信頼度成長モデルを用いた安全関連系ソフトウェアの安全度解析

On Safety Analysis for Safety-Related Software Based on Reliability Growth Models

関西大学・総合情報学部　井上 真二
SRATECH Laboratory 株式会社　藤原 隆次
鳥取大学・大学院工学研究科　山田　茂

Shinji Inoue (Faculty of Informatics, Kansai University)
Takaji Fujiwara (SRATECH Laboratory Inc.)
Shigeru Yamada (Graduate School of Engineering, Tottori University)

1　はじめに

電気・電子・プログラマブル電子（electrical/electronic/programmable electronic：E/E/PE）安全関連系は，全体システムの安全性を機能的側面から担うシステムとして，プラントや自動車関連など，近年その適用分野や急激に拡大している．この中で，2020 年には，E/E/PE 安全関連系に対する国際基本機能安全規格である IEC 61508 [1] が発行され，各産業界では，この規格に準拠した E/E/PE 安全関連系の設計・開発が国際的に広く求められるようになってきている．IEC 61508 は，E/E/PE 安全関連系の概念設定から使用終了フェーズまでの「全安全ライフサイクル」を設定しながら，各フェーズにおける安全要求事項を規定している．特に，IEC 61508 では，各安全要求事項に対して，安全度水準（SIL）を基盤とした機能安全評価を求めている点が特徴的であると言える．

一般的に，E/E/PE 安全関連系は，全体システムが果たすべき意図機能を担う非制御装置（equipment under control: EUC）や基本制御システム（basic control system：BCS）とは別に，それらに対して付加的に取り付けられるシステムである．つまり，E/E/PE 安全関連系は，EUC や BCS から成るシステムが担うべき所定の安全機能が遂行できなくなったとき，それらから作動要求を受け，全体システムの安全性を機能的側面から最終的に確保するようなシステムとして認識されている．したがって，通常，E/E/PE 安全関連系のハードウェアシステムに焦点を当てた安全性評価では，E/E/PE 安全関連系への作動要求頻度（低頻度作動要求モードと高頻度作動要求・連続モード）に従い，それぞれに定められた目標機能失敗尺度によって，安全度水準（safety integrity level：SIL）と呼ばれる定義された安全度水準への割り当てが行われる．このように IEC 61508 では，特に，E/E/PE 安全関連系のハードウェアシステムに対して，その摩耗や劣化によるランダムハードウェア故障の概念に基づき，各々の目標機能失敗尺度を定量的に算出するための基本的な枠組みが与えられている．また，実務への適用に関しては，当該規格に沿ったテーラリングも必要であり，特に，E/E/PE 完全関連系のハードウェアシステムにおける様々な内部構造を意識した新たな目標機能失敗尺度の導出方法など研究面からの注目も浴びつつある [4]．一方，ソフトウェア故障に起因する E/E/PE 安全関連系の安全性評価では，そもそも，その故障原因がソフトウェア要求定義，設計，コーディング時において作り込まれた誤りによるものであり，いわゆる決定論的原因によるものであるとの基本的な考え方がある [7]．したがって，ハードウェアシステムに対する定量的な安全性評価手法とは異なり，ソフトウェアに対しては，一定の SIL を達成するためにソフトウェア開発過程において求められる開発技法が提示されるのみで，目標機能失敗尺度など定量的な尺度に基づいた SIL の割り当ては求められていない．

しかしながら，安全性を脅かすソフトウェア故障の原因が「決定論的原因」であったとしても，数あるソフトウェアパスの実行頻度にはばらつきがあり，バグが作り込まれたソフトウェアパスが実行されソフトウェア故障を観測する時間（もくしは時間間隔）はランダムに振る舞い，不確実性を有することが考えられる．また，求められる開発技法を適用したとしても，SIL との関係性は必ずしも担保されるとは限らないことも指摘されている [2]．したがって，IEC 61508 に準拠した一定の安全度水準を満たしたソフトウェアを開発するためには，ソフトウェア開発過程において規格で定められた開発技法の適用を行うだけでなく，最終的な成果物（ソフトウェアシステム）に対する定量的な安全性評価手法の開発が求められる．な

表 1： IEC 61508 において定義された安全度水準.

SIL	Low Demand Mode	High Demand or Continuous Mode
4	$10^{-5} \leq \text{PFD} < 10^{-4}$	$10^{-9} \leq \text{PFH (1/hour)} < 10^{-8}$
3	$10^{-4} \leq \text{PFD} < 10^{-3}$	$10^{-8} \leq \text{PFH (1/hour)} < 10^{-7}$
2	$10^{-3} \leq \text{PFD} < 10^{-2}$	$10^{-7} \leq \text{PFH (1/hour)} < 10^{-6}$
1	$10^{-2} \leq \text{PFD} < 10^{-1}$	$10^{-6} \leq \text{PFH (1/hour)} < 10^{-5}$

お，ソフトウェア故障を観測する時間（もくしは時間間隔）の不確実性を意識しながら，確率・統計則に従った定量的なソフトウェア信頼性評価技術については，従前より精力的に研究が行われており，これらの技術を定量的なソフトウェア安全性評価へ活用するアプローチは，その第 1 歩として考え得る．本研究では，E/E/PE 安全関連系のソフトウェアについて，既存のソフトウェア信頼度成長モデルに基づいた信頼性評価技術 [6,8] を適用かつ軽微な改良を施しながら，SIL として規定された安全度水準に準拠した定量的なソフトウェア安全性評価を行うための目標機能失敗尺度の近似的算出方法について議論する．また，ソフトウェア信頼性データを用いた提案手法の適用例についても示す.

2　ソフトウェア目標機能失敗尺度

SIL は，運用時における E/E/PE 安全関連系への作動要求頻度に応じた所定の目標機能失敗尺度に基づいて，4 つのレベルに等級化された安全度水準である．E/E/PE 安全関連系への作動要求頻度が年に 1 回以下であれば「低頻度作動要求モード」として運用時の作動要求モードを定め，安全機能の作動要求時危険側機能失敗平均確率（average probability of failure on demand of the safety function：PFD）に基づいて SIL の割り当てが行われる．PFD は，作動要求時において E/E/PE 安全関連系がフォールト状態（ソフトウェアフォールトとは異なる）にある確率であり，E/E/PE 安全関連系のアンアベイラビリティ（不可用性）を確率的に表現する尺度として解釈できる．本研究では，既存のソフトウェア信頼性評価技術を活用することを念頭に，可用性と信頼性の定義は本質的に異なるが，近似的な PFD の算出枠組みとして，

$$\text{PFD} \approx 1 - \text{Software Reliability}, \tag{1}$$

のように与えること考える．一般的に不可用性は，平均故障発生時間間隔や平均修理時間に基づいて算出されるべき一種の信頼性評価尺度であるが，式 (1) は，ソフトウェア故障の平均修復時間がある程度短く一定である状況を想定し，ソフトウェアの信頼性が高ければ可用性も高くなるような関係性に基づいて近似的に与えた枠組みとして与えている．

一方，E/E/PE 安全関連系への作動要求頻度が年に 1 回より多い場合は,「高頻度作動要求・連続モード」に該当し，安全機能の危険側故障の平均頻度（average frequency of a dangerous failure of the safety function [1/h]：PFH）が目標機能失敗尺度として，SIL の割り当てに利用される．PFH の単位は 1/hour であり，いわゆるハザードレード（故障率）として解釈できる．したがって，本研究では PFH に対して，

$$\text{PFH} = \frac{1}{\text{MTBF}}, \tag{2}$$

を，PFH を近似的に基本的に算出するための枠組みとして与えることにする.

3　ソフトウェア信頼度成長モデルの適用

目標機能失敗尺度に関する前述した基本的算出枠組みに含まれるソフトウェア信頼性評価尺度は，ソフトウェア信頼度成長モデルに基づいて与えることを考える．しかしながら，既存のソフトウェア信頼性評価技術では，一般的に，安全側もしくは危険側故障を区別せず，これをすべてソフトウェア故障として扱うた

め，新たに，危険側ソフトウェア故障発生率 DFR($0 < \mathrm{DFR} \leq 1$) を導入しながら，当初枠組みに基づいた改良を行う必要がある．また，テスト環境と運用環境におけるソフトウェア故障観測強度も異なることが容易に想定されるため，双方におけるソフトウェア実行環境の違いとこれらの関係性を環境関数 $EF(\cdot)$ を用いて表現することにする．本研究ではこれらの点を考慮しながら，ソフトウェア信頼度成長モデルとして，ソフトウェアフォールト発見数モデルとソフトウェア故障発生時間モデルを用いた場合について，それぞれ，目標機能失敗尺度の近似的算出式を与える．

いま，$\{N(t), t \geq 0\}$ を任意のテスト時刻 t までに発見・修正されたソフトウェアフォールト数を表す計数過程とする．なお，ここでは，ソフトウェア故障の観測後，その原因となるソフトウェアフォールトは直ちにかつ完全に修正・除去されるものと仮定する．特に，この計数過程 $N(t)$ が非同次ポアソン過程（nonhomogeneous Poisson process：NHPP）：

$$\Pr\{N(t) = n\} = \frac{\{\Lambda(t)\}^n}{n!} e^{-\Lambda(t)} \quad (n = 0, 1, 2, \cdots), \tag{3}$$

に従うと仮定するモデルは，NHPP モデル [6, 8] と呼ばれ，広く実用に供されているソフトウェアフォールト発見数モデルの一つとして知られている．式 (3) において，$\Lambda(t)$ は計数過程 $N(t)$ の期待値を表す平均値関数を呼ばれ，任意の時刻 t までの発見された総期待フォールト数を表す．これより，式 (3) に従い，ソフトウェア信頼性評価に有用な様々な信頼性評価尺度が導出できる．ソフトウェア信頼度関数 $R(x \mid t)$ は，テスト作業が時刻 t まで実施された後の時間区間 $(t, t + x]$ においてソフトウェア故障が発生しない確率として定義され，式 (3) から，

$$R(x \mid t) \equiv \Pr\{N(t + x) - N(t) = 0\} = \exp[-\{\Lambda(t + x) - \Lambda(t)\}], \tag{4}$$

として導出される．また，平均ソフトウェア故障発生時間間隔（MTBF）の代替的尺度として知られる瞬間 MTBF は，

$$MTBF_I(t) = \frac{1}{\lambda(t)}, \tag{5}$$

と与えられる．式 (5) において，$\lambda(t)(= d\Lambda(t)/dt)$ は，NHPP の強度関数であり，任意の時刻における瞬間的なソフトウェア故障発生頻度を表す．ソフトウェアフォールト発見数モデルにおける上述したソフトウェア信頼性評価尺度から，式 (1) および式 (2) のソフトウェア目標機能失敗尺度をそれぞれ，

$$\begin{aligned} PFD(x \mid t) &= 1 - \exp\left[-\mathrm{DFR} \cdot \{\Lambda(t + EF(x)) - \Lambda(t)\}\right] \\ &= 1 - \exp\left[-\mathrm{DFR}\{\Lambda(t + \mathrm{EC}_{fc} \cdot x) - \Lambda(t)\}\right], \end{aligned} \tag{6}$$

$$PFH(t) = \frac{\mathrm{DFR}}{EF^{-1}(MTBF_I(t))} = \frac{\mathrm{DFR} \cdot \mathrm{EC}_{fc}}{MTBF_I(t)}, \tag{7}$$

のように与える．式 (6) および式 (7) において，環境関数は $EF(x) = \mathrm{EC}_{fc} \times x$ と与え，$\mathrm{EC}_{fc}(> 0)$ は，テスト環境と運用環境の関係性を示す環境係数である．この環境関数は，設計運用期間 x をテスト環境下における時間に換算している．また，$EF^{-1}(\cdot)$ は環境関数の逆関数である．

ソフトウェア故障発生時間モデルは，各ソフトウェア故障発生時間間隔の確率的挙動に焦点を当てたモデルであり，各ソフトウェア故障発生時間間隔に対するハザードレートを適切に規定する必要がある．本研究では，テスト環境下における $(k-1)$ 番目から k 番目のソフトウェア故障発生時間間隔を表す確率変数 X_k のハザードレートを $z_k^T(x)$ として，運用環境下でのハザードレート $z_k^O(x)$ との関係を，

$$z_k^O(x) = EF(z_k^T(x)) = \mathrm{EC}_{ft} \cdot z_k^T(x) \qquad (\mathrm{EC}_{ft} > 0), \tag{8}$$

と与える．ここで，$\mathrm{EC}_{ft}(> 0)$ は，テスト環境下と運用環境下におけるハザードレートの関係性を示す環境係数である．これより，ソフトウェア故障発生時間分布に対する目標機能失敗尺度を，それぞれ，式 (1) および式 (2) に基づいて，

$$PFD_k = \mathrm{DFR}\left\{1 - (1 - F_k^O(x))\right\} = \mathrm{DFR} \cdot \left[1 - \exp\left\{-\mathrm{EC}_{ft} \int_0^x z_k^T(\theta) \mathrm{d}\theta\right\}\right], \tag{9}$$

$$PFH_k = \frac{\mathrm{DFR}}{MTBF_k^O} = \frac{\mathrm{DFR} \cdot \mathrm{EC}_{ft}}{MTBF_k^T}, \tag{10}$$

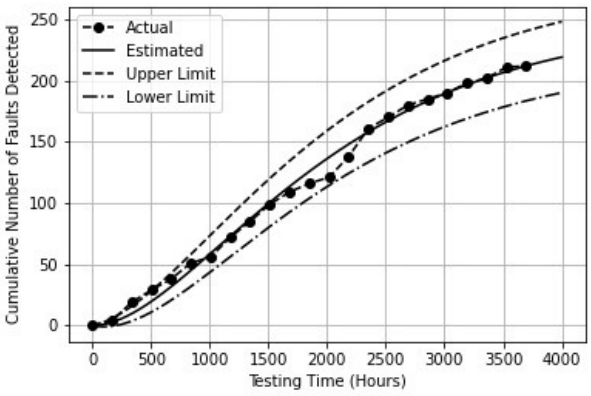

図 1： 推定された遅延 S 字形ソフトウェア信頼度成長モデルと 95%信頼限界.

のように与える．ここで，$F_k^O(x)$ および $MTBF_k^O$ は，それぞれ，運用環境下における $(k-1)$ 番目から k 番目のソフトウェア故障発生時間間隔に対する確率分布関数および MTBF であり，ハザードレートの一般的な定義式および式 (8) からそれぞれ導出できる．例えば，$z_k^T(x)$ に対して，Moranda モデル [5]：$z_k^T(x) \equiv z_k^T = Dc^{k-1} \quad (D > 0, 0 < c < 1; k = 1, 2, \cdots)$ を適用した場合，目標機能失敗尺度はそれぞれ，

$$PFD_k = DFR \cdot F_k^T(\mathrm{EC}_{ft} \times x), \tag{11}$$

$$PFH_k = DFR \cdot \mathrm{EC}_{ft} \cdot z_k^T(x), \tag{12}$$

と求められる．ここで，D は 1 番目のソフトウェア故障発生に対する初期ハザードレート，c はハザードレートの減少係数である．

4 適用例

本稿では，紙面の都合上，フォールト発見数モデルを適用した場合における提案アプローチの適用例を示す．適用する実測データは，S 字形ソフトウェア信頼度成長曲線を示す 22 組のフォールト発見数データ [3]：$(t_k, y_k)(k = 1, 2, \cdots, 22; t_{22} = 22(\text{週}), y_{22} = 212)$ であり，y_k はテスト時刻 t_k までに修正・除去されたフォールト数の累積値を示す．ただし，E/E/PE 安全関連系ソフトウェアのテスト工程にて収集されたデータは現時点で収集できず，今回は，文献等にて掲載されているデータを適用したことを付記しておく．いま，式 (3) の平均値関数に対して，広く実用に供されているフォールト発見数モデルの一つとし知られている遅延 S 字形ソフトウェア信頼度成長モデル [8] ：$\Lambda(t) = a[1 - (1 + bt)\exp[-bt]](a > 0, b > 0)$ を適用した場合を考える．ここで，a はテスト開始前にソフトウェア内に潜在する総期待フォールト数，b はフォールト 1 個当たりのフォールト発見率を表す．なお，モデル含まれるパラメータ a および b は，PFH の単位に従うように単位時間の変換を行い適用した上述のソフトウェアフォールト発見数データから，NHPP の確率的性質に基づいた最尤法を用いて推定した．まず，図 1 に，推定された遅延 S 字形ソフトウェア信頼度成長モデル $\widehat{\Lambda}(t)$ とその 95% 信頼限界の挙動を示す．また，コルモゴロフ・スミルノフ適合度検定の結果，推定された遅延 S 字形ソフトウェア信頼度成長モデルは実測データに対して有意水準 5% で適合していることを確認できた．

本稿において，提案手法の適用例を示すにあたり，今回は一例として $\mathrm{DFR} = 0.01$ および $\mathrm{EC} = 0.01$ とそれぞれ設定した．なお，DFR や EC の設定については，E/E/PE 安全関連系ソフトウェアに対するテスト工程で得られる DFR に関する情報，およびテスト工程および運用段階におけるパス実行網羅度の時間的達成率などの情報に基づいて EC を決定する手法が考えられる．また，設計上の運用期間については，E/E/PE 完全関連系の運用時に一般的に実施される保守作業（プルーフテスト）の実施間隔などに基づいて与えられるが，今回は，1 年間（$= 8,760$ 時間）とした．これより，PFD および PFH の算出式は，式 (6) および

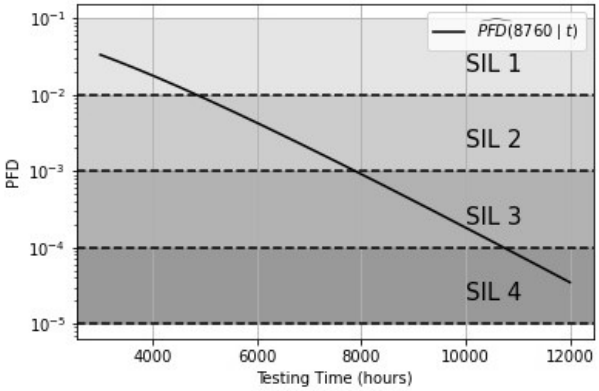

図 2： 推定された PFD の挙動.

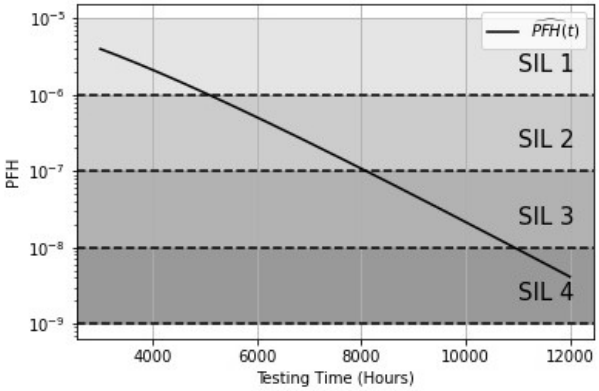

図 3： 推定された PFH の挙動.

式 (7) から，それぞれ，

$$PFD(8,760 \mid t) = 1 - \exp[-0.01\,\{\Lambda(t + 0.01 \times 8,760) - \Lambda(t)\}], \tag{13}$$

$$PFH(t) = \frac{\mathrm{DFR}}{MTBF_I(t)/\mathrm{EC}} = \frac{0.01}{100 \times MTBF_I(t)}. \tag{14}$$

となる．図 2 および図 3 に，それぞれ，式 (13) および式 (14) に基づいて推定された PFD および PFH の挙動をそれぞれ示す．図 2 および図 3 から，テストの実施に伴い，目標機能失敗尺度は次第に減少していく様子がわかる．また，これらの推定結果から，ある一定の SIL を満たすために必要なテスト期間についても推定できる．表 2 に，図 2 および図 3 に基づいて，一定の SIL を達成するために必要なテスト期間を示す．例えば，図 2 および表 2 から，所定の安全性要求下において，例えば低頻度作動要求モードにて SIL 2 を満たすためには，当初のテスト終了時刻（22 週 $= 3,696$ 時間）から，さらに，約 7 週間ほど引き続きテスト作業が必要であることがわかる．また，当該ソフトウェアが高頻度作動要求・連続モードにて運用され，同様に SIL 2 を達成する必要がある場合，図 3 および表 2 から，当初のテスト終了時刻からさらに約 8 週間ほど引き続きテスト作業が必要となることがわかる．このように，今回議論した手法は，一定の SIL を満たすために必要なテスト実施期間を推定することができ，SIL に基づいたソフトウェア開発管理を支援する方法を与えるものと考えられる．

表 2： 一定の SIL を満たすために最低限必要なテスト期間

SIL	Testing Time Required at Least (Hours)	
	Low Demand Mode	High Demand or Continuous Mode
4	10728	10932
3	7887.5	8098.5
2	4849.4	5084.4
1	1045.4	1088.9

5 おわりに

本研究では，E/E/PE 安全関連系ソフトウェアに対し，より客観的な安全性評価の実現を目的として，ソフトウェア信頼度成長モデルを活用しながら，IEC 61508 において定義される SIL を割り当てるための近似的な目標機能失敗尺度の算出アプローチについて議論した．現行の IEC 61508 では，ソフトウェア開発プロセスにおいて適用した技法等など，定性的なアプローチによって SIL の割り当てがなされるが，データに基づいた SIL の割り当ては，客観的な安全性評価や開発されたソフトウェアの実態に即した SIL の割り当てが可能となる．また，今回のアプローチは，要求される SIL を満足するソフトウェアを開発するために必要なテスト時間の推定など，SIL に基づいた一定の安全性要求を満たすソフトウェア開発を支援する管理技術としても期待できる．一方で，DFR や $EF(\cdot)$ の数理的構造をはじめ，EC の設定方法など，実際の環境への適用手法に関して取り組むべき課題が残る．また，PFD に対する近似的な算出式については，本来の定義に準拠しつつ，ソフトウェアの可用性を陽に意識したモデルの改良が求められる．今後は，E/E/PE 安全関連系ソフトウェアのテスト工程から得られる実際のソフトウェア信頼性データも適用しながら，上述の課題について取り組み，E/E/PE 安全関連系ソフトウェアについて，SIL 割り当てのためのより適切な目標機能失敗尺度の算出方法について議論する必要がある．

参考文献

[1] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, Edition 2.0, 2010.

[2] T. Fujiwara, M. Kimura, Y. Satoh and S. Yamada, “A method of calculating safety integrity level for IEC 61508 conformity software,” *Procceedings of the 17th IEEE Pacific Rim International Symposium on Dependable Computing*, IEEE Computer Society, 2011, pp. 296–301.

[3] S. Inoue and S. Yamada, “Discrete program-size dependent software reliability assessment: modeling, estimation, and goodness-of-fit comparisons,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E90-A, No. 12, pp. 2891–2902, 2007.

[4] E. Kato and Y. Satoh, “Safety integrity level model for IEC 61508 — Examination of modes of operation –,” *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E83-A, No. 5, 863–865, 2000.

[5] P.B. Moranda, “Event-altered rate models for general reliability analysis,” *IEEE Transactions on Reliability*, Vol. R-28, pp. 376–381, 1979.

[6] H. Pham, *Software Reliability*, Springer-Verlag, Singapore, 2000.

[7] 佐藤吉信, 「機能安全の基礎」, 日本規格協会，東京，2014.

[8] 山田茂,「ソフトウェア信頼性の基礎 —モデリングアプローチ」, 共立出版，東京，2011.