ベイズ推定を用いた時系列状態推定

-GNSS を模したシミュレータを用いた自己位置推定の実装例-

赤井 直紀

2022 年 12 月 21 日

本稿では，ロボットの状態推定に焦点を当てる．機械系の学問でロボットについて議論するときは，しばしば扱う対象のモデル化が良く議論される．例えばロボットを制御するにあたり，ロボットがどのように動くかや，センサの観測値から自身や周囲の状態をどのように認識するかのモデルを考える．当然このモデル化は重要ではあるが，実世界にはモデル化困難な要因が多数存在する．例えば，ある制御入力をロボットに与えても，実世界には摩擦などの様々な要因が存在し，ロボットは記述されたモデル通りには動かない．また，センサの観測にも誤差が含まれるため，安易にセンサの観測値を信じて種々の状態を求めることは好ましくない．

この様な不確かさに対処しながらロボット動かすことを目的とした，確率ロボティクス [1] と呼ばれる学問分野がある．確率ロボティクスでは，ベイズ推定を基盤として，様々な不確かさに対処するための方法が議論されている．本稿では，確率ロボティクスで扱われる状態推定（ロボットの位置を推定する問題など）について述べる．そして簡易的な全球測位衛星システム（Global Navigation Satellite System: GNSS）による位置測位を模したシミュレータを用いて，カルマンフィルタとパーティクルフィルタによる移動ロボットの自己位置推定の実装方法を述べる．さらに最適化に焦点を当て，これと最尤推定の関係性についても述べる．そして，単に最適化を行うのではなく，確率論を用いて時系列状態推定を行うことの利点について述べる．

1　確率の基礎

1.1　確率

ある確率変数が存在するとする．今この変数は，0 か 1 のどちらかの値（2 値）を取るものとする．このとき，$x = 0$ または $x = 1$ となる確率を

$$\begin{aligned} p(x = 0) \\ p(x = 1) \end{aligned} \tag{1}$$

と表す．今，$x = 0$，または $x = 1$ となる確率が等しいとき，

$$p(x = 0) = p(x = 1) = \frac{1}{2} \tag{2}$$

となる．$p(x)$ は，x の確率分布と呼ばれる．

今，x が取りうる値の集合を $\mathcal{X}$ と表す（は離散値であると仮定している）．このとき，確率，および確率

分布は，以下の制約を満たさなければならない．

$$
\begin{gathered}
0 \leq p(x) \leq 1 \quad (x \in \mathcal{X}) \\
\sum_{x \in \mathcal{X}} p(x) = 1
\end{gathered} \tag{3}
$$

つまり，確率の値は 0 以上，1 以下である必要があり，確率分布の総和は 1 でなければならない．馴染み深い例では，例えば 6 面のサイコロを投げるとき，ある目が出る確率はであり，6 つの目が出る確率の総和は 1 となっている．

なお上記の例では，x は離散変数であると仮定しているが，x が連続変数の場合は，$p(x)$ は確率密度，もしくは確率密度関数と呼ばれる．確率密度からある確率を得るためには，任意の区間で確率密度を積分する必要がある．

$$
p_{a,b} = \int_a^b p(x)\mathrm{d}x \quad (a \leq b) \tag{4}
$$

$p_{a,b}$ は，x が区間 $[a, b]$ に収まる確率を表す．なお，確率密度も常に 0 以上である必要がある．また x が離散であるときと同様に，以下の制約を満たす必要がある．

$$
\int_{\mathcal{X}} p(x)\mathrm{d}x = 1 \tag{5}
$$

式 (5) において $\mathcal{X}$ は，x の定義域全体を表す集合である．確率密度が常に 0 以上かつ，x の定義域全体での積分値が 1 になるという制約から，すべての確率の値が 0 以上かつ 1 以下になるという制約も同様に満たす．

1.2 同時確率と条件付き確率

確率にはさらに，同時確率と条件付き確率が存在する．同時確率とは，2 つ以上の変数について考えるものである．例えば，x と y という 2 つの変数が存在する場合，その同時確率は

$$
p(x, y) \tag{6}
$$

と表される．これは，x と y が同時に起こる確率を表す．例えば 6 面のサイコロを投げるとき，1 回目のサイコロの出目を，2 回目のサイコロの出目をで表すとき，それぞれ出目が 1 となる同時確率はとなる．なお同時確率も，確率の値が 0 以上，1 以下になる，および総和が 1 になるという制約を満たさなければならない．ただし総和の計算にあたっては，x と y 両方の値が取りうる組み合わせ（連続値であればそれぞれの定義域）を考慮して総和を計算しなければならない．

条件付き確率とは，ある変数が固定された下での確率を表す．例えば，y という値が固定（もしくは観測されたともいう）されたときに，x の条件付き確率を

$$
p(x|y) \tag{7}
$$

と表す．この確率の値も，0 以上，1 以下の値とならなければならない．ただし同時確率と異なり，総和を計算する際には，についてのみ考慮すれば良い．あくまで条件付き確率は，ある変数が決められた下での，についての確率分布を考えていると理解すればよい．

なおここまでは，離散か連続のスカラー値を取る変数を考えていたが，ベクトルを引数とする確率分布も考えられる．例えば，$\mathbf{x} = (x_1 \; x_2)^\top$ となるベクトルを考える．この確率分布は

$$
p(\mathbf{x}) = p(x_1, x_2) \tag{8}
$$

というように，各ベクトルの要素の同時確率を表すと理解すれば良い．

1.3　加法定理と乗法定理

確率論においては，式 9 に示す加法定理と，式 10 に示す乗法定理という 2 つの重要な定理が存在する．

$$\sum_{y\in\mathcal{Y}} p(x,y) = p(x) \tag{9}$$

$$p(x,y) = p(x|y)p(y) = p(y|x)p(x) \tag{10}$$

なお式 9 では，離散変数を考えているが，連続変数となるときは総和の計算が積分の計算になる．これは以下で述べる全確率の定理やベイズの定理でも同様である．

加法定理は，同時確率のある変数に関する総和（連続値であれば積分）を計算すると，その変数を消すことができることを意味する．ここで，例えば $x = x_1$ と固定して加法定理を計算すると，$x = x_1$ のときの場合のすべてのについて考慮した確率を得ることができる．このような操作を周辺化と呼び，得られた確率を周辺化確率と呼ぶ．

乗法定理は，同時確率を条件付き確率と通常の確率を用いた形に書き変えられることを意味している．つまり，2 つのことが同時に起こる確率はわからないが，片方の変数に対する確率，また片方の変数が固定された場合の条件付き確率が分かれば，同時確率が計算できることを意味している．なを，同時確率をどちらの条件付き確率を用いて書き変えられるかは，仮定する問題やモデルにより定まることに注意しなければならない．

1.4　全確率の定理とベイズの定理

乗法定理を加法定理に代入すると，全確率の定理を得ることができる．

$$\sum_{y\in\mathcal{Y}} p(x|y)p(y) \tag{11}$$

全確率の定理は，y に関する確率と，y が起きた下での，x に関する条件付き確率が計算できれば，x に関する確率が計算できるということを意味している．すなわち，x に関する確率を求めるにあたり，y の情報を活用できることを意味している．

また，乗法定理からはベイズの定理を導くことができる．

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{12}$$

ベイズの定理は，全確率の定理を用いて以下のように書くこともできる．

$$p(x|y) = \frac{p(y|x)p(x)}{\sum_{x\in\mathcal{X}} p(y|x)p(x)} \tag{13}$$

このとき，$\sum_{x\in\mathcal{X}} p(y|x)p(x)$ は，$p(y|x)p(x)$ の x に関して総和を取ったものとなっている．つまり，$\sum_{x\in\mathcal{X}} p(y|x)p(x)$ は正規化のための係数となり，ベイズの定理による計算結果は，確率分布としての制約を満たす．なおベイズの定理においては，$p(x)$ を事前確率，$p(y|x)$ を尤度，$p(x|y)$ を事後確率と呼ぶ．$p(x|y)$ は，y が観測された後での x に関する確率を表しており，その意味で事後確率と呼ばれている．これに対して $p(x)$ を事前確率と呼ぶ．また $p(y|x)$ は，x を固定した場合に y が得られる確率を意味している．ここで y は観測されているものなので，$p(y|x)$ は y が得られるモデルに基づいて記述される確率分布となる．すなわち，y が得られる尤もらしさを表現する確率分布となるため，尤度と呼ばれる．ベイズの定理は，事後確率が直接求めることが難しい場合でも，事前確率と尤度を用いて，それを計算できるということを意味している．

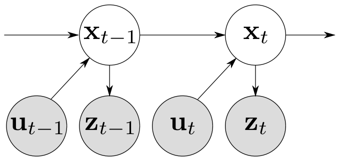

図 1　対象とする問題に対応するグラフィカルモデル．

2　時系列状態推定

2.1　再帰的ベイズフィルタの導出

ここから，時系列の状態推定問題について考えていく．例題として，GNSS レシーバーを有して移動するロボットの位置を求める問題を考える．ここで，ロボットは連続的に移動することを前提としており，その意味で時系列を考慮しなければならない．また，ロボット位置が，推定する状態となる．

時刻 t におけるロボットの位置を $\mathbf{x}_t$ とする．またこの時の GNSS により測位された位置を $\mathbf{z}_t$ とする．ロボットはある制御入力に従って動くものとし，時刻 t の時点で与えられた入力を $\mathbf{u}_t$ とする．今，初期時刻 $t=1$ から，現時刻 $t=T$ まで，GNSS による測位を行いながら移動したロボットの，時刻 T における位置に関する確率分布を求める問題を考える．この問題は，以下の分布を求める問題となる．

$$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}) \tag{14}$$

ここで $1:T$ は，初期時刻から現時刻までのデータをまとめて記述したものであり，例えば $\mathbf{u}_{1:T} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_T)$ である．なお図 1 に，扱う問題に対応するグラフィカルモデルを示す．グラフィカルモデルとは，変数間の依存関係を図的に表したものである．グラフィカルモデルにおいては，矢印の先の変数は，矢印の根元の変数に依存していることを意味する．また灰色の変数は観測可能な変数，白色が推定される変数を表している．グラフィカルモデルを用いることで，求める分布の式変形を図的に理解することもできるが，本稿ではその詳細までは触れない．

式 (14) に示す分布を求めるにあたり，まず時刻 T における GNSS の測位結果 $\mathbf{z}_T$ を用いて，ベイズの定理を適用する．

$$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}) = \frac{p(\mathbf{z}_T|\mathbf{x}_T, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})}{p(\mathbf{z}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})} \tag{15}$$

ここで，$p(\mathbf{z}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})$ は正規化のための係数となるため，これを正規化係数として省略して，

$$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}) = \eta p(\mathbf{z}_T|\mathbf{x}_T, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) \tag{16}$$

と記述する．η が正規化係数である．

今，$p(\mathbf{z}_T|\mathbf{x}_T, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})$ について考える．この分布は，時刻 T における GNSS の測位結果に関する尤度分布である．GNSS の測位は，ロボットの現在位置にのみ依存するものと考えることができる（実際には周囲の環境形状など様々な要因に依存するが，ここでは簡略化のために現在位置のみに依存すると考える）．この仮定により，不要な条件変数を削除することが可能となり，

$$p(\mathbf{z}_T|\mathbf{x}_T, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) = p(\mathbf{z}_T|\mathbf{x}_T) \tag{17}$$

となる．ここで，$p(\mathbf{z}_T|\mathbf{x}_T)$ を GNSS の観測モデルと呼ぶ．これを用いれば，

$$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}) = \eta p(\mathbf{z}_T|\mathbf{x}_T) p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) \tag{18}$$

と書き変えることができる．

今，ここでは連続的に移動するロボットの位置について考えている．しかしながら，上記の確率分布には，過去のロボットの位置に関する変数が含まれていない．すなわち，連続的なロボットの移動が考慮されていない．ここで，$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})$ の確率分布に対して，1 時刻前のロボットの位置 $\mathbf{x}_{T-1}$ を用いて，全確率の定理を適用する．

$$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) = \int p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) \mathrm{d}\mathbf{x}_{T-1} \tag{19}$$

ここで，$p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})$ について考える．この分布では，1 時刻前のロボットの位置 bfx_{T-1} が条件として与えられ，かつ現時刻の制御入力 $\mathbf{u}_t$ が反映され，現時刻 T におけるロボットの位置 $\mathbf{x}_T$ に関する確率分布を表している．すなわち，ある制御入力に対してロボットがどの様に動いたかを記述すればよく，この考えに基づき不要な条件変数を削除すると，

$$p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_{1:T}, \mathbf{z}_{1:T-1}) = p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_T) \tag{20}$$

となる．ここで $p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_T)$ をロボットの動作モデルと呼ぶ．

また $p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T-1})$ について考えると，これは 1 時刻前のロボット位置 $\mathbf{x}_{T-1}$ に関する確率分布である．この分布においては未来の制御入力である $\mathbf{u}_T$ は，明らかに影響を与えない．そのためこれを削除し，$p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T-1}, \mathbf{z}_{1:T-1})$ と記述する．

以上より，求める確率分布 $p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T})$ は，以下の様に書き換えられる．

$$p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}) = \eta p(\mathbf{z}_T|\mathbf{x}_T) \int p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_T) p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T-1}, \mathbf{z}_{1:T-1}) \mathrm{d}\mathbf{x}_{T-1} \tag{21}$$

ここで，左辺 $p(\mathbf{x}_T|\mathbf{u}_{1:T}, \mathbf{z}_{1:T})$ の T を $T-1$ とすれば，右辺の積分項内に存在する $p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T-1}, \mathbf{z}_{1:T-1})$ となる．つまりこの式は，再帰式となっている．この再帰式は，これまでの制御入力と GNSS の測位結果を基に定めたロボットの位置に関する確率分布 $p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T-1}, \mathbf{z}_{1:T-1})$ に対して，ロボットの動作モデル $p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_T)$ を適用し，かつ GNSS の観測モデル $p(\mathbf{z}_T|\mathbf{x}_T)$ による尤度付けを経て，ロボットの位置に関する事後確率の分布が得られるということを意味している．また特に，尤度計算の部分はベイズの定理を適用して得られたものであり，この再帰式は再帰的ベイズフィルタとも呼ばれる（フィルタとはこし器であり，更新式を繰り返すことで不要な値を削除し，正しい値を得るという意味である）．

2.2　シミュレーション環境

ここまでの話では，すべて $p(x)$ というような式を用いてきた．$p(x)$ とは x に関する確率分布であり，これは $y = f(x)$ という様な関数を考えていることと同じである．すなわち，関数の形が定まらない限り，具体的にどのように計算されるか，もしくは計算できるかどうかなどはわからない．そこで，前述のロボットの位置推定問題に焦点を当て，実際に再帰的ベイズフィルタの実装を行う．本節では，そのためのシミュレーション条件を整理する．

図 2 に，想定するシミュレーションモデルの座標系の関係を示す．ロボットは 2 次元の xy 平面（世界座標）を動くものとし，その位置と姿勢をを $\mathbf{x} = (x\ y\ \theta)^\top$ と表す．ロボットには，ロボット座標上で見た xy

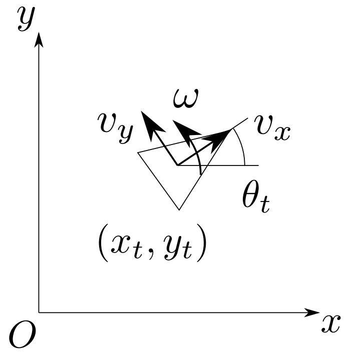

図 2　対象とする問題の座標系．

方向の並進の移動速度と，角速度 ω が与えられる．すなわち制御入力は $\mathbf{u} = (v_x \ v_y \ \omega)^\top$ である．このとき，制御入力に対するロボットの移動は，以下の様に記述できる．

$$\begin{pmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{\theta}_t \end{pmatrix} = \begin{pmatrix} \hat{x}_{t-1} \\ \hat{y}_{t-1} \\ \hat{\theta}_{t-1} \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \Delta t \tag{22}$$

ここで $\hat{\mathbf{x}} = (\hat{x}_t \ \hat{y}_t \ \hat{\theta}_t)^\top$ は，制御入力により更新（予測）されたロボットの位置であり，Δt は更新の時間間隔である．ただし，実際にロボットが与えられられた制御入力に従って動くことはなく，実際のロボットはある一定のノイズの影響を受けて移動するものとする．

$$\begin{pmatrix} x_t^{\mathrm{GT}} \\ y_t^{\mathrm{GT}} \\ \theta_t^{\mathrm{GT}} \end{pmatrix} = \begin{pmatrix} x_{t-1}^{\mathrm{GT}} \\ y_{t-1}^{\mathrm{GT}} \\ \theta_{t-1}^{\mathrm{GT}} \end{pmatrix} + \begin{pmatrix} (1+r_x)v_x \\ (1+r_y)v_y \\ (1+r_\omega)\omega \end{pmatrix} \Delta t \tag{23}$$

$\mathbf{x}^{\mathrm{GT}} = (x^{\mathrm{GT}} \ y^{\mathrm{GT}} \ \theta^{\mathrm{GT}})^\top$ はロボットの真の位置であり，r_x，r_y，r_ω は制御入力に加わるノイズの率を表すパラメータである．すなわち，時系列状態推定を行うための予測値 $\hat{\mathbf{x}}_t$ は，与えられた制御入力 $\mathbf{u}_t$ の通りに更新されるが，実際のロボットはこれにノイズが加えられた分移動するため，予測値と真値の誤差は時間経過とともに増大していく．

次に GNSS の測位について考える．GNSS は，ロボットの真値の位置を計測するセンサであると仮定する．ただし，GNSS の観測にも誤差が含まれると仮定する．

$$\begin{pmatrix} x_t^{\mathrm{GNSS}} \\ y_t^{\mathrm{GNSS}} \\ \theta_t^{\mathrm{GNSS}} \end{pmatrix} = \begin{pmatrix} \mathcal{N}(x_t^{\mathrm{GT}}, \sigma_x^2) \\ \mathcal{N}(y_t^{\mathrm{GT}}, \sigma_y^2) \\ \mathcal{N}(\theta_t^{\mathrm{GT}}, \sigma_\theta^2) \end{pmatrix} \tag{24}$$

ここで $\mathcal{N}(x, \sigma^2)$ は，平均 x，分散 σ^2 の正規乱数を発生させる関数である．以上のシミュレーションを基に，ある制御入力に対してノイズが加わりながら動くロボットの正しい位置を，GNSS を用いながら推定する時系列状態推定の実装を行う．

2.3　カルマンフィルタを用いた時系列状態推定の実装

上記の時系列状態推定を実装するために，再帰的ベイズフィルタが直接利用できるのではなく，再帰的ベイズフィルタはあくまで時系列状態推定のために一般形式である．これを実際に解くために，カルマンフィルタやパーティクルフィルタと呼ばれるベイズフィルタが用いられる．ここではまず，カルマンフィルタを用いた実装を述べる．なおカルマンフィルタを用いる場合，ロボットの位置に関する確率分布は，正規分布

で表現できると仮定する．すなわち，

$$p(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t, \Sigma_t) = \frac{1}{(\sqrt{2\pi})^3\sqrt{|\Sigma_t|}} \exp\{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_t)^\top \Sigma_t^{-1}(\mathbf{x}-\mathbf{x}_t)\} \tag{25}$$

とする．

まず，時刻 $t-1$ におけるロボット位置の分布 $p(\mathbf{x}t-1) = \mathcal{N}(\mathbf{x}_{t-1}, \Sigma_{t-1})$ が求められているとする．ここからまず，$\int p(\mathbf{x}_T|\mathbf{x}_{T-1}, \mathbf{u}_T)p(\mathbf{x}_{T-1}|\mathbf{u}_{1:T-1}, \mathbf{z}_{1:T-1})\mathrm{d}\mathbf{x}_{T-1}$ を計算する必要がある．詳細な導出は省略するが，ロボットの位置が正規分布により表現されると仮定した場合，この分布は以下のように計算できる．

$$\begin{aligned} \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1} &= \mathcal{N}(\hat{\mathbf{x}}_t, \hat{\Sigma}_t) \\ \hat{\mathbf{x}}_t &= A\mathbf{x}_{t-1} + B\mathbf{u}_t \\ \hat{\Sigma}_t &= A\Sigma_{t-1} + A^\top BRB^\top \end{aligned} \tag{26}$$

ここで，前述のロボットの運動に関する式から，A と B は 3 次の単位行列である．また R は，制御入力に加わると考えられる正規乱数の共分散行列である．

次に，これに尤度分布を掛け，事後分布に更新することを考える．カルマンフィルタを用いる場合，事後分布は以下のように計算される．

$$\begin{aligned} \eta p(\mathbf{z}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1} &= \mathcal{N}(\mathbf{x}_t, \Sigma_t) \\ \mathbf{x}_t &= \hat{\mathbf{x}}_t + K_t(\mathbf{z}_t - C\hat{\mathbf{x}}_t) \\ \Sigma_t &= (I - K_tC)\hat{\Sigma}_t \end{aligned} \tag{27}$$

ここで，C は観測行列であり，ロボット位置を GNSS の測位結果と同じ状態空間に変換する行列である．なおこれも前述のシミュレーションモデルから，3 次の単位行列となる．また，I も同様に 3 次の単位行列である．K_t はカルマンゲインと呼ばれ，以下のように計算される．

$$K_t = \hat{\Sigma}_t C^\top (C\hat{\Sigma}_t C^\top + Q) \tag{28}$$

ここで Q は，GNSS の測位結果に加わると考えられるノイズの共分散行列である．カルマンフィルタを用いる場合，このような行列計算により再帰的ベイズフィルタが実装される．

2.4 パーティクルフィルタを用いた時系列状態推定の実装

次に，パーティクルフィルタを用いた実装について述べる．パーティクルフィルタとは，パーティクルと呼ばれる状態の候補を表したサンプルセットの集合により，確率分布を近似する方法である．すなわち，ロボットの位置に関する確率分布を

$$p(\mathbf{x}_t) \simeq \frac{1}{M}\sum_{i=1}^{M} p_t^{[i]}\delta(\mathbf{x} - \mathbf{x}_t^{[i]}) \tag{29}$$

と表す．ここで，M はパーティクル数，$\mathbf{x}_t^{[i]}$ と $p_t^{[i]}$ は，i 番目のパーティクルの状態と重み，$\delta(\cdot)$ はディラックのデルタ関数であり，括弧内が 0 の時に 1，そうでない場合に 0 となる．なおパーティクルは，重みは総和が 1 となる．

カルマンフィルタの実装で述べた様に，まず時刻 $t-1$ の位置に関する分布が求められているとし，すでに $p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1})$ を近似するパーティクル群が得られているとする．このとき

$\int p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1},\mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1}$ は，以下のようにパーティクル群の位置を更新することで近似できる．

$$
\begin{aligned}
&\int p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1},\mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1} \simeq \frac{1}{M}\sum_{i=1}^{M} p_t^{[i]}\delta(\mathbf{x}-\hat{\mathbf{x}}_t^{[i]}) \\
&\begin{pmatrix} \hat{x}_t^{[i]} \\ \hat{y}_t^{[i]} \\ \hat{\theta}_t^{[i]} \end{pmatrix} = \begin{pmatrix} x_{t-1}^{[i]} \\ y_{t-1}^{[i]} \\ \theta_{t-1}^{[i]} \end{pmatrix} + \begin{pmatrix} \mathcal{N}(v_x, \alpha_1 v_x^2 + \alpha_2 v_y^2 + \alpha_3\omega^2) \\ \mathcal{N}(v_y, \alpha_4 v_x^2 + \alpha_5 v_y^2 + \alpha_6\omega^2) \\ \mathcal{N}(\omega, \alpha_7 v_x^2 + \alpha_8 v_y^2 + \alpha_9\omega^2) \end{pmatrix} \Delta t
\end{aligned} \tag{30}
$$

ここで α_1-α_9 は，非負の任意定数であり，ある制御入力に対してパーティクル群がどれだけ広がるかを調整するパラメータであり，移動に対する不確かさを表現するパラメータとなる．

次に，各パーティクルの重みを観測モデルを用いて定める．すなわち，

$$
p_t^{[i]} = p(\mathbf{z}_t|\hat{\mathbf{x}}_t) \tag{31}
$$

とする．各パーティクルに対して観測モデルを用いた重み（尤度）の計算を終えると，そのパーティクル群が $\eta p(\mathbf{z}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1},\mathbf{z}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1}$ を近似することになる．なお，GNSS の測位に加わるノイズが正規分布に従うと仮定する場合，観測モデルは以下のように実装される．

$$
p(\mathbf{z}_t|\hat{\mathbf{x}}_t^{[i]}) = \frac{1}{(\sqrt{2\pi})^3\sqrt{|Q|}}\exp\{-\frac{1}{2}(\mathbf{z}_t-\hat{\mathbf{x}}_t^{[i]})^\top Q^{-1}(\mathbf{z}_t-\hat{\mathbf{x}}_t^{[i]})\} \tag{32}
$$

重み計算の後に，パーティクル群の重みを正規化し，パーティクル群の重み付き平均を計算する．

$$
\begin{aligned}
p_t^{[i]} &\leftarrow \frac{p_t^{[i]}}{\sum_{j=1}^{M} p_t^{[j]}} \\
\mathbf{x}_t &= \sum_{i=1}^{M} p_t^{[i]}\hat{\mathbf{x}}_t^{[i]}
\end{aligned} \tag{33}
$$

なおこの重み付き平均は，パーティクル群が近似する事後分布から，期待値を取り出す操作となっている．重み付き計算の後，必要に応じて，不要なパーティクルを削除し，有効なパーティクルの複製を作成するというリサンプリングを行う．この操作を繰り返すことで，パーティクルフィルタを用いた状態推定が実現される．

2.5　時系列状態推定結果の一例

図 3 に，カルマンフィルタを用いて移動軌跡の推定結果の例を示す．緑色の点が GNSS の測位結果である．GNSS の測位は確かに真値（Ground truth）周辺に存在するが，ノイズが含まれ正しく真値を追跡できてはいない．また黒線は制御入力を積分して推定した軌跡（Odometry）は，移動に伴い誤差が増大していることがわかる．これらに対して，カルマンフィルタを用いることで正しく真値を追跡できていることがわかる．なおパーティクルフィルタを用いた場合でも，同様の結果を得ることが可能である．

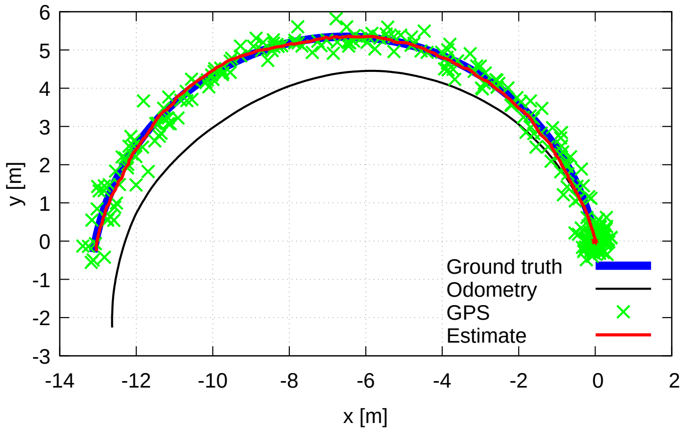

図 3　カルマンフィルタを用いた自己位置推定の結果.

3　確率論による状態推定の利点

3.1　最適化と最尤推定

最後に，最適化と最尤推定の関係に関して述べ，状態推定のために確率論を用いることの利点について述べる．今，あるパラメータを求める最小二乗問題について考えにあたり，以下の関数を考える．

$$y = f(\mathbf{x}; \boldsymbol{\theta}) \tag{34}$$

この関数は，あるパラメータベクトル $\boldsymbol{\theta}$ を持ち，ある変数 $\mathbf{x}$ を y に写像する（y はベクトルでも問題ないが，簡単のためにスカラーとする）．今観測されたデータ集合として，$X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N), Y = (y_1, y_2, ..., y_N)$ という N 個の対応する入出力データが得られているとする．このとき，パラメータベクトル $\boldsymbol{\theta}$ を求めるために，以下の最適化問題を考える．

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta} \in \Theta} \frac{1}{2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \boldsymbol{\theta}))^2 \tag{35}$$

Θ はパラメータベクトル $\boldsymbol{\theta}$ が取りうる値を表す集合である．$y - f(\mathbf{x}; \boldsymbol{\theta})$ は誤差となり，誤差の 2 乗和を最小にするパラメータベクトルを求めることができれば，それは与えられた入出力データ集合の関係性を最適に説明するパラメータベクトルとなる．

今ここで，誤差 e を以下のように定める．

$$e = y - f(\mathbf{x}; \boldsymbol{\theta}) + \epsilon \tag{36}$$

ここで ϵ は，平均ゼロ，分散 σ^2 従う正規分布から発生されるノイズとする．すなわち，得られた誤差には，正規分布に従うノイズが含まれるという関係を設ける．この誤差も考慮しながら，パラメータベクトル $\boldsymbol{\theta}$ を求めることを考える．

誤差 e には，正規分布に従うノイズが含まれると仮定している．そのため e に関する確率分布は，以下のように定義される．

$$p(e; \theta) = \mathcal{N}(e; 0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e^2}{2\sigma^2}\right) \tag{37}$$

この確率分布は，未知のパラメータベクトル $\boldsymbol{\theta}$ を固定した下で，誤差 e が得られる尤もらしさ表す確率であるため，$\boldsymbol{\theta}$ に関する尤度となる．

ここで，得られている入出力データの対応から，それぞれ誤差を定義し，その誤差の集合 $E = (e_1, e_2, ..., e_N)$ が同時に得られる同時確率を考える．この同時確率を考えるにあたり，それぞれの誤差は独立している（それぞれの誤差の発生に起因する要因が関係していない）と仮定する．するとこの同時分布は，以下のように因数分解できる．

$$p(E;\theta) = \prod_{i=1}^{N} p(e_i;\theta) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) \tag{38}$$

今，この尤度を最大化することを考える．すなわち，

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}\in\Theta} p(E;\boldsymbol{\theta}) \tag{39}$$

となるパラメータベクトルについて考える．この最大化は，最大の尤度を得られるパラメータを求めるため，最尤推定と呼ばれる．この最大化を行うにあたり，対数尤度 $\log p(E;\boldsymbol{\theta})$ について考える（log は自然対数である）．$\log p(E;\boldsymbol{\theta})$ は以下のように書き換えられる．

$$\log p(E;\boldsymbol{\theta}) = \sum_{i=1}^{N} -\frac{1}{\sqrt{2\pi}\sigma} \frac{e_i^2}{2\sigma^2} \tag{40}$$

ここで，尤度を最大化することと，負の対数尤度を最小化することは等しいので，

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}\in\Theta} p(E;\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}\in\Theta}(-\log p(E;\boldsymbol{\theta})) = \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \frac{e_i^2}{2\sigma^2} \tag{41}$$

となる．ここで，

$$\frac{1}{\sqrt{2\pi}\sigma} \frac{e_i^2}{2\sigma^2} \propto \frac{e_i^2}{2\sigma^2} \tag{42}$$

であるので，負の対数尤度を最小化するパラメータベクトルは，誤差の 2 乗和を最小にするパラメータベクトルと等しいことがわかる．すなわち，誤差が正規分布に従うノイズを含むと仮定した最尤推定は，最適化と同様の計算結果となる．特に，$\sigma^2 = 1$ と仮定した場合は，最適化と一致する．一見すると，最適化は誤差を最小にしているだけの操作に見えるが，正規分布に従う誤差が含まれると仮定した際の最尤推定と一致するのである．

3.2 最適解とベイズ推定の解の違い

最尤推定とは，観測データが得られる尤もらしいパラメータを求めることである．時系列状態推定の観点からこれを見ると，これは尤度分布である観測モデル $p(\mathbf{z}_t|\mathbf{x}_t)$ を最大化する $\mathbf{x}_t$ を求めることである．もし，GNSS の測位が正規分布に従うノイズを含むと仮定すると，GNSS の測位結果 $\mathbf{z}_t$ が尤度を最大化する解となる．そのため，上述したシミュレーションにおいて最尤推定を用いることは，毎ステップ GNSS の測位を解として採用することと等価になる．一方で，図 3 に示すカルマンフィルタの計算結果からわかる様に，GNSS の測位は真値を正しく追従していない．しかし，カルマンフィルタによる推定結果は，GNSS の測位が含むノイズに対して頑健性を持つことがわかる．これは，カルマンフィルタが事後分布，すなわち $p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t})$ の期待値を求めているためである．すなわち，最尤推定（最適化）とベイズ推定により求められる解は異なり，一般にベイズ推定により求められる解は，観測が含むノイズの影響に対して頑健である．ただし，解の精度を比較した場合は，最適化により得られる解の方が精度が高くなることも多い．

3.3 観測の柔軟なモデル化

最後に，確率論を用いて状態推定を考える利点に関して，観測モデルの視点から述べる．一般に最適化は，ノイズに対して脆弱である．ノイズを含むデータに対して最適化を適用するにあたり，ロバスト推定と呼ばれる方法を利用することがある．ロバスト推定では，以下の目的関数の最小化を考える．

$$\frac{1}{2}\sum_{i=1}^{N}\rho(e_i)e_i^2 \tag{43}$$

ここで $\rho(e_i)$ は，誤差の大きさに依存して振る舞いを変える重み関数である．例えば，誤差が小さい場合は正しいデータであると考え，重みを大きくし，逆に誤差が大きい場合は，誤ったデータであると仮定し，重みを小さくするなどすると，最適化を行うにあたり，データに含まれるノイズの影響が小さくなる．ただし，実際にはこの様な安易な仮定のみでデータに含まれるノイズの影響を取り除くことはできない．

これに対して確率論を用いる場合，尤度関数の設計をより柔軟に行うことができる．例えば，尤度を以下のような線形結合で記述することもできる．

$$p(e;\boldsymbol{\theta}) = \sum_{i\in\mathcal{C}} c_i p_i(e;\boldsymbol{\theta}) \tag{44}$$

ここで $\mathcal{C}$ は，誤差が発生する要因の集合，c_i は i 番目の誤差要因が発生する割合，$p_i(e;\boldsymbol{\theta})$ は i 番目の誤差要因が発生するモデルである．各 $p_i(e;\boldsymbol{\theta})$ が確率分布としての制約を満たす，かつ $\sum_{i\in\mathcal{C}} c_i = 1$ を満たせば，$\sum_{i\in\mathcal{C}} c_i p_i(e;\boldsymbol{\theta})$ も確率分布となる．すなわちこの負の対数尤度の最小化が行えれば，様々な誤差の発生要因を考慮したパラメータベクトルの推定が可能になる．

ただし，通常この様な線形結合により表現された分布は，安易に最適化することはできない．しかし，再帰的ベイズフィルタの 1 種であるパーティクルフィルタを用いるなどすると，最適解を得ることはできないが，効率的にこれらの要因を考慮したフィルタリングが実装できる．すなわち確率論を用いることで，実環境で起こりうる不確かさを考慮した状態推定が可能になるのである．なお，自己位置推定に焦点を当て，ベイズ推定を用いてより高度な推定を行った取り組みを [2] にまとめている．

課題

- ロボットやその他の実世界で用いられるシステムを含め，ある制御入力に対して，確実にその状態の変化を記述できるモデルを構築することはできない．その理由を述べよ．
- カルマンフィルタでは，推定される状態が正規分布で表されると仮定して状態推定を行う．一方でパーティクルフィルタでは，パーティクル群により表されると仮定して状態推定を行う．それぞれの利点，欠点を述べよ．
  ヒント：確率分布の形状，推定される状態の次元数を考えよ．
- 式 (21) に示すように，ベイズフィルタは 1 時刻前の状態の分布を考える．一方で最適化，または最尤推定は，観測モデルの最大化のみを考えている．今，どのような状況でも確実な状態を観測できるセンサが使用できるとする（実際にそのようなものはない）．このとき，このセンサを用いて最適化を行うべきか，ベイズフィルタを適用するべきか，どちらが正しいといえるかの理由を述べよ．なお，推定すべき状態の遷移については確実に知ることはできないものとする．
  ヒント：どのような状況でも確実な状態を観測ができるセンサが持つ誤差の分布形状について考えよ．

参考文献

[1] S. Thrun, W. Burgard, and D. Fox. “Probabilistic Robotics,” *MIT Press*, 2005.

[2] 赤井直紀．“LiDAR を用いた高度自己位置推定システム: 移動ロボットのための自己位置推定の高性能化とその実装例”，コロナ社，2022.