ソフトウェア信頼度成長曲線に関する統合モデルのＺグラフとその応用

古　山　恒　夫†

ソフトウェア信頼度成長曲線に関する統合モデルは，これまで提案された多くのモデルをカバーできるだけでなく，これまでモデル化されていなかった領域もカバーする．そのため，このモデルを用いれば既存の SRGM より高い精度で残存フォールト数を推定することができる．統合モデルのパラメータ推定法としては，確率過程に基づく最尤推定法と解析的に近似解を求めるＹ方程式法がすでに提案されている．これらの推定法は，一般に時刻 $t = 0$ から最新時刻までの全データを使ってパラメータを推定することが主たる目的であり，データの断片から各時点における局所的なパラメータを求めてその結果を表示することに対する配慮はなされていなかった．本論文では，局所的なパラメータの変動を２次元グラフ上で視覚的にとらえることのできるＺグラフを提案する．Ｚグラフは実データの断片からでも簡単な計算で得ることができるところに特徴がある．これを用いることによりデバッグ状況の時々刻々の変化を容易に把握することができる．実データに対してＺグラフを描いたところ，各時点でどの従来モデルに最も近いかが容易に把握できた．また，得られたパラメータを用いて残存バグを推定した結果，従来の２つの推定方法による推定結果と推定精度に差がないことが分かった．

Z-graph of the Manifold Growth Model and Its Application

Tsuneo Furuyama†

The manifold growth model that unifies existing software reliability growth models can cover a wide range of accumulated fault data including various types of data which are difficult to treat with existing models. Therefore, the remaining faults can be more accurately estimated by using this model than by using the existing models. Parameters of the manifold model are estimated by using the maximum likelihood method based on the stochastic process, which needs to solve transcendental equations, or analytically by using the Y-equation method. These methods generally use all data ranging from $t = 0$ until recent time and estimate the parameters of the whole range. Therefore, it is difficult to calculate and display parameter values at every instantaneous time. This paper proposes Z-graph, which can show directly and visibly the parameter values at every instantaneous time. Z-graph can be depicted easily if there is a fragment of actual data. The dynamical changes of the instantaneous parameter values of the manifold growth model, which are considered to be the changes of debugging conditions, are easily observed with Z-graph gotten from actual data. The ability of estimating the remaining faults of Z-graph is equal to those of the two existing estimation methods.

1. はじめに

信頼性の高いソフトウェアを予定期間内に開発するためには，残存フォールト数を高い精度で推定することが不可欠である．この課題を解決するために，これまで多くの方法が提案されてきたが，最も広く利用されているのは，試験工程で検出されるフォールトの累積値の傾向から残存フォールト数を推定する方法である．すなわち，時間対検出フォールト累積数の特性をもとにしたソフトウェア信頼度成長モデル（SRGM）を利用する方法である．

SRGM はすでに Jelinski ら[1)]，Musa[2)]，Littlewood[3)]，Goel ら[4)]，Yamada ら[5)]，Ohba[6)]，Tohma ら[7)]，Kanoun ら[8)]，Karunanithi ら[9)]など多くの研究者によって論じられている．

これらは，指数形や S 字形などそれぞれある特定の累積バグデータ群に対しては適切な総バグ数や残存バグ数の推定を行うことができる．しかし，どのモデルがどの累積バグデータに対して最もよく適合するかは，それぞれのモデルごとにパラメータの推定を行った後に比較する必要があった．

† 東海大学開発工学部
School of High-Technology for Human Welfare, Tokai University

この問題を解決する手段の 1 つとして上記のような代表的なモデルを包含する統合モデルが提案された[10),11)]．このモデルの中には，これまで提案された代表的なモデルのどれに合致するかを示すパラメータ γ が含まれており，γ を推定することにより，最も適合する従来モデルを容易に選択することができる．また，γ の値によって従来モデルの中間的なモデルも表すことができるため，累積バグデータによっては従来モデルよりも高い精度で残存バグを推定することができる．このことから，統合モデルは実用的には次のような場合に有効であると考えられる．

(1) テスト環境が理想に近づき，完全な確率過程で記述できるような世界が来たときに，より高精度で総バグおよび残存バグを推定する．

(2) 従来モデルのちょうど中間的な傾向を持つ累積バグデータが与えられたときに，従来モデルより高い精度で総バグおよび残存バグを推定する．

(3) 従来モデルのどのモデルが与えられた累積バグデータに最も適合するかを一度で判定する．

統合モデルのパラメータを求める方法として，フォールト発生を確率過程ととらえて確率的な尤度関数を最小とするようなパラメータを求める最尤推定法がある．しかし，この方法は超越方程式を解く必要があり[11)]，解の収束に時間がかかったり，解そのものが求まらなかったりする場合がある．それを解決する 1 つの方法として，累積フォールト数およびその微分値それぞれの対数をとることにより，パラメータを解析的に求める Y 方程式法が提案された[12)]．しかしながらこれらの 2 つの方法は，いずれも時刻 $t=0$ から最新時刻までの全データを使っていわゆる全所的なパラメータを推定することが主たる目的であった．

全所的なパラメータに対して，各時刻ごとの瞬間的なあるいは局所的なパラメータを考えることができる．局所的なパラメータはそれらを平均化した全所的なパラメータより各時刻でのデバッグ状況をより正確に反映していると考えられるので，これを求めることにより対象プロジェクトのデバッグ状況の変化をより正確に把握できる可能性が高くなる．しかしながら，すでに提案された上記 2 つのパラメータ推定法は，各時点における局所的なパラメータを求めてその結果を表示することには必ずしも適してはいない．すなわち，最尤推定法では最低 4 点のデータがあれば統合モデルに必要な 4 つのパラメータを求めることができるが，各時点の局所的パラメータを求めるのに毎回超越方程式を解くことになり効率的とはいえない．一方，3 次元空間内での回帰平面からパラメータを求める Y 方程式法では，各時点での局所的パラメータを計算することは容易であるが，それを視覚的，直感的にとらえることは 3 次元空間内での回帰平面に頼っているため一般に難しい．したがって，計算で求めたパラメータを改めて別のグラフで表示する必要がある．

本論文では，これらの問題を解決するために新たに Z 方程式およびそのグラフ表示である Z グラフを提案する．Z グラフは実データの断片からでも簡単な計算で得ることができるところに特徴がある．これを用いることによりデバッグ状況の時々刻々の変化を視覚的，直感的に容易に把握することができる．

2 章で統合モデルの概要を紹介し，3 章で新しいパラメータ推定法——Z 方程式と Z グラフを提案する．4 章で実データを用いて Z グラフの有効性を示す．

2. 統合モデルの概要[10),11)]

これまでに提案されているあるいは利用されている代表的な SRGM としては，指数形モデル，超指数形モデル，遅延 S 字形モデル，習熟 S 字形モデル，ゴンペルツ曲線，ロジスティック曲線などがある（**図 1**）．統合モデルはそれらのモデルあるいは曲線をカバーできるモデルである．

2.1 微分方程式

時刻 t における累積バグ数を y とする．統合モデルは次の微分方程式として定義される：

$$\frac{d(y+\delta)}{dt}\cdot(y+\delta)^{(\gamma-1)}=\alpha e^{-\beta t} \qquad (1)$$

ここで，$\alpha>0$，$\beta \geqq 0$，$\delta \geqq 0$ である．α は y 軸に関するスケールファクタ，β は t 軸に関するスケールファクタである．δ はこの微分方程式の解となる関数の y 軸方向の平行移動量を示すパラメータである．

2.2 一 般 解

式 (1) の一般解は次のようになる．

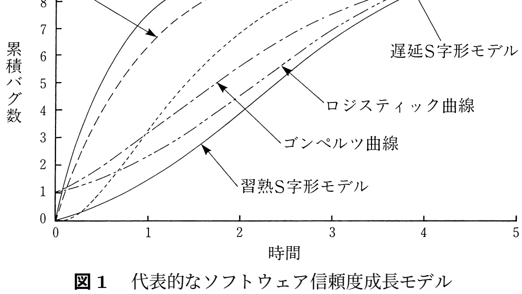


図 1 代表的なソフトウェア信頼度成長モデル
Fig. 1 Examples of representative software reliability growth models.

(A) $\beta > 0$, $\gamma \neq 0$ の場合

$$\begin{aligned} y &= N(1 - a \cdot e^{-bt})^c - \delta \\ &= N\{(1 - a \cdot e^{-bt})^c - (1-a)^c\} + y_0 \end{aligned} \tag{2}$$

ただし，y_0 は $t=0$ のときの y の値であり，

$$N = \left\{ \frac{\alpha \cdot \gamma + \beta(y_0+\delta)^\gamma}{\beta} \right\}^{\frac{1}{\gamma}} \tag{3}$$

$$a = \frac{\alpha \cdot \gamma}{\alpha \cdot \gamma + \beta(y_0+\delta)^\gamma} \tag{4}$$

$$b = \beta \tag{5}$$

$$c = \frac{1}{\gamma} \tag{6}$$

である．ここで，N は〔定数分（$a=1$ の場合は 0）を差し引いて〕試験開始時点に含まれるバグ総数，b はフォールトの検出速度を規定するパラメータ，a と c は成長曲線の形状（たわみ）を決定するプロジェクトごとのフォールト検出特性と解釈できる．

(B) $\beta > 0$, $\gamma = 0$ の場合

$$y = N \exp(-ke^{bt}) - \delta \tag{7}$$

ただし，

$$N = (y_0 + \delta)\ \exp\left(\frac{\alpha}{\beta}\right) \tag{8}$$

$$k = \frac{\alpha}{\beta} \tag{9}$$

$$b = \beta \tag{10}$$

式 (8) の N は式 (3) で $\gamma \to 0$ とした場合であり，k は式 (4) の a と式 (6) の c の積をとり，$\gamma \to 0$ とした値である．

3. Z 方程式と Z グラフ

3.1 定　　義

以下では，式を見やすくするために，y の t による一次微分を y'，二次微分を y'' で表す．一般に t を変数とするある関数 $y = f(t)$ に対して，z_1 と z_2 を

$$z_1 = z_1(t) = \frac{y'}{y} = \frac{f'(t)}{f(t)} \tag{11}$$

$$z_2 = z_2(t) = \frac{y''}{y'} = \frac{f''(t)}{f'(t)} \tag{12}$$

と定義し，さらに z_1 と z_2 の関係を表す式

$$F(z_1(t), z_2(t); t) = 0 \tag{13}$$

を Z 方程式と定義する．ここで t は媒介変数である．また，z_1 を横軸に z_2 を縦軸にとることにより Z 方程式の解を表示したグラフを Z グラフと呼ぶことにする．

3.2 統合モデルの Z 方程式

図 1 に示した従来の代表的な SRGM は，習熟 S 字形モデルを除いて $\delta = 0$ の場合の統合モデルの解，あるいはその近似で表されることが知られている[10),11)]．以下では，$\delta = 0$ の場合の統合モデルの解の Z 方程式を求める．式 (1) は次のように表される．

$$y' \cdot y^{\gamma - 1} = \alpha e^{-\beta t} \tag{1$'$}$$

式 (1)$'$の両辺の対数をとり式を整理すると，次の式が得られる．

$$ln(y') + (\gamma - 1)ln(y) + \beta t - ln(\alpha) = 0^{☆} \tag{14}$$

式 (14) を t で微分すると，

$$\frac{y''}{y'} + (\gamma - 1)\frac{y'}{y} + \beta = 0 \tag{15}$$

が得られる．式 (15) は式 (11) および式 (12) より次のように書き換えることができる．

$$z_2 + (\gamma - 1)z_1 + \beta = 0 \tag{16}$$

あるいは，式 (5) および式 (6) より次の式が得られる．

$$z_2 + \left(\frac{1}{c} - 1\right)\ z_1 + b = 0 \tag{17}$$

式 (17) は統合モデルの解の Z 方程式が時刻 t とは無関係に線形となることを示している．また，逆に Z 方程式が線形となる方程式は統合モデルの解となることは式 (17) の導出過程から明らかである．Z グラフの傾き dz_2/dz_1 は，式 (17) から

$$\frac{dz_2}{dz_1} = 1 - \gamma = 1 - \frac{1}{c} \tag{18}$$

となる．式 (18) から傾きは時間軸のスケールファクタ $b (= \beta)$ と独立であることが分かる．式 (17) と式 (18) は次のことを意味する．

『統合モデルを表す方程式 (1) のパラメータ β とγ（言い換えると統合モデルの一般解である式 (2) のパラメータ b と c，あるいは式 (7) のパラメータ b）は，$\delta = 0$ と仮定すれば，「ある異なる 2 つの時刻での y，y'，y'' の値から求めることができる」ことを示している．これは，推定すべきの曲線の「断片」さえあれば時間軸の平行移動に無関係にパラメータ β と γ を推定できることを意味している．特にパラメータ γ は，時間軸のスケール変換（拡大/縮小）にも無関係に推定できる．』

3.3 統合モデルの Z グラフ

Z グラフ上からは，パラメータ $\beta\ (= b)$ は z_2 軸の切片にマイナスを付けたものとして，パラメータ

☆ ここで，$y_1 = t$，$y_2 = l_n(y)$，$y_3 = l_n(y')$ と変数変換したものが Y 方程式である[12)]．

表 1 従来の代表的な SRGM とその Z 方程式
Table 1 Existing SRGM and their Z-equation.

モデル(曲線)	一般式	統合モデルの微分方程式のパラメータ α	β	γ	δ	$z_1=\frac{y'}{y}$	$z_2=\frac{y''}{y'}$	Z 方程式
指数形	$y=N(1-e^{-bt})$	Nb	b	1	0	$\frac{be^{-bt}}{(1-e^{-bt})}$	−b	$z_2+b=0$
ゴンペルツ	$y=N\exp(-ke^{-bt})$ (注 1)	kb	b	0	0	$k\ b\ e^{-bt}$	$b(ke^{-bt}-1)$	$z_2-z_1+b=0$
ﾛｼﾞｽﾃｨｯｸ	$y=\frac{N}{1+\varphi e^{-bt}}$	$\frac{b\varphi}{N}$	b	−1	0	$\frac{b\varphi e^{-bt}}{1+\varphi e^{-bt}}$	$\frac{b(\varphi e^{-bt}-1)}{1+\varphi e^{-bt}}$	$z_2-2z_1+b=0$
習熟 S 字形	$y=N\frac{1-e^{-bt}}{1+\varphi e^{-bt}}$ (注 2)	$\frac{b\varphi^2}{N(\varphi+1)}$	b	−1	$\frac{N}{\varphi}$	$\frac{b(\varphi+1)e^{-bt}}{(1+\varphi e^{-bt})(1-e^{-bt})}$	$\frac{b(\varphi e^{-bt}-1)}{1+\varphi e^{-bt}}$	$z_2-\frac{b(\varphi-1)}{\varphi+1}=0$ $(t\to 0)$ $z_2+(\frac{1}{\varphi+1}-1)z_1+b=0$ $(t\to\infty)$
遅延 S 字形	$y=N\{1-(1+bt)e^{-bt}\}$	$\approx 2.2N^{0.46}b$	$\approx 0.78b$	≈ 0.46	0	$\frac{b^2te^{-bt}}{1-(1+bt)e^{-bt}}$	$\frac{1-bt}{t}$	$z_2-\frac{1}{2}z_1=0$ $(t\to 0)$ $z_2-0.54z_1+0.78b=0$
超指数形	$y=\sum_{i=1}^{n}N_i\{1-\exp(-b_it)\}$	$\approx\frac{N^{\gamma}\beta}{\gamma}$	——	>1	0	$\frac{\sum_{i=1}^{n}N_ib_i\exp(-b_it)}{\sum_{i=1}^{n}N_i\{1-\exp(-b_it)\}}$	$-\frac{\sum_{i=1}^{n}N_ib_i^2\exp(-b_it)}{\sum_{i=1}^{n}N_ib_i\exp(-b_it)}$	$z_2+\frac{\sum_{i=1}^{n}N_ib_i^2}{\sum_{i=1}^{n}N_ib_i}=0$ $(t\to 0)$
ﾜｲﾌﾞﾙ過程	$y=\lambda t^m$	$m\lambda^{1/m}$	0	$\frac{1}{m}$	0	$\frac{m}{t}$	$\frac{m-1}{t}$	$z_2+\frac{1-m}{m}z_1=0$

(注 1) 普通ｺﾞﾝﾍﾟﾙﾂ曲線は $y=N\cdot p^{q^t}$ で表わされるが，exp を使った式に変形した方が扱い易いので，この式にように変形する．ここで k=−ln(p), b=−ln(q)である．

(注 2) $y=N\frac{\varphi+1}{\varphi}\left(\frac{1}{1+\varphi\exp(-bt)}-\frac{1}{\varphi+1}\right)$ と変形することにより，ﾛｼﾞｽﾃｨｯｸ曲線を y 軸方向に移動したものと等価であることが分かる．

$\gamma\ (=1/c)$ は $(1-dz_2/dz_1)$ として求められる．なお，以下では β と b，γ と c は文脈によりそれぞれ適宜使い分けることとする．

もし，異なる 3 つ以上の時刻での y，y'，y'' の値が得られるなら，計測誤差の影響を最も小さくするパラメータを統計的手法を用いて求めることができる．式 (17) が一次式であるので，最小自乗法を用いることができる．

3.4 従来の SRGM の Z 方程式と Z グラフ

これまでに提案された代表的な SRGM の Z 方程式は，次のいずれかの方法によって求めることができる．

(a) Y 方程式を微分する

(b) 統合モデルの微分方程式のパラメータを式 (16) に代入する．

これまでに提案された主な SRGM に対する Z 方程式を**表 1** に示す．指数形モデル，ゴンペルツ曲線，ロジスティック曲線，ワイブル過程モデルの場合は統合モデルのパラメータ δ が 0 であることから，式 (16) より線形の Z 方程式が得られる．しかし，習熟 S 字形モデルでは δ が 0 でないため，線形の Z 方程式を得ることはできない．遅延 S 字形モデルと超指数形モデルはいずれも統合モデルの解ではないが，これらのモデルは $\delta=0$ の場合の統合モデルの一般解で近似できるため[10)]近似的な線形の Z 方程式を得ることができる．

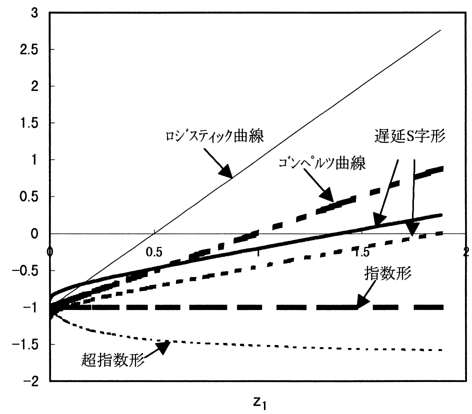


図 2 既存モデルの Z グラフ（$b=1$ の場合）
Fig. 2 Relationships between the Z-graphs of the existing models ($b=1$).

これまでに提案された主な SRGM に対する Z グラフの例を**図 2** に示す．指数形モデル，ゴンペルツ曲線，ロジスティック曲線に対する Z グラフは直線となり，遅延 S 字形モデルと超指数形モデル対する Z グラフは近似的に直線となる．

4. Z グラフの応用

Z グラフの応用として，(a) デバッグ状況の時々刻々の変化の把握と (b) 残存バグ推定が考えられる．前者は，Z グラフが SRGM の断片から統合モデルのパラメータを容易にかつ視覚的に推定することができることを利用したものであり，後者はグラフ全体から統合モデルのパラメータを推定できることを利用したものである．本論文では中川データ[13)]および三觜データ[14)]を用いてこれら 2 つの応用を示す．

Z グラフを描くためには，一次および二次の微分値が必要であるが，累積バグデータからは微分値が得られないので，近似的に差分値を用いる．以下では，各時刻が等間隔の場合を扱い，一次差分および二次差分として最も簡単な次の式を用いる．

$$y'(t_i) = \frac{1}{2\Delta}\{y(t_{i+1}) - y(t_{i-1})\} \quad (19)$$

$$y''(t_i) = \frac{1}{2\Delta}\{y'(t_{i+1}) - y'(t_{i-1})\} \quad (20)$$

ただし，Δ は単位時間間隔である．以下では $\Delta = 1$ とするが，Z グラフは時間軸の伸縮に依存しないことから一般性を失うことはない．

4.1 バグ検出状況の推移の把握

各デバッグ時点ごとに Z グラフをプロットすると，統合モデルのパラメータの時間的な変化，すなわちあるその時点で既存モデルのどのモデルと似た状態にあるかを視覚的，直感的にとらえることができる．

(1)　中川データによる検証

Z グラフの応用例として**図 3** に中川データを，**図 4** にその Z グラフを示す．図 4 の (a) では，$t = 2$ から 6 までは傾きが 1（ゴンペルツ曲線風）であるが，$t = 7$ から 10 までは傾きはマイナスに転じている（超指数形風）．次に図 4 の (b) で z_1 軸を伸長してみると，$t = 10$ から 15 までは傾きが 12 程度でパラメータ c の値はロジスティック曲線における値 (−1) をはるかに超えた (−11) 程度のグラフになる．その後，グラフの傾きはプラスとマイナスの間を大きく揺れ動いているが，グラフそのものはマクロに見ると，z_2 のマイナス領域で z_2 軸に近づいている．すなわち，t の値が大きなところ（たとえば $t = 16$ 以上）ではマクロには中川データは指数形モデルになっていることが分かる．

(2)　三觜データによる検証

ゴンペルツ曲線が適合する例として知られている三觜データを**図 5** に示す．$t = 0$ で $y = 248$ の場合（シフト前データ）と，$t = 0$ で $y = 0$ となるように y 軸方向に座標移動した場合（シフト後データ）のそれぞれの Z グラフを**図 6** に示す．分析対象データを y 軸方向に移動した場合，Z グラフでは z_1 の座標値が非線形だが単調に変化するだけで，全体のパターンは変わらない．

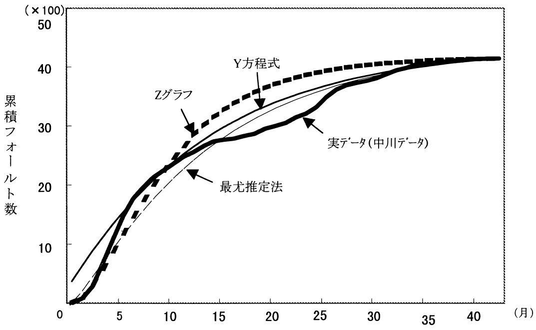
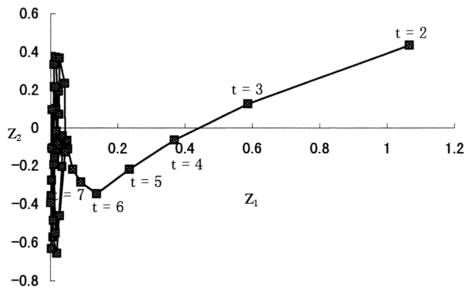
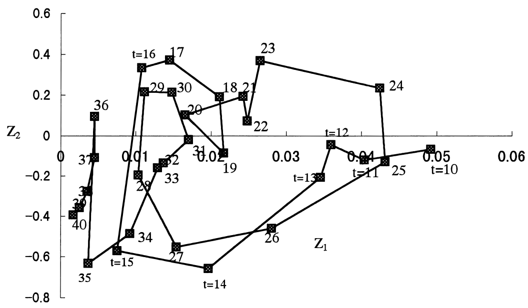


図 3　実データ（中川データ）と Endpoint 推定曲線
Fig. 3　Nakagawa data with estimated curves.



(a) 全体（$t = 2$ から 40）
(a) Overall graph for $t = 2$ to 40



(b) 一部分（$t = 10$ から 40）
(b) Detailed graph for $t = 10$ to 40

図 4　中川データにおける Z グラフ
Fig. 4　Z-graph for Nakagawa data.

(a) シフト前データの Z グラフ：$t = 5$ から 9 までの間は γ は −1.5 程度であり，ロジスティック曲線（$\gamma = -1$）に近い．しかし，$t = 11$ から急激に γ の値は 0 に近くなって，ゴンペルツ曲線に近づくことが分かる．確かに従来のモデルの中ではゴンペルツ曲線で推定するのが最も妥当であることが分かる．

(b) シフト後データの Z グラフ：$t = 2$ から 9 までは

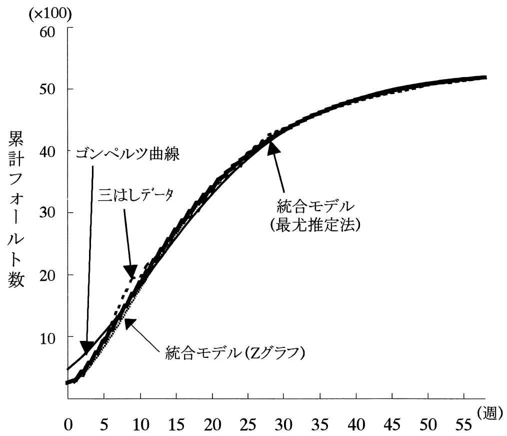
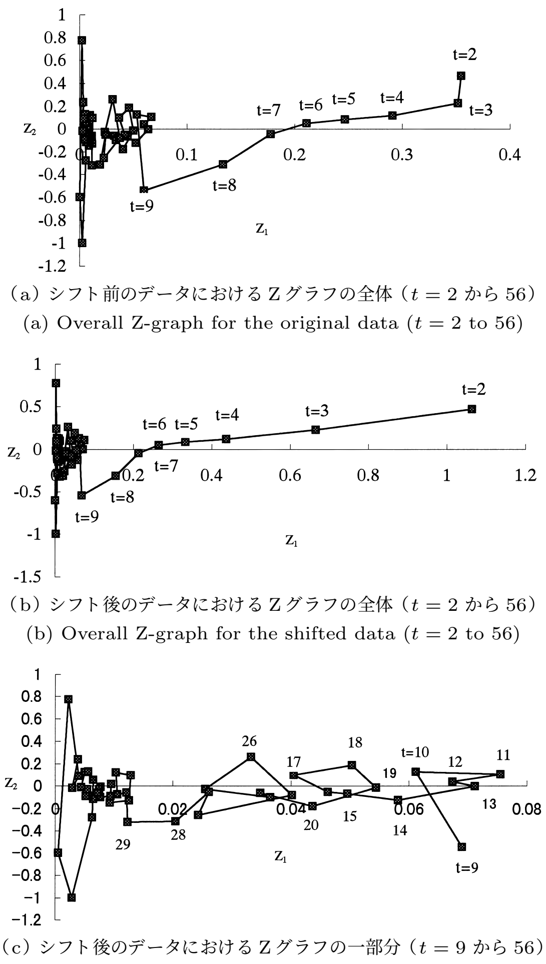


図 5 実データ（三觜データ）と Endpoint 推定曲線
Fig. 5 Mitsuhashi data with estimated curves.



（a）シフト前のデータにおける Z グラフの全体（$t = 2$ から 56）
(a) Overall Z-graph for the original data ($t = 2$ to 56)

（b）シフト後のデータにおける Z グラフの全体（$t = 2$ から 56）
(b) Overall Z-graph for the shifted data ($t = 2$ to 56)

（c）シフト後のデータにおける Z グラフの一部分（$t = 9$ から 56）
(c) Partial Z-graph for shifted data ($t = 9$ to 56)

図 6 三觜データにおける Z グラフ
Fig. 6 Z-graph for Mitsuhashi data.

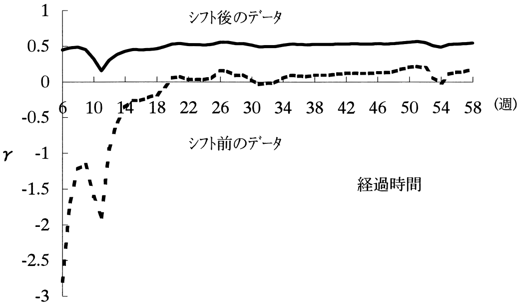


図 7 シフト前とシフト後の三觜データにおけるパラメータ γ の比較
Fig. 7 Comparison of variation of parameter γ between the original and shifted Mitsuhashi data.

z_1，z_2 ともに単調に減少している．特に $t = 2$ から 6 までは，傾き（$1 - \gamma$）がほぼ 0.5 であり，遅延 S 字形モデルとほぼ等しい．

シフト前とシフト後それぞれのデータに対するパラメータ γ の推定値の比較を**図 7** に示す．図 7 はシフト前のデータではゴンペルツ曲線が，シフト後のデータでは遅延 S 字形モデルが，いずれも従来型モデルの中で最も近いことを示している．

このように Z グラフでバグ検出状況の変化を分析してみると，デバッグ期間中 1 つの従来型モデルの仮定を満たし続けているわけではなく，さまざまな従来型モデルの間を推移していることが分かる．

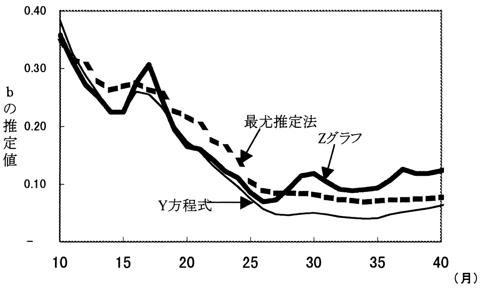
4.2 残存バグ推定への応用

統合モデルのパラメータ推定法としては，すでに確率過程として厳密な裏付けのある最尤推定法と解析的にパラメータを求めることのできる Y 方程式法が提案されている．この節では従来の推定方法を用いて統合モデルの全所的パラメータを推定した場合と，Z グラフにより推定した場合を，実データを用いて比較する．推定精度の評価尺度としては，サンプル時系列データのうちの Endpoint（最後）のデータを途中の時系列までのサンプルデータを用いて，どの程度の誤差内で推定できるか（これを Endpoint 推定と呼ぶことにする）を用いる．特にここでは最尤推定法による推定誤差からの差に着目して比較評価する．

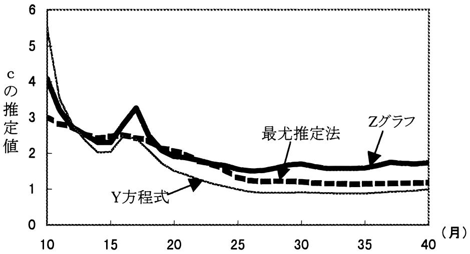
（1） パラメータの推定法

パラメータ b と c は式 (17) をもとに，Z グラフ上の回帰直線から求める．得られたパラメータの有意性は普通の回帰分析の検定方法を用いる．

パラメータ a と N は，最尤推定法でパラメータを求めるとデータ系列の両端で推定値と実績値が等しくなるという性質を利用して推定する．この方法を用いるとパラメータ a と N は次のようにして解析的に求め

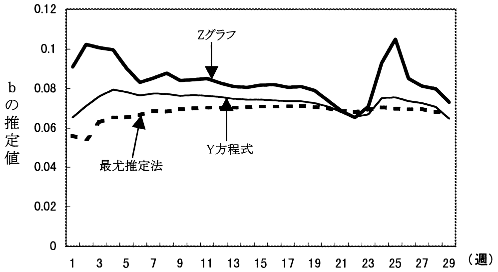


（a）パラメータ b の推定
(a) Estimation of parameter b

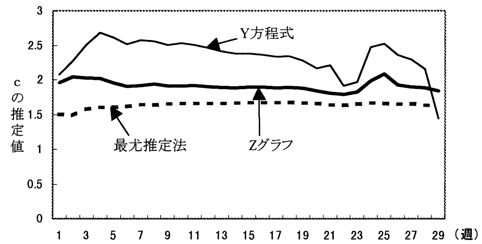


（b）パラメータ c の推定
(b) Estimation of parameter c

図 8　中川データに対するパラメータの推定値の変動
Fig. 8　Variation of estimated value of parameter with Nakagawa data.



（a）パラメータ b の推定
(a) Estimation of parameter b



（b）パラメータ c の推定
(b) Estimation of parameter c

図 9　シフト後の三觜データに対するパラメータの推定値の変動
Fig. 9　Variation of estimated value of parameters with the shifted Mitsuhashi data.

ることができる．

$$\begin{cases} y_0 = N(1-a)^c \\ y_n = N(1-ae^{-bt_n})^c \end{cases} \tag{21}$$

より

$$a = \frac{1-q}{1-qe^{-bt_n}} \tag{22}$$

$$N = y_n\left(\frac{1-qe^{-bt_n}}{1-e^{-bt_n}}\right)^c \tag{23}$$

となる．ただし $q = \left(\frac{y_0}{y_n}\right)^{\frac{1}{c}}$ である．特に $y_0 = 0$ の場合は

$$a = 1 \tag{24}$$

$$N = \frac{y_n}{(1-e^{-bt_n})^c} \tag{25}$$

となり，推定式は

$$y = y_n\left(\frac{1-e^{-bt}}{1-e^{-bt_n}}\right)^c \tag{26}$$

となる．

（2）　全所的パラメータの推定値の比較

中川データおよび三觜データに対して最尤推定法，Y 方程式，Z グラフのそれぞれの方法で統合モデルの全所的パラメータを推定し比較したものを**図 8** および**図 9** に示す．推定パラメータの変動は中川データで見た目にも急激な変化が見られる $t = 24$ 以降でパラメータ b と c の差がやや大きくなっている以外は全体としては，各推定法で同じような動きをしている．

表 2　パラメータ推定値の比較
Table 2　Comparison of estimated values of each parameter.

データ (推定時点)	推定方式	b	c (γ)
中川データ (t=42)	最尤推定法	0.079	1.19 (0.84)
	Y方程式	0.076	1.07 (0.93)
	Zグラフ	0.139	1.82 (0.55)
三觜データ (t=58)	最尤推定法	0.068	1.62 (0.61)
	Y方程式	0.065	1.44 (0.69)
	Zグラフ	0.073	1.83 (0.55)

次に両方のデータに対してそれぞれ最終時刻でのパラメータの推定値を比較したものを**表 2** に示す．実績データにこぶがあるなど変動の大きい中川データではパラメータ b，c とも推定方式で差がみられる．すなわち，パラメータ b と c の Z グラフでの推定値は，最尤推定法で得られた結果のそれぞれ 1.8 倍と 1.5 倍である．一方，変動の少ない三觜データでは，2 つの方式で得られた結果の比率はパラメータ b，c とも 1.1 倍であり，推定方式で差がないことを示している．

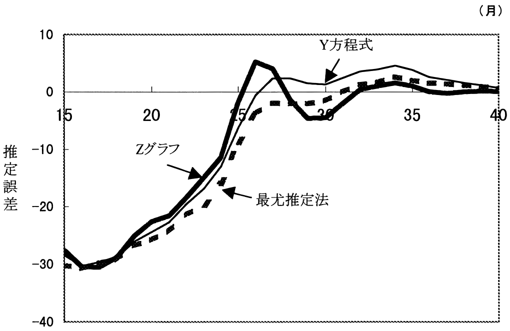


図 10　中川データに対する Endpoint 推定誤差の比較
Fig. 10　Comparison of endpoint estimation error over time for Nakagawa data.

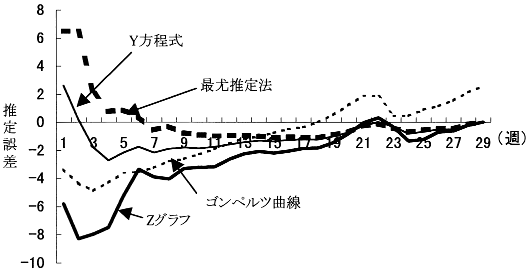


図 11　三菁データに対する Endpoint 推定誤差の比較
Fig. 11　Comparison of endpoint estimation errorover time for Mitsuhashi data.

(3)　Endpoint 推定誤差の比較

2つのデータに対して最尤推定法，Y 方程式，Z グラフのそれぞれの方法で求めた Endpoint 推定値の変動を**図 10** と**図 11** に示す．Endpoint 推定誤差の変動は，いずれのデータでも推定法によらず同じ傾向を示している．ただし，いずれのデータでも Y 方程式による推定誤差の方が Z グラフより最尤推定法の推定誤差の変動により近い動きをしている．特に三菁データにおける最尤推定法と各方式の差は Y 方程式で $t = 39$ 以降，Z グラフでも $t = 46$ 以降で 1%以内となり，統合モデルに関する推定方式による推定誤差との差は非常に小さいといえる．なお，参考のために示した三菁データにおけるゴンペルツ曲線での推定誤差が Endopoint に近づいても 0 にならないのは，データ全体に対して最小自乗法で曲線推定を行っているためである．

4.3　考　　察

(1)　局所的パラメータ推定法の比較と評価

統合モデルを用いてある短い時間間隔でのパラメータを推定する方法として次の 4 つの方法が考えられる(**表 3**)．

(a)　異なる 4 つの時刻での y から求める．

(b)　異なる 3 つの時刻での y と y' から Y 方程式を用いて求める．

(c)　異なる 2 つの時刻での y，y'，y'' から Z グラフを用いて求める．

(d)　ある 1 点での y および y の 1 次から 3 次までの微分係数を用いて求める．

以下に (c) Z グラフを用いた方法以外の方法の特徴を述べる．

(a)　異なる 4 つの時刻での y から求める方法

統合モデルの一般解のパラメータは 4 つあるので，理論的には 4 つの異なる時刻でのそれぞれの y の値があれば一般解が求まる．これをある短い時間間隔の区間内の 4 点ごとに順次適用していけば，時間軸上に連続的にパラメータを求めることができる．しかし，パラメータを求めるための 4 本の方程式は超越方程式となり，次々と解を求めていくのは簡単ではない．

(b)　異なる 3 つの時刻での y と y' から Y 方程式を用いて求める方法

3 つの異なる時点での y と y' の値が分かれば，Y 方程式を用いて解析的にパラメータを求めることができる．したがって，ある短い時間間隔の区間内の 3 点ごとに移動しながら Y 方程式を適用していけば，時間軸上に連続的にパラメータを求めることができる．また，その結果をグラフにプロットすれば，パラメータの推移を見ることはできる．しかし，この場合は ① 計算の基礎となる時刻が 3 つという時間的なひろがりを持つことと，② Z グラフのように直感的に理解できるグラフは描けないため，改めて計算結果をグラフ表示しないといけないという問題がある．

(d)　ある 1 点での y および y の 1 次から 3 次までの微分係数を用いて求める方法

理論的には，1 点での y と y の 3 次までの微分係数が知られていれば，その時点での統合モデルのパラメータを求めることができる．しかし，① 求まるパラメータは γ だけであること，② 実際に計算に必要となるデータ数が 1 時刻あたり 7 点必要となること，および ③ γ の値を求める計算式が y と y の 3 次までの微分係数の有理式でそれほど簡単ではないことから，Z グラフ法より優れているとは言い難い．具体的な γ の計算式は式 (15) の両辺を t で微分して γ について解くことにより得られる．

$$\gamma = \frac{\{y'''y' - (y'')^2\}y^2}{\{y''y - (y')^2\}(y')^2} + 1 \qquad (27)$$

以上のことから局所的パラメータを推定・表示するための方法としては，Z グラフ法が最も優れているといえる．

表 3 各パラメータの局所的な値の推定方法の比較
Table 3 Comparison of four methods for estimating parameter values at each instantaneous time.

推定方法	求まるﾊﾟﾗﾒｰﾀ	ﾊﾟﾗﾒｰﾀ計算に必要な最小限のﾃﾞｰﾀ	計算に必要な実際のデータの個数（注１）	解くべき方程式	ﾃﾞｰﾀ数が多い場合のパラメータ推定法	特徴
４点法	N, a, b, c	異なる４つの時刻の(t,y)の組	○○○○ t_1 t_2 t_3 t_4	超越方程式	最尤推定法	確率過程に基づいた厳密な裏付けがある．
Ｙ方程式	$\alpha, \beta(=b), \gamma(=\frac{1}{c})$	異なる３つの時刻の(t,y,y')の組	×○○○× t_{-1} t_1 t_2 t_3 t_{+1}	３ 元１次連立方程式	(重) 回帰分析	最小のﾃﾞｰﾀ個数で解析解を求めることができる．
Ｚグラフ	$\beta(=b), \gamma(=\frac{1}{c})$	異なる２つの時刻の(y,y',y'')の組	××○○×× t_{-2} t_{-1} t_1 t_2 t_{+1} t_{+2}	２元１次連立方程式		ﾊﾟﾗﾒﾀ β と γ の変動の様子を２次元ｸﾞﾗﾌ上で視覚的に捉えることができる．
３次微分利用	$\gamma(=\frac{1}{c})$	ある時刻の(y,y',y'',y''')	×××○××× t_{-3} t_{-2} t_{-1} t_1 t_{+1} t_{+2} t_{+3}	１次方程式		理論的にある１時刻での γ を求めることができる．

(注１) ○印：ﾊﾟﾗﾒｰﾀを求めるために必要な各時刻 $t_i = (i = 1, \ldots, n)$ の y の値（求めたﾊﾟﾗﾒｰﾀがｶﾊﾞｰする時間の範囲に相当）
×印：y の（高次）微分を求めるために必要となる各時刻の前後のﾃﾞｰﾀ（ここでは簡単のために各 t_i が等間隔で並んでいるものとする）

(２) 推定方式によるパラメータ推定値の違い

実際の推定対象データは理想的な曲線と異なる．パラメータを直接推定するためのデータは生のデータを変換したものであり，最尤推定法，Ｙ方程式法，Ｚグラフ法の３つの方式で異なる．Ｙ方程式では，時刻 t と累積バグ数 y の対数と y' の対数の組を３次元空間上にプロットした値から重回帰分析により最適平面を求めて（全所的）パラメータを推定する．一方，Ｚグラフでは２次元平面にプロットしたデータから回帰分析により最適直線を求めて（全所的）パラメータを推定する．すなわち，パラメータの推定は変形されたデータのある種の平均から求めることであり，最尤推定法，Ｙ方程式法，Ｚグラフ法に差があるのはその平均のとりかたの違いにあるといえる．そのため，特に変動の大きな実データでは全体パラメータの推定値が推定方式によってかなり異なるように見えることがある．しかし多くの場合，各パラメータどうしでそれらの差を相殺し合って最終的な目的である Endpoint 推定への影響は小さくなっていると考えられる．特に三菱データのように比較的滑らかに変動するデータの場合は，推定方式による差は小さい．このことは，Ｚグラフによる推定が，その簡便さ，視覚性，適用データ範囲の広さ（収束範囲の広さ）などの点から見て威力を発揮できることを意味する．

5. ま と め

(１) $\delta = 0$ とした統合モデルでは，Ｚグラフを用いることにより，累積バグ曲線の断片からパラメータ β と γ，すなわち推定すべきフォールトの検出速度と曲線の形状の局所的な値を容易に推定できる．パラメータ β の推定値は，累積バグ曲線の時間軸方向の平行移動に対して不変であり，パラメータ γ は累積バグ曲線の時間軸方向の平行移動および時間軸のスケール変換（縮尺/拡大）に対して不変である．

(２) 実データをＺグラフ上にプロットすることにより，パラメータ γ の局所的な値を示すグラフの傾きの変化点を容易に把握できる．２つの実データに対してＺグラフを描いて分析したところ，グラフの傾きが時刻 t により大きく変動することが分かった．このことは実データが各テスト段階を通して１つの従来モデルの仮定を満たし続けてはいないことを示している．

(３) Ｚグラフを用いて統合モデルのパラメータを推定し，残存バグ数を推定した結果，従来の最尤推定法あるいはＹ方程式法と推定誤差に差はないことが分かった．

参 考 文 献

1) Jelinski, Z. and Moranda, P.: Software Reliability Reseach, *Statistical Computer Performance Evaluation*, Freiberger, W. (Ed.), pp.465–484, Academic Press, New York (1972).
2) Musa, J.D.: A Theory of Software Reliability and Its Application, *IEEE Trans. Softw. Eng.*, Vol.SE-1, No.3, pp.312–327 (1975).
3) Littlewood, B.: Theories of Software Reliability: How Good Are They and How Can They Be Improved?, *IEEE Trans. Softw. Eng.*, Vol.SE-6, No.5, pp.489–500 (1980).
4) Goel, A.L. and Okumoto, K.: Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, *IEEE Trans. Rel.*, Vol.R-28, No.3, pp.206–211 (1979).
5) Yamada, S., Ohba, M. and Osaki, S.: S-

Shaped Reliability Growth Modeling for Software Error Detection, *IEEE Trans. Rel.*, Vol.R-32, No.5, pp.475–478 (1983).
6) Ohba, M.: Software Reliability Analysis Modeles, *IBM J. Res. Dev.*, Vol.28, No.4, pp.428–443 (1984).
7) Tohma, M., Tokunaga, K., Nagase, S. and Murata, Y.: Structural Approach to the Estimation of the Number of Residual Software Faults Based on the Hyper-geometric Distribution, *IEEE Trans. Softw. Eng.*, Vol.SE-15, No.3, pp.345–355 (1989).
8) Kanoun, K., Martini, M.R.B. and Souza, J.M.: A Method for Software Reliability Analysis and Prediction Application to the TROPICO-R Swiching System, *IEEE Trans. Softw. Eng.*, Vol.SE-17, No.4, pp.334–344 (1991).
9) Karunanithi, N., Whitly, D. and Malaiya, Y.K.: Prediction of Software Reliability Using Connectionist Models, *IEEE Trans.Softw.Eng.*, Vol.SE-18, No.7, pp.563–574 (1992).
10) 古山恒夫，中川　豊：ソフトウエア信頼度成長曲線に関する統合モデルと有効性の検証，情報処理学会ソフトウエア工学研究会，Vol.97-10, pp.73–80 (1994).
11) Furuyama, T. and Nakagawa, Y.: A Manifold Growth Model that Unifies Software Reliability Growth Models, *Int. J. of Reliability, Quality and Safety Engineering*, Vol.1, No.2, pp.161–184 (1994).
12) 古山恒夫：ソフトウエア信頼度成長モデルに関する統合モデルの解析的パラメータの推定法，情報処理学会論文誌，Vol.37, No.12, pp.2326–2333 (1996).
13) Nakagawa, Y. and Hanata, S.: An Error Complexity Model for Software Reliability Measurement, *Proc. 11th ICSE*, pp.230–236 (1989).
14) 三觜　武：ソフトウエアの品質評価法，日科技連出版社 (1981).

(平成 11 年 7 月 30 日受付)
(平成 12 年 7 月 5 日採録)

古山　恒夫（正会員）

1945 年生．1968 年東京大学工学部計数工学科卒業．1973 年同大学院博士課程修了．同年日本電信電話公社入社．横須賀電気通信研究所で，拡張型言語，Ada，Common LISP 等の言語処理プログラムの研究実用化に従事．日本電信電話（株）ソフトウエア研究所でソフトウエアプロジェクト管理法，ソフトウエア品質保証法，ソフトウエア見積り法等の研究実用化に従事．1996 年より東海大学開発工学部情報通信工学科教授．工学博士．平成 6 年度山下記念研究賞受賞．IEEE，電子情報通信学会，ヒューマンインタフェース学会，日本ロボット学会各会員．