Improving the predictability of a popular software reliability growth model

Peter A. Keiller

Howard University

Washington DC.

Motivation for the Research

- Once a software product is shipped, it is estimated to cost over $10,000 to correct one software defect
- Assessment of software quality is often obtained using software reliability growth models
- There are over 40 software reliability growth models used by software development shops to access the quality of software
- There is currently no established “best model”
- There is currently little methodology for model selection

Purpose of the Study

Improve the estimation procedures and predictive performance for a member of an important large family of software reliability growth models

Research Composition

The Research was arranged in two (2) stages:

- **Stage 1:** involved the collection of failure data and the selection of the popular family of software reliability models
- **Stage 2:** addressed the improvement in estimation and performance for a selected member of the software reliability models with four phases of analysis

Collection of Failure Data

- 41 sets of failure data were used in the research
- The data sets were compiled from published articles and correspondence with prominent researchers in the software reliability field in Europe and the U.S.A.
- The data sets represent a wide variety of software ventures ranging from commercial to military projects

Data Set Breakdown

	Number of Failures			
CPU TIME (Secs)	Small(<50)	Medium (50-200)	Large (>200)	Total
Small (0-100K)	5 (M:1;L:1)	7 (M:5;L:0)	1 (M:0;L:0)	13
Medium (100K-10 mill)	6 (M:2;L:3)	8 (M:1;L:6)	4 (M:1;L:3)	18
Large (> 10 mill)	1 (M:1;L:0)	5 (M:5;L:0)	4 (M:3;L:1)	10
Total	12	20	9	41

M - obtained from Musa, L - obtained from Littlewood through CSR

Types of Software Reliability Models

- **Times Between Failure Models**

  Times between failures follow a distribution whose parameters depend on the number of faults remaining in the program during the interval. (K/L; L/V; J/M; …)

- **Failure Count Models**

  Number of faults detected in a given testing interval follow a Poisson Distribution (M/O; Shooman; G/O; …)

Nonhomogeneous Poisson Process

A class of models of the form:

$$\Pr\{N(t) = n \mid M(t)\} = \frac{M(t)^n}{n!} e^{-M(t)}$$

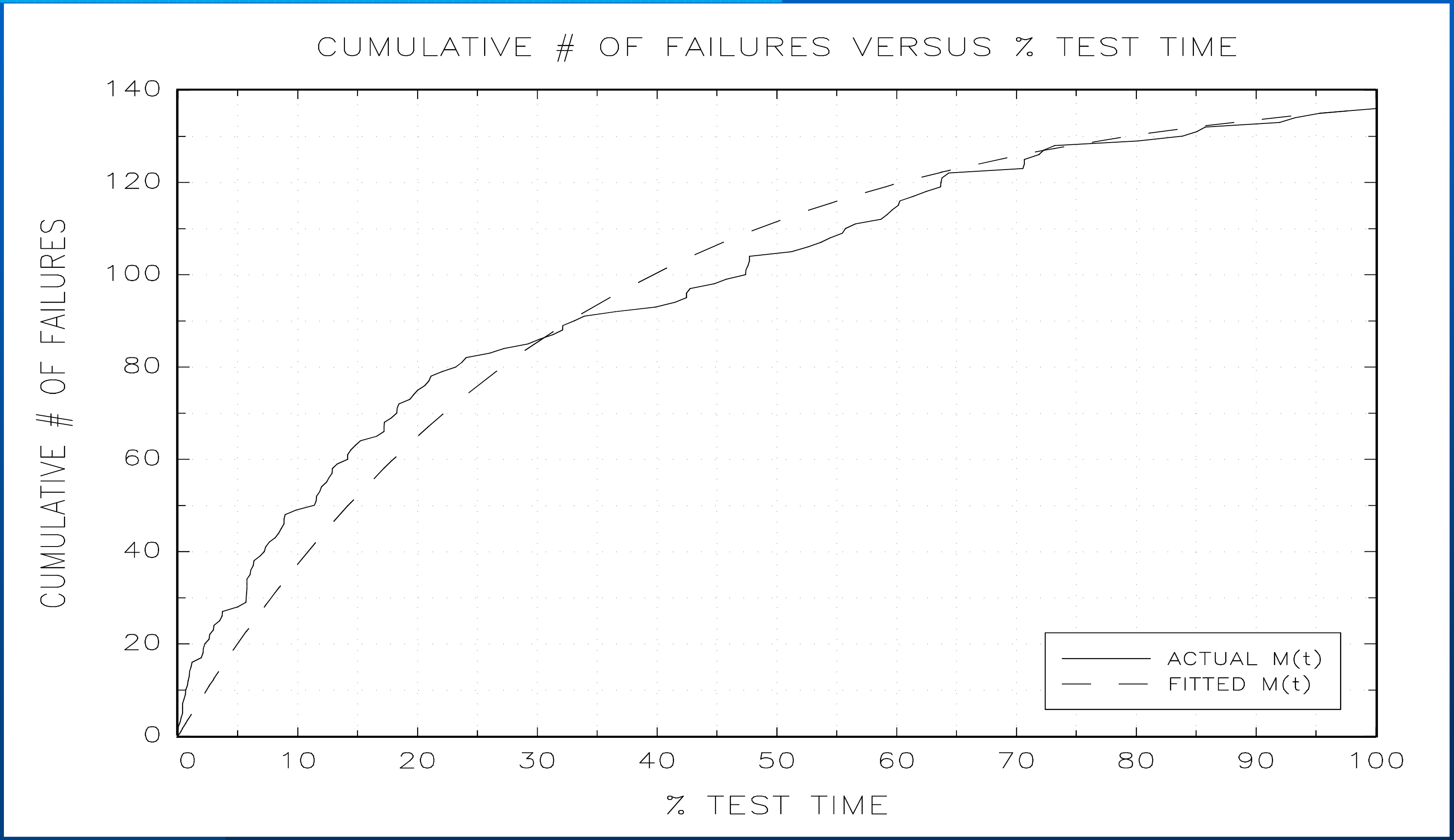
where N(t) is the # of software failures in [0,t]

$$M(t) = \int_0^t \lambda(u)\, du = E[N(t)] \text{ or the Mean function}$$

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P\{N(t+\Delta t) - N(t) > 0\}}{\Delta t} = \frac{dM(t)}{dt}$$

is the failure intensity function

Cumulative # Of Failures Versus % Test Time



Assessing Software Quality With The NNHP

Given testing up to time t with h failures:

1. The probability of k additional failures in [t, t+s] is

$$\Pr\{N(t+s) = h+k \mid N(t) = h\} = \frac{[M(t+s) - h]^k}{k!} e^{-[M(t+s) - h]}$$

2. Probability distribution of time until the next failure, T_{h+1} is

$$\Pr\{T_{h+1} > s\} = \Pr\{N(t+s) = h \mid N(t) = h\}$$

Selected Models (Nonhomogeneous Poisson Process)

M1 Logarithmic:	$M_1(t) = \gamma \log(1 + \beta t)$	$, 0 < \beta$
M2 Pareto:	$M_2(t) = \gamma (1 - (1 + \beta t)^{-\alpha})$	$, 0 < \alpha, 0 < \beta$
M3 Exponential:	$\mathbf{M_3(t) = \gamma (1 - e^{\eta t})}$	$\mathbf{, 0 < \eta}$
M4 Weibull:	$M_4(t) = \gamma (1 - \exp(-\eta t^{\alpha}))$	$, 0 < \alpha, 0 < \eta$
M5 Gen. Power:	$M_5(t) = \gamma ((1 + \beta t)^{-\alpha} - 1)$	$, -1 < \alpha < 0, 0 < \beta$
M6 Power:	$M_6(t) = \gamma t^{\alpha}$	$, -1 < \alpha \leq 0$
		$, 0 < \gamma$

Estimating Parameters

- To completely define the appropriate NHHP model for a particular data set, the analyst must:
  - Select a Mean function, M(t)
  - Use a statistical estimation procedure to evaluate the unknown parameters.
- Estimation Procedure used in the research
  - Maximum Likelihood Estimation
- Why ?
  - No substantial differences were detected when using other classical procedures (eg. Least Squares)
  - Recommended by experts in the field:Littlewood, Musa, Jelinski, Moranda, Miller, etc.

The Research Question

Can the estimation procedure of the Goel-Okumoto software reliability model be modified to improve its predictive performance?

Research Stage 2

PHASE 1: Analysis of the failure data and the development of trend procedures.

PHASE 2: Selection of performance measures for the evaluation process in the research.

PHASE 3: Comparison and ranking of the procedures.

PHASE 4: Use of statistical analysis methods in the overall selection of procedures

Failure Data Analysis

- **Exploratory Data Analysis**

For each data set the following analyses were conducted:

Basic plots

Trend Tests

Trend Test
Arithmetic

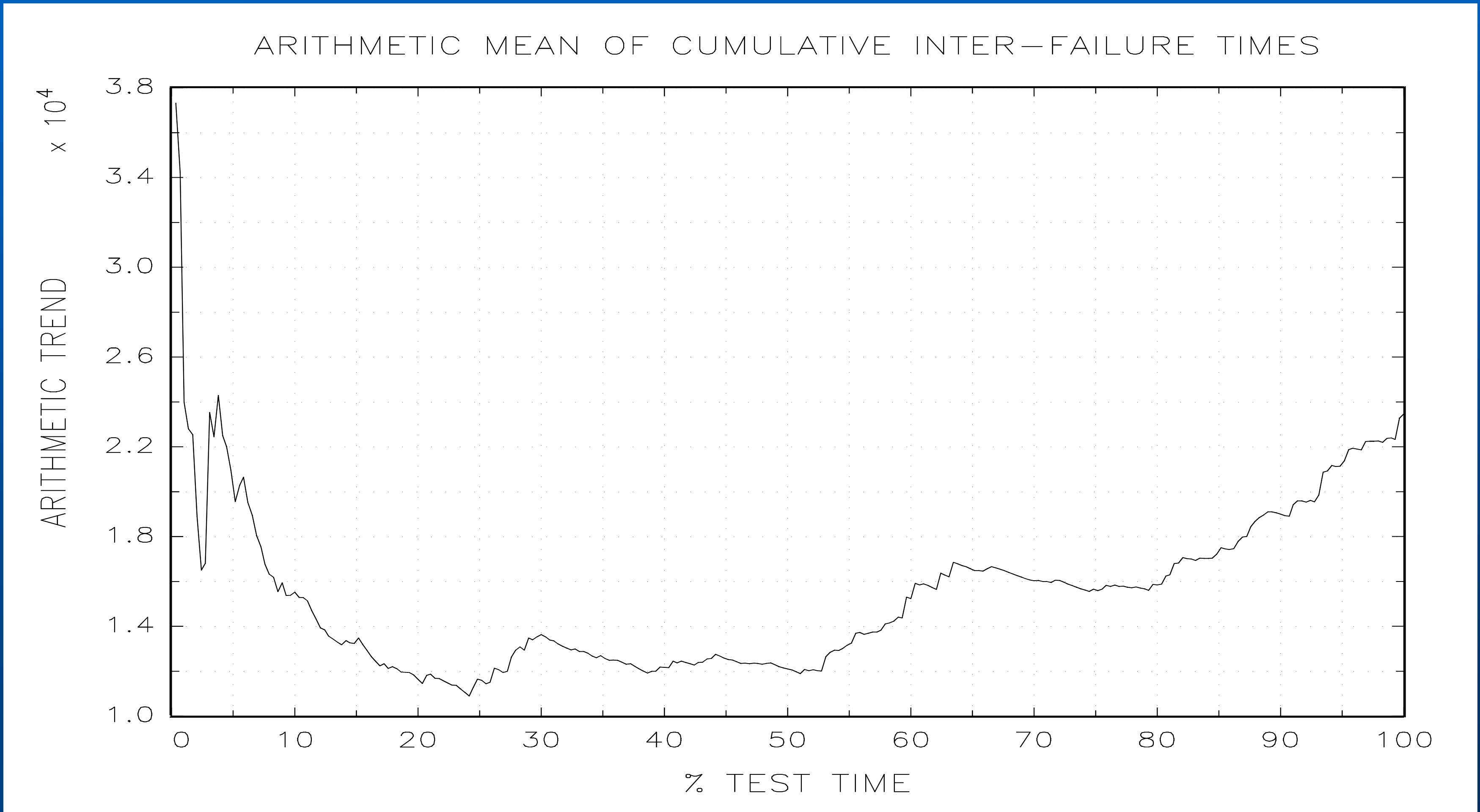
$$\tau_{j} = \frac{1}{n} \sum_{i=1}^{j} x(i), \; j = 1, 2, 3, \ldots, n.$$

where j = # of inter failure times;

x(i) are the inter-failure times

=> (Reliability growth is presumed if τ_j form an increasing series)

Arithmetic Mean Of Cumulative Inter-Failure Times SYS5



Trend Test
Laplace

$$u(t_0) = \frac{\frac{1}{n}\sum_{i=1}^{n} s_i - \frac{t_0}{2}}{t_0\sqrt{\frac{1}{12n}}}$$

where n = # of failures in $(0, t_0)$;

s_i is the time of occurrence of failure i

=>(Reliability growth is presumed if $u(t_0)$ is negative)

Trend Procedures

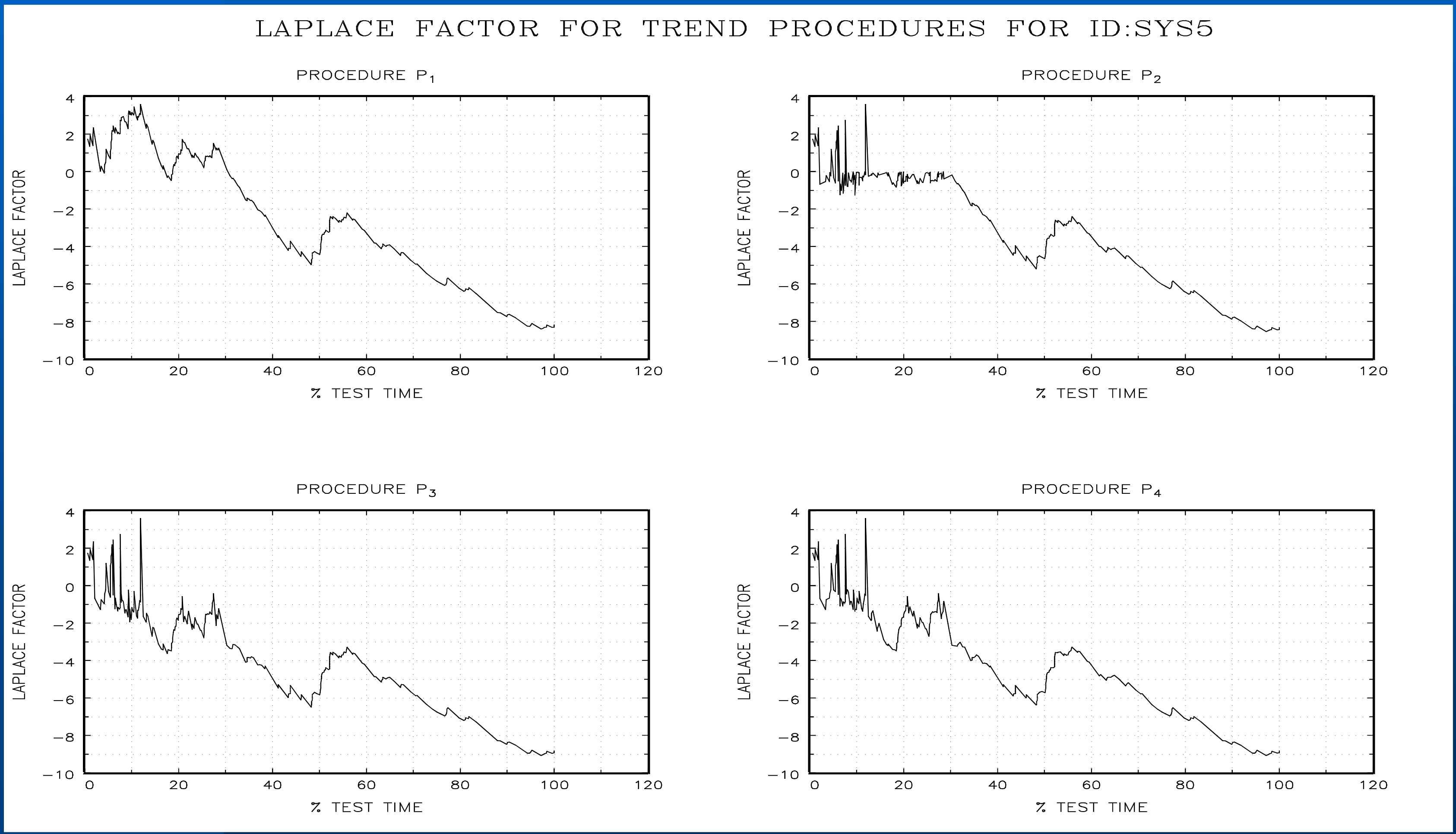
Procedure P_1: Using all the data within the current interval $(0,s_k)$, fit the model

Procedure P_2: Using the Laplace Test statistic, construct the interval that has the maximum size window with reliability growth within the current interval $(0,s_k)$.

Procedure P_3: Using the Laplace Test statistic, construct the interval that has the maximum reliability growth window within the current interval $(0,s_k)$.

Procedure P_4: Using the results from Procedure P_1 and Procedure P_3 as dynamic constraints, construct a window within the current interval $(0,s_k)$ that has improve reliability growth over the previous interval.

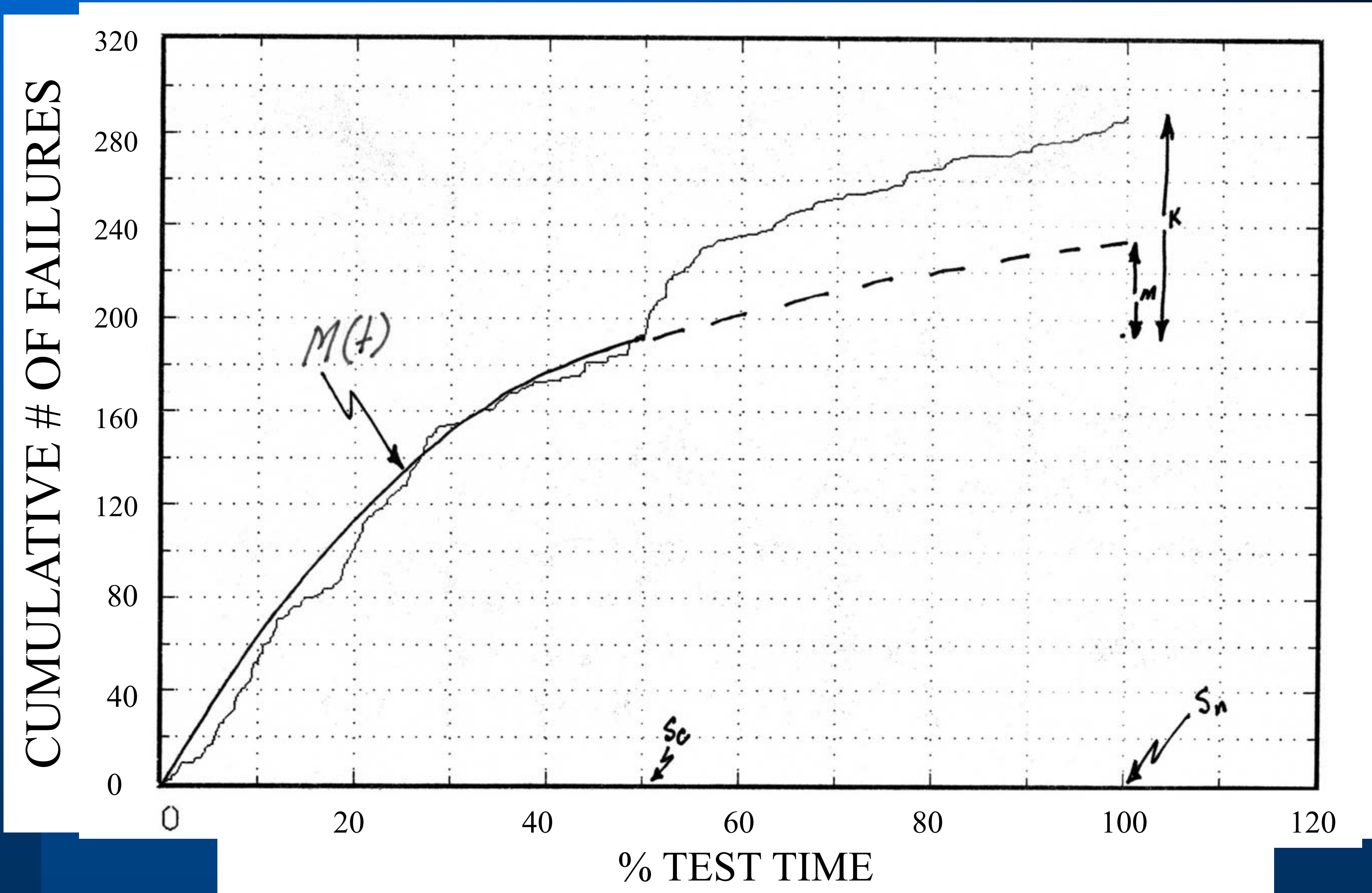
Laplace Trend Procedures



Performance Methodology
Estimation for Model-Procedure Pair

- For a given data set, failures are observed until some time s_n
- The failure data up to time s_c (< s_n) is used with the procedure to make maximum likelihood estimates of the model parameters
- The number of failures remaining, m, between (s_c , s_n) is estimated using $\hat{M}(s_n) - \hat{M}(s_c)$ and compared against the actual remainder, k

Cumulative # Of Failures Versus % Test Time

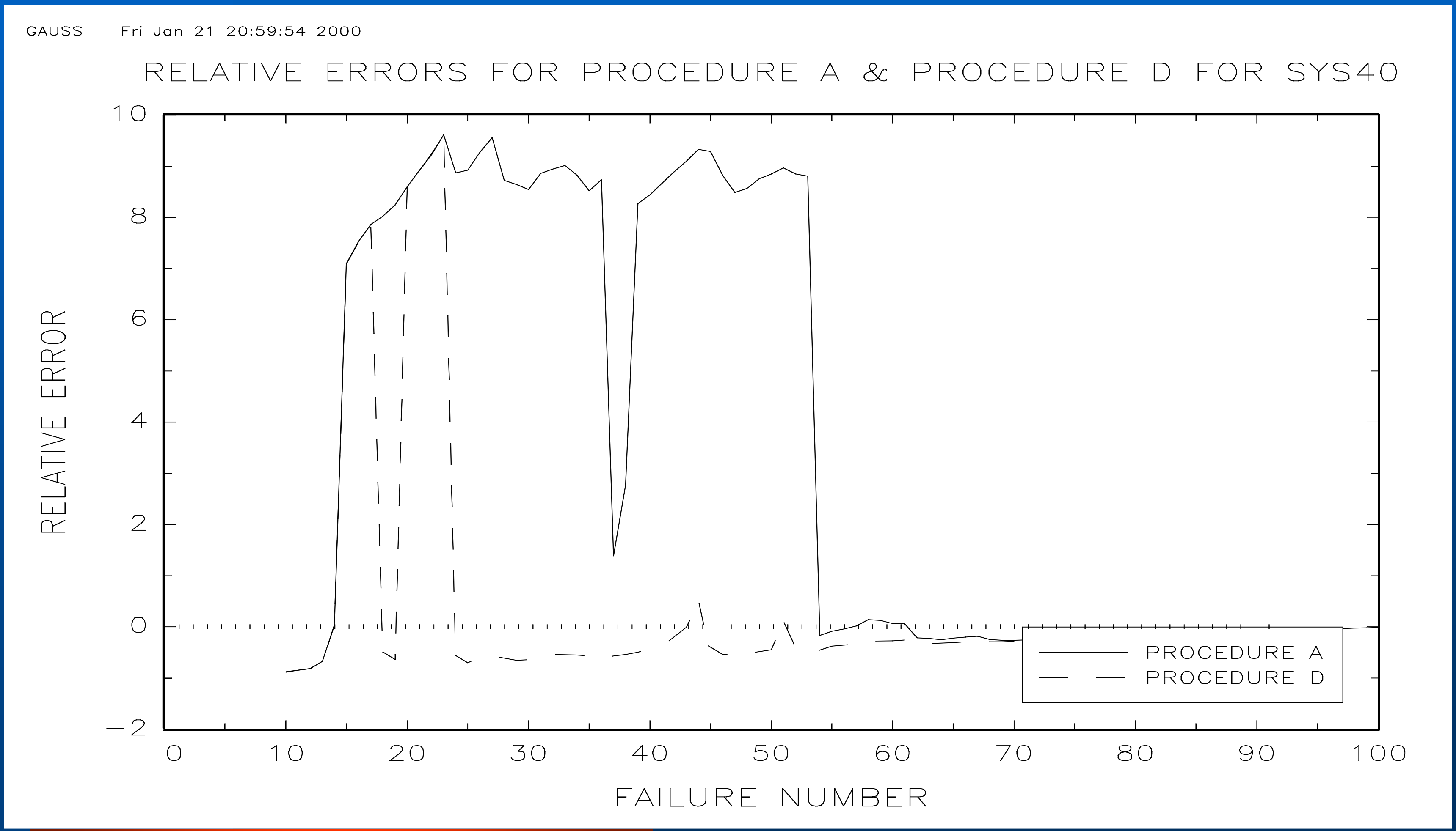


Performance Methodology (continued...)

- The relative predicted error, RE = (m - k)/k, is computed
- This is repeated for different values to s_n and summed (BIAS)
- The absolute value of RE at each value to s_n is summed (ARE)
- The value of RE is squared at each value to s_n and summed (RESQ)
- For each data set the performance values (ARE, BIAS, and RESQ) are computed, normalized and averaged

Comparison of Relative Errors

Procedures A and D



Comparison Of Procedures

- For each data set and for a specific performance measure the procedures are compared and ranked. (weights of 1-4 are awarded - smallest performance error to the largest performance error)
- Hypothesis testing to determine the bias of the procedure is conducted
- For all unbiased procedures, a statistical Sign Test utilizing the average absolute relative errors (ARE) is conducted
- To account for both bias and variability, the Sign Test utilizing the average relative errors squared (RESQ) is also conducted

Findings

Tests	Procedures			
	P1	P2	P3	P4
<u>Hypothesis Test (BIAS)</u>				
H_0 : Average relative errors = 0				
H_α : Average relative errors $\neq$ 0	----	99%	----	99%
<u>Sign Test (ARE)</u>				
{Compared to Procedure P_1}	ND	99%	ND	95%
<u>Sign Test (RESQ)</u>				
{Compared to Procedure P_1}	ND	99%	99%	99%

---- : Bias

ND : No Difference

Procedure Performance Ranking

Model	Procedures			
	P1	P2	P3	P4
Goel-Okumoto	4	1	3	2

Average Percentage Improvement Over Procedure P_1

Model	Performance Measures		
	BIAS	ARE	RESQ
Goel-Okumoto	81%	46%	57%

Conclusions

- Overall the research was successful in showing that we can improve the predictive performance of the Goel-Okumoto model
- Procedures P_2 and P_4 performed better than procedure P_1 in both the Hypothesis Test (BIAS) and the Sign Test (ARE)
- Procedures P_2, P_3, and P_4 performed better than procedure P_1 in the Sign Test (RESQ)

Recommendations

- Additional procedures should be developed using the Laplace Trend statistic technique
- Additional analysis is needed during the early stages of the failure data sets
- Better optimization techniques are needed for the parameter estimates of the likelihood function
- Additional classes of software reliability models should be tested using the research procedures
- Additional performance measures must be introduced in the comparison process
- A more systematic starting failure number approach should be developed