

粒子フィルタにおける粒子の位置情報を用いた移動ロボットの誘導制御

Guidance Control of Mobile Robots by Localization of Particles in Particle Filter

○篠崎佑介*, 村松鋭一*

○Yusuke Shinozaki*, Eiichi Muramatsu*

*山形大学

*Yamagata University

キーワード : 粒子フィルタ (particle filter), 状態推定 (state estimation), 移動ロボット (mobile robot), 誘導制御 (guidance control)

連絡先 : 〒 992-8510 米沢市城南町 4-3-16 山形大学大学院理工学研究科機械システム工学専攻
村松鋭一, Tel.: (0238)26-3327, Fax.: (0238)26-3327, E-mail: muramatu@yz.yamagata-u.ac.jp

1. 研究の背景と目的

動的システムの状態推定法として粒子フィルタの研究が進んでいる。移動ロボットに粒子フィルタを適用するとき、その目的は自己位置推定であり、多数の粒子の分布はロボットの位置と姿勢に関する確率分布を近似するものとなる。各粒子はロボットの状態に関する尤度を持っており、それはロボットの位置と姿勢に関する情報を信頼度とともに数値化したものとなっている。

本研究では粒子フィルタの粒子をロボットの自己位置推定のみだけでなく、障害物回避に用いる方法を提案する。各粒子の尤度を利用し、障害物に対する危険度を算出し、それに応じたベクトルを用いてロボットを誘導する。

2. 移動ロボットの運動方程式

xy 平面上を移動するロボットの位置と姿勢を状態とし、 (x, y, θ) と表す。その時間微分 $(\dot{x}, \dot{y}, \dot{\theta})^T$ は、前方方向の速度 ν と角速度 ω を用いて

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \nu \cos \theta \\ \nu \sin \theta \\ \omega \end{pmatrix} \quad (1)$$

で記述される。これを刻み時間 Δt で離散化した方程式は、 $\omega = 0$ のとき、

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \nu_t \cos \theta_{t-1} \\ \nu_t \sin \theta_{t-1} \\ \omega_t \end{pmatrix} \Delta t \quad (2)$$

$\omega \neq 0$ のとき,

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \nu_t \omega_r^{-1} \{\sin(\theta_{t-1} + \omega_t \Delta t) - \sin \theta_{t-1}\} \\ \nu_t \omega_r^{-1} \{-\cos(\theta_{t-1} + \omega_t \Delta t) + \cos \theta_{t-1}\} \\ \omega_t \Delta t \end{pmatrix} \quad (3)$$

と表される.

3. ランドマーク

本研究での環境では, ロボットがマップに点在しているランドマークを認識することにより, 位置や角度を把握し, パーティクルに生かすことができる. 点ランドマークの座標は, $m_j = (m_{j,x}, m_{j,y})^T$, ロボットの姿勢 $x = (x, y, \theta)^T$ であり, ロボットの姿勢から, $z_j = (l_j, \phi_j)^T$ を求めていく. l_j は, ロボットとランドマークの距離であるため,

$$\begin{aligned} m_j &= |m_j - x|^2 \\ &= \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \end{aligned} \quad (4)$$

となる. ϕ_j は, 2つの線分のなす角になるため,

$$\begin{aligned} \phi_j + \theta &= \text{atan2}(m_{j,x} - y, m_{j,x} - x) \\ \phi_j &= \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \end{aligned}$$

となる. 上記式をまとめると,

$$\begin{pmatrix} l_j \\ \phi_j \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix} \quad (5)$$

$$z_j = h(x, m_j) \quad (6)$$

$$h(x, m_j) = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix} \quad (7)$$

といった関数で表せる. ランドマーク m_j を特化した関数 h_j を考えると,

$$z_j = h_j(x) \quad (8)$$

と表せる.

4. パーティクルフィルタ

4.1 パーティクルフィルタについて

本研究では, ロボットの自己位置推定のためにパーティカルフィルタ³⁾を適応する. この手法では, ロボットの姿勢が示すパラメータ (位置 (x, y) , 方向 θ), による空間に, 確率的な重みを持たせた点を数多く分布させる. そして, その空間内の任意の場所にロボットが存在する確率をその領域内に存在するパーティクルの重みの合計で近似表現する. そのパーティクルの重みは, センサ情報が入力されると, ベイズの定理により, パーティクルの重みを変更される. この手法は, Monte Carlo localization (MCL) と呼ばれているものである.

4.2 状態推移モデル

本研究における雑音は, 4つの標準偏差をパラメータとして用いており, $\sigma_{\nu\nu} = 0.19, \sigma_{\nu\omega} = 0.00, \sigma_{\omega\nu} = 0.13, \sigma_{\omega\omega} = 0.20$ とし, それぞれ, 直線で生じる道のりのばらつきの標準偏差, 回転で生じる道のりのばらつきの標準偏差, 直線で生じる向きのばらつきの標準偏差, 回転で生じる向きのばらつきの標準偏差となっている. これにより, 雑音を考慮した速度, 角速度は,

$$\begin{pmatrix} \nu' \\ \omega' \end{pmatrix} = \begin{pmatrix} \nu \\ \omega \end{pmatrix} + \begin{pmatrix} \delta_{\nu\nu} \{\sqrt{|\nu| \Delta t} + \delta_{\nu\omega} \{\sqrt{|\omega| \Delta t}\}\} \\ \delta_{\omega\nu} \{\sqrt{|\nu| \Delta t} + \delta_{\omega\omega} \{\sqrt{|\omega| \Delta t}\}\} \end{pmatrix} \quad (9)$$

と表される. この際, パーティクルの分布は, 信念分布に近似していると言え,

$$\begin{aligned} P(x_t^* \in X) &= \int_{x \in X} b_t(x) dx \\ &\approx \frac{1}{N} \sum_{i=0}^{N-1} \delta(x_t^{(i)} \in X) \end{aligned} \quad (10)$$

となる.

4.3 センサ値の反映

パーティクルは, ロボットがランドマークを観測したときのセンサ値, $z_j = (l_j \phi_j)^T$, パーティクルの姿勢 $x^{(i)}, x^{(k)}$ によって尤もらしさを求め,

$$L_j(x|z) = \eta p_j(z|x) \quad (11)$$

となる. L_j は尤度関数であり, η は正規化定数である.

MCL では, それぞれのパーティクルに重みと呼ばれる変数 w_t をもたせる. MCL のパーティクルは姿勢と重みの 2 つの変数の組で,

$$\xi_t^{(i)} = (x_t^{(i)}, w_t^{(i)}) \quad (12)$$

とされ, 全パーティクルの合計は,

$$\sum_{i=0}^{N-1} w_t^{(i)} = 1 \quad (13)$$

となり, センサ値に重みを反映させると,

$$w_t^{(i)} = L_j(x_t^{(i)} | z_{j,t}) \hat{w}_t^{(i)} \quad (14)$$

となる. $\hat{w}_t^{(i)}$ はセンサ値を反映する前のパーティクルの重みである.

4.4 尤度関数

姿勢 x において, ランドマーク m_j を観測した時のセンサ値 $z_j = (l_j \phi_j)^T$ のばらつきが, 2次元ガウス分布に従うとして, ばらつきの共分散行列を x の関数として $Q_j(x)$ と表すと,

$$Q_j(x) = \begin{pmatrix} [l_j(x)\sigma_l]^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix} \quad (15)$$

と定義し, $l_j(x)$ は $h_j(x)$ の距離の成分, $l_j(x)\sigma_l, \sigma_\phi$ はそれぞれ l_j, ϕ_j に期待されるばらつきの標準偏差である. また l_j, ϕ_j の雑音はお互いに独立と仮定する. $l_j(x)\sigma_l, \sigma_\phi$ の値はどのランドマークでも共通であるとする. 尤度関数は, ガウス分布を使用し,

$$L_j(x|z_j) = N[z = z_j | h_j(x), Q_j(x)] \quad (16)$$

と定義される. 本研究では, $l_j(x)\sigma_l, \sigma_\phi$ の値は $\sigma_l = 0.14, \sigma_\phi = 0.05$ とする.

4.5 リサンプリング

リサンプリングとは, 重みの大きいパーティクルに偏ってしまった現象を重みがあまりに小さすぎるパーティクルを消して, 重みが大きいパーティクルを分割することで, 重みが集中している領域のパーティクルの数を多く保つ処理のことである. 本研究では, 系統リサンプリングを採用している. 系統リサンプリング³⁾ は, パーティクルの重みを積み上げ, その先頭から重みの累積値が r のところに目安をつける. ($r \sim U(0, W/N')$) そして, 以下の処理を N 回繰り返す. 目安をつけていたリストの要素を選び, その要素となったパーティクルを選び, 新しいリストに追加 (重みは $1/N'$) し, r に W/N' を足す. これを採用することで, 増えやすい計算量を比較的抑えめにし, サンプリングバイアスを考慮することができる.

5. ロボット制御

5.1 ロボット制御について

本研究では, 離散時間において (2), (3) で示した通りに動作を行っており, 動作を加えなければ, 雑音を考慮して速度と角速度の設定通りにほぼ直進し続ける. つまり, 制御しなければ, 障害物があった際, 回避することができない. そこで, 本研究では, パーティクルそれぞれに対し, 危険度 D_j を算出させ, その値をもとに回避する方法を提案する.

5.2 危険度について

危険度 D を算出する方法は、リサンプリング前の尤度をすべて足し合わせて正規化し、その数値と障害物とパーティクルの距離 s_j の逆数を取り、それを足し合わせるにより算出される。(16) から、

$$\sum_{i=0}^j L_j(x|z_j) = w \quad (17)$$

$$D_j = \frac{L_j(x|z_j)}{w} + \frac{1}{s_j} \quad (18)$$

と算出される。

5.3 危険度による回避制御

危険度 D_j と障害物に対して一定の距離になったときに、回避を行うようにする。その際、危険度 D_j によって回避するベクトル方向を決定する。例えば、 $L_1(x|z_1) = 0.10, s_1 = 2.0$ であるとき、(18) から、 $D_1 = 0.60$ であると算出される。現在のロボットの進行方向を 0° とし、 D_j が 1.0 の時に 180° パーティクルの角度を変更すると、 $D_1 = 0.60$ であるため、この計算上では、 $s_1 \leq 2.0$ の時においてロボットの進行方向から 108° 変化する。この変化したパーティクルが存在した際、パーティクルとロボットのベクトル計算を行い、ロボットの進行方向は、以前の状態から、 54° 傾いた状態で回避運動を行う。

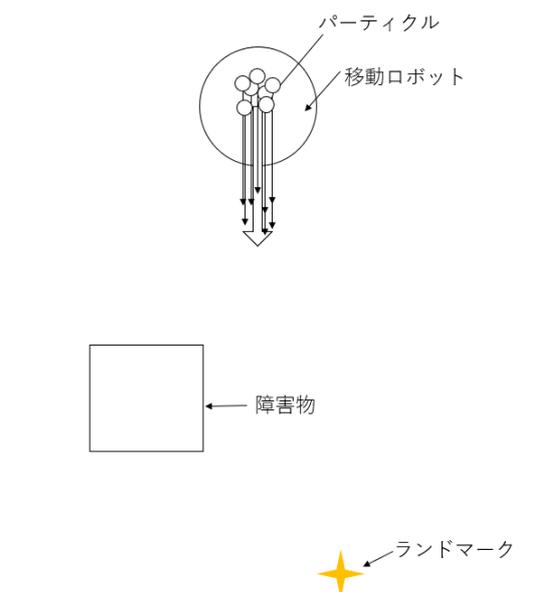


Fig. 1 回避前の移動ロボットと粒子の様子

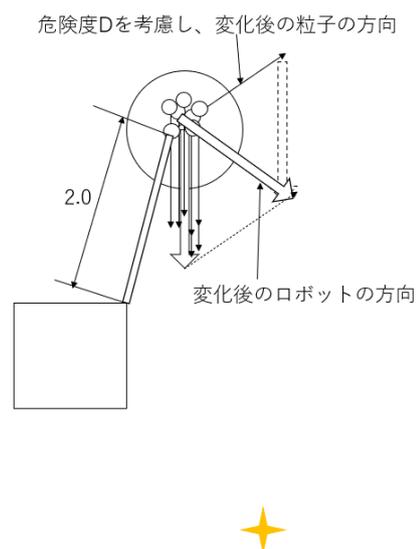


Fig. 2 危険度による変化の様子

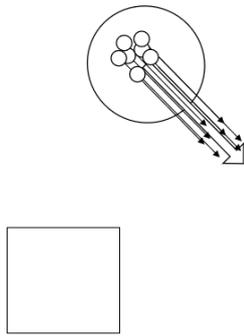


Fig. 3 回避後の移動ロボットと粒子の様子

6. シミュレーション

下図は,5.3での回避運動を採用した移動ロボットとパーティクルの様子である. なお, 図中上に描かれている (x, y, θ) は, 移動ロボットの座標と角度であり, 図中下に描かれている (x, y, θ) は, 最大尤度のパーティクルの座標と角度である.

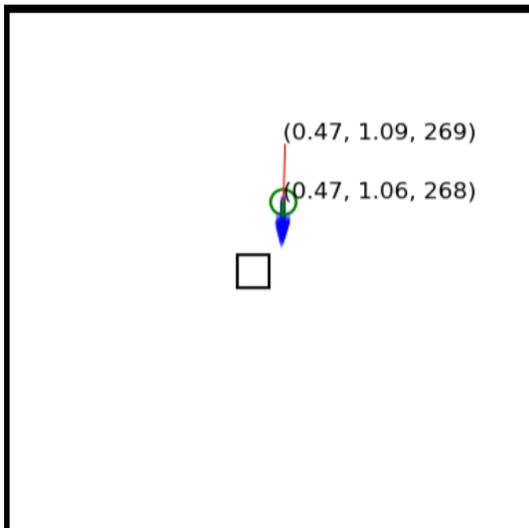


Fig. 4 回避前

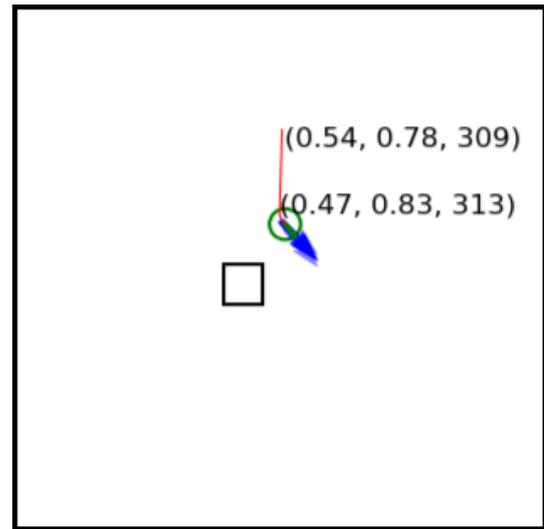


Fig. 5 回避後

以上から, 危険度 D_j を採用した回避運動を行うことができた.

7. まとめ

本研究では粒子フィルタの粒子をロボットの自己位置推定のみだけでなく, 障害物回避に用いる方法を提案した. そして, 各粒子の尤度を利用し, 障害物に対する危険度を算出し, それに応じたベクトルを用いてロボットを障害物から回避を行うことができた.

参考文献

- 1) 足立, 丸田: カルマンフィルタの基礎, 東京電機大学出版局 (2012)
- 2) 片山徹: 非線形カルマンフィルタ, 朝倉書店 (2011)
- 3) 上田隆一: 詳解確率ロボティクス, 講談社 (2019)
- 4) 上田隆一, 新井民夫, 浅沼和哉, 上田和昇, 大隈久: パーティクルフィルタを利用した自己位置推定に生じる致命的な推定誤りからの回復法, 日本ロボット学会誌 Vol.23 No.4 pp466 473, 2005

- 5) 小林亮、星野智史：移動ロボットのための環境変化にロバストなパーティクルフィルタに基づく自己位置推定の観測モデル、日本ロボット学会誌 Vol.41 No.1, pp.92-101, 2023