	82	83	114	4973
19	24	51	328	83	614	115	3500
20	44	52	3	84	352	116	59
21	180	53	9	85	673	117	98
22	397	54	12	86	4179	118	2439
23	19	55	18	87	111	119	1812
24	145	56	9	88	75	120	6203
25	36	57	75	89	407	121	385
26	54	58	15	90	288	122	3500
27	1337	59	366	91	894	123	4892
28	163	60	428	92	1314	124	687
29	8	61	212	93	845	125	62
30	1	62	115	94	55	126	2796
31	17	63	264	95	409	127	3268
32	16	64	269	96	36	128	3845

ISSN 2229-5518

5 Estimated Parameters and Their Control Limits

The estimated parameters and the calculated control limits of the Mean Value Chart for Data Set#1 to Data Set #2 with the false alarm risk, α = 0.0027 are given in Table 5.1. Using the estimated parameters and the estimated limits, we calculated the control limits UCL=$m(t_U)$, CL=$m(t_C)$ and LCL=$m(t_L)$. They are used to find whether the software process is in control or not. The estimated values of 'a' and 'b' and their control limits for both 4th-order and 5th-order statistics are as follows.

Table 5.1: Parameter estimates and Control limits of 4 & 5 order

Data Set	Order	Estimated Parameters		Control Limits		
		a	b	UCL	CL	LCL
11	4	4.913480	0.000008	4.906847	2.456738	0.006631
	5	3.777591	0.000009	3.772491	1.888794	0.005098
12	4	2.497335	0.000005	2.493964	1.248667	0.003370
	5	1.998837	0.000007	1.996139	0.999418	0.002698

6 Distribution of Time Between Failures

The mean value successive differences of rth order cumulative time between failures data of the considered data sets are tabulated in Table 6.1 to 6.4. Considering the mean value successive differences on y axis, failure numbers on x axis and the control limits on Mean Value chart, we obtained figure 6.1 to 6.4. A point below the control limit $m(t_L)$ indicates an alarming signal. A point above the control limit $m(t_U)$ indicates better quality. If the points are falling within the control limits it indicates the software process is in stable.

Table 6.1: Successive differences of 4 order mean values of Data Set 1

F. No	4-order C_TBF	m(t)	SD
1	227	0.008491	**0.008104**
2	444	0.016595	**0.011741**
3	759	0.028336	**0.011046**
4	1056	0.039382	**0.034434**
5	1986	0.073816	**0.025398**
6	2676	0.099214	**0.064139**
7	4434	0.163353	**0.023688**
8	5089	0.187041	**0.010812**
9	5389	0.197853	**0.035549**
10	6380	0.233402	**0.037990**
11	7447	0.271392	**0.016817**
12	7922	0.288209	**0.081864**
13	10258	0.370073	**0.031756**
14	11175	0.401829	**0.047528**
15	12559	0.449357	**0.031567**
16	13486	0.480924	**0.060385**
17	15277	0.541309	**0.036065**
18	16358	0.577374	**0.063652**
19	18287	0.641026	**0.074081**
20	20567	0.715107	**0.113216**
21	24127	0.828323	**0.133852**
22	28460	0.962175	**0.118284**
23	32408	1.080459	**0.151915**
24	37654	1.232374	**0.121858**
25	42015	1.354233	**0.007717**
26	42296	1.361950	**0.160977**
27	48296	1.522927	**0.096905**
28	52042	1.619832	**0.035550**
29	53443	1.655382	**0.075916**
30	56485	1.731299	**0.148659**
31	62651	1.879958	**0.052366**
32	64893	1.932324	**0.247962**
33	76057	2.180287	**0.256267**
34	88682	2.436553	

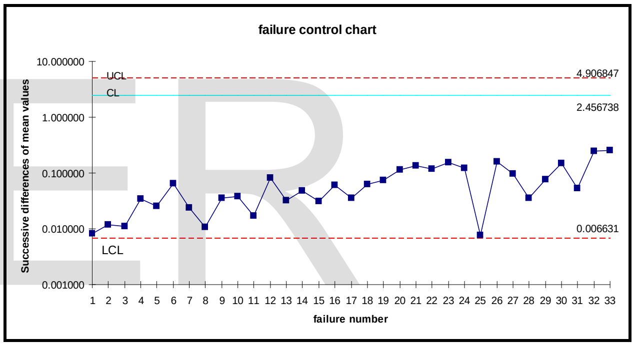

Figure 6.1: 4 order Mean Value Chart for Data Set 1

Table 6.2: Successive differences of 5 order mean values Data Set 1

F. No	5-order C_TBF	m(t)	SD
1	342	0.011058	**0.0073874**
2	571	0.018446	**0.0127743**
3	968	0.031220	**0.0325679**
4	1986	0.063788	**0.0352675**
5	3098	0.099055	**0.0611080**
6	5049	0.160163	**0.0085353**
7	5324	0.168699	**0.0325975**
8	6380	0.201296	**0.0386494**
9	7644	0.239946	**0.0736335**
10	10089	0.313579	**0.0265272**
11	10982	0.340106	**0.0463728**
12	12559	0.386479	**0.0622315**
13	14708	0.448711	**0.0421367**
14	16185	0.490847	**0.0443142**
15	17758	0.535161	**0.0777152**
16	20567	0.612877	**0.1429209**

17	25910	0.755798	**0.0889949**
18	29361	0.844793	**0.2033849**
19	37642	1.048177	**0.1018462**
20	42015	1.150024	**0.0764395**
21	45406	1.226463	**0.0876204**
22	49416	1.314084	**0.0825227**
23	53321	1.396606	**0.0648931**
24	56485	1.461499	**0.1217613**
25	62661	1.583261	**0.2138904**
26	74364	1.797151	**0.1697761**
27	84566	1.966927	

22	31174	0.345651	**0.029752**
23	34077	0.375403	**0.013650**
24	35422	0.389053	**0.020684**
25	37476	0.409737	**0.018562**
26	39336	0.428299	**0.081416**
27	47688	0.509716	**0.023114**
28	50119	0.532829	**0.079603**
29	58707	0.612432	**0.093561**
30	69259	0.705993	**0.080090**
31	78723	0.786083	**0.080635**
32	88694	0.866718	

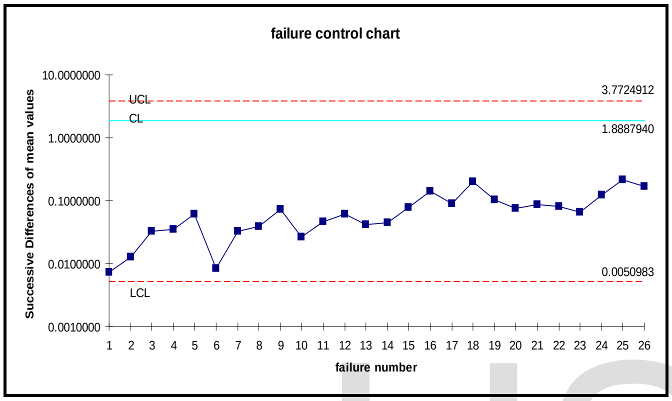

Figure 6.2: 5 order Mean Value Chart for Data Set 1

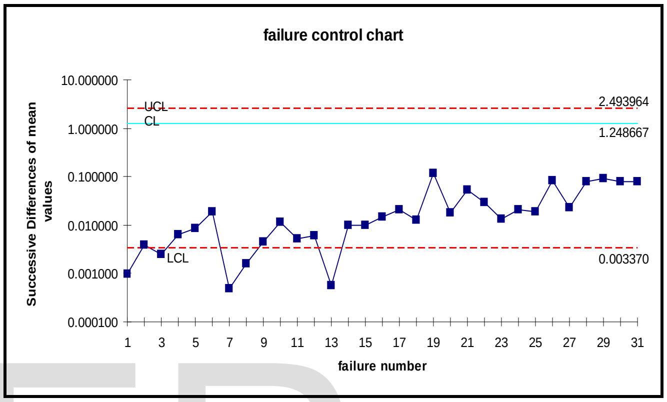

Figure 6.3: 4 order Mean Value Chart for Data Set 2

Table 6.3: Successive differences of 4 order mean values of Data Set 2

F. No	4-order C_TBF	M(t)	SD
1	1557	0.018451	**0.000968**
2	1639	0.019419	**0.003940**
3	1973	0.023359	**0.002474**
4	2183	0.025833	**0.006245**
5	2714	0.032078	**0.008690**
6	3455	0.040767	**0.018548**
7	5045	0.059315	**0.000488**
8	5087	0.059804	**0.001568**
9	5222	0.061372	**0.004479**
10	5608	0.065851	**0.011464**
11	6599	0.077315	**0.005108**
12	7042	0.082423	**0.006006**
13	7564	0.088428	**0.000552**
14	7612	0.088980	**0.010136**
15	8496	0.099115	**0.009822**
16	9356	0.108937	**0.014842**
17	10662	0.123779	**0.020997**
18	12523	0.144776	**0.012697**
19	13656	0.157473	**0.118045**
20	24480	0.275518	**0.017551**
21	26136	0.293069	**0.052582**

Table 6.4: Successive differences of 5 order mean values of Data Set 2

F. No	5-order C_TBF	m(t)	SD
1	1579	0.020936	**0.0020966**
2	1738	0.023033	**0.0038449**
3	2030	0.026878	**0.0089787**
4	2714	0.035856	**0.0101523**
5	3491	0.046009	**0.0202711**
6	5054	0.066280	**0.0021669**
7	5222	0.068447	**0.0049699**
8	5608	0.073417	**0.0127422**
9	6602	0.086159	**0.0080472**
10	7233	0.094206	**0.0047036**
11	7603	0.098910	**0.0113067**
12	8496	0.110216	**0.0142908**
13	9632	0.124507	**0.0248726**
14	11629	0.149380	**0.0143520**
15	12793	0.163732	**0.1383243**
16	24480	0.302056	**0.0263464**
17	26809	0.328402	**0.0558953**
18	31869	0.384298	**0.0377882**
19	35386	0.422086	**0.0220518**
20	37476	0.444138	**0.0999356**
21	47320	0.544073	**0.0224413**
22	49620	0.566515	**0.0849005**

ISSN 2229-5518

23	58648	0.651415	**0.0935747**
24	69259	0.744990	**0.0786178**
25	78785	0.823608	

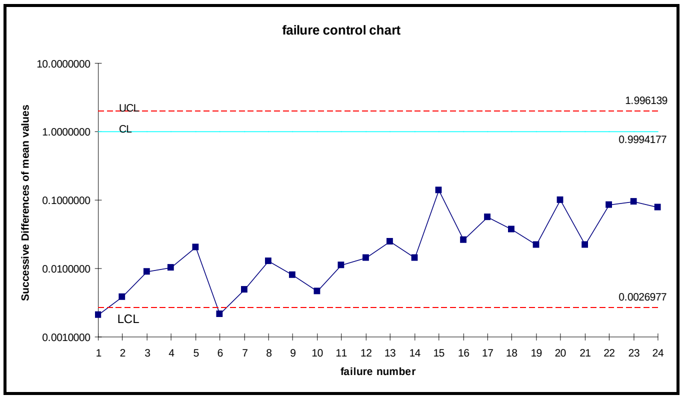

Figure 6.4: 5 order Mean Value Chart for Data Set 2

7 CONCLUSION

The 4 and 5 order Time Between Failures are plotted through the estimated mean value function against the r^{th} failure (i.e 4 & 5) serial order. The parameter estimation is carried out by Newton Raphson Iterative method. Data Set#1 and Data Set#2 have shown that, some of the mean value successive differences have gone out of calculated control limits i.e below LCL at different instants of time. In Data Set#1, the successive differences of mean values are within the control limits for both 4^{th} and 5^{th} order. In Data Set#2, the failure process is detected at an early stage for both 4th and 5th order i.e. in between UCL and LCL, which indicates a stable process control. The early detection of software failure will improve the software Reliability. Hence we conclude that our method of estimation and the control chart are giving a Positive recommendation for their use in finding out preferable control process or desirable out of control signal. When the successive differences of failures are less than LCL, it is likely that there are assignable causes leading to significant process deterioration and it should be investigated. On the other hand, when the successive differences of failures have exceeded the UCL, there are probably reasons that have lead to significant improvement.

REFERENCES

[1]. Balakrishnan. N., Clifford Cohen. A., (1991). "Order Statistics and Inference", Academic Press, INC. page 13.

[2]. MacGregor, J.F., Kourti, T., (1995). "Statistical process control of multivariate processes". Control Engineering Practice Volume 3, Issue 3, 403-414.

[3]. Musa, J.D., Iannino, A., Okumoto, k., (1987). "Software Reliability: Measurement Prediction Application". McGraw-Hill, New York.

[4]. Ohba, M., (1984). "Software reliability analysis model". IBM J. Res. Develop. 28. 428-443.

[5]. Pham. H., (2006). "System software reliability", Springer.

[6]. Swapna S. Gokhale and Kishore S.Trivedi, (1998). "Log- Logistic Software Reliability Growth Model". The 3rd IEEE International Symposium on High-Assurance Systems Engineering. IEEE Computer Society.

[7]. Lyu, M. R., (1996). "Handbook of Software Reliability Engineering". McGraw-Hill publishing, ISBN 0-07-039400-8.

[8]. Oakland. J., (2008). "Statistical Process Control", Sixth Edition, Butterworth-Heinemann, Elsevier.

[9]. Malaiya, Y. K., Sumit sur, Karunanithi, N. and Sun, Y. (1990). "Implementation considerations for software reliability", 8th Annual Software ReliabilitySymposium, June 1.

Author's profile:

Dr. R. Satya Prasad Received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Andhra Pradesh. He received gold medal from Acharya Nagarjuna University for his outstanding performance in a first rank in Masters Degree. He is currently working as Associative Professor and H.O.D, in the Department of Computer Science & Engineering, Acharya Nagarjuna University. His current research is focused on Software Engineering. He published 70 research papers in National & International Journals.

Mr. K.Prasad Rao, Working as a Professor, Dept. of M.C.A, CMR Institute of Technology. He is having 22 years of experience as a Head & Lecturer in Computer Science field. He Published papers in 3 National and 1 International journals. He is pursuing Ph.D at Acharya Nagarjuna University. His research interests lies in Software Engineering.

Mr. G. Krishna Mohan, working as a Reader in the Department of Computer Science, P.B.Siddhartha College, Vijayawada. He obtained his M.C.A degree from Acharya Nagarjuna University, M.Tech from JNTU, Kakinada, M.Phil from Madurai Kamaraj University and pursuing Ph.D from Acharya Nagarjuna University. He qualified AP State Level Eligibility Test. His research interests lies in Data Mining and Software Engineering. He published 21 research papers in various National and International journals.