

卒業論文

地磁気方位センサと地図情報を用いるパーティクル フィルタによる屋外移動ロボットの自己姿勢推定結果 の修正

指導教官：黒木 秀一 教授

九州工業大学 工学部

機械知能工学科 知能制御工学コース

学籍番号：14104018

提出者氏名：遠藤 光

提出日：2018年2月14日

摘要

本研究では、移動ロボットが屋外で活動した後、得られた自己姿勢推定結果を、複数のセンサ情報を用いて修正を行った。移動ロボットには MobileRobots 社の PioneerP3-AT を使用し、Kinect センサにより得られる 3 次元 SLAM の自己姿勢推定結果を、地磁気方位センサの情報と地図情報を用いるパーティクルフィルタにより、修正を行うという手法を試みた。

これまで、我々の研究室では、オドメトリ情報よりダイナミクスを構成し、GPS と地磁気方位センサおよび Kinect センサを用いるパーティクルフィルタによる自己姿勢推定を行った。その結果、Kinect センサより、精度の高い自己姿勢推定結果を得た。しかし、GPS の測定誤差が大きいことや、Kinect センサを用いて得られた 3 次元データの高さ情報を無視して尤度計算を行うという問題があった。そこで、本研究の手法に取り組み、その結果、精度の高い、自己姿勢推定結果の修正を行うことができた。

目 次

1 序論	2
2 地磁気方位センサと地図情報を用いるパーティクルフィルタ (PF) による自己姿勢推定の修正	3
2.1 地図情報を用いる PF による自己姿勢推定	3
2.1.1 地図情報の作成	3
2.1.2 方位センサの信頼度	4
2.1.3 PF による自己姿勢推定アルゴリズム	5
3 実験	9
3.1 使用機器	9
3.2 実験方法	10
3.3 実験結果と考察	11
3.3.1 PF による推定結果と考察	11
4 結論	15
5 付録	18

1 序論

ロボットが屋外で活動するためには、正確な自己姿勢推定が必要である。我々の研究室ではこれまで、オドメトリ情報よりダイナミクスを構成し、GPS と地磁気方位センサおよび Kinect センサを用いるパーティクルフィルタによる自己姿勢推定を行った。その結果、Kinect センサより、精度の高い自己姿勢推定結果を得た。しかし、GPS の測定誤差が大きいことや、Kinect センサを用いて得られた 3 次元データの高さ情報を無視して尤度計算を行うという問題があった。

そこで、本研究では、Kinect センサにより得られる 3 次元 SLAM の自己姿勢推定結果を、地磁気方位センサの情報と地図情報を用いるパーティクルフィルタにより、修正を行うという手法を試みた。本稿では、2 章で自己姿勢推定結果を修正する手法を示し、3 章で実験結果を示すとともに考察を行い、4 章で結論を述べる。

2 地磁気方位センサと地図情報を用いるパーティクルフィルタ(PF)による自己姿勢推定の修正

2.1 地図情報を用いる PF による自己姿勢推定

2.1.1 地図情報の作成

本研究で用いる地図情報はグーグルマップを用いて作成する。グーグルマップより、実際に走行した道路の中央付近の複数点の緯度、経度を取得し、北を y 軸方向、東を x 軸方向とした世界座標へと変換する。時刻 k での北緯、東経を N_k, E_k とすると、東経の変換は

$$x_k = (E_k - E_0) \frac{\beta}{360} \cos\left(\frac{\pi N_0}{180}\right) \quad (1)$$

となる。ここで β は赤道長で $40075.017[\text{km}]$ である。同時に北緯は

$$y_k = (N_k - N_0) \frac{\pi \gamma}{360} \quad (2)$$

となる。ここで γ は北極と南極を結んだ直径で $12756.274[\text{km}]$ である。

取得した点 $\mathbf{m}_i = (x_i, y_i) (i = 1, 2, 3, 4, 5)$ を結び、地図情報 $R = \{x \mid x = \alpha(\mathbf{m}_{j+1} - \mathbf{m}_j) + \mathbf{m}_j\} (0 < \alpha \leq 1, j = 1, 2, 3, 4, \mathbf{m}_5 = \mathbf{m}_1)$ として使用する。作成した経路情報をグーグルマップに表示した図を図 1 に示す。



図 1: 地図情報

2.1.2 方位センサの信頼度

先行研究では GPS と地磁気方位センサおよび Kinect センサを用いる PF により自己姿勢推定を行い, Kinect センサより良い結果が得られることが分かった. 本研究では, Kinect センサのデータから得られた自己姿勢推定結果を地図情報と地磁気方位センサを用いて修正することを目的とする. 地磁気方位センサにより得られたデータを図 2 に示す. この図から一定時間おきにロボットがおよそ 90 度ずつ回転しており, ロボットが実際に移動した経路に沿ったデータを取得していることがわかる.

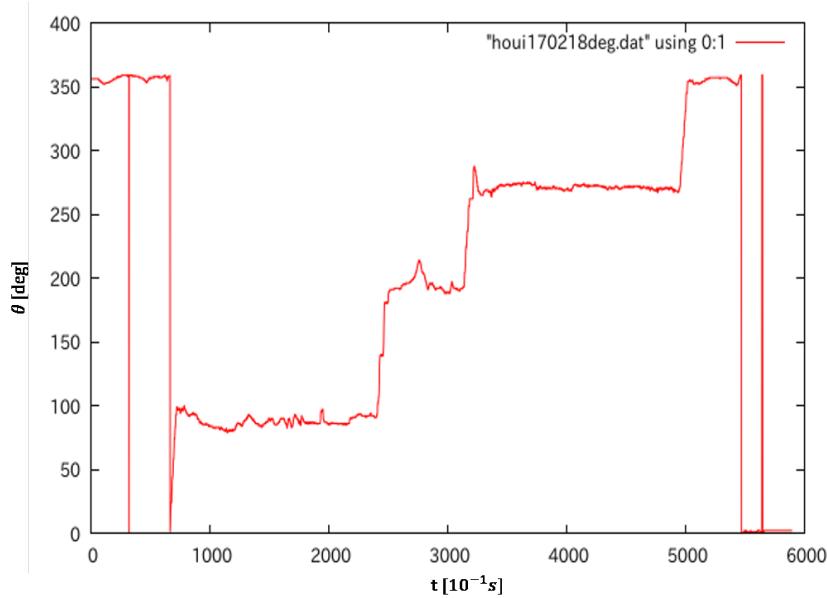


図 2: 地磁気方位センサのデータ

2.1.3 PFによる自己姿勢推定アルゴリズム

パーティクルフィルタ(PF)とは確率密度分布を多数のサンプルによって近似する手法の一つである。状態空間中のパーティクルと呼ばれる多数の粒子により、ロボットの姿勢に関する仮説を複数立てることにより分布を近似し、それを追跡、時間更新するアルゴリズムである。本研究で実装するPFのアルゴリズムを以下に簡潔に示す。尚、本研究に合わせて我々の研究室で従来の自己姿勢推定に用いられていたものを改良しており、先行研究論文でも紹介されているものを一部参考にしている。^[1]

まず、時刻 k でのロボットの姿勢を $\mathbf{x}_k = (x_k, y_k, z_k, \theta_k)^T$ とし、その時刻の姿勢についての各パーティクルの分布を $r(x_k)$ とする。このとき、パーティクル群 $\langle X_k^{[m]} \rangle_{m=1}^M$ による近似は $\langle X_k^{[m]} \rangle_{m=1}^M = \langle \mathbf{x}_k^{[m]}, w_k^{[m]} \rangle_{m=1}^M$ のように与えられているものとする。ここで、 M はパーティクル数であり、各々の姿勢 $x_k^{[m]}$ とその重み(尤度) $w_k^{[m]}$ をもつ複数のパーティクルにより、パーティクル群が構成される。ここで上添字 $[m] (m = 1, 2, \dots, M)$ はパーティクルのインデックスを表しており、各重み $w_k^{[m]}$ は $0 \leq w_k^{[m]} \leq 1$, $\sum_{m=1}^M w_k^{[m]} = 1$ を満たすものとする。

PFは以下の3ステップの処理を行う。ただし、時刻0における初期状態についてはすべてのパーティクルに対し、 $\langle X_0^{[m]} \rangle_{m=1}^M = \langle \mathbf{x}_0^{[m]}, w_0^{[m]} \rangle_{m=1}^M = \langle (0, 0, 0, 0)^T, \frac{1}{M} \rangle_{m=1}^M$ と与えておくものとする。

Step 1: 姿勢推定

ロボットの位置 $\mathbf{x}_k = (x_k, y_k, z_k)^T$ と時刻 T_k が与えられているとすると、 \mathbf{x}_k から \mathbf{x}_{k+1} に行く速度は次のように推定される。

$$\mathbf{v}_k = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T_{k+1} - T_k} = \frac{\Delta \mathbf{x}_k}{\Delta T_k} \quad (3)$$

よって、位置の時間変化は

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k \Delta T_k \quad (4)$$

と推定できる。また、 θ_k は図3に示した角度とした。

次に、第 i パーティクルの位置は $\hat{\mathbf{x}}_{k+1}^{(i)} = \hat{\mathbf{x}}_k^{(i)} + \Delta T_t (\mathbf{v}_k + \boldsymbol{\varepsilon}_{\mathcal{S}_k}^{(i)})$ により生成させる。ただし、 $\boldsymbol{\varepsilon}_{\mathcal{S}_k}^{(i)}$ は進行方向 \mathbf{v}_k とその垂直方向で異なるガウス分布 $N(0, \sigma_{\mathbf{v}}^2)$ と $N(0, \sigma_{\mathbf{v}^\perp}^2)$ を持つよう生成させる。

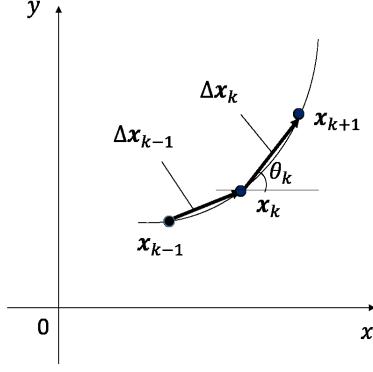


図 3: 速度の導出

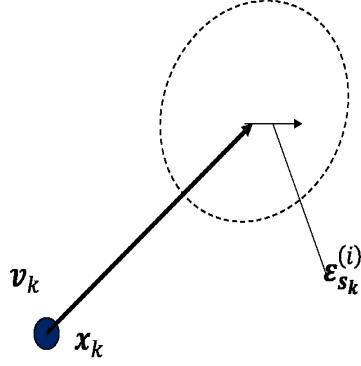


図 4: パーティクルの生成

Step 2: 重み (尤度) の計算

道路情報として, $R = \{x \mid x = \alpha(\mathbf{m}_{i+1} - \mathbf{m}_i) + \mathbf{m}_i\} (0 < \alpha \leq 1, i = 1, 2, 3, 4)$ があるとき, $\hat{x}_k^{(i)} = x_r$ とすると, $\mathbf{m}_{i+1} - \mathbf{m}_i$ と $x_r - \mathbf{m}_i$ の角度 θ は, $\cos \theta = \frac{(\mathbf{m}_{i+1} - \mathbf{m}_i) \cdot (\mathbf{x}_r - \mathbf{m}_i)}{|\mathbf{m}_{i+1} - \mathbf{m}_i| |\mathbf{x}_r - \mathbf{m}_i|}$ ($x_r \neq \mathbf{m}_i$) より, 求められる. したがって, ロボットとの距離 d_k は,

$$d_k = \begin{cases} |\mathbf{x}_r - \mathbf{m}_i| & \textcircled{1} (\cos \theta < 0) \\ |\mathbf{x}_r - \mathbf{m}_i| |\sin \theta| & \textcircled{2} (0 \leq |\mathbf{x}_r - \mathbf{m}_i| \cos \theta < |\mathbf{m}_{i+1} - \mathbf{m}_i|) \\ |\mathbf{x}_r - \mathbf{m}_{i+1}| & \textcircled{3} (|\mathbf{x}_r - \mathbf{m}_i| \cos \theta \geq |\mathbf{m}_{i+1} - \mathbf{m}_i|) \end{cases} \quad (5)$$

と求められる. これを道路全てで計算し, 1番小さいものを採用する. これより, 地図情報の尤度を

$$w_{1,k}^{[m]} = \begin{cases} 1 & (d_k \leq W_{road}) \\ (\exp(-\frac{(d_k - W_{road})^2}{A}) + 0.1)/1.1 & (d_k > W_{road}) \end{cases} \quad (6)$$

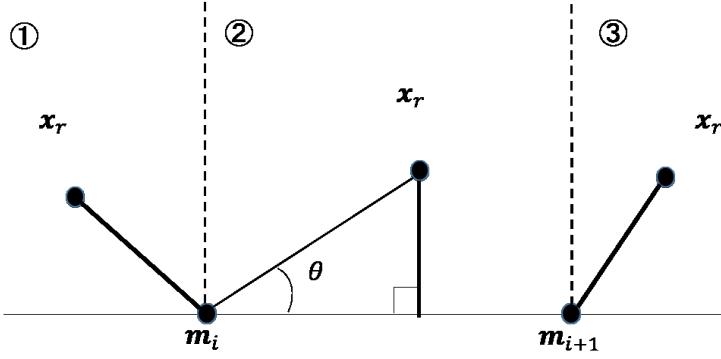


図 5: 口ポットとの距離

と求める。ただし A は調整すべきパラメータ, W_{road} は道幅である。また, 方位センサの計測値を θ_k とする。方位センサの計測値と推定値との尤度を

$$w_{2,k}^{[m]} = \exp\left(-\frac{\theta_k^{[m]} - \theta_k)^2}{B}\right) \quad (7)$$

と求める。ここで, B は調整すべきパラメータである。これらの尤度を使い正規化前のパーティクルの尤度を

$$w_{3,k}^{[m]} = C w_{1,k}^{[m]} + (1 - C) w_{2,k}^{[m]} \quad (8)$$

とする。ここで, C は調整すべきパラメータである。式 (8) を正規化することで各パーティクルの尤度を次のように求める。

$$w_k^{[m]} = \frac{w_{3,k}^{[m]}}{\sum_{m=1}^M w_{3,k}^{[m]}} \quad (9)$$

Step 3: リサンプリング

最後に尤度計算によって各パーティクルの分布が偏ってしまった場合リサンプリングを行う。本研究ではリサンプリングを行う条件として次式に示す有効サンプルサイズ (ESS:Effective Sample Size) を利用しこの値がパーティクル数の半分以下になったときのみリサンプリングを行う。

$$N_{\text{ess}} = \left(\sum_m (w_k^{[m]})^2 \right)^{-1} \quad (10)$$

リサンプリングは仮のパーティクル集合 $\langle \tilde{\mathbf{X}}_k^{[m]} \rangle_{m=1}^M$ の中の状態変数の各要素 $\tilde{x}_k^{[m]}$ を各パーティクルの尤度 $\tilde{w}_k^{[m]}$ に比例した確率で復元抽出を行う。

$$\mathbf{x}_k^{[m]} \sim \begin{cases} \tilde{x}_k^{[1]} & \text{with prob. } \propto \tilde{w}_k^{[1]} \\ \tilde{x}_k^{[2]} & \text{with prob. } \propto \tilde{w}_k^{[2]} \\ \vdots \\ \tilde{x}_k^{[M]} & \text{with prob. } \propto \tilde{w}_k^{[M]} \end{cases} \quad (11)$$

$$w_k^{[m]} := \frac{1}{M} \quad (12)$$

$\tilde{w}_k^{[m]}$ が大きいほど抽出される確率は高くなるので、リサンプリング後の集合には重複するパーティクルが多く含まれている。以上の手順により時刻 k の PF の分布を近似するパーティクル群が得られる。

$$\langle \mathbf{X}_k^{[m]} \rangle_{m=1}^M = \langle \mathbf{x}_k^{[m]}, w_k^{[m]} \rangle_{m=1}^M \quad (13)$$

以上の 3 ステップの処理を行うことにより、各時刻における目標分布、すなわち時刻 k の自己姿勢を次式によって推定できる。

$$\bar{\mathbf{x}}_k = \begin{pmatrix} \bar{x}_k \\ \bar{y}_k \\ \bar{z}_k \\ \bar{\theta}_k \end{pmatrix} = \sum_{m=1}^M w_k^{[m]} \mathbf{x}_k^{[m]} \quad (14)$$

3 実験

3.1 使用機器

本研究は、移動ロボットに MobileRobots 社の P3-AT 図 6、地磁気方位センサに Honeywell 社の HMC6352、RGB-D カメラ兼深度センサに Xbox One Kinect センサ図 7 を用いる。Kinect センサの仕様について表 1 に示す。

表 1: Kinect センサの仕様

色	解像度	1920 × 1080
	フレームレート	30[fps]
	フレームレート(暗所)	15[fps]
深度	測定範囲	0.5 ~ 8.0[m]
	解像度	512 × 424
	深度フレームレート	30[fps]
水平角度	70[deg]	
垂直角度	60[deg]	



図 6: P3-AT



図 7: Kinect センサ

3.2 実験方法

実験は九州工業大学教育研究 3 号棟周囲を P3-AT をジョイスティックによる手動操作で走行させ，地磁気方位センサと Kinect センサのデータを取得する。Kinect センサから得られるデータより，自己姿勢推定を行う。また，Kinect センサから得られる各時刻での自己位置を PF による自己姿勢推定に用いるために，Kinect 直交座標を北を y 軸方向，東を x 軸方向とした，世界座標系へと変換する。走行経路を図 8 に示す。



図 8: 走行経路

3.3 実験結果と考察

3.3.1 PF による推定結果と考察

走行実験後, Kinect センサにより得られた 3 次元 SLAM とその 3 次元 SLAM の自己姿勢推定結果をパーティクルフィルタにより修正を行った結果を図 9 に示す. これより, 尤度計算に地図情報が用いられたものは, 3 次元 SLAM の自己姿勢推定結果を修正し, 地図データに近いところを推移していることがわかる. また, 尤度計算に地磁気方位センサしか用いられていないものは, うまく 3 次元 SLAM の自己姿勢推定結果を修正できていないことがわかる. 図 9 を xy 平面上に射影し, 真の経路とした地図情報の値と, 自己姿勢推定結果を PF により修正した値との, 平均誤差と最大誤差を示すと表のようになった. これより, 地図情報に関する尤度の係数を大きくした方がより精度の高い修正結果が得られることが分かる.

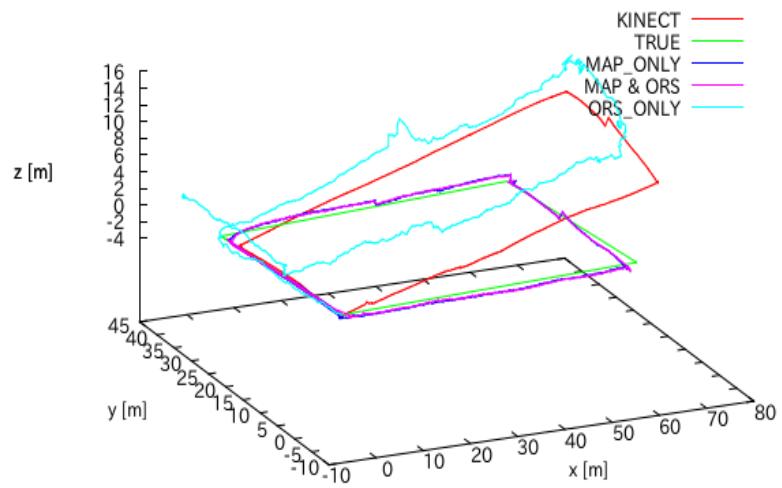


図 9: 3 次元 SLAM の自己姿勢推定結果の修正結果

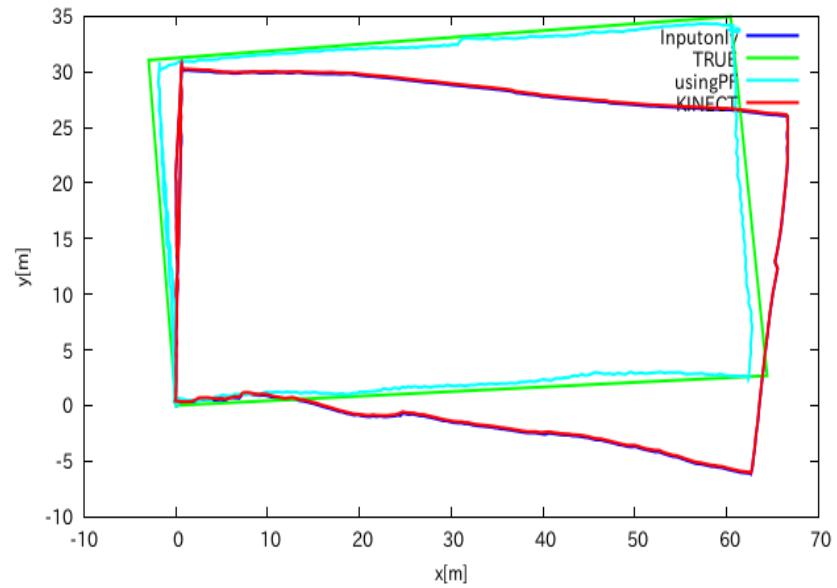


図 10: (a)MAP_ONLY($C=1$)

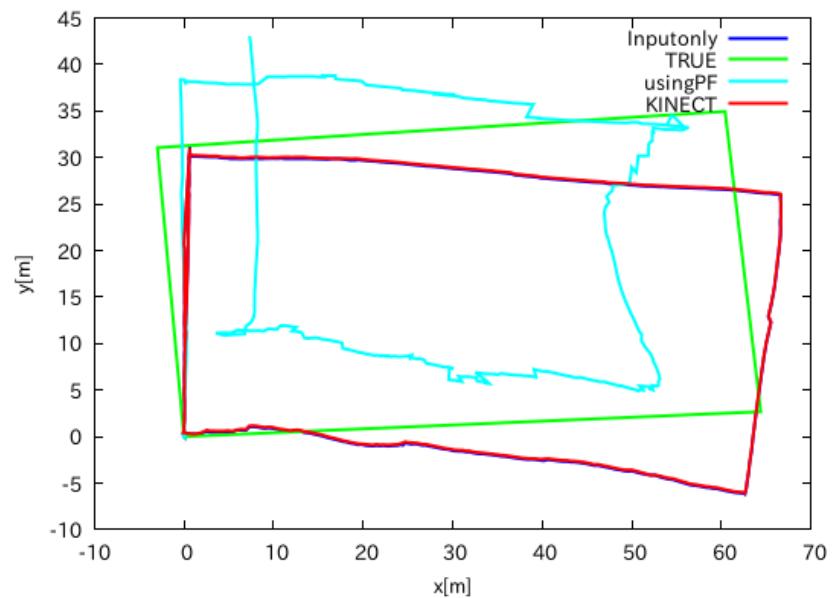


図 11: (b)ORS_ONLY($C=0$)

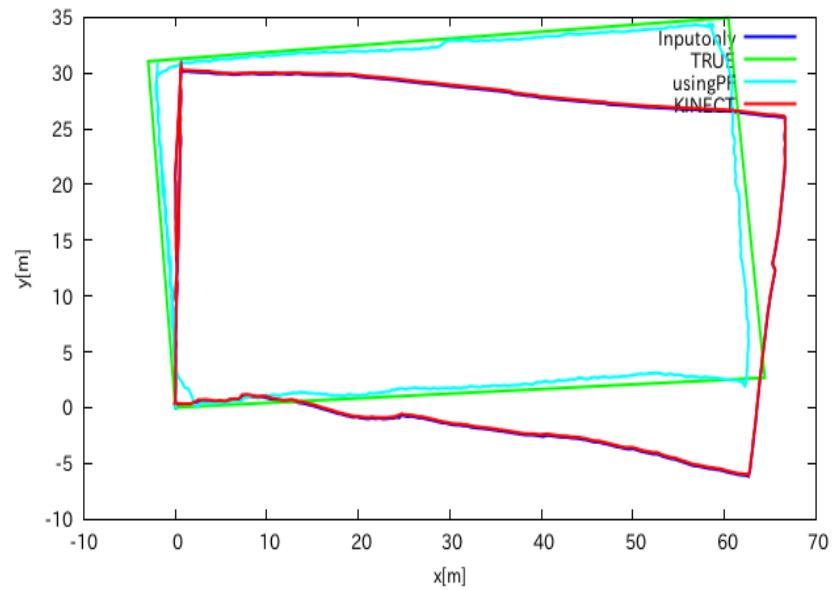


図 12: (c)MAP&ORS($C=0.5$)

表 2: 真値との誤差

	平均誤差 [m]	最大誤差 [m]
(a)MAP_ONLY	0.527	1.51
(b)ORS_ONLY	5.28	14.2
(c)MAP&ORS	0.561	1.68

4 結論

本実験で、地磁気方位センサと地図情報を用いたパーティクルフィルタによる3次元自己姿勢推定結果を修正する方法を提案し、その精度を比較した。その結果、尤度計算に地図情報のみを用いた修正結果では、平均誤差 0.527[m]、最大誤差 1.51[m] と精度の高い修正結果が得られた。また、地磁気方位センサは、3次元自己姿勢推定結果の修正にあまり役立たないことがわかった。

今後の課題として、PF に用いるための GPS の測定誤差を減少させる方法の検討、また、自己姿勢推定における、地磁気方位センサの有用な適用方法の検討があげられる。

参考文献

- [1] 前田拓耶 , ”GPS と地磁気方位センサ, 地図情報を用いるパーティクルフィルタによる屋外移動ロボットの自己姿勢推定” 九州工業大学卒業論文 , 2017 .
- [2] 一木爽 , ”GPS と地磁気方位センサおよび Kinect センサを用いるパーティクルフィルタによる屋外移動ロボットのためのオンライン SLAM” 九州工業大学卒業論文 , 2017 .
- [3] S.Thrun 他 , ” 確率ロボティクス ” , 毎日コミュニケーションズ , 2004.

謝辞

今回この論文の作成及び本研究を進めるに当たり，多大なる御協力，御指導を戴きました黒木秀一教授，松尾一矢助教，浅川和彦技術職員に深く感謝するとともに厚く御礼申し上げます.

また，いろいろと応援，協力，助言をしてくれた黒木研究室構成員一同にも重ねて感謝します．

5 付録

A ロボットの速度動作モデル

本論文で用いたロボットの速度動作モデルについて簡潔に述べる。時刻 k におけるロボットの姿勢を $\mathbf{x}_k = (x_k, y_k, \theta_k)$ 、入力を $\mathbf{u}_k = (v_k, \omega_k)^T$ とする。入力 \mathbf{u}_k にノイズが含まれない場合、ロボットの速度は (v_{k-1}, ω_{k-1}) で一定であり、ロボットは半径 $r_k - 1 = \left| \frac{v_{k-1}}{\omega_{k-1}} \right|$ の円弧上を移動する。このとき円弧の中心座標 (x_c, y_c) が

$$x_c = x_{k-1} - \frac{v_{k-1}}{\omega_{k-1}} \sin \theta_{k-1}, \quad y_c = y_{k-1} + \frac{v_{k-1}}{\omega_{k-1}} \cos \theta_{k-1} \quad (15)$$

であることを用いると、 Δt_{k-1} 後の姿勢 \mathbf{x}_k は $\omega_{k-1} \neq 0$ のとき

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} -\frac{v_{k-1}}{\omega_{k-1}} \sin \theta_{k-1} + \frac{v_{k-1}}{\omega_{k-1}} \sin (\theta_{k-1} + \omega_{k-1} \Delta t_{k-1}) \\ \frac{v_{k-1}}{\omega_{k-1}} \cos \theta_{k-1} - \frac{v_{k-1}}{\omega_{k-1}} \cos (\theta_{k-1} + \omega_{k-1} \Delta t_{k-1}) \\ \omega_{k-1} \Delta t_{k-1} \end{bmatrix} \quad (16)$$

と求められる。なお、この式は $\omega_{k-1} = 0$ のときも満たす。走路の起伏や斜面のある屋外環境下では実際の入力はこのモデルに加えノイズが含まれる場合がある。そのためノイズが加わる場合へと拡張して考えるために、平均がゼロで有限な分散に従う確率変数でモデル化を行う。したがって、このときの時刻 $k-1$ におけるロボットの制御入力は $\hat{\mathbf{u}}_{k-1} = (\hat{v}_{k-1}, \hat{\omega}_{k-1})^T = (v_{k-1}, \omega_{k-1})^T + (\varepsilon_{\alpha_1 v^2 + \alpha_2 \omega^2}, \varepsilon_{\alpha_3 v^2 + \alpha_4 \omega^2})^T$ と表せる。これを式 (16) に代入したものを本論文での速度動作モデルとして扱う。

B パーティクルフィルタの数学的導出

離散時刻 $k = 0, 1, 2, \dots$ における状態 \mathbf{x}_k , 入力 \mathbf{u}_k , 計測(観測)値 Z_k に対し , \mathbf{x}_k は Markov 性 , すなわち $p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ を満たし , さらにすべての k に対し , 観測列 Z_1, \dots, Z_k は状態列 $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k$ が既知の条件下で独立 (条件付独立: CI: conditionally independent) であると仮定する . 例えば , 状態推移モデル $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ と計測モデル $p(Z_k | \mathbf{x}_k)$ が , それぞれ ,

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \epsilon_k \quad (17)$$

$$Z_k = g(\mathbf{x}_k) + \delta_k \quad (18)$$

で与えられる制御系等が考えられる . ここでシステム関数 $f(\cdot)$ と計測関数 $g(\cdot)$ は決定論的 (deterministic) であり , システム誤差 ϵ_k と計測誤差 δ_k は確率論的 (stochastic) である . 式 (17) を状態推移モデル (システムモデル , 動作モデル , 運動学的モデル) , 式 (18) を計測モデルという .

PF で予測 (推定) すべき状態の列 $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k$ を $\mathbf{x}_{0:k}$ と表すと , その同時確率分布は以下のようになる . まず , 条件付確率の定義式に前向き予測のみにより状態推定すると仮定することにより , $p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}, Z_{1:k})$ についての漸化式 (再帰式)

$$p(\mathbf{x}_{0:k} | \mathbf{u}_{1:k}, Z_{1:k}) = p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}, Z_{1:k})p(\mathbf{x}_{0:k-1} | \mathbf{u}_{1:k-1}, Z_{1:k-1}) \quad (19)$$

を得る . さらに , この式の右辺の第 1 因子の \mathbf{x}_k の分布は , まず Markov 性により \mathbf{x}_k の分布を予測 (推定) し , Bayes の定理より計測値 Z_k を用いて更新 (修正) して求めることとすると ,

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}, Z_{1:k}) &= \frac{p(Z_k | \mathbf{x}_k, \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}, Z_{1:k-1})p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}, Z_{1:k-1})}{p(Z_k | \mathbf{x}_{0:k-1}, \mathbf{u}_{1:k}, Z_{1:k-1})} \\ &= \eta p(Z_k | \mathbf{x}_k)p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \end{aligned} \quad (20)$$

が得られる . 式 (20) を式 (19) に代入し , $p(\mathbf{x}_{0:k} | \mathbf{u}_{1:k}, Z_{1:k})$ を $r(\mathbf{x}_k)$ とおいて漸化式

$$r(\mathbf{x}_k) := \eta p(Z_k | \mathbf{x}_k)p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)r(\mathbf{x}_{k-1}) \quad (21)$$

が得られる . この式から時刻 $k = 1, 2, \dots$ に対して順次得られる $r(\mathbf{x}_k)$ を時刻 k における PF の目標分布 (target distribution) とする .

PF は , 前時刻で得られた目標分布 (またはその近似) から , ある提案分布 (proposal distribution) $q(\mathbf{x}_k)$ を粒子 (particle) の集合として予測生成し , 重み w_k により修正した $w_k q(\mathbf{x}_k)$ により各時刻の PF の目標分布 $p(\mathbf{x}_{0:k} | \mathbf{u}_{1:k}, Z_{1:k})$ を推定する . また明らかに

$$w_k = \frac{p(\mathbf{x}_{0:k} | \mathbf{u}_{1:k}, Z_{1:k})}{q(\mathbf{x}_k)} = \frac{\text{目標分布}}{\text{提案分布}} \quad (22)$$

とすればよいことがわかる。最も直接的な方法は、状態推移モデル $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)$ を用いて提案分布を生成し、計測モデル $p(\mathbf{Z}_k|\mathbf{x}_k)$ を用いた重みで修正する。すなわち

$$q(\mathbf{x}_k) := p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)q(\mathbf{x}_{k-1}) \quad (23)$$

$$w_k := \eta p(\mathbf{Z}_k|\mathbf{x}_k)w_{k-1} \quad (24)$$

として、初期条件を $q(\mathbf{x}_0) = p(\mathbf{x}_0)$ と $w_0 = 1/M$ として $k = 1, 2, \dots$ について、 $w_k q(\mathbf{x}_k)$ により目標分布を推定する場合、 $w_k q(\mathbf{x}_k) = p(\mathbf{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)[w_{k-1}q(\mathbf{x}_{k-1})]$ となるので、 $w_k q(\mathbf{x}_k)$ を正規化すれば、式(21)の目標分布と等価になる。なお、上記の重みは Bayes の定理の尤度 (likelihood) $p(\mathbf{Z}_k|\mathbf{x}_k)$ に相当するので、そう呼ぶ場合もある。

さらに、上式の $q(\mathbf{x}_k)$ と w_k を中間の分布 $\tilde{q}(\mathbf{x}_k)$ と重み \tilde{w}_k とし、 $\tilde{q}(\mathbf{x}_k)$ から重み \tilde{w}_k でリサンプリングして得た $q(\mathbf{x}_k) \sim \tilde{w}_k \tilde{q}(\mathbf{x}_k)$ を提案分布、新しい重みを $w_k = 1/M$ とすると、 $\sum w_k q(\mathbf{x}_k)$ で目標分布が推定できる。ただし、重み w_k の分布の偏りが小さいのにリサンプリングを行うと、モンテカルロ誤差が無用に混入することになり、かえって推定の精度が悪くなるので、リサンプリングは重み w_k の分布の偏りが大きい場合のみに行えばよい。