

# アンサンブルカルマンフィルタ, 粒子フィルタ, ガウシアン粒子フィルタについて\*

村田 眞哉<sup>†</sup>・平松 薫<sup>†</sup>

## On Ensemble Kalman Filter, Particle Filter, and Gaussian Particle Filter\*

Masaya MURATA<sup>†</sup> and Kaoru HIRAMATSU<sup>†</sup>

In this paper, we clarify theoretical aspects of the representative non-Gaussian filters: the ensemble Kalman filter (EnKF) and the particle filter (PF). We first show that the EnKF is a realization algorithm of the linear optimal filter for nonlinear problems. We also show that under the Gaussian assumption for the predicted state, the EnKF provides a realization algorithm of the Gaussian filter. We next propose the multiple distribution estimation approach which is a novel framework for designing non-Gaussian filters and show that the PFs are special cases. We then propose a new PF algorithm to address the particle impoverishment problem inherent in the standard PF algorithms. We also show that by applying the proposed algorithm, we can improve the filtering accuracy of the Gaussian particle filter. We finally confirm the performance of each filter using two benchmark simulation models.

### 1. はじめに

本論文では非線形問題に対する最適フィルタリング問題を取り扱う。以下に準備として各用語を定義し、既存のフィルタの性質を説明する。そして本論文の貢献と構成について記す。

#### 1.1 最適フィルタリング問題

確率的に時間変化する未知変数に対する推定量を求める問題の中で、現時刻までの観測量をすべて使用する推定量の計算法をフィルタとよぶ。本論文では推定量、観測量を確率変数として考え、それらの実現値を推定値、観測値とよぶ。そして以下の二乗誤差を最小化するフィルタを、最適フィルタとよぶ。

$$E[(X_t - g(\mathcal{Y}_{1:t}))(X_t - g(\mathcal{Y}_{1:t}))^T] \quad (1)$$

ここで、 $X_t$  が時刻  $t$  の未知変数であり、状態とよばれる。 $\mathcal{Y}_{1:t}$  は時刻 1 から  $t$  までの観測量の集合（系列） $\{Y_1, Y_2, \dots, Y_t\}$  であり、 $g(\mathcal{Y}_{1:t})$  が状態の推定量（フィルタ）である。 $X_t$ ,  $Y_t$  は任意の次元の確率ベクトルとする。よく知られているように、(1) 式の二乗誤差を最小化する  $g(\mathcal{Y}_{1:t})$  は以下の条件付き期待値である。

$$E[X_t | \mathcal{Y}_{1:t}] = \int X_t p(X_t | \mathcal{Y}_{1:t}) dX_t$$

したがって、 $p(X_t | \mathcal{Y}_{1:t})$  の計算法の設計が最適フィルタリング問題の主題になる。

$p(X_t | \mathcal{Y}_{1:t})$  は（観測量の集合が与えられた後の）状態の事後分布とよばれ、これをフィルタで推定したものを状態のフィルタ分布とよぶ（以降、単にフィルタ分布とよぶ）。 $E[X_t | \mathcal{Y}_{1:t}]$  は事後分布の 1 次モーメントであり、未知状態に対する不偏推定量である。つまり以下の数式が成立する。

$$E[X_t - E[X_t | \mathcal{Y}_{1:t}]] = 0$$

また、 $E[(X_t - E[X_t | \mathcal{Y}_{1:t}])(X_t - E[X_t | \mathcal{Y}_{1:t}])^T]$  は推定誤差共分散行列を与え、事後分布の 2 次モーメントになる。したがって状態の事後分布を正確に推定したフィルタ分布をもとめることができれば、その 1 次モーメントを計算することで最適な状態推定量を得ることができる。

\* 原稿受付 2016 年 4 月 4 日

<sup>†</sup> NTT コミュニケーション科学基礎研究所

NTT Communication Science Laboratories;

3-1, Morinosato Wakamiya, Atsugi-shi, Kanagawa 243-0198, JAPAN

**Key Words:** nonlinear filter, Gaussian filter, non-Gaussian filter, linear optimal filter, ensemble Kalman filter, particle filter, Gaussian particle filter, sequential importance sampling, data assimilation.

## 1.2 モデルに対する仮定

状態の事後分布は以下であるとする.

$$p(X_t|\mathcal{Y}_{1:t}) = \frac{p(Y_t|X_t)p(X_t|\mathcal{Y}_{1:t-1})}{p(Y_t|\mathcal{Y}_{1:t-1})} \quad (2)$$

ここで以下を仮定した.

$$p(Y_t|X_t, \mathcal{Y}_{1:t-1}) = p(Y_t|X_t)$$

つまり観測量は同時刻の状態のみに依存すると仮定する.  $p(Y_t|X_t)$  は状態と観測量の関係を与えるもので、観測モデルとよばれる. 観測の際にのるノイズ (モデリング誤差も含む) を観測ノイズ  $\Omega_t$  とよび、 $p(Y_t|X_t)$  は  $X_t$  の実現値と  $\Omega_t$  の確率密度関数 (probability density function, PDF) の汎関数とする.  $p(X_t|\mathcal{Y}_{1:t-1})$  は状態の事前分布とよばれ、以下であるとする.

$$\begin{aligned} p(X_t|\mathcal{Y}_{1:t-1}) &= \int p(X_t, \mathcal{X}_{1:t-1}|\mathcal{Y}_{1:t-1})d\mathcal{X}_{1:t-1} \\ &= \int p(X_t|X_{t-1})p(X_{t-1}|\mathcal{Y}_{1:t-1})dX_{t-1} \end{aligned} \quad (3)$$

ここで  $\mathcal{X}_{1:t-1}$  は時刻 1 から  $t-1$  までの状態の集合 (系列) であり、以下を仮定した.

$$p(X_t|\mathcal{X}_{1:t-1}, \mathcal{Y}_{1:t-1}) = p(X_t|X_{t-1})$$

つまり時刻  $t$  における状態は 1 期前の状態のみに依存すると仮定する.  $p(X_t|X_{t-1})$  は時刻  $t-1$  の状態と現時刻  $t$  の状態の関係を与えるもので、システムモデルとよばれる. システムモデルのモデリング誤差をシステムノイズ  $\Psi_t$  とよび、 $p(X_t|X_{t-1})$  は  $X_{t-1}$  の実現値と  $\Psi_t$  の PDF の汎関数とする. したがって  $p(Y_t|\mathcal{Y}_{1:t-1})$  は以下になる.

$$p(Y_t|\mathcal{Y}_{1:t-1}) = \int p(Y_t|X_t)p(X_t|\mathcal{Y}_{1:t-1})dX_t \quad (4)$$

以上のことから、(2) 式のフィルタ分布を計算するためには (3) 式の状態の事前分布も推定する必要がある、この推定分布を状態の予測分布とよぶ (以降、単に予測分布とよぶ). また (4) 式の  $p(Y_t|\mathcal{Y}_{1:t-1})$  を、観測量の予測分布とよぶ. 上記二つの観測モデルおよびシステムモデルに対する仮定に加え、観測ノイズ、システムノイズ、および初期状態  $X_0$  は互いに独立であるとする. そしてこれらの確率変数の PDF は既知とし、各ノイズは白色とする.

## 1.3 代表的な非線形フィルタ

これまで提案されてきた代表的な非線形フィルタは

- (1) ガウシアンフィルタ (Gaussian filter, GF)
- (2) 線形最適フィルタ (linear optimal filter, LOF)
- (3) 非ガウシアンフィルタ (non-Gaussian filter, NGF)

である. 以下、各フィルタの性質を説明する.

### 1.3.1 GF

GF では予測分布と観測量の予測分布に正規性が仮定され、その結果、フィルタ分布を正規分布としてもとめる. そして GF のアルゴリズムはこのフィルタ分布の 2 次モーメントまでを与える [1]. 非線形問題に対しては通常、状態の事前分布および事後分布は正規分布に従わないため、これらの仮定は成立しない. そのため、予測分布の 1 次モーメントは不偏性を満たさなくなり (同時刻の未知状態の期待値と一致しなくなり)、GF の状態推定量も不偏にはならない. 一方、線形ガウス型の問題で、初期状態のガウス分布が既知の場合はこれらの正規性の仮定が成立するため、GF はこの問題に対して最適な状態推定量を与える (このアルゴリズムはカルマンフィルタとよばれる).

このように GF は準最適かつ偏った状態推定量を一般的には与えるが、そのアルゴリズムのシンプルさとフィルタリング速度の高速性から、多くの非線形問題に対して適用されてきた. GF の代表的な実現アルゴリズムは拡張カルマンフィルタ (extended Kalman filter, EKF) と unscented Kalman filter (UKF) であるが、それぞれのアプローチは異なる. EKF では非線形モデルを、観測値が与えられた後の状態推定値の周りで線形化することで正規性を保証するのにに対し、UKF では各分布に直接正規性が課される. したがって、状態推定誤差共分散行列やモデルの非線形性が大きい場合は後者のアプローチの方が有効である [2,3].

UKF より推定精度や数値的安定性が優れる GF の実現アルゴリズムとして、Gauss-Hermite Kalman filter (GHKF)[4,5], cubature Kalman filter (CKF)[6], transformed UKF (TUKF)[7] が提案されている. また、フィルタ分布にだけ正規性が仮定される (フィルタ分布の 3 次以上のモーメントを無視する) ガウシアン粒子フィルタ (Gaussian particle filter, GPF) とよばれる手法も提案されている [1]. GPF は GF より仮定が少ないが、1.3.3 節で説明する粒子の重みの偏りによる推定精度の劣化が問題になることがある [8,9]. また、この問題を回避するためには多数の粒子を用いる必要があるため、通常、GPF のフィルタリング速度は GF より遅い.

### 1.3.2 LOF

LOF は観測量に対して線形なフィルタの中で、二乗誤差の意味で最適な状態推定量  $L(\mathcal{Y}_{1:t})$  を与える. 通常、予測分布の 1 次モーメントに不偏性が仮定され、結果として以下のように線形フィルタも不偏になる.

$$E[X_t - L(\mathcal{Y}_{1:t})] = 0$$

また

$$0 = E[X_t] - E[L(\mathcal{Y}_{1:t})] = E[E[X_t|\mathcal{Y}_{1:t}] - L(\mathcal{Y}_{1:t})]$$

であるから、この不偏性の仮定のもとでは、 $L(\mathcal{Y}_{1:t})$  は状態の事後分布の 1 次モーメント (最適フィルタ) の不偏推定量であることもわかる。したがって、その推定誤差共分散行列は最適フィルタのものより大きくなるが、これらの量を使用して、状態の事後分布の 2 次モーメントまでを近似することができる。ここで、3 次以上のモーメントに対しては何も近似されていないことに注意が必要である。つまり、状態の事後分布に強い非ガウス性が生じる場合、LOF のフィルタ分布の推定精度は良くない。以上をまとめると、LOF は予測分布に対する不偏性と状態の事後分布に正規性が仮定できる場合、良好な状態推定量を出力し、GF は状態の事前分布と事後分布に正規性が仮定できる場合、良好な状態推定量を出力することになる。

本論文ではアンサンブルカルマンフィルタ (ensemble Kalman filter, EnKF)[1] が非線形問題に対する LOF の実現アルゴリズムであることを示す。また、予測分布に正規性を仮定する場合、EnKF は GF の実現アルゴリズムを与えることを示す。またこの関係から、EnKF を GF の一般的な (非ガウス性をもつ) 予測分布への拡張とみなすことが可能になる。EnKF は初期状態の分布が正規分布である線形ガウス型の状態空間モデルにおいては漸近的にカルマンフィルタに一致する。その際、EnKF では予測分布およびフィルタ分布の共分散行列を保持する必要がないため、高次元のフィルタリング問題を扱うデータ同化の分野で標準的なアプローチの一つになっている。

また 2.3 節では、EnKF の状態推定量とその推定誤差共分散行列からなる正規分布をフィルタ分布とみなすことで、GPF を近似するアルゴリズムが設計可能なことを示す。このアルゴリズムには、次節で述べる粒子の重みの偏りに起因する問題が存在しないため、特に高次元の推定問題に対して有効であることが期待される。

### 1.3.3 NGF

NGF には GF や LOF のような仮定は存在しない。NGF の代表的な実現アルゴリズムは粒子フィルタ (particle filter, PF) である。PF では状態の事前分布と事後分布が粒子とよばれる各分布の実現値の集合で近似される (ディラックのデルタ関数の加重平均で近似される)。状態の事後分布が正確に表現されれば、その期待値をとることで最適フィルタを実現することができる。しかしながら、PF では時間が進むにつれ各粒子の重みのバランスが崩れ、ごく少数の粒子に重みが偏る現象が生じる。重みが偏ると、ごく少数の粒子で各分布を表現することになるため、各分布の近似精度や状態の推定精度が大きく劣化する。しかしながら、全粒子の重みのバランスを保ちながら PF を実行し続けることは困難であり [8,9]、これが PF の本質的な問題になっている。

この問題は連続な分布を有限個の粒子で近似しているために起こり、PF のアイデアそのものであるため避け

ることはできない。その結果、重みの再配分を行う粒子のリサンプリングとよばれる処理が適宜追加されることになる。しかしながらこの処理では通常、重みが大きい粒子を複数回選択することにより重みのバランスを回復するため、同一の粒子が複数回選択されることになり、粒子の多様性が失われてしまう。この問題は particle impoverishment problem (PIP) とよばれ、重みの偏りと同様に、PF の推定精度の劣化に大きな影響を与えている。これらの重みの偏りや PIP により、PF で最適フィルタを実現することは困難である [5]。

通常、粒子数を大きくとったり、提案分布とよばれる状態の事後分布を粗く近似した分布から粒子をサンプリングする、逐次重点サンプリング法 (sequential importance sampling, SIS)[9] とよばれる手法を適用することで重みのバランスを保つが、遅かれ早かれ時間の経過により重みのバランスは崩れるため、粒子のリサンプリングが必要になり、本質的な問題の解決には至らない。また、状態の次元が高次元になると有限個の粒子による連続分布の近似精度が下がり、重みの偏りの問題がより深刻になり、結果として最適フィルタからの逸脱が大きくなる。つまり現状、高次元の状態推定問題への PF の適用は困難である [8]。

1.3.1 節と 1.3.2 節で述べた GPF は、PF のフィルタ分布の 3 次以上のモーメントを無視してフィルタ分布に正規性を課すアルゴリズムであるため、同様な重みの偏りの問題が存在する。この問題が生じると、フィルタ分布の 2 次モーメントまでをごく少数の粒子に基づき推定することになり、とくに共分散行列の推定精度が劣化してしまう。本論文では PF のアルゴリズムを含むより一般的な NGF の設計法を提案し、PIP に対する新しいアルゴリズムを提案する。またこのアルゴリズムの適用により、上記の GPF の問題に対処し、GPF のフィルタリング精度を改善するアルゴリズムを提案する。

以下に本節で説明した各フィルタの性質についてまとめる。

- GF は予測分布および観測量の予測分布に正規性を仮定し、この仮定のもとで最適な状態推定量を算出。
- LOF は予測分布の 1 次モーメントに不偏性を仮定し、線形フィルタの中で最適な状態推定量を算出。
- NGF は予測分布に対して何も仮定しないが、フィルタ分布の計算において重みの偏りや PIP が発生。

### 1.4 本論文の貢献と構成

上述したようにさまざまな非線形フィルタがこれまで提案されてきたが、これらのフィルタは厳密に状態の事前分布および事後分布を計算できるわけではない。つまり現状、すべて準最適なフィルタ (最適フィルタの近似) であるため、実際の観測値を与えた際の状態推定値の信頼性の議論には注意を要する。GF と LOF は NGF(PF) と比べて高速であるが、上述したような強い仮定に基づ

いている。一方、PF には重みの偏りの問題やそれに伴う PIP があり、とくに高次元の問題に対して非常に難があるため、このような問題に対しては GF や LOF の方が良い結果を与えることが期待される。

重みの偏りや PIP への対処法として merging particle filter[10,11] や Monte Carlo filter particle filter[12] が提案されているが、これらはより多くの粒子数を考慮することで PF の推定精度を保つアルゴリズムであるため、処理コストが増大してしまう。また、merging particle filter には設定パラメータがあるため、アルゴリズムが複雑化する。Feedback particle filter[13] とよばれる、粒子の時間更新に入力項を追加するアルゴリズムも提案されている。入力項は状態の事後分布とフィルタ分布の Kullback-Leibler divergence、つまり各 PDF 間の距離を最小化するように設計されるため、この入力項の追加により各粒子の重みの偏りが低減することが期待される。これは 1.3.3 節で述べた SIS と同様な効果であり、新たな PF の理論として興味深い。しかしながら入力項の計算コストが大きく、離散型のアルゴリズムも発展途上であるため、さらなる研究が必要である。一方、本論文の PIP に対する提案法は SIS に基づくものであり、計算コストの観点からも実用的であるといえる。

本論文の貢献と構成は以下である。まず、EnKF が非線形問題に対する LOF の実現アルゴリズムであることを 2 章で示す。そして予測分布に正規性を仮定すると、EnKF は GF の実現アルゴリズムを与えることを示す。また、EnKF のフィルタ分布に正規性を仮定することで GPF を近似するアルゴリズムを設計する。3 章で複数分布推定法 (multiple distribution estimation, MDE) に基づく NGF の設計法を提案し、PF が MDE から導出されることを示す。そして、PIP に対する新たな PF のアルゴリズムを 4 章で提案する。また、この提案アルゴリズムを GPF に適用することで、GPF のフィルタリング精度を改善する。5 章で各フィルタの性能を二つのベンチマーク問題を使用した数値シミュレーションにより確認する。そして 6 章で本論文をまとめる。

## 2. LOF の実現アルゴリズム

まず非線形問題に対する LOF のアルゴリズムを説明する。そして粒子数 (アンサンブル数) が増えるにつれ、EnKF が漸的に LOF に一致することを示す。

### 2.1 LOF

LOF は以下である。

$$\begin{aligned} Y_{t|t-1} &= E[Y_t | \mathcal{Y}_{1:t-1}] \\ X_{t|t-1} &= E[X_t | \mathcal{Y}_{1:t-1}] \\ X_{t|t} &= X_{t|t-1} + L_t(Y_t - Y_{t|t-1}) \\ L_t &= U_t(V_t)^{-1} \\ U_t &= E[(X_t - X_{t|t-1})(Y_t - Y_{t|t-1})^T | \mathcal{Y}_{1:t-1}] \\ V_t &= E[(Y_t - Y_{t|t-1})(Y_t - Y_{t|t-1})^T | \mathcal{Y}_{1:t-1}] \end{aligned} \quad (5)$$

$$\begin{aligned} P_{t|t-1} &= E[(X_t - X_{t|t-1})(X_t - X_{t|t-1})^T | \mathcal{Y}_{1:t-1}] \\ P_{t|t} &= P_{t|t-1} - L_t U_t^T \end{aligned} \quad (6)$$

ここで

$$E[X_{t|t-1}] = E[X_t]$$

つまり事前分布の 1 次モーメントが未知状態の期待値に等しいという不偏性が仮定され、(5) 式の状態推定量も以下のように不偏になる。

$$E[X_t - X_{t|t}] = 0$$

すると (6) 式により LOF の推定誤差共分散行列が計算される。このアルゴリズムからわかるように、LOF の実現には各条件付き期待値の計算が必要になる。

### 2.2 EnKF

本章では EnKF が LOF の実現アルゴリズムを漸的に与えることを示す。つまり、EnKF の粒子数を大きくすると、EnKF のアルゴリズムが LOF に漸的に一致することを示す。状態の事前分布  $p(X_t | \mathcal{Y}_{1:t-1})$  が  $L$  個の粒子を使用して

$$p(X_t | \mathcal{Y}_{1:t-1}) \approx \frac{1}{L} \sum_{l=1}^L \delta(X_t - X_{t|t-1}^{(l)})$$

で近似されている場合の EnKF は以下である。

$$Y_t^{(l)} = h(X_{t|t-1}^{(l)}) + \Omega_t^{(l)} \quad (7)$$

$$Y_t^L = \frac{1}{L} \sum_{l=1}^L Y_t^{(l)} \quad (8)$$

$$X_{t|t-1}^L = \frac{1}{L} \sum_{l=1}^L X_{t|t-1}^{(l)} \quad (9)$$

$$X_{t|t}^{(l)} = X_{t|t-1}^{(l)} + K_t^L(Y_t - Y_t^{(l)}) \quad (10)$$

$$K_t^L = U_t^L(V_t^L)^{-1} \quad (11)$$

$$U_t^L = \frac{1}{L-1} \sum_{l=1}^L (X_{t|t-1}^{(l)} - X_{t|t-1}^L)(Y_t^{(l)} - Y_t^L)^T \quad (12)$$

$$V_t^L = \frac{1}{L-1} \sum_{l=1}^L (Y_t^{(l)} - Y_t^L)(Y_t^{(l)} - Y_t^L)^T \quad (13)$$

ここで  $h(\cdot)$  は状態の非線形観測関数を表す。 $X_{t|t}^{(l)}$  が EnKF により観測更新 (フィルタリング) された  $l$  番目の粒子である。以下に、粒子数  $L$  を大きくすることで、 $X_{t|t}^L$  が LOF の  $X_{t|t}$  ((5) 式) に漸的に一致することを示す。

$$\begin{aligned} X_{t|t}^L &= \frac{1}{L} \sum_{l=1}^L X_{t|t}^{(l)} \\ &= X_{t|t-1}^L + K_t^L(Y_t - Y_t^L) \end{aligned} \quad (14)$$

(14) 式は  $L$  が大きくなると (5) 式に一致することがわかる。

つぎに、

$$P_{t|t}^L = \frac{1}{L-1} \sum_{l=1}^L (X_{t|t}^{(l)} - X_{t|t}^L)(X_{t|t}^{(l)} - X_{t|t}^L)^T$$

が LOF の  $P_{t|t}$  (6) 式) に漸近的に一致することを示す。以下の記号を定義すると

$$\begin{aligned} \delta Y_t^{(l)} &\equiv Y_t^{(l)} - Y_t^L \\ \delta X_{t|t}^{(l)} &\equiv X_{t|t}^{(l)} - X_{t|t}^L = \delta X_{t|t-1}^{(l)} - K_t^L \delta Y_t^{(l)} \end{aligned}$$

$P_{t|t}^L$  は以下になる。

$$\begin{aligned} P_{t|t}^L &= \frac{1}{L-1} \sum_{l=1}^L \left( \delta X_{t|t-1}^{(l)} (\delta X_{t|t-1}^{(l)})^T \right. \\ &\quad \left. - \delta X_{t|t-1}^{(l)} (\delta Y_t^{(l)})^T (K_t^L)^T - K_t^L \delta Y_t^{(l)} (\delta X_{t|t-1}^{(l)})^T \right. \\ &\quad \left. + K_t^L \delta Y_t^{(l)} (\delta Y_t^{(l)})^T (K_t^L)^T \right) \\ &= P_{t|t-1}^L - U_t^L (K_t^L)^T - K_t^L (U_t^L)^T \\ &\quad + K_t^L V_t^L (K_t^L)^T \\ &= P_{t|t-1}^L - K_t^L (U_t^L)^T \end{aligned} \quad (15)$$

ここで、 $P_{t|t-1}^L, K_t^L, U_t^L$  は  $L$  が十分大きくなると  $P_{t|t-1}, L_t, U_t$  に漸近するため、(15) 式は (6) 式に一致することがわかる。つまり EnKF で計算される粒子  $X_{t|t}^{(l)}, (l=1, 2, \dots, L)$  は、LOF の状態推定量とその推定誤差共分散行列の情報を保持している。したがって、1.3.2 節で述べたようにこれらの量は状態の事後分布の 2 次モーメントまでを近似していると考えられるため、 $X_{t|t}^{(l)}, (l=1, 2, \dots, L)$  を状態の事後分布の  $L$  個の実現値とみなすことができる。つまり

$$p(X_t | \mathcal{Y}_{1:t}) \approx \frac{1}{L} \sum_{l=1}^L \delta(X_t - X_{t|t}^{(l)}) \quad (16)$$

とみなすことができ、フィルタ分布が算出される。

(16) 式を使用すると、次の時刻  $t+1$  の予測分布は以下で計算される。

$$\begin{aligned} p(X_{t+1} | \mathcal{Y}_{1:t}) &= \int p(X_{t+1} | X_t) p(X_t | \mathcal{Y}_{1:t}) dX_t \\ &\approx \frac{1}{L} \sum_{l=1}^L p(X_{t+1} | X_{t|t}^{(l)}) \\ &\approx \frac{1}{L} \sum_{l=1}^L \delta(X_{t+1} - X_{t+1|t}^{(l)}) \end{aligned} \quad (17)$$

ここで  $X_{t+1|t}^{(l)}$  は  $p(X_{t+1} | X_{t|t}^{(l)})$  からサンプリングされた粒子である。フィルタ分布が正しいと仮定すると、(17) 式も  $L$  の増加とともに漸近的に状態の事前分布に近づくため、EnKF は非線形問題に対する LOF の実現アルゴリズムになる。

## 2.3 GF の実現アルゴリズムの設計

1.3.1 節で述べたように、GF は予測分布と観測量の予測分布に正規性を仮定した際の、フィルタ分布の 2 次モーメントまでを与える。この場合、フィルタ分布は正規分布になる。実は GF のアルゴリズムは 2.1 節の LOF と同じである。したがって 2.2 節の結果をそのまま使用すると、予測分布に正規性を仮定した際、つまり、 $L$  個の粒子がその正規分布からサンプリングされている場合、EnKF の  $X_{t|t}^L$  および  $P_{t|t}^L$  は GF が与える 2 次モーメントまでに漸近的に一致する。そして、 $X_{t|t}^L, P_{t|t}^L$  を期待値および共分散行列とする正規分布  $N(X_t | X_{t|t}^L, P_{t|t}^L)$  から、 $L$  個の粒子をリサンプリングすることで、GF のフィルタ分布を EnKF により表現することが可能になる。そして (17) 式に従いこれらの粒子を時間更新し、以下の期待値および共分散行列に従う正規分布から  $L$  個の粒子をリサンプリングすることで、正規性が仮定された予測分布を表現することができる。

$$X_{t+1|t}^L = \frac{1}{L} \sum_{l=1}^L X_{t+1|t}^{(l)}$$

$$P_{t+1|t}^L = \frac{1}{L-1} \sum_{l=1}^L (X_{t+1|t}^{(l)} - X_{t+1|t}^L)(X_{t+1|t}^{(l)} - X_{t+1|t}^L)^T$$

このフィルタ分布と予測分布の繰り返し計算により GF のアルゴリズムを EnKF で実現することができる。1.3.1 節に示したように、GF の実現アルゴリズムとしてすでに UKF, GHKF, CKF, TUKF が提案されているが、これらはすべて GF の近似であるのに対し、本節の EnKF は粒子数を増やすと漸近的に GF に一致する。この GF の実現アルゴリズムを Gaussian ensemble Kalman filter (GEnKF) とよぶことにする。GEnKF は予測分布とフィルタ分布に正規性を仮定した EnKF である。

GEnKF とは異なり、フィルタ分布にのみ正規性を仮定するアルゴリズム (GEnKF2 とよぶ) は GPF と類似する。GPF は GF の非正規性をもつ予測分布への拡張とみなすことができるため、GEnKF2 も GF の拡張と考えることができる。これらの違いは、GPF は最適フィルタの推定量とその推定誤差共分散行列からなる正規分布から粒子をサンプリングするのに対し、GEnKF2 は LOF (準最適フィルタ) の推定量とその推定誤差共分散行列から成る正規分布から粒子をサンプリングするところにある。つまり、GEnKF2 は GPF の近似であるといえる。しかしながら GPF には 1.3.3 節で説明したような粒子の重みの偏りの問題が存在し、その結果、少数の粒子によるフィルタ分布の共分散行列の推定が発生し、フィルタリング精度が劣化する恐れがある。したがって GPF と GEnKF2 はどちらが良いか一概にはいえない。とくに重みの偏りがより顕著になる高次元の問題に対しては、GEnKF2 の方が優れる可能性が高い。これらのフィルタの性能差は 5 章のシミュレーションにより数値

的に確認する。

以下に本節で説明した各フィルタの性質についてまとめる。また、 $L$  個のアンサンブルを使用した GEnKF と GEnKF2 のアルゴリズムを示す。

- EnKF は LOF の実現アルゴリズムであり、予測分布の 1 次モーメントに不偏性を仮定し、この仮定のもとで不偏な状態推定量を算出。粒子数の増加とともに LOF に漸近する。
- GEnKF は GF の実現アルゴリズムであり、予測分布および観測量の予測分布に正規性を仮定し、この仮定のもとで最適な状態推定量を算出。粒子数の増加とともに GF に漸近する。
- GEnKF2 は GPF の近似アルゴリズムであり、予測分布の 1 次モーメントの不偏性とフィルタ分布の正規性を仮定。EnKF のフィルタ分布の 3 次以上のモーメントを無視するアルゴリズムに対応する。

#### — GEnKF のアルゴリズム —

予測分布の計算（時間更新）

$$p(X_t | \mathcal{Y}_{1:t-1}) = \frac{1}{L} \sum_{l=1}^L \delta(X_t - X_{t|t-1}^{(l)})$$

$$X_{t|t-1}^L = \frac{1}{L} \sum_{l=1}^L X_{t|t-1}^{(l)}$$

$$P_{t|t-1}^L = \frac{1}{L-1} \sum_{l=1}^L \left( (X_{t|t-1}^{(l)} - X_{t|t-1}^L)(X_{t|t-1}^{(l)} - X_{t|t-1}^L)^T \right)$$

$L$  個の粒子をリサンプリング

$$X_{t|t-1}^{(l)} \sim N(X_t | X_{t|t-1}^L, P_{t|t-1}^L), (l=1, 2, \dots, L)$$

フィルタ分布の計算（観測更新）

(7) ~ (13) 式を計算

$L$  個の粒子をリサンプリング

$$X_{t|t}^{(l)} \sim N(X_t | X_{t|t}^L, P_{t|t}^L), (l=1, 2, \dots, L)$$

$$p(X_t | \mathcal{Y}_{1:t}) \approx \frac{1}{L} \sum_{l=1}^L \delta(X_t - X_{t|t}^{(l)})$$

#### — GEnKF2 のアルゴリズム —

予測分布の計算（時間更新）

$$p(X_t | \mathcal{Y}_{1:t-1}) = \frac{1}{L} \sum_{l=1}^L \delta(X_t - X_{t|t-1}^{(l)})$$

フィルタ分布の計算（観測更新）

(7) ~ (13) 式を計算

$L$  個の粒子をリサンプリング

$$X_{t|t}^{(l)} \sim N(X_t | X_{t|t}^L, P_{t|t}^L), (l=1, 2, \dots, L)$$

$$p(X_t | \mathcal{Y}_{1:t}) \approx \frac{1}{L} \sum_{l=1}^L \delta(X_t - X_{t|t}^{(l)})$$

### 3. MDE による NGF の設計法

本章では MDE に基づく NGF の設計法を提案する。まず MDE の定式化を示し、PF が導出されることを示す。

#### 3.1 MDE の定式化

時刻  $t$  の状態の事前分布と事後分布は以下で表現できる。

$$p(X_t | \mathcal{Y}_{1:t-1}) = \int p(X_t, \alpha_t | \mathcal{Y}_{1:t-1}) d\alpha_t$$

$$p(X_t | \mathcal{Y}_{1:t}) = \int p(X_t, \alpha_t | \mathcal{Y}_{1:t}) d\alpha_t$$

ここで  $\alpha_t$  は確率変数のパラメータであり、 $X_t$  と独立ではないとする。つまり、上式は以下になる。

$$p(X_t | \mathcal{Y}_{1:t-1}) = \int p(X_t | \alpha_t, \mathcal{Y}_{1:t-1}) p(\alpha_t | \mathcal{Y}_{1:t-1}) d\alpha_t \quad (18)$$

$$p(X_t | \mathcal{Y}_{1:t}) = \int p(X_t | \alpha_t, \mathcal{Y}_{1:t}) p(\alpha_t | \mathcal{Y}_{1:t}) d\alpha_t$$

$$= \frac{1}{\int p(Y_t | \alpha_t, \mathcal{Y}_{1:t-1}) p(\alpha_t | \mathcal{Y}_{1:t-1}) d\alpha_t} \times \int p(Y_t | \alpha_t, \mathcal{Y}_{1:t-1}) p(\alpha_t | \mathcal{Y}_{1:t-1}) \times p(X_t | \alpha_t, \mathcal{Y}_{1:t}) d\alpha_t \quad (19)$$

ここで、 $p(\alpha_t | \mathcal{Y}_{1:t-1})$  も  $M$  個の粒子で

$$p(\alpha_t | \mathcal{Y}_{1:t-1}) \approx \sum_{j=1}^M s_t^{(j)} \delta(\alpha_t - \alpha_t^{(j)}),$$

where  $\sum_{j=1}^M s_t^{(j)} = 1$

のように近似されているとすると、(18), (19) 式は以下になる。

$$p(X_t | \mathcal{Y}_{1:t-1}) \approx \sum_{j=1}^M s_t^{(j)} p(X_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t-1}) \quad (20)$$

$$p(X_t | \mathcal{Y}_{1:t}) \approx \frac{1}{\sum_{k=1}^M s_t^{(k)} p(Y_t | \alpha_t^{(k)}, \mathcal{Y}_{1:t-1})} \times \sum_{j=1}^M s_t^{(j)} p(Y_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t-1}) p(X_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t})$$

$$= \sum_{j=1}^M u_t^{(j)} p(X_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t}), \quad (21)$$

$$\text{where } u_t^{(j)} = \frac{s_t^{(j)} p(Y_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t-1})}{\sum_{k=1}^M s_t^{(k)} p(Y_t | \alpha_t^{(k)}, \mathcal{Y}_{1:t-1})}$$

(20), (21) 式は、状態の事前分布および事後分布が  $M$  個の条件付き事前分布  $p(X_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t-1}), (j=1, 2, \dots, M)$  および条件付き事後分布  $p(X_t | \alpha_t^{(j)}, \mathcal{Y}_{1:t}), (j=1, 2, \dots, M)$  の加重平均で近似されることを示している。本定式化による各分布の表現を MDE とよぶ。MDE を実際に計算するため、 $\alpha_t^{(j)}, (j=1, 2, \dots, M)$  の設定について述べる。

### 3.2 $\alpha_t$ の設定

時刻  $t-1$  における状態の事後分布が  $N$  個の粒子を使用して、以下で近似されているとする。

$$p(X_{t-1}|Y_{1:t-1}) \approx \sum_{i=1}^N w_{t-1}^{(i)} \delta(X_{t-1} - X_{t-1}^{(i)}), \quad (22)$$

where  $\sum_{i=1}^N w_{t-1}^{(i)} = 1$

1.2 節のモデルに対する仮定のもとでは、 $\alpha_t$  の候補は  $\Psi_t$  か  $X_{t-1}$ 、もしくはその両方になる。以下、それぞれの場合について MDE を計算する。

#### 3.2.1 $\alpha_t = \Psi_t$ の場合

$\alpha_t$  として  $\Psi_t$  を採用すると、(20) 式の  $p(X_t|\alpha_t^{(j)}, \mathcal{Y}_{1:t-1})$  は (22) 式から以下になる。

$$\begin{aligned} p(X_t|\Psi_t^{(j)}, \mathcal{Y}_{1:t-1}) &= \int p(X_t|X_{t-1}, \Psi_t^{(j)}) \\ &\quad \times p(X_{t-1}|\mathcal{Y}_{1:t-1}) dX_{t-1} \\ &\approx \sum_{i=1}^N w_{t-1}^{(i)} p(X_t|X_{t-1}^{(i)}, \Psi_t^{(j)}) \end{aligned}$$

ここで  $\Psi_t^{(j)}$  は  $j$  番目の  $\Psi_t$  の実現値である ( $\Psi_t^{(j)}, (j=1, 2, \dots, M)$ )。1.2 節で述べたように  $p(X_t|X_{t-1})$  は  $X_{t-1}$  の実現値と  $p(\Psi_t)$  の汎関数であるが、この場合システムノイズも与えられているため、上記のようにディラックのデルタ関数  $\delta(X_t - X_t^{(i,j)})$  になる。すると (20) 式は以下になる。

$$p(X_t|\mathcal{Y}_{1:t-1}) \approx \sum_{j=1}^M s_t^{(j)} \left( \sum_{i=1}^N w_{t-1}^{(i)} \delta(X_t - X_t^{(i,j)}) \right) \quad (23)$$

また、(21) 式の  $p(X_t|\alpha_t^{(j)} = \Psi_t^{(j)}, \mathcal{Y}_{1:t})$  は以下になる。

$$\begin{aligned} p(X_t|\Psi_t^{(j)}, \mathcal{Y}_{1:t}) &= \frac{p(Y_t|X_t)p(X_t|\Psi_t^{(j)}, \mathcal{Y}_{1:t-1})}{p(Y_t|\Psi_t^{(j)}, \mathcal{Y}_{1:t-1})} \\ &\approx \frac{\sum_{i=1}^N w_{t-1}^{(i)} p(Y_t|X_t^{(i,j)}) \delta(X_t - X_t^{(i,j)})}{\sum_{k=1}^N w_{t-1}^{(k)} p(Y_t|X_t^{(k,j)})} \end{aligned}$$

したがって (21) 式は以下になる。

$$\begin{aligned} p(X_t|\mathcal{Y}_{1:t}) &\approx \sum_{j=1}^M \frac{u_t^{(j)}}{\sum_{k=1}^N w_{t-1}^{(k)} p(Y_t|X_t^{(k,j)})} \\ &\quad \times \left( \sum_{i=1}^N w_{t-1}^{(i)} p(Y_t|X_t^{(i,j)}) \delta(X_t - X_t^{(i,j)}) \right) \\ &\approx \sum_{j=1}^M \frac{s_t^{(j)}}{\sum_{k=1}^M s_t^{(k)} \left( \sum_{l=1}^N w_{t-1}^{(l)} p(Y_t|X_t^{(l,k)}) \right)} \\ &\quad \times \left( \sum_{i=1}^N w_{t-1}^{(i)} p(Y_t|X_t^{(i,j)}) \delta(X_t - X_t^{(i,j)}) \right) \quad (24) \end{aligned}$$

(23) 式および (24) 式は各分布が  $M \times N$  個の粒子で近似されることを意味しており、この  $\alpha_t$  の設定では予測分

布およびフィルタ分布の計算コストが増大してしまうことがわかる。

#### 3.2.2 $\alpha_t = \{X_{t-1}, \Psi_t\}$ の場合

$\alpha_t$  として  $\{X_{t-1}, \Psi_t\}$  を採用すると、 $p(X_t|\alpha_t^{(j)}, \mathcal{Y}_{1:t-1})$  は以下になる。

$$\begin{aligned} p(X_t|X_{t-1}^{(j)}, \Psi_t^{(j)}, \mathcal{Y}_{1:t-1}) &= p(X_t|X_{t-1}^{(j)}, \Psi_t^{(j)}) \\ &= \delta(X_t - X_t^{(j,j)}) \end{aligned}$$

ここで以下であるから

$$\begin{aligned} p(\alpha_t|\mathcal{Y}_{1:t-1}) &= p(X_{t-1}, \Psi_t|\mathcal{Y}_{1:t-1}) \\ &= p(X_{t-1}|\mathcal{Y}_{1:t-1}) p(\Psi_t) \\ &\approx \sum_{j=1}^M \left( s_t^{(j)} \delta(X_{t-1} - X_{t-1}^{(j)}) \right. \\ &\quad \left. \times \delta(\Psi_t - \Psi_t^{(j)}) \right) \end{aligned}$$

$M=N$  とすると (20) 式は以下になる。

$$\begin{aligned} p(X_t|\mathcal{Y}_{1:t-1}) &\approx \sum_{j=1}^N s_t^{(j)} \delta(X_t - X_t^{(j,j)}), \quad (25) \\ \text{where } s_t^{(j)} &= w_{t-1}^{(j)} \end{aligned}$$

また、 $p(X_t|X_{t-1}^{(j)}, \Psi_t^{(j)}, \mathcal{Y}_{1:t})$  は以下であるので、

$$p(X_t|X_{t-1}^{(j)}, \Psi_t^{(j)}, \mathcal{Y}_{1:t}) = \delta(X_t - X_t^{(j,j)})$$

(21) 式は以下になる。

$$\begin{aligned} p(X_t|\mathcal{Y}_{1:t}) &\approx \sum_{j=1}^N u_t^{(j)} \delta(X_t - X_t^{(j,j)}), \quad (26) \\ \text{where } u_t^{(j)} &= \frac{s_t^{(j)} p(Y_t|X_t^{(j,j)})}{\sum_{k=1}^N s_t^{(k)} p(Y_t|X_t^{(k,k)})} \end{aligned}$$

(25), (26) 式は bootstrap filter (BF) や Monte Carlo filter (MCF) とよばれる PF のアルゴリズム [14] に相当する。BF と MCF は同一のアルゴリズムである。 $u_t^{(j)}$  が  $j$  番目の粒子の重みを表す。

1.3.3 節で説明した粒子の重みの偏り度合いは以下で数値化される。

$$N_{\text{eff}} = \frac{1}{\sum_{j=1}^N \left( u_t^{(j)} \right)^2}$$

これは有効粒子数ともよばれ、重みが均一にバランスしている場合、つまり

$$u_t^{(j)} = \frac{1}{N}, \quad (j=1, 2, \dots, N)$$

である場合、最大の  $N$  を与える。また、一つの粒子に重みが集中している場合、つまりある粒子の重みが 1 で、その他の粒子の重みが 0 の場合、 $N_{\text{eff}}$  は最小の 1 になる。有効粒子数があるしきい値より小さい場合、各粒子  $X_t^{(j,j)}, (j=1, 2, \dots, N)$  を  $u_t^{(j)}, (j=1, 2, \dots, N)$  に従いサン

プリングする処理（粒子のリサンプリング処理）が行われ、その結果、サンプリングされた粒子の重みがすべて同じになる。粒子のリサンプリング処理を行うことで重みのバランスを回復することができるが、その結果、同一の粒子が複数回選択されてPIPが生じるため、各問題に対する最適なきい値の設定は困難である。そのためフィルタリング精度ではなく、フィルタリング速度の観点からしきい値が設定されることもある（粒子のリサンプリング処理は一般的に処理コストが大きいので、必要な時にだけ実行する）。

(25), (26) 式の導出に際して強い仮定は存在しないが、

$$\alpha_t^{(j)} = \{X_{t-1}^{(j)}, \psi_t^{(j)}\}, \quad (j=1, 2, \dots, N)$$

は状態とシステムノイズの結合（同時）分布からサンプリングする必要があるため、重みの偏りの問題を軽減するためには粒子数  $N$  を十分大きくとる必要がある。したがって、BF(MCF) にはフィルタリング速度の課題があることがわかる。

### 3.2.3 $\alpha_t = X_{t-1}$ の場合

$\alpha_t$  として  $X_{t-1}$  をとると

$$p(X_t | X_{t-1}^{(j)}, \mathcal{Y}_{1:t-1}) = p(X_t | X_{t-1}^{(j)})$$

である。したがって  $M = N$  とすると、(20) 式は以下になる。

$$p(X_t | \mathcal{Y}_{1:t-1}) \approx \sum_{j=1}^N s_t^{(j)} p(X_t | X_{t-1}^{(j)}), \quad (27)$$

where  $s_t^{(j)} = w_{t-1}^{(j)}$

また、以下の等式が成り立つ。

$$p(Y_t | X_{t-1}^{(j)}) = \frac{p(Y_t | X_t) p(X_t | X_{t-1}^{(j)})}{p(X_t | X_{t-1}^{(j)}, Y_t)} \quad (28)$$

(28) 式の右辺は  $X_t$  の関数に見えるが、任意の  $X_t$  の値に対して同じ確率密度を与える。(28) 式を使用すると (21) 式は以下になる。

$$p(X_t | \mathcal{Y}_{1:t}) \approx \frac{1}{C_f} \sum_{j=1}^N \frac{s_t^{(j)} p(Y_t | X_t) p(X_t | X_{t-1}^{(j)})}{p(X_t | X_{t-1}^{(j)}, Y_t)} \times p(X_t | X_{t-1}^{(j)}, Y_t)$$

$$= \sum_{j=1}^N u_t^{(j)} p(X_t | X_{t-1}^{(j)}, Y_t), \quad (29)$$

$$\text{where } C_f = \sum_{k=1}^N \frac{s_t^{(k)} p(Y_t | X_t) p(X_t | X_{t-1}^{(k)})}{p(X_t | X_{t-1}^{(k)}, Y_t)},$$

$$u_t^{(j)} = \frac{s_t^{(j)} p(Y_t | X_t) p(X_t | X_{t-1}^{(j)})}{C_f \times p(X_t | X_{t-1}^{(j)}, Y_t)}$$

ここで、 $p(X_t | X_{t-1}^{(j)}, Y_t)$  からサンプリングされた粒子  $X_t^{(j)}$  で

$$p(X_t | X_{t-1}^{(j)}, Y_t) \approx \delta(X_t - X_t^{(j)})$$

という近似をおき (29) 式を書き換えると、(29) 式は最適な提案分布  $p(X_t | X_{t-1}^{(j)}, Y_t)$  を使用した PF[5,15] とよばれるアルゴリズムになる。 $p(X_t | X_{t-1}^{(j)}, Y_t)$  は状態の事後分布  $p(X_t | \mathcal{Y}_{1:t})$  を粗く近似した分布とみなせる。しかしながら通常、 $p(X_t | X_{t-1}^{(j)}, Y_t)$  を容易に計算することは困難なため、以下の仮定がおかれる。

$p(X_t, Y_t | X_{t-1}^{(j)})$  に以下の正規性を仮定すると

$$p(X_t, Y_t | X_{t-1}^{(j)}) = N \left( \begin{pmatrix} X_t \\ Y_t \end{pmatrix} \middle| \begin{pmatrix} \bar{X}_t^{(j)} \\ \bar{Y}_t^{(j)} \end{pmatrix}, \begin{pmatrix} \bar{P}_t^{(j)} & V_{X_t Y_t}^{(j)} \\ V_{Y_t X_t}^{(j)} & V_{Y_t Y_t}^{(j)} \end{pmatrix} \right) \quad (30)$$

$p(X_t | X_{t-1}^{(j)}, Y_t)$  は  $N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)})$  になる。ここで  $\hat{X}_t^{(j)}$  および  $\hat{P}_t^{(j)}$  は (30) 式の仮定に基づき GF で計算される。そして  $N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)})$  からサンプリングされた粒子  $X_t^{(j)}$  で

$$p(X_t | X_{t-1}^{(j)}, Y_t) \approx \delta(X_t - X_t^{(j)})$$

という近似をおくと、(29) 式は以下になる。

$$p(X_t | \mathcal{Y}_{1:t}) \approx \sum_{j=1}^N u_t^{(j)} \delta(X_t - X_t^{(j)}), \quad (31)$$

$$\text{where } u_t^{(j)} = \frac{s_t^{(j)} p(Y_t | X_t^{(j)}) N(X_t^{(j)} | \bar{X}_t^{(j)}, \bar{P}_t^{(j)})}{C_f \times N(X_t^{(j)} | \hat{X}_t^{(j)}, \hat{P}_t^{(j)})}$$

$$C_f = \sum_{k=1}^N \frac{s_t^{(k)} p(Y_t | X_t^{(k)}) N(X_t^{(k)} | \bar{X}_t^{(k)}, \bar{P}_t^{(k)})}{N(X_t^{(k)} | \hat{X}_t^{(k)}, \hat{P}_t^{(k)})}$$

(31) 式は提案分布  $N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)})$  を使用した PF のアルゴリズム (SIS) に相当する [9]。  $N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)})$  を EKF で計算するアルゴリズムは extended Kalman particle filter (EKPF)[16] とよばれる。同様に UKF で計算した場合、そのアルゴリズムは unscented Kalman particle filter (UKPF)[17] とよばれる。提案分布を NGF で近似する手法もあるが、処理コストが増大するためあまり実用的ではない [12,18,19]。

(26) 式の BF(MCF) と比較すると、 $X_t^{(j)}$  は状態の事後分布を近似した分布（提案分布）からサンプリングされているため、 $j=1, 2, \dots, N$  に対して  $p(Y_t | X_t^{(j)})$  は  $p(Y_t | X_t^{(j,j)})$  より大きくなることが期待でき、重みの偏りの問題が改善しているように思える。しかしながら、(31) 式の状態の事後分布の近似精度は提案分布の精度に大きく依存している。提案分布が最適な提案分布から逸脱している場合、(31) 式の近似精度は大きく劣化する。つまり、SIS は状態の事後分布の近似精度を犠牲にしながら、重みの偏りの問題を軽減させる手法であるといえる。それでも重みの偏りを完璧に除去することは不可能であり、実際、SIS においても粒子のリサンプリング処



理が適宜追加され、PIPが生じる。次章でこの問題に対する新しいアプローチを提案する。

#### 4. 提案分布の選択と粒子のサンプリング

3.2.3 節で導出した SIS は重みの偏りに対処しているが、全粒子の重みを均一にすることは不可能であり、粒子のリサンプリング処理後に PIP が生じる。そこでわれわれは PIP に対して以下のアプローチを提案する。 $\alpha_t$  として  $X_{t-1}$  を採用し、(21) 式に (30) 式の仮定を適用すると以下になる。

$$p(X_t|\mathcal{Y}_{1:t}) \approx \sum_{j=1}^N u_t^{(j)} N(X_t|\hat{X}_t^{(j)}, \hat{P}_t^{(j)}), \quad (32)$$

$$\text{where } u_t^{(j)} = \frac{s_t^{(j)} N(Y_t|\bar{Y}_t^{(j)}, V_{Y_t Y_t}^{(j)})}{\sum_{k=1}^N s_t^{(k)} N(Y_t|\bar{Y}_t^{(k)}, V_{Y_t Y_t}^{(k)})}$$

ここで

$$p(Y_t|X_{t-1}^{(j)}) = \int p(X_t, Y_t|X_{t-1}^{(j)}) dX_t$$

を使用した。(32) 式は状態の事後分布が  $N$  個の提案分布の重ねあわせ（正規分布の重ねあわせ）で近似されることを示しており、各分布の重みは  $u_t^{(j)}, (j=1, 2, \dots, N)$  で計算される。

提案法ではまず (32) 式を計算する。そして (22) 式のような  $N$  個の粒子によるフィルタ分布の表現を得るが、これを以下の手順で実行する。

- (1) 重み  $u_t^{(j)}, (j=1, 2, \dots, N)$  に基づき、各提案分布を選択する。
- (2) 選択された提案分布から一つずつ粒子をサンプリングする。

BF(MCF) や SIS と同様に、重み  $u_t^{(j)}, (j=1, 2, \dots, N)$  には偏りが生じ、結果として手順 1 により同じ提案分布が複数回選択されることもあるが、手順 2 により異なった粒子が必ずサンプリングされる。したがって、この手法には PIP が存在しない。たとえば一つの提案分布が重み 1 を獲得し、その他の提案分布の重みがゼロという極端な場合においても、手順 2 により異なった  $N$  個の粒子がサンプリング可能になる。

本アプローチによるフィルタ分布の近似誤差は (32) 式の近似誤差、つまり最適な提案分布に対する提案分布  $N(X_t|\hat{X}_t^{(j)}, \hat{P}_t^{(j)})$  の設定誤差だけになる。このような提案分布の選択および粒子のサンプリング法を importance selection and sampling (ISS) とよび、以下に ISS を使用した PF のアルゴリズム (importance selection and sampling filter, ISSF) を示す。

##### 4.1 ISSF

ISSF のアルゴリズムは以下になる。アルゴリズム中の

$$p(X_t|X_{t-1}^{(j)}) \approx N(X_t|\bar{X}_t^{(j)}, \bar{P}_t^{(j)})$$

は、 $p(X_t|X_{t-1}^{(j)})$  の 3 次以上のモーメントを無視することを得る。SIS を使用したフィルタリングアルゴリズムでは、提案分布からの粒子のサンプリング後に重みを計算する必要があるが、ISSF では GF による提案分布の計算の際に重みも計算されるため、処理コストも改善する。GF としては EKF や UKF を選択することが多いが、必ずしも精度が高い GF の実現アルゴリズムを選択することが ISSF の精度改善につながるとは限らない。たとえば 2.3 節で提案した GEnKF は漸近的に GF に一致するが、最適な提案分布が非ガウシアンの場合は、ガウス性を追求することで最適な提案分布からの逸脱が大きくなってしまう。このような理由から 5 章のシミュレーションでは、GF としてその実現アルゴリズムの中で最速な EKF を採用する。

##### ISSF のアルゴリズム

予測分布の計算（時間更新）

$$p(X_t|\mathcal{Y}_{1:t-1}) \approx \frac{1}{N} \sum_{j=1}^N p(X_t|X_{t-1}^{(j)})$$

フィルタ分布の計算（観測更新）

$j=1, 2, \dots, N$  に対して

$$p(X_t|X_{t-1}^{(j)}) \approx N(X_t|\bar{X}_t^{(j)}, \bar{P}_t^{(j)})$$

GF で  $N(Y_t|\bar{Y}_t^{(j)}, V_{Y_t Y_t}^{(j)})$  を計算

GF で  $N(X_t|\hat{X}_t^{(j)}, \hat{P}_t^{(j)})$  を計算

$$u_t^{(j)} = \frac{s_t^{(j)} N(Y_t|\bar{Y}_t^{(j)}, V_{Y_t Y_t}^{(j)})}{\sum_{k=1}^N s_t^{(k)} N(Y_t|\bar{Y}_t^{(k)}, V_{Y_t Y_t}^{(k)})}$$

$$p(X_t|\mathcal{Y}_{1:t}) \approx \sum_{j=1}^N u_t^{(j)} N(X_t|\hat{X}_t^{(j)}, \hat{P}_t^{(j)})$$

ISS を実行し、 $N$  個の粒子をサンプリング

$$p(X_t|\mathcal{Y}_{1:t}) \approx \frac{1}{N} \sum_{j=1}^N \delta(X_t - X_t^{(j)})$$

##### 4.2 IGPf

(32) 式から以下のようにフィルタ分布の 2 次モーメントまでを計算し、それらが作る正規分布から粒子をサンプリングすることも考えられる。

$$\begin{aligned} \hat{X}_t &= E[X_t|\mathcal{Y}_{1:t}] = \int X_t p(X_t|\mathcal{Y}_{1:t}) dX_t \\ &\approx \sum_{j=1}^N u_t^{(j)} \left( \int X_t N(X_t|\hat{X}_t^{(j)}, \hat{P}_t^{(j)}) dX_t \right) \\ &= \sum_{j=1}^N u_t^{(j)} \hat{X}_t^{(j)} \\ \hat{P}_t &= E[(X_t - \hat{X}_t)(X_t - \hat{X}_t)^T|\mathcal{Y}_{1:t}] \\ &\approx \sum_{j=1}^N u_t^{(j)} \times \end{aligned}$$

$$\begin{aligned}
& \left( \int (X_t - \hat{X}_t)(X_t - \hat{X}_t)^T N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)}) dX_t \right) \\
&= \sum_{j=1}^N u_t^{(j)} \times \left( \int \left( (X_t - \hat{X}_t^{(j)}) + (\hat{X}_t^{(j)} - \hat{X}_t) \right) \right. \\
&\quad \times \left( (X_t - \hat{X}_t^{(j)}) + (\hat{X}_t^{(j)} - \hat{X}_t) \right)^T \\
&\quad \times N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)}) dX_t \Big) \\
&= \sum_{j=1}^N u_t^{(j)} \left( \hat{P}_t^{(j)} + (\hat{X}_t^{(j)} - \hat{X}_t)(\hat{X}_t^{(j)} - \hat{X}_t)^T \right)
\end{aligned}$$

つまり複数の提案分布に基づき状態の事後分布を正規分布  $N(X_t | \hat{X}_t, \hat{P}_t)$  で近似表現し、このフィルタ分布から  $N$  個の粒子をサンプリングする。この手法は 2.3 節で説明した GPF の改良になっており、実際、重み  $u_t^{(j)}, (j=1, 2, \dots, N)$  が偏ったとしても、提案分布の共分散行列がある程度大きい場合、フィルタ分布の 2 次モーメントとして十分な共分散行列が保証される。このアルゴリズムを importance GPF (IGPF) とよび、以下にまとめる。時間更新は ISSF と同じであるため省略する。

#### IGPF のアルゴリズム

フィルタ分布（観測更新）

$$\begin{aligned}
& j = 1, 2, \dots, N \text{ に対して} \\
& p(X_t | X_{t-1}^{(j)}) \approx N(X_t | \bar{X}_t^{(j)}, \bar{P}_t^{(j)}) \\
& \text{GF で } N(Y_t | \bar{Y}_t^{(j)}, V_{Y_t Y_t}^{(j)}) \text{ を計算} \\
& \text{GF で } N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)}) \text{ を計算} \\
& u_t^{(j)} = \frac{s_t^{(j)} N(Y_t | \bar{Y}_t^{(j)}, V_{Y_t Y_t}^{(j)})}{\sum_{k=1}^N s_t^{(k)} N(Y_t | \bar{Y}_t^{(k)}, V_{Y_t Y_t}^{(k)})} \\
& p(X_t | \mathcal{Y}_{1:t}) \approx \sum_{j=1}^N u_t^{(j)} N(X_t | \hat{X}_t^{(j)}, \hat{P}_t^{(j)}) \\
& \hat{X}_t = \sum_{j=1}^N u_t^{(j)} \hat{X}_t^{(j)} \\
& \hat{P}_t = \sum_{j=1}^N u_t^{(j)} \left( \hat{P}_t^{(j)} + (\hat{X}_t^{(j)} - \hat{X}_t)(\hat{X}_t^{(j)} - \hat{X}_t)^T \right) \\
& N(X_t | \hat{X}_t, \hat{P}_t) \text{ から } N \text{ 個の粒子をサンプリング} \\
& X_t^{(j)} \sim N(X_t | \hat{X}_t, \hat{P}_t), (j=1, 2, \dots, N) \\
& p(X_t | \mathcal{Y}_{1:t}) \approx \frac{1}{N} \sum_{j=1}^N \delta(X_t - X_t^{(j)})
\end{aligned}$$

IGPF では  $N$  個の提案分布を GF (EKF) で計算する必要があるため、GPF と比べて処理コストが高いように思える。しかしながら GPF で重みの偏りに対処するには粒子数を多くとる必要があり、結果としてどの程度フィルタリング速度に差が出るかは自明でない。2.3 節の GEnKF2 は GPF の近似であるのに対して、IGPF は GPF の改善である。次章で各フィルタの性能をシミュ

レーションにより確認する。

以下に ISSF と IGPF の性質についてまとめる。

- ISSF は選択された提案分布から粒子をサンプリングし (ISS を実行し)、PF の PIP を改善。
- IGPF は ISSF のフィルタ分布に正規性を仮定し、GPF の重みの偏りの問題を改善。

## 5. シミュレーション

以下の提案フィルタの性能を、二つの代表的なシミュレーションモデルを使用して数値的に確認する。

- (1) GEnKF
- (2) GEnKF2
- (3) ISSF
- (4) IGPF

GEnKF は 2.3 節で説明した GF の実現アルゴリズムであり、粒子数が増えると GF に一致する。つまり漸近的ではあるが、GF の厳密な実現アルゴリズムである。GEnKF2 は 2.3 節で説明したように GPF の近似アルゴリズムであるが、アルゴリズム中に粒子の重みの偏りが存在しないため、問題によっては GPF より高い性能を出すことが期待される。とくに高次元の問題に対して効果が期待できる。ISSF は 4.1 節で提案した ISS に基づく PF のアルゴリズムであり、PF の本質的な問題である PIP を改善することで SIS より優れた性能を出す。IGPF は 4.2 節に提案した複数提案分布に基づく GPF のアルゴリズムであり、状態の事後分布の 2 次モーメントまでの推定精度が改善されるため、GPF より優れた性能を出すことが期待される。

### 5.1 スカラーモデル

以下のモデル [20] を使用する。この問題は非線形成長モデル (nonlinear growth model) とよばれ、システムモデル、観測モデルともに状態に対して強い非線形変換がかかるため、状態の事前分布、事後分布ともに非ガウシアンになることが予想される。

$$\begin{aligned}
x_t &= 0.5x_{t-1} + \frac{25x_{t-1}}{1 + (x_{t-1})^2} \\
&\quad + 8 \cos(1.2(t-1)) + \psi_t, \quad \psi_t \sim N(0, 1) \\
y_t &= \frac{(x_t)^2}{20} + \omega_t, \quad \omega_t \sim N(0, 1)
\end{aligned}$$

$x_0 = 0$  とする。観測値は  $t=1$  から  $t=100$  まで生成した。比較したフィルタは TUKF, GPF, EKPF, EnKF である。TUKF, EKPF は GF および NGF (SIS) の代表的な実現アルゴリズムである。各フィルタは  $N(x_0 | 0, 0)$  の初期状態、つまり、確定的な値 0 から開始し、システムノイズ、観測ノイズの分散値は 1 とした。EKPF の粒子のリサンプリング処理と、ISSF の提案分布の選択処理は毎ステップ、観測更新（フィルタリング）後に実行した。Table 1 に、100 回のモンテカルロランにおけるフィルタリング精度（平均 RMSE）とフィルタリング速度（一つの観測値のフィルタリングに要した平均時間 [秒]）の

Table 1 スカラーモデルに対する各フィルタのフィルタリング精度 (平均 RMSE) とフィルタリング速度 (一つの観測値のフィルタリングに要した平均時間 [秒]). TUKF は状態の次元によって粒子数 (シグマ点列の数) が決まるため, 性能値は一つのみである.  $N$  は粒子数であり, 同一粒子数において最も高い性能値を太字で示す.

	TUKF	GPF	EKPF	EnKF
$N = 10$	7.5 0.001s	5.8 0.001s	4.8 0.003s	5.0 0.001s
$N = 50$		3.5 0.006s	3.1 0.02s	3.4 0.006s
$N = 100$		2.8 0.01s	2.7 0.04s	3.4 0.01s
	GEnKF	GEnKF2	ISSF	IGPF
$N = 10$	7.3 0.002s	5.6 0.002s	<b>4.2</b> <b>0.002s</b>	4.4 0.002s
$N = 50$	6.3 0.008s	3.5 0.007s	<b>2.7</b> <b>0.01s</b>	3.1 0.01s
$N = 100$	5.7 0.02s	3.5 0.01s	<b>2.6</b> <b>0.02s</b>	2.7 0.02s

結果を示す. 平均 RMSE は以下の式で与えられる.

$$\frac{1}{100} \sum_{t=1}^{100} \text{RMSE}(t), \quad \text{where}$$

$$\text{RMSE}(t) = \sqrt{E[(x_t - \hat{x}_t)^2 | \mathcal{Y}_{1:t}]}$$

$$\approx \sqrt{\frac{1}{100} \sum_{k=1}^{100} (x_t - \hat{x}_t^{(k)})^2}$$

ここで  $\hat{x}_t^{(k)}, (k=1, 2, \dots, 100)$  は  $k$  番目のランにおける時刻  $t$  の状態の推定値である.

Table 1 から ISSF が最も高いフィルタリング性能を示していることがわかる. つぎに良い性能を示したのは IGPF と EKPF である. IGPF はフィルタ分布の 3 次以上のモーメントを無視し, フィルタ分布に正規性を仮定しているにもかかわらず, NGF である EKPF とほぼ同等の性能を示している. これは IGPF が状態の事後分布の 2 次モーメントまでを高精度に推定し, 結果として予測分布の推定精度が向上していることを示している. このことはつぎに良い性能を示した GPF の結果と比較すると明らかで, IGPF は GPF のフィルタリング精度を改善していることがわかる. EnKF および GEnKF2 はそれに次ぐ性能であり, その数値は GF である TUKF, GEnKF のものより優れており, この問題に対しては状態と観測値の予測分布に正規性を仮定することは無理であることを示している.

平均 RMSE の下限は平均 Cramer-Rao lower bound (average CRLB)[5] を計算することでもとめられる. こ

Table 2 システムノイズと観測ノイズを小さくしたスカラーモデルに対する各フィルタのフィルタリング精度 (平均 RMSE) とフィルタリング速度 (一つの観測値のフィルタリングに要した平均時間 [秒]). 各数値の見方は Table 1 と同じである.

	TUKF	GPF	EKPF	EnKF
$N = 10$	2.53 0.001s	1.35 0.001s	1.07 0.004s	1.31 0.001s
$N = 50$		0.92 0.006s	0.67 0.02s	0.73 0.006s
$N = 100$		0.6 0.01s	0.53 0.04s	0.44 0.01s
	GEnKF	GEnKF2	ISSF	IGPF
$N = 10$	2.67 0.002s	1.71 0.002s	<b>0.88</b> <b>0.003s</b>	1.1 0.002s
$N = 50$	0.99 0.008s	0.88 0.006s	0.65 0.01s	<b>0.58</b> <b>0.01s</b>
$N = 100$	1.1 0.02s	0.68 0.01s	0.44 0.03s	<b>0.42</b> <b>0.02s</b>

の数値は最適フィルタの平均 RMSE と一致するため, 各フィルタの平均 RMSE と比べることで, 現状, 各フィルタがどの程度最適なものに近づいているか評価することが可能になる. この問題に対する average CRLB は 0.76 であった. 各フィルタの中で最良なものであった ISSF との差は約 1.8 であり, この差をさらに縮めることが今後の課題になる.

#### システムノイズ, 観測ノイズが小さい場合

つぎに, 5.1 節のスカラーモデルに対して  $\psi_t \sim N(0, 0.01)$ ,  $\omega_t \sim N(0, 0.01)$  とした場合の問題に対するシミュレーション結果を Table 2 に示す. Table 1 と同様に, 100 回のモンテカルロシミュレーションの結果である. Table 2 からこの問題に対しては IGPF が最も高い性能を示していることがわかる. つぎは ISSF である. この結果は, 各ノイズが小さい場合はフィルタ分布の 3 次以上のモーメントを無視し, 正規性を仮定しても精度が劣化しないことを示している. したがって EnKF の結果も良い. GPF の精度は重みの偏りの問題 (全粒子の重みがゼロになる問題も含む) により劣化している. GEnKF2 と GEnKF に顕著な性能差が確認できることから, この問題に対しては予測分布に正規性を課することはできないことがわかる. この問題に対する average CRLB は 0.09 であり, 最良の IGPF の平均 RMSE 値との差は 0.33 であった. 前節の問題に対する差は 1.8 であったことから, 各ノイズが小さくなると提案フィルタはより最適フィルタに近づくことがわかる.

以上の結果から, 各ノイズが大きい場合は ISSF が最良のフィルタリング性能を出すことがわかった. 逆に小さい場合は IGPF で十分な結果が得られることがわかつ

た。また、このような問題に対して EnKF も高い性能（とくに高速なフィルタリング速度）を記録していることから、有効なフィルタの一つであることが確認された。

## 5.2 ベクトルモデル

以下のモデル [21] を使用する。このモデルは人工衛星の地球への再突入モデルである。

$$\begin{pmatrix} \dot{x}_{1,t} \\ \dot{x}_{2,t} \\ \dot{x}_{3,t} \\ \dot{x}_{4,t} \\ \dot{x}_{5,t} \end{pmatrix} = \begin{pmatrix} x_{3,t} \\ x_{4,t} \\ G(t)x_{1,t} + D(t)x_{3,t} \\ G(t)x_{2,t} + D(t)x_{4,t} \\ 0 \end{pmatrix} + A\tilde{\Psi}_t$$

ここで  $(x_{1,t}, x_{2,t}, x_{3,t}, x_{4,t}, x_{5,t})^T$  は人工衛星のポジション  $(x_{1,t}, x_{2,t})^T$  と速度ベクトル  $(x_{3,t}, x_{4,t})^T$ 、そして衛星の空気力学的な性質  $(x_{5,t})$  から成る 5 次元の状態ベクトルである。  $G(t)$  と  $D(t)$  は重力と空気抵抗に関する項であり、それぞれ以下で計算される。

$$G(t) = -\frac{Gm_0}{|R_t|^3}$$

$$D(t) = \xi_0 \exp\left(x_{5,t} + \frac{|R_0| - |R_t|}{H_0}\right) |V_t|$$

ここで各定数は

$$Gm_0 = 3.9860 \times 10^5, \quad \xi_0 = -0.59783,$$

$$H_0 = 13.406, \quad |R_0| = 6374$$

であり、典型的な大気環境および人工衛星の性質を反映している。また

$$|R_t| = \sqrt{(x_{1,t})^2 + (x_{2,t})^2}, \quad |V_t| = \sqrt{(x_{3,t})^2 + (x_{4,t})^2}$$

である。単位は km, kg, second である。  $A$  と  $\tilde{\Psi}_t$  は以下である。

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{\Psi}_t \sim N\left(0, \begin{pmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{pmatrix}\right)$$

### 5.2.1 離散化モデル

上記の連続形のモデルを Euler-Maruyama 近似で離散化し、概 2 次近似すると以下になる。

$$X_t = f(X_{t-1}) + \Psi_t$$

ここで  $X_t$  は

$$X_t = (x_{1,t}, x_{2,t}, x_{3,t}, x_{4,t}, x_{5,t})^T$$

であり、  $f(X_{t-1})$  は以下である。

$$\begin{pmatrix} x_{1,t-1} + x_{3,t-1}\Delta_t + \frac{1}{2}(G(t-1)x_{1,t-1} + D(t-1)x_{3,t-1})\Delta_t^2 \\ x_{2,t-1} + x_{4,t-1}\Delta_t + \frac{1}{2}(G(t-1)x_{2,t-1} + D(t-1)x_{4,t-1})\Delta_t^2 \\ x_{3,t-1} + (G(t-1)x_{1,t-1} + D(t-1)x_{3,t-1})\Delta_t \\ x_{4,t-1} + (G(t-1)x_{2,t-1} + D(t-1)x_{4,t-1})\Delta_t \\ x_{5,t-1} \end{pmatrix}$$

ここで  $\Delta_t$  は時間幅である。初期状態  $X_0$  は以下の正規分布に従うものとする。

$$N\left(\begin{pmatrix} 6400.4 \\ 50.0 \\ -1.8093 \times 1.5 \\ -6.7967 \times 1.5 \\ 0.6932 \end{pmatrix}, \begin{pmatrix} 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right)$$

上記の設定は、地球半径 6374km 付近からシミュレーションを開始することを意味し、システムモデルにおける大気抵抗による非線形項をより大きくするため、突入速度が大きくとられている。  $\Psi_t$  は期待値ゼロの正規白色ノイズで、その共分散行列は以下になる。

$$Q = \begin{pmatrix} \frac{q_1\Delta_t^3}{3} & 0 & \frac{q_1\Delta_t^2}{2} & 0 & 0 \\ 0 & \frac{q_2\Delta_t^3}{3} & 0 & \frac{q_2\Delta_t^2}{2} & 0 \\ \frac{q_1\Delta_t^2}{2} & 0 & q_1\Delta_t & 0 & 0 \\ 0 & \frac{q_2\Delta_t^2}{2} & 0 & q_2\Delta_t & 0 \\ 0 & 0 & 0 & 0 & q_3\Delta_t \end{pmatrix}$$

ここで

$$q_1 = q_2 = 2.4064 \times 10^{-3}, \quad q_3 = 0$$

である。

観測モデルは以下である。

$$\begin{pmatrix} y_{1,t} \\ y_{2,t} \end{pmatrix} = \begin{pmatrix} \sqrt{(x_{1,t} - 6374)^2 + (x_{2,t})^2} \\ \tan^{-1}\left(\frac{x_{2,t}}{(x_{1,t} - 6374)}\right) \end{pmatrix} + \Omega_t$$

つまり、(6374 km, 0) の座標にある観測所からの人工衛星の距離と角度が観測される。観測ノイズ  $\Omega_t$  は期待値ゼロの正規白色ノイズで、その共分散行列は  $\text{diag}((3.0 \times 0.001)^2, (3.0 \times 0.017)^2)$  とする。

### 5.2.2 各フィルタの性能比較

フィルタの初期状態は以下とする。

$$N\left(\begin{pmatrix} 6400.4 \\ 50.0 \\ -1.8093 \times 1.5 \\ -6.7967 \times 1.5 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 10^{-6} & 0 & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 & 0 \\ 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}\right)$$

つまり人工衛星の空気力学的性質 ( $\hat{x}_{5,0}$ ) は未知とし、フィルタリングにより人工衛星の状態とともに同時推定する。フィルタに対するシステムノイズは 5.2.1 節の  $\Psi_t$  の  $Q$  に

$$q_1 = q_2 = 2.4064 \times 10^{-3}, \quad q_3 = 0.001^2$$

を代入したものを使用し、観測ノイズは 5.2.1 節の  $\Omega_t$  と同じものを使用した。

$\Delta_t = 1s$  とし、観測値を  $t=1$  から  $t=100$  まで生成した。つまり約 100 秒間シミュレーションを行った。比較したフィルタはスカラーモデルのものと同じである。5.1 節と同様に、以下の Table 3 に 100 回のモンテカルロランにおけるフィルタリング精度と速度の結果を示す。フィルタリング精度は以下の衛星のポジションと空気力学的性質に対する平均 RMSE で数値化した。

$$\text{RMSE}^{(1)}(t) = \sqrt{E[(x_{1,t} - \hat{x}_{1,t})^2 + (x_{2,t} - \hat{x}_{2,t})^2 | \mathcal{Y}_{1:t}]}$$

$$\text{RMSE}^{(2)}(t) = \sqrt{E[(x_{5,t} - \hat{x}_{5,t})^2 | \mathcal{Y}_{1:t}]}$$

各 RMSE の期待値および平均の計算は 5.1 節と同じである。

Table 3 の結果からスカラーモデルと同様に、ISSF が最も高い性能を出していることがわかる。EKPF はそれに次ぐが、ISSF と比べて処理コストが高いことがわかる。IGPF はつぎに良い性能を示しているが、フィルタ分布の 3 次以上のモーメントを無視したことによる推定精度の劣化が見られる。しかしながら EnKF, GEnKF2, GEnKF には大きな性能差がみられないため、状態の事前分布および事後分布にそこまで強い非ガウス性は存在していないことがわかる。このことは、衛星の速度が空気抵抗により急激に減少し、システムモデルの非線形項が弱まることや、衛星のポジションが比較的高精度に観測されることから説明できる。

GEnKF は粒子数を増やすことで漸近的に GF を実現するため、GF の近似アルゴリズムである TUKF より高いフィルタリング精度を記録している。GPF は粒子の重みの偏りの問題により、フィルタが発散した (粒子数を 2000 まで増やしても発散)。この結果は、GPF の代わりとしての GEnKF2 や EnKF の有効性を示している。

システムノイズ、観測ノイズが小さい場合

つぎにスカラーモデルの時と同様に、システムノイズと観測ノイズが小さい場合に対する各フィルタの性能を確認する。5.2.1 節のモデルのシステムノイズの分散値を

$$q_1 = q_2 = 20.0 \times 2.4064 \times 10^{-5}, \quad q_3 = 0$$

とし、観測ノイズの共分散行列を  $\text{diag}(0.001^2, 0.017^2)$  とした。この問題では衛星のポジションの観測誤差が小さいため、状態の事後分布がさらに正規分布に近づくことが予想される。フィルタにのるシステムノイズは

Table 3 ベクトルモデルに対する各フィルタのフィルタリング精度 (平均 RMSE) とフィルタリング速度 (一つの観測値のフィルタリングに要した平均時間 [秒])。上の数値が衛星のポジションの平均 RMSE、下の数値が衛星の空気力学的性質の平均 RMSE である。div は発散を示す。TUKF は状態の次元によって粒子数 (シグマ点列の数) が決まるため、その性能値だけを記している。N は粒子数であり、同一粒子数において最も高い性能値を太字で示す。

	TUKF	GPF	EKPF	EnKF
$N = 100$	0.91 0.19 0.002s	div 0.72 0.02s	0.85 0.04 0.07s	0.83 0.07 0.02s
$N = 300$		div 0.81 0.06s	0.76 0.004 0.26s	0.79 0.07 0.06s
$N = 500$		div 0.76 0.09s	0.73 0.004 0.53s	0.76 0.09 0.1s
$N = 700$		div 0.72 0.13s	0.73 0.004 0.88s	0.76 0.09 0.15s
$N = 1000$		div 0.72 0.19s	0.72 0.004 1.58s	0.76 0.09 0.22s
	GEnKF	GEnKF2	ISSF	IGPF
$N = 100$	0.95 0.08 0.03s	0.81 0.08 0.03s	<b>0.81</b> <b>0.04</b> <b>0.05s</b>	0.85 0.04 0.03s
$N = 300$	0.78 0.06 0.1s	0.78 0.07 0.08s	<b>0.73</b> <b>0.004</b> <b>0.2s</b>	0.76 0.004 0.1s
$N = 500$	0.76 0.07 0.16s	0.75 0.07 0.13s	<b>0.73</b> <b>0.003</b> <b>0.43s</b>	0.76 0.004 0.17s
$N = 700$	0.74 0.07 0.24s	0.75 0.08 0.19s	<b>0.72</b> <b>0.003</b> <b>0.75s</b>	0.75 0.003 0.25s
$N = 1000$	0.75 0.08 0.36s	0.74 0.07 0.28s	<b>0.71</b> <b>0.003</b> <b>1.4s</b>	0.75 0.003 0.36s

$$q_1 = q_2 = 20.0 \times 2.4064 \times 10^{-5}, \quad q_3 = 0.001^2$$

であり、観測ノイズはモデルと同じものを使用した。100 回のモンテカルロシミュレーションを行った結果を以下の Table 4 に示す。

Table 4 の結果から、この問題に対してはスカラーモデルの時と同様に、IGPF が最も高い性能を示している

Table 4 システムノイズと観測ノイズを小さくしたベクトルモデルに対する各フィルタのフィルタリング精度（平均 RMSE）とフィルタリング速度（一つの観測値のフィルタリングに要した平均時間 [秒]）。各数値の見方は Table 3 と同じである。

	TUKF	GPF	EKPF	EnKF
$N = 100$	0.42	div	0.34	<b>0.28</b>
	0.18	0.72	0.04	<b>0.04</b>
	0.002s	0.02s	0.07s	<b>0.02s</b>
$N = 300$		div	0.28	0.27
		0.8	0.004	0.04
		0.05s	0.26s	0.06s
$N = 500$		div	0.27	0.26
		0.75	0.004	0.06
		0.09s	0.5s	0.1s
$N = 700$		div	0.27	0.26
		0.72	0.004	0.06
		0.13s	0.86s	0.15s
$N = 1000$		div	0.27	0.26
		0.72	0.003	0.05
		0.18s	1.56s	0.21s
	GEnKF	GEnKF2	ISSF	IGPF
$N = 100$	0.33	0.29	0.32	0.31
	0.05	0.04	0.04	0.04
	0.03s	0.02s	0.05s	0.03s
$N = 300$	0.28	0.28	0.27	<b>0.27</b>
	0.04	0.04	0.004	<b>0.005</b>
	0.1s	0.08s	0.2s	<b>0.09s</b>
$N = 500$	0.27	0.27	0.26	<b>0.27</b>
	0.04	0.04	0.004	<b>0.004</b>
	0.16s	0.13s	0.4s	<b>0.17s</b>
$N = 700$	0.27	0.27	0.26	<b>0.26</b>
	0.04	0.04	0.003	<b>0.003</b>
	0.23s	0.19s	0.72s	<b>0.24s</b>
$N = 1000$	0.27	0.26	0.26	<b>0.26</b>
	0.05	0.04	0.003	<b>0.003</b>
	0.35s	0.27s	1.39s	<b>0.36s</b>

ことがわかる。その精度は ISSF とほぼ同じであるが、速度が優れている。前節のシミュレーションと同様に、EnKF, GEnKF2, GEnKF には大きな性能差がみられないため、状態の事前分布および事後分布に強い非ガウス性は存在していない。GEnKF は粒子数を増やすことで性能が改善し、その数値は ISSF や IGPF のものと比較しても良好である。これは、このような問題に対しては GF の適用も有効であることを示している。

一方、GPF は Table 3 の結果と同じ理由によりフィルタが発散した。GEnKF2 および EnKF は良好な結果

を示していることから、GPF の代わりとしてこれらのフィルタを適用することも効果的であることを確認した。この問題に対しては衛星の空気力学的性質を同時推定しているため、average CRLB との比較はできない。しかしながら、最良のフィルタリング精度である IGPF の 0.26 と 0.003 は、衛星の性質が未知の状況においても約 260m の誤差で衛星のポジションを特定でき、かつその空気力学的性質も高精度で同時推定可能であることを示している。

また、ISSF と IGPF は少ない粒子数 ( $N = 300$ ) でも高い精度の状態推定が可能であることを示しており、高速かつ高精度な NGF として期待できる。最適フィルタをさらに追求し、これらの数値を下げるのが今後の課題である。以上の結果から、ベクトルモデルに対しても各ノイズが大きい場合は ISSF が最良の性能を出すことがわかった。また各ノイズが小さい場合、IGPF の推定精度は ISSF とほぼ同等であり、そのフィルタリング速度は比較的高速なため、IGPF の適用が効果的であることを確認した。これらのシミュレーションの結果を踏まえ、次節で本論文をまとめる。

## 6. おわりに

本論文ではまず、EnKF が非線形問題に対する線形最適フィルタの実現アルゴリズムを漸近的に与えることを示した。そして状態の予測分布に正規性を仮定する場合、EnKF が GF の実現アルゴリズムを与えることを示し、このフィルタを GEnKF とよんだ。この関係性から EnKF は GPF と同様に、GF の非正規性をもつ予測分布への拡張とみなすことができ、その際、予測分布に対する不偏性の仮定が必要になることを述べた。また、EnKF に基づく GPF の近似法 (GEnKF2) を提案し、粒子の重みの偏りが生じない GPF のアルゴリズムを設計した。

つぎに、NGF を与える新しい枠組みである MDE を提案し、BF(MCF) や SIS の PF アルゴリズムがその特殊な例であることを示した。そして SIS で問題になる PIP への対処法として ISS を提案し、ISS に基づく新たな PF のアルゴリズムである ISSF を提案した。ISS ではまず重みに基づき提案分布が選択され、それらの分布から異なった粒子がサンプリングされるため、PIP の問題は発生しない。また、複数提案分布に基づく GPF の改善アルゴリズムである IGPF を提案した。最適な提案分布に近い、より高精度な提案分布を高速に計算することができれば、これらのフィルタの性能はさらに向上する。

そして、各フィルタの性能を二つの代表的なシミュレーションモデルを使用して数値的に比較し、ISSF がすべての問題に対して優れたフィルタリング精度を記録することを確認した。GEnKF は漸近的に GF を実現するため、通常の GF の近似アルゴリズムと比べ、粒子数の増加とともに高い精度を示すことを確認した。また、

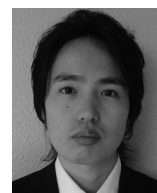
GEnKF2およびEnKFの性能値をGPFの結果と比較することで、とくにベクトルモデルに対してこれらのフィルタが優れた精度を示すことを確認した。システムノイズおよび観測ノイズが小さい場合は、IGPFが十分な性能を示すことを確認し、そのフィルタリング速度は高速なため、広範な実問題への応用が期待できる。

### 参考文献

- [1] 片山: 非線形カルマンフィルタ, 朝倉書店 (2011)
- [2] S. J. Julier, J. K. Uhlmann and H. F. Durrant-Whyte: A new approach for filtering nonlinear systems; *Proc. 1995 American Control Conference*, Vol. 3, pp. 1628–1632 (1995)
- [3] S. J. Julier and J. K. Uhlmann: New extension of the Kalman filter to nonlinear systems; *Proc. AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Vol. Multi Sensor Fusion, Tracking and Resource Management II (1997)
- [4] K. Ito and K. Xiong: Gaussian filters for nonlinear filtering problems; *IEEE Trans. Autom. Control*, Vol. 45, No. 5, pp. 910–927 (2000)
- [5] A. J. Haug: *Bayesian Estimation and Tracking*, John Wiley & Sons, Inc. (2012)
- [6] I. Arasaratnam and S. Heykin: Cubature Kalman filters; *IEEE Trans. Autom. Control*, Vol. 54, No. 6, pp. 1254–1269 (2009)
- [7] L. Chang, B. Hu, A. Li and F. Qin: Transformed unscented Kalman filter; *IEEE Trans. Autom. Control*, Vol. 58, No. 1, pp. 252–257 (2013)
- [8] 樋口, 上野, 中野, 中村, 吉田: データ同化入門 一次世代のシミュレーション技術一, 朝倉書店 (2011)
- [9] S. Sarkka: *Bayesian Filtering and Smoothing*, Cambridge Univ. Press (2013)
- [10] S. Nakano, G. Ueno and T. Higuchi: Merging particle filter for sequential data assimilation; *Nonlinear Processes in Geophysics*, Vol. 14, pp. 395–408 (2007)
- [11] 中野, 上野, 中村, 樋口: Merging particle filter とその特性; 統計数理, Vol. 56, No. 2, pp. 225–234 (2008)
- [12] M. Murata, H. Nagano and K. Kashino: Monte Carlo filter particle filter; *Proc. 2015 European Control Conference*, pp. 2836–2841 (2015)
- [13] T. Yang, P. G. Mehta and S. P. Meyn: Feedback particle filter; *IEEE Trans. Autom. Control*, Vol. 58, No. 10, pp. 2465–2480 (2013)
- [14] A. Doucet, N. de Freitas and N. Gordon: *Sequential Monte Carlo Methods in Practice*, Springer-Verlag New York (2001)
- [15] B. Ristic, S. Arulampalam and N. Gordon: *Beyond the Kalman Filter*, Artech House Publishers (2004)
- [16] J. F. G. de Freitas, M. Niranjan, A. H. Gee and A. Doucet: Sequential Monte Carlo methods to train neural network models; *Neural Computation*, Vol. 12, No. 4, pp. 955–993 (2000)
- [17] R. Van der Merwe, N. de Freitas, A. Doucet and E. Wan: The unscented particle filter; *Advances in Neural Information Processing Systems*, pp. 584–590 (2000)
- [18] P. M. Djuric, T. Lu and M. F. Bugallo: Multiple particle filtering; *Proc. 2007 IEEE Conf. Acoustics, Speech and Signal Processing*, pp. 1181–1184 (2007)
- [19] P. Closas and M. F. Bugallo: Improving accuracy by iterated multiple particle filtering; *IEEE Signal Processing Letters*, Vol. 19, No. 8, pp. 531–534 (2012)
- [20] D. Simon: *Optimal State Estimation*, John Wiley & Sons, Inc. (2006)
- [21] S. J. Julier and J. K. Uhlmann: Unscented filtering and nonlinear estimation; *Proc. IEEE*, Vol. 92, No. 3, pp. 401–422 (2004)

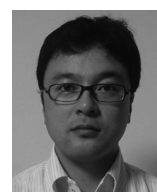
### 著者略歴

村田 眞哉 (正会員)



1982年10月13日生。2007年3月早稲田大学大学院理工学研究科修士課程物理学および応用物理学専攻修了。同年4月日本電信電話株式会社入社。現在NTTコミュニケーション科学基礎研究所研究主任。フィルタリング, データ同化, 情報検索などの研究に従事。2015年度システム制御情報学会奨励賞, SSS'14 Young Author Prize 受賞。IEEE CSS, IFAC, SICEの会員。

平松 薫



1971年4月21日生。1996年3月慶應義塾大学大学院理工学研究科修士課程計算機科学専攻修了。同年4月日本電信電話株式会社入社。現在NTTコミュニケーション科学基礎研究所主幹研究員, グループリーダー。Semantic Web, センサネットワーク, メディア探索技術などの研究に従事。博士(情報学)。情報処理学会(IPSJ), 人工知能学会(JSAI)の会員。