# 連続状態ソフトウェア信頼性モデル

安藤隆夫, 土肥正 (01307065), 岡村寛之 (01013754)
広島大学大学院工学研究科情報工学専攻

## 1. はじめに

従来までソフトウェア信頼性を定量的に評価するために，非定常ポアソン過程 (NHPP) に代表される点過程に基づいたソフトウェア信頼性モデルが数多く提案されてきた [1, 2]. これらは，テスト工程においてソフトウェアに含まれるフォールトが発見･除去される時間的挙動をモデル化したもので，ソフトウェア信頼度，期待残存フォールト数，瞬間 MTBF などのソフトウェア信頼性評価尺度を定量的に評価するために有用である．反面，離散状態空間上で定義されるこれらのソフトウェア信頼性モデルを，管理図法などに代表されるテスト工程の進捗度管理（例えば [3]）に応用する場合，最終的に正規近似などを適用して連続状態モデルとして扱う方が便利なことが多い.

これまでに，対数正規過程（もしくは幾何 Brown 運動過程）に基づいた連続状態ソフトウェア信頼性モデルが提案されている [4] が，それらは従来までの NHPP を基本としたソフトウェア信頼性モデルとは全く異質の枠組みで定式化されている．本稿では，ソフトウェアの逐次的品質評価への適用を念頭に入れ，従来までの NHPP に基づいたモデル化の枠組みと矛盾することのないより自然なアプローチにより，文献 [4] とは若干異なる連続状態ソフトウェア信頼性モデルを提案する．提案された連続状態ソフトウェア信頼性モデルは O-U 過程 (Ornstein-Uhlenbeck processes) をベースにした非定常拡散過程であり，ソフトウェアの品質評価に適用することが容易であるだけでなく，従来からの NHPP モデルとの整合性も保たれている.

## 2. NHPP モデル

まず最初に，NHPP に基づいたソフトウェア信頼性モデルについて概説する．テストを開始する前にソフトウェア内に残存する初期フォールト数の期待値を $a\,(>0)$ とし，$\{X(t), t \geq 0\}$ を時刻 $t$ までに発見される累積フォールト数を表す確率過程とする．ここで，$X(0) = 0$ および $\lim_{t\to\infty} \mathrm{E}[X(t)] = a$ とする．Goel and Okumoto [1] に従って，単位テスト時間当たりに発見されるフォールト数はその時点でソフトウェア内に残存するフォールト数に比例するものと仮定し，形式的に

$$dX(t) = b(a - X(t))dt + dM(t), \quad X(0) = 0 \tag{1}$$

のような確率微分方程式を定義する [5]．ここで，確率過程 $\{M(t), t \geq 0\}$ は連続時間マルチンゲールであり，$b\,(>0)$ はフォールト発見率を表す正定数である．$M(t)$ がポアソン測度上のマルチンゲールであれば $X(t)$ は平均 $\mathrm{E}[X(t)] = \Lambda(t)$ のポアソン過程となり，式 (1) の両辺の期待値をとることにより常微分方程式 $d\Lambda(t) = b(a - \Lambda(t))dt$ を得る．これを初期条件 $\Lambda(0) = X(0) = 0$ の下で解くことにより，平均値関数

$$\Lambda(t) = a(1 - e^{-bt}) \tag{2}$$

を求めることができ，$X(t)$ の確率関数は

$$\Pr\{X(t) = x \mid X(0) = 0\} = \frac{\Lambda(t)^x e^{-\Lambda(t)}}{x!} \tag{3}$$

となる．これは指数形ソフトウェア信頼性モデル [1] と呼ばれ，$\mathrm{Var}[X(t)] = \Lambda(t)$ であり，初期フォールト数 $\lim_{t\to\infty} X(t)$ は平均 $a$ のポアソン分布に従うことがわかる.

同様に，障害の発生過程とフォールトの認知過程をモデル化した次のような確率微分方程式

$$dX(t) = b\{a(1 - e^{-bt}) - X(t)\}dt + dM(t) \tag{4}$$

を考えると，両辺の期待値をとることにより

$$\Lambda(t) = a\{1 - (1 + bt)e^{-bt}\} \tag{5}$$

を得る．このモデルは遅延 S 字形ソフトウェア信頼性モデル [2] と呼ばれ，指数形モデルと並んで最も古典的な NHPP モデルのひとつである.

実際のテスト工程において採集されたフォールトデータに基づいて NHPP の平均値関数の推定値 $\hat{\Lambda}(t)$ を導出し，ソフトウェア信頼度，期待残存フォールト数，MTBF などのソフトウェア信頼性評価尺度を求める．また，確率 $\alpha$ で推定される $\hat{\Lambda}(t)$ の存在範囲として $100\alpha\%$ 信頼限界を求め，テスト工程の進捗度管理に適用する．最も直接的な方法は，$\Pr\{X(t) \leq x_l\} = \alpha$ と $\Pr\{X(t) \geq x_u\} = \alpha$ を満たす $(x_l, x_u)$ を各時点 $t$ において求め，$100\alpha\%$ 管理限界線をプロットすることである．しかしながらこの方法では，フォールトデータを取得しながら，将来における任意の時点で連立方程式を解く必要があるため，あまり効率の良い方法ではない．よって，最も簡便な代替案は NHPP の正規近似を行うことである．すなわち，

$$X(t) \sim N\left(\hat{\Lambda}(t), \hat{\Lambda}(t)\right). \tag{6}$$

ここで，$N(\cdot,\cdot)$ は正規分布を表す．$k_\alpha$ を標準正規分布の $100(1+\alpha)/2$ パーセント点とすれば，累積発見フォールト数の $100\alpha\%$ 信頼限界は，近似的に

$$\hat{\Lambda}(t) \pm k_\alpha \sqrt{\hat{\Lambda}(t)} \tag{7}$$

によって与えられる.

## 3. 連続状態モデル

ここでは, NHPP を正規近似するのではなく, 式 (1) の連続時間マルチンゲール $\{M(t), t \geq 0\}$ を Wiener 測度上のマルチンゲールとして考える. すなわち, 任意のスケールパラメータ $\sigma$ $(>0)$ に対し $M(t) = \sigma B(t)$ と仮定する. ここで, $\{B(t), t \geq 0\}$ は標準 Wiener 過程であり, $\mathrm{E}[B(t)] = 0$, $\mathrm{Var}[B(t)] = t$ を満たす. これより, 式 (1) の確率微分方程式は

$$dX(t) = b(a - X(t))dt + \sigma dB(t) \tag{8}$$

となり, 平均復帰 O-U 過程 (mean-reverting O-U process) と呼ばれる. $X(0) = 0$ の下でこれを解くと,

$$X(t) = a(1 - e^{-bt}) + \sigma \int_0^t e^{-b(t-s)} dB(s) \tag{9}$$

を得る. 明らかに,

$$\begin{aligned} \mathrm{E}[X(t)] &= \Lambda(t) = a(1 - e^{-bt}) \to a \quad (t \to \infty) \\ \mathrm{Var}[N(t)] &= \frac{\sigma^2}{2b}(1 - e^{-2bt}) \to \frac{\sigma^2}{2b} \quad (t \to \infty) \end{aligned}$$

である. 式 (9) から明らかに,

$$\begin{aligned} X(t) &\sim N\Big(a(1 - e^{-bt}), (\sigma^2/2b)(1 - e^{-2bt})\Big), \\ \lim_{t \to \infty} X(t) &\sim N(a, \sigma^2/(2b)) \end{aligned}$$

であり, 文献 [4] で議論されたモデルとは大きく異なる. スケールパラメータが $\sigma = \sqrt{2ab}$ のとき初期フォールト数は $N(a, a)$ に従うが, 任意の $t$ に対する $X(t)$ の分散は Goel and Okumoto [1] モデルとは若干ずれることがわかる.

式 (4) の遅延 S 字形ソフトウェア信頼性モデルと同様に, 次のような確率微分方程式

$$dX(t) = b\{a(1 - e^{-bt}) - X(t)\}dt + \sigma dB(t) \tag{10}$$

を考えれば, その解過程は

$$X(t) = a\{1 - (1 + bt)e^{-bt}\} + \sigma \int_0^t e^{-b(t-s)} dB(s) \tag{11}$$

となり, やはり正規分布に従う非定常拡散過程に帰着される. このように, ポアソン測度上で表現される代表的な NHPP モデルは, Wiener 測度上の連続状態モデルに変換することが可能である.

## 4. 数値例

2 種類のソフトウェアフォールトデータ (DS-1, DS-2) を用いてモデルの比較を行った. DS-1 と DS-2 は, それぞれ実際のテスト工程で採集された 137 個と 397 個のフォールト発見時間データ (CPU time) である. 観測された全てのデータを使用して最尤法によりパラメータの推定を行い, 各モデルごとに情報量基準 AIC を算出する. 表 1 において, NHPP, Normal, Lognormal はそれぞれ非定常ポアソン過程, O-U 過程, 対数正規過程に基づいたモデルを表しており, Exponential, S-shaped は **2.** と **3.** で述べた指数形モデルと S 字形モデルを意味している.

表 1: AIC に基づいたモデル比較.

| DS-1 | | | |
|---|---|---|---|
| Model | NHPP | Normal | Lognormal |
| Exponential | 1970 | 616 | 572 |
| S-shaped | 2094 | 623 | |
| **DS-2** | | | |
| Model | NHPP | Normal | Lognormal |
| Exponential | 4793 | 1691 | 1699 |
| S-shaped | 5078 | 1716 | |

この表より, 非定常ポアソン過程に基づいたモデルよりも Brown 運動をベースにしたモデルの方が AIC の値を低く抑えることが可能であり, 情報量基準の観点からは良好であると言える. このような傾向は他のフォールトデータに対しても観測されることから, AIC を基本としたモデル評価においては連続状態モデルの方が点過程モデルよりも有利であると結論付けることが出来る. Normal と Lognormal の比較についてはデータの種類に依存するため一概にそれらの優位性について結論付けることは出来ないが, あまり複雑なモデル化を行うよりも単純なモデルの方が AIC を低めに抑える傾向にあることがわかった.

一般に NHPP などの点過程に基づいたソフトウェア信頼性モデルでは, 平均値関数を累積発見フォールト数のデータによく適合させることが出来るので, 平均値の意味では元のデータをよく説明出来ていると言えるが, 確率分布自体の適合性を見る AIC などの比較では必ずしも良い結果を与えるとは限らない. フォールト発見過程という離散状態空間上で定義されるべき確率過程をより精密に記述するためには, さらに説得力のある離散状態モデルを開発するか, 近似モデルとしての O-U 過程モデルをさらに理論武装することが必要であろう.

## 参考文献


[1] A. L. Goel and K. Okumoto, Time-dependent error-detection rate model for software reliability and other performance measures, *IEEE Trans. Reliab.*, **R-28** (3), 206–211, 1979.

[2] S. Yamada, M. Ohba and S. Osaki, S-shaped reliability growth modeling for software error detection, *IEEE Trans. Reliab.*, **R-32** (5), 475–478, 1983.

[3] K. Okumoto, A statistical method for software quality control, *IEEE Trans. Software Eng.*, **SE-11** (12), 1424–1430 (1985).

[4] S. Yamada, M. Kimura, H. Tanaka and S. Osaki, Software reliability measurement and assesment with stochastic differential equations, *IEICE Trans. Fundamentals*, **E77-A** (1), 109–116, 1994.

[5] G. Koch and P. J. C. Spreij, Software reliability as an application of martingale & filtering theory, *IEEE Trans. Reliab.*, **R-32** (4), 342–345 (1983).