

カルマンフィルタによる

ソフトウェア開発時誤り予測の数値シミュレーション

Kalman Filtering for Error Prediction in Software Development: Numerical Simulations

矢田部 学* 岡野 麻子**
Manabu Yatabe, Asako Okano

カルマンフィルタ処理は、状態推移を記述する方程式に基づいて現フェーズから次フェーズにおける状態の予測と次フェーズにおける測定結果を反映した予測の修正を交互に行うアルゴリズムである。ソフトウェア開発時の誤りを予測するカルマンフィルタを試作した。それを擬似データに適用し数値シミュレーションを行った。本稿は試作したカルマンフィルタとそのシミュレーション結果について紹介する。

Kalman filtering is an algorithm based on two processes; one predicts next-phase states from the present ones using state equations and the other corrects the predicted states with reference to measurements at the next phase. We designed a Kalman filter experimentally and applied it to predicting errors in software development. Numerical simulations of the Kalman filter were carried out using pseudo data. This report mentions the Kalman filter and its simulations.

1. まえがき

ソフトウェア開発における品質分析・予測に使用されるメトリクスの一つとして、一般的に誤り数が用いられる。我々は、誤り数を誤り検出率という尺度に変換し、プロジェクトの開発期間内で測定される誤り検出率の累積値による予測を行っている。その際、ゴンペルツ曲線を予測モデルとして利用している⁽¹⁾。ソフトウェア開発の誤り予測においてゴンペルツ曲線が採用されるのは、ウォーターフォール開発における誤り検出率の累積傾向と形状が似ているという理由による⁽²⁾。しかし、ゴンペルツ曲線のパラメータは、モデル式を線形化した後で最小二乗法により決定されるため、元のモデル式による予測値の誤差を統計学的に評価しにくいという問題がある。また、ゴンペルツ曲線による予測モデルは測定データのみに基づき、誤りを支配する法則は考慮していない。そこで、それを記述する方程式（状態方程式）があるという仮説を立てた。

状態方程式を考慮した予測モデルとしてカルマンフィルタがある。それは、測定を記述する観測方程式の他に状態推移を記述する状態方程式に基づいて、状態量の予測・推定と誤差共分散を評価する計算アルゴリズムであり、航空宇宙分野において実績がある⁽³⁾⁽⁴⁾。本稿では、

ソフトウェア開発時の誤りを記述する状態方程式と観測方程式に基づくカルマンフィルタによる誤り予測手法を提案する⁽⁵⁾。

統計学的な議論のためには多数の類似プロジェクトのデータが必要である。現実には、組織や開発チームの変更などにより、類似プロジェクトのデータを多数用意することが困難である。そのため、擬似データを用いて試作したカルマンフィルタの数値シミュレーションを行った。

2. 仮説と試み

2.1 仮説

次の仮説を立てた。

- ソフトウェア開発において、開発規模・処理対象・作業者などが類似したプロジェクトでの誤り検出率は同じ統計学的な特性を持つ（同じ母集団の標本）。
- 各開発フェーズにおける誤り検出率を記述できる方程式が存在する。

2.2 試み

以下のように各開発フェーズにおける誤り検出率を推定・予測するカルマンフィルタの試作と数値シミュレー

ションを試みた。

- (a) プロジェクト計画時にゴールとして設定した誤り検出率（計画値）に基づき、それを記述する方程式を定式化する。
- (b) これを状態方程式とし、誤り検出率を推定・予測するカルマンフィルタを設計する。
- (c) ソフトウェア開発の誤り検出率の擬似データを作成し、各開発フェーズにおける誤り検出率を推定・予測するカルマンフィルタモデルの数値シミュレーションを行う。

3. 誤り予測モデル

3.1 前提

開発フェーズにおける誤り検出率（error detection rate: 以下 EDR）の計画値が与えられている。この計画値に基づいて EDR の変化を記述する状態方程式をモデル化する。EDR の測定過程は観測方程式でモデル化する。

3.2 状態方程式

開発フェーズを n で表記する。そこでの状態量 x_n は EDR だけからなるスカラー量とする。この状態量の推移を記述する状態方程式⁽⁶⁾について述べる。これは過去の類似プロジェクトから得られた EDR の計画値 p_n に基づく。計画値の比率 $a_{n-1} \equiv p_n/p_{n-1}$ は、フェーズ $n-1$ から n への EDR の類似プロジェクト固有の変化率を表していると考えられる；フェーズ n における EDR は前フェーズの状態量 x_{n-1} の a_{n-1} 倍とする。状態量は確率的な要素だけでは決まらない。開発環境の背景からの確率的な影響（作りこみ誤りなど）を考慮する必要がある。そこで、状態方程式にはシステムノイズ u_{n-1} を考慮する。このノイズは正規分布 $N(0, \sigma_{sys}^2)$ に従うとする。以上より、状態方程式を以下のように表す：

$$x_n = a_{n-1}x_{n-1} + u_{n-1}. \quad (1)$$

3.3 観測方程式

測定の状況を表す数学モデルが観測方程式⁽⁶⁾である。開発フェーズ n における EDR が直接測定されるとする。

観測方程式には測定環境の確率的な要素（レビュー・テストの観点もれなど）を測定ノイズ v_n として考慮する。このノイズは正規分布 $N(0, \sigma_{obs}^2)$ に従うとする。以上より、EDR の測定値 y_n に関する観測方程式を以下のように表す：

$$y_n = x_n + v_n. \quad (2)$$

3.4 カルマンフィルタ

状態方程式(1)と観測方程式(2)で表される処理系にカルマンフィルタ⁽⁶⁾を適用する。図1において、「システム」は状態方程式、「測定」は観測方程式の処理に対応し、測定の結果が「カルマンフィルタ」に入力される。カルマンフィルタは状態量に対して、状態方程式に基づく予測（時間更新）と測定に基づく推定（観測更新）を交互に行うアルゴリズムである。このため、更新を表す2種類の記号を必要とする。状態の予測に関する更新を、開発フェーズを指定する“ n ”で記述する。測定に関する更新を“(±)”で記述する。ここで“(+)”は測定後、“(-)”は測定前の状態を表す。以下に時間更新と観測更新の処理をまとめる。なお、推定量であることを明示するために状態量にはハット“ $\hat{\cdot}$ ”をつけて表す。

3.4.1 時間更新

現フェーズ n から次フェーズ $n+1$ の状態量（EDR）を予測する。状態量 x_n とその誤差共分散^(*) P_n に関して次の処理を行う⁽⁶⁾。

- (a) 状態量（予測） $\hat{x}_{n+1}(-) = a_n \hat{x}_n(+)$
- (b) 誤差共分散（予測） $P_{n+1}(-) = a_n^2 P_n(+)$

ここで、誤差共分散の中の Q_n はシステムノイズ共分散^(*)と呼ばれるもので、式(1)のシステムノイズが従う

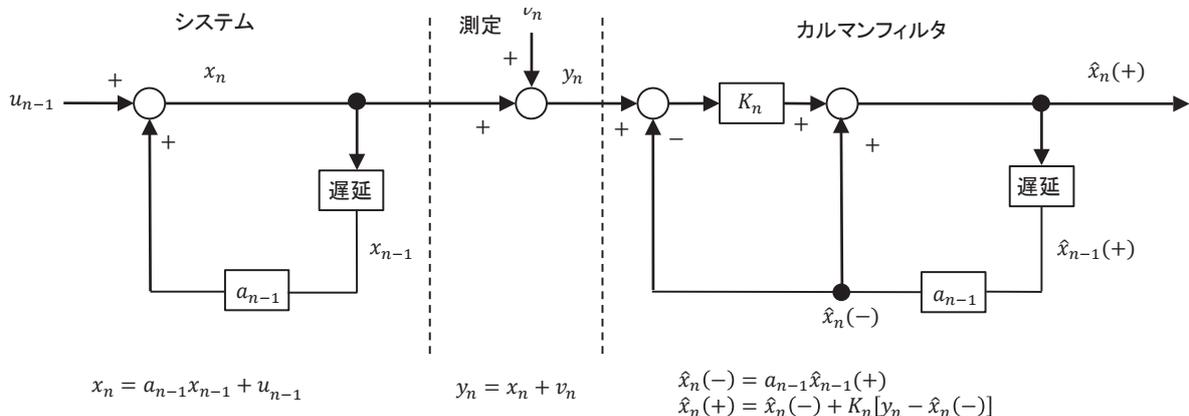


図1 カルマンフィルタの処理

正規分布の分散と $Q_n = \sigma_{\text{sys}}^2$ の関係がある。この処理では、状態方程式(1)に基づいて、開発フェーズ n における状態量 $\hat{x}_n(+)$ から次フェーズの状態量 $\hat{x}_{n+1}(-)$ を予測する。現フェーズの誤差共分散 $P_n(+)$ にシステムノイズ共分散 Q_n を考慮して、次のフェーズの誤差共分散 $P_{n+1}(-)$ を予測する。

3.4.2 観測更新

測定結果に基づいて状態量 (EDR) の補正を行い状態量の真値を推定する。状態量とその誤差共分散に関して次の処理を行う⁽⁶⁾。

- (a) カルマンゲイン $K_n = P_n(-) / [P_n(-) + R_n]$
- (b) 状態量 (推定) $\hat{x}_n(+)=\hat{x}_n(-)+K_n[y_n-\hat{x}_n(-)]$
- (c) 誤差共分散 (推定) $P_n(+)=(1-K_n)P_n(-)$

ここで、カルマンゲインの中に現れている R_n は測定ノイズ共分散^(*)と呼ばれるもので、式(2)の測定ノイズが従う正規分布の分散と $R_n = \sigma_{\text{obs}}^2$ の関係がある。測定前の誤差共分散と測定ノイズ共分散からカルマンゲインが決まる。このゲインを測定値と予測値の差分 $y_n - \hat{x}_n(-)$ に乗じた量で予測値を修正して、測定を反映した推定値 $\hat{x}_n(+)$ が得られる。測定後の誤差共分散 $P_n(+)$ は、 $(1 - K_n) < 1$ が成り立つので、測定前 $P_n(-)$ より小さくなる。

3.5 累積誤り検出率とその誤差

EDR の推定値 $\hat{x}_k(+)$ ($k = 1, 2, \dots$) を累積して、開発フェーズ n における累積誤り検出率 (cumulative error detection rate : 以下 cEDR) を算出する :

$$c_n = [\hat{x}_1(+) + \hat{x}_2(+) + \dots + \hat{x}_{n-1}(+) + \hat{x}_n(+)] \quad (3)$$

カルマンフィルタが出力する誤差共分散 $P_n(+)$ は EDR の推定値 $\hat{x}_n(+)$ の分散を意味する。これより、開発フェーズ n における EDR 推定値の誤差を表す標準偏差は $\sigma_{x_n} = \sqrt{P_n(+)}$ である。これに基づいて cEDR の標準偏差を算出する。各開発フェーズにおける EDR の測定は独立であると仮定する。誤差伝播則⁽⁷⁾を cEDR の定義式(3)に適用すると、フェーズ n における cEDR の標準偏差は以下のように表現できる :

$$\sigma_{c_n} = \sqrt{\sigma_{x_1}^2 + \sigma_{x_2}^2 + \dots + \sigma_{x_n}^2} \quad (4)$$

4. 数値シミュレーション

擬似データを用いて、前章で述べた誤り予測のシミュレーションを行った。EDR の測定は、類似プロジェクトの母集団からそれに属するプロジェクトの標本を抽出することに相当する。この母集団は正規分布 $N(\mu, \sigma^2)$ に

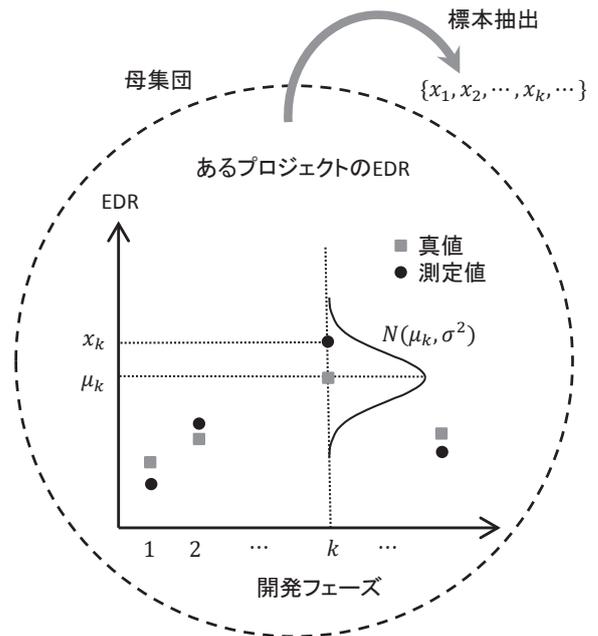


図2 EDR の測定

従うとする。

図2に母集団に属する一つのプロジェクトに着目したときの EDR 測定概念を示す。このプロジェクトの開発フェーズ k における EDR は真値 μ_k を中心として標準偏差 σ で広がっている。このため、一つの開発フェーズ k に着目しても、複数回測定を行うと測定するたびに標本値が異なる。ここでの分散 σ^2 は状態方程式(1)にあるシステムノイズが従う正規分布の分散と同一とする ($\sigma \equiv \sigma_{\text{sys}}$)。

4.1 プロジェクトの擬似データ

以下の手順で EDR の擬似データを作成する。擬似データのノミナル値 (現実のソフトウェア開発では真値) はソフトウェア開発関連の書籍⁽²⁾⁽⁸⁾を参考にした。

4.1.1 擬似データ作成手順

- (a) EDR のノミナル値 μ_n ($n = 1, 2, \dots, 6$) を与える。各 n は実際の開発フェーズ (設計やテストなど) に相当する。
- (b) (a) で与えた μ_n のまわりに正規分布 $N(\mu_n, \sigma^2)$ に従う擬似乱数を発生させて擬似プロジェクトのデータを作成する (過去のプロジェクトの模擬)。
- (c) (b) のデータセットの平均をとり、新たなプロジェクトの計画値とする。
- (d) 新たなプロジェクトの EDR 測定を模擬するデータは、(b) と同様に $N(\mu_n, \sigma^2)$ に従う擬似乱数を発生させて作成する。

4.2 シミュレーション結果

4.1節に述べた手順でEDRを模擬する擬似データを生成しcEDRを評価するシミュレーションを行った。代表的な例として、推定・予測値のエラーバー内にノミナル値が入るケース1と入らないケース2を以下に示す。さらに、これらの結果を踏まえてケース3では、多数のデータに対して推定・予測値のエラーバー内にノミナル値が含まれる割合を調べる。

4.2.1 ケース1

表1のデータを用いてcEDRの変化を算出した。過去の類似プロジェクトの数は5とした。また、EDRの測定は $n=3$ までとした。つまり、 $n=3$ までは時間更新(予測)と観測更新(推定)を行うが、 $n=4$ 以降の測定は行わず、予測値 $\hat{x}_n(-)$ を推定値 $\hat{x}_n(+)$ とした。状態量の初期値は計画値で代用し、誤差共分散の初期値は $P_0(-) = 10$ とした。また、システムノイズと測定ノイズの標準偏差は、それぞれ $\sigma_{\text{sys}} = 0.7$ 及び $\sigma_{\text{obs}} = 0.7$ とした。

そのときのcEDRの変化を図3に示す。ここで示した推定・予測値のエラーバーは式(4)から算出した。

ここでは、推定値の両側に1シグマをとっている。最終フェーズ($n=6$)のノミナル値はこのエラーバーの範囲の中に入った。

表1 擬似 EDR データ (ケース1)

開発フェーズ	1	2	3	4	5	6
ノミナル値	6.8	8.8	25.0	8.5	4.5	2.6
プロジェクト1	6.64	8.68	26.22	7.68	4.93	1.85
プロジェクト2	6.79	9.23	24.85	8.18	4.85	3.65
プロジェクト3	6.40	8.89	23.32	8.36	4.53	2.07
プロジェクト4	7.23	9.04	24.94	9.61	4.13	2.78
プロジェクト5	6.41	8.43	23.83	7.37	3.80	2.24
計画値	6.69	8.85	24.63	8.24	4.45	2.52
測定値	6.38	8.72	25.14	-	-	-

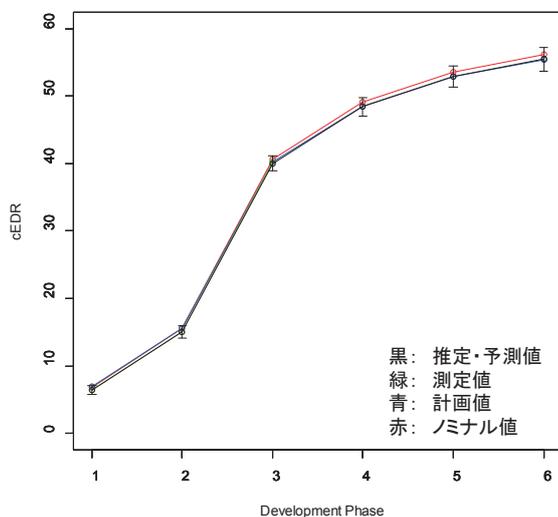


図3 cEDRの変化 (ケース1)

4.2.2 ケース2

表2のデータを用いて、ケース1と同様のシミュレーションを行った。その結果が図4である。シミュレーションに関する条件はケース1と同様である。最終フェーズのノミナル値は両側1シグマのエラーバー内には入らなかった。この例では両側2シグマのエラーバーを考えるとノミナル値はエラーバー内に入る。

4.2.3 ケース3

ケース1と2のシミュレーション結果から、推定・予測値のエラーバーとノミナル値の関係はデータにより異なる。そこで、多数の擬似データに対してエラーバー内に入るノミナル値の割合を調べた。擬似データのセットを 10^4 個用意した(生成方法はケース1、2と同様)。各データセットは、開発フェーズ $n=3$ までEDRの測定を行い、 $n=4$ 以降は測定を行わない条件とした(ケース1、2と同様)。これら 10^4 個のデータセットについて、最終フェーズ $n=6$ におけるEDRの推定・予測値のエラーバー内にノミナル値が入ったデータセットの個数を調べた。その結果を次に示す。

- (a) 両側1シグマ内 6,856個 (68.6%)
- (b) 両側2シグマ内 9,542個 (95.4%)
- (c) 両側3シグマ内 9,978個 (99.8%)

表2 擬似 EDR データ (ケース2)

開発フェーズ	1	2	3	4	5	6
ノミナル値	6.8	8.8	25.0	8.5	4.5	2.6
プロジェクト1	7.72	9.61	24.10	8.80	3.96	2.18
プロジェクト2	7.60	9.89	24.02	8.04	3.96	3.97
プロジェクト3	7.85	10.02	25.59	8.01	4.37	3.35
プロジェクト4	6.38	9.73	25.33	9.43	5.65	2.86
プロジェクト5	6.13	7.78	25.04	8.52	3.77	2.60
計画値	7.14	9.41	24.82	8.56	4.34	2.99
測定値	5.36	8.64	23.98	-	-	-

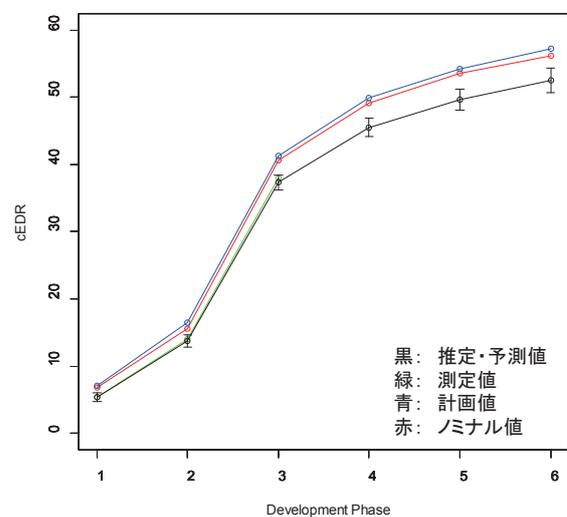


図4 cEDRの変化 (ケース2)

カルマンフィルタ理論の誤差は正規分布に従うノイズを前提としている。したがって、EDRの推定・予測値 \hat{x}_n は正規分布に従う確率変数の実現値とみなすことができる。確率変数が正規分布に従うとき、これらの一次結合で定義される新たな確率変数も正規分布に従う⁽⁹⁾(再生性)。その結果、式(3)で算出されるcEDRも正規分布に従う確率変数の実現値になる。

理論上、正規分布に従うデータに対しては、平均値の両側1シグマ内には68.3%、2シグマ内には95.5%、3シグマ内には99.7%のデータが入る⁽⁹⁾。シミュレーション結果を、ノミナル値を中心とした推定・予測値の分布とみなすと、理論値との比較ができる；ノミナル値のまわりに推定・予測値が正規分布に従い広がっている。上の結果はこれを満たしているように見えるが、注意が必要である。これについては4.3節で議論する。

4.3 考察

4.2節のシミュレーション結果の考察をまとめる。

4.3.1 ケース1と2の比較

ケース1では、最終開発フェーズ($n=6$)において、EDRのノミナル値がエラーバー内(両側1シグマ)に入った。しかし、ケース2ではエラーバー内に入らなかった。ケース1とケース2の違いは以下のように考えられる。カルマンフィルタの初期値($n=1$ の計画値)とそこでの測定値の差分を見ると、ケース1のほうがケース2より小さい。そのため、ケース1の方がカルマンフィルタによる予測が良くなったと考えられる。

ケース2におけるノミナル値と推定・予測値の差異の改善方法として、例えば状態方程式(1)にバイアス項 b を追加して

$$x_n = a_{n-1}x_{n-1} + b_{n-1} + u_{n-1}$$

とする方式などが考えられる。ただし、このバイアス項はケース1の結果を含めて決定しなければならず、状態方程式のモデル化が複雑になる。現実のソフトウェア開発では、このバイアス項は計画値の妥当性に関連していると考えられる。適切な状態方程式の表現については今後の検討が必要である。

4.3.2 ケース3

誤差共分散 $P_n(\pm)$ には、状態方程式(1)のシステムノイズと観測方程式(2)の測定ノイズが反映されている。測定が行われると、その結果に基づいて予測値が修正されるため誤差共分散が $P_n(-)$ から $P_n(+)$ へ減少する。この減少は次フェーズのシステムノイズと測定ノイズに影響を与える。そのため、これらノイズを変更せずに測定条件

だけを変更すると、4.2節で述べた「ノミナル値がエラーバー内に入る割合」が崩れる。一般に、測定の回数を増やすほど、精度が上がりエラーバーは短くなる。これは、ベイズ統計の観点では「事前確率分布から事後確率分布への更新に伴う分散の減少」と解釈できる⁽¹⁰⁾。

例として、測定をする開発フェーズを増やして $n=4$ 以降も測定する場合を考える。測定はシステムに影響する外乱を制御する行為とみなせるので、システムノイズはケース3のシミュレーションより小さめに設定する必要がある。他方、測定ノイズは、一回の測定ごとに加算されるので、ケース3より大きめに設定する必要がある。測定条件が変更されても、状況に対応したシステムノイズと測定ノイズを再設定すれば、「ノミナル値がエラーバー内に入る割合」は保存される。カルマンフィルタ理論のノイズは正規性を前提としているので、このことを意識することは重要である。

今回のシミュレーションでは、システムノイズと測定ノイズは全開発フェーズで一定値を用いた。モデルの精密化のためには、測定の有無を考慮して開発フェーズごとに設定することなどが考えられる。

上述のことは、カルマンフィルタのパラメータ設定とも関連している。カルマンフィルタを処理するためには、次に示す4種類のアプリアリなパラメータを設定する必要がある。

- (a) 初期値に関して
 - (i) 状態量： $\hat{x}_0(-)$
 - (ii) 誤差共分散： $P_0(-)$
- (b) ノイズに関して
 - (iii) システムノイズ共分散： Q_n
 - (iv) 測定ノイズ共分散： R_n

適切でないパラメータを設定するとカルマンフィルタは有効に機能しない。これらを設定するための一般的な手段はない。シミュレーション結果に基づいてパラメータの試行錯誤的なチューニングが必要である。その際の手段として、「ノミナル値がエラーバー内に入る割合」を意識することは有用と考えられる。

5. むすび

EDRを状態量として、これを推定・予測するカルマンフィルタを試作した。この中には、EDRの推移を記述した状態方程式と統計学に基づく誤差の評価(誤差共分散)が考慮されている。その試作ツールを用いて、数値シミュレーションを実施し、ソフトウェア開発時の誤り予測へのカルマンフィルタの利用可能性を示した。今後は、実際のプロジェクトの実績データを使って検証していく予定である。

本稿ではEDRのみからなるスカラーの状態量に関するカルマンフィルタについて述べた。カルマンフィルタの

状態量としては、例えば EDR と cEDR の 2 成分からなるベクトル量を考えることもできる。出発点となる状態量、状態方程式、観測モデルの決め方は一意ではない。有効なカルマンフィルタを設計するために、これらの検討は重要である。

* 1 本モデルは EDR のみからなる 1 成分の状態量を扱っているため、 $\{P_n, Q_n, R_n\}$ は実際には「…共分散」ではなく「…分散」である。一般にはベクトルと行列で記述されている数式が、本モデルではスカラー量の関係式となっている。

参考文献

- (1) 岡野 麻子, 岡田 政之, 土屋 義兼: 定量的プロジェクト管理におけるソフトウェア品質予測モデルの構築と適用, MSS 技報, 26 (2016)
http://www.mss.co.jp/technology/report/pdf/26_06.pdf
- (2) 三觜 武: ソフトウェアの品質評価法—統計的管理へのアプローチ, 181 ~ 191, 日科技連出版社 (1981)
- (3) 西村 敏充: カルマン・フィルタ理論の飛翔体システムへの応用, システムと制御, 22, No.1, 10 ~ 19 (1978)
- (4) 岩田 隆敬: 宇宙分野におけるカルマンフィルタの応用, 計測と制御, 56, No.9, 668 ~ 674 (2017)
- (5) 矢田部 学: 擬似データを用いたカルマンフィルタ

による不具合予測, Fy18 品証生産性向上セミナー Q-2, 社内講座資料 (2018)

- (6) Gelb, A., ed.: Applied Optimal Estimation, 107 ~ 119, MIT Press (1974)
- (7) Lichten, W. (村上 雅章 訳): 計測データと誤差解析の入門, 54 ~ 55, ピアソン・エデュケーション (2004)
- (8) Jones, C. (鶴保 征城, 富野 壽 監訳): ソフトウェア開発の定量化手法 第 2 版, 394 ~ 399, 共立出版 (1998)
- (9) 東京大学教養学部統計学教室 編: 統計学入門, 122 ~ 151, 東京大学出版会 (1991)
- (10) 松原 望: 入門ベイズ統計—意思決定の理論と発展, 17 ~ 19, 東京図書 (2008)

執筆者紹介

矢田部 学

1986 年入社。つくば事業部で宇宙分野の解析や金融工学に従事。2004 年から鎌倉事業部で宇宙・防衛分野のモデリングや統計解析に従事。博士 (理学)。

岡野 麻子

1997 年入社。入社以来、鎌倉事業部で防衛分野に従事。2005 年から品質保証に従事。2012 年から生産技術部門でプロセス改善に従事。