

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

小林 亮* 星野 智史*

Observation Model for Robust Localization of Mobile Robots based on Particle Filter in Changing Environments

Ryo Kobayashi* and Satoshi Hoshino*

For autonomous mobile robot navigation, localization is an essential capability. Given a mobile robot equipped with a 3D LiDAR sensor, an environment map composed of point cloud is built beforehand. The robot is thus allowed to localize the position in the map using the sensor scan data. However, the environment sometimes changes due to obstacles. Under the changing environment, the localization capability of the robot might be decreased. For this challenge, we propose a sensor observation model in a framework of particle filter based localization. In the observation model, we focus on the distance and distribution of point clouds of the map and sensor scan data. In the experiments, a mobile robot is moved by an operator in a virtual environment with obstacles. The robot based on the proposed observation model is able to localize the position in both the original and changing environments with the same accuracy. From the results, we finally show the robustness of the localization capability for changing environments.

Key Words: Autonomous Mobile Robots, Localization, Particle Filter, Observation Model

1. 緒 言

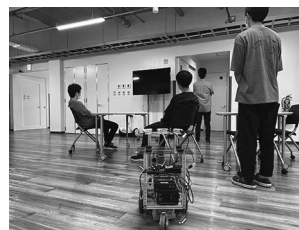
案内や運搬,そして警備ロボットには,目的地に向けた自律移動が求められる。これら車輪型のロボットが目的地に向け自律的に移動するためには,対象となる環境地図の構築と地図上での自己位置推定,ならびに目的地までの経路・動作計画が必要となる。本研究では,ロボットの自己位置推定に焦点を当てる。ロボットには自己位置推定のために,エンコーダーとLiDARが搭載されていることを前提とする。

一般的にロボットの自己位置推定は,事前に移動環境でSLAM [1][2]により構築した環境地図と,センサ観測で取得したスキャンデータを用いて行われる[3]。しかしながらロボットが移動する際,地図構築時には存在しなかった停止車両や歩行者などの出現に加え,移動環境のレイアウト変化によるランドマークの消失が考えられる。なお,本研究では地図構築時に環境に存在しない物体を障害物とし,時間的に位置が変化しない障害物を準動的障害物,時間的に位置が変化する障害物を動的障害物と定義する。Fig. 1に,環境変化の例を示す。

Fig. 1 (a)は,地図構築時の環境である。事前のLiDAR SLAMにより,移動環境の点群地図が構築される。そのため,壁に加え机や椅子といった物体の点群がランドマークとして自己位置推定に用いられることとなる。Fig. 1 (b)では,同じ環境に人が存在している。そのため,ロボットは地図に存在しない人のス



(a) SLAM environment



(b) Environment with people



(c) Environment without landmarks (desks and chairs)

Fig. 1 Example of changes in environment

キャンデータを観測することとなる。また,人による遮蔽で,後方のランドマークを観測することができない。Fig. 1 (c)では,レイアウトの変化として,机や椅子がなくなっている。そのため,ロボットはランドマークの一部を観測することができない。Fig. 1 (a)の地図構築時から変化したFig. 1 (b)やFig. 1 (c)の環境では,ロボットにとって環境地図とセンサ観測で取得したスキャンデータの対応をとることが困難となり,自己位置の推定精度が低下してしまう恐れがある。

原稿受付 2022年3月3日

*宇都宮大学

*Utsunomiya University

■ 本論文は有用性(要素分野)で評価されました。

上述の問題に対して、自己位置推定そのもののロバスト性を向上させるアプローチ [4] [5] と、自己位置推定の誤りを検知するアプローチ [6] [7] が提案されている。本研究では前者のアプローチに着目し、環境変化に対してロバストな自己位置推定を目的とする。なお、環境変化に対して、地図構築時の環境と同程度の精度で自己位置が推定できることをロバストと定義する。

自己位置推定のロバスト性を向上させるためには、準動的障害物や動的障害物の出現とランドマークの消失といった環境変化を想定したうえで、環境地図とセンサ観測で取得したスキャンデータに共通するランドマークを対応付ける必要がある。そこで本研究では、パーティクルフィルタ [8] [9] を用いた自己位置推定において、環境地図とスキャンデータそれぞれの距離情報と分布情報に基づく観測モデルを提案する。

本論文の構成は、以下のとおりである。第 2 章では、関連研究に言及したうえで、環境変化にロバストな自己位置推定の課題、ならびに課題に対するアプローチについて述べる。第 3 章では、パーティクルフィルタを用いた自己位置推定について述べる。第 4 章では、提案する観測モデルについて述べる。第 5 章では、提案する観測モデルを用いた自己位置推定について述べる。第 6 章では、シミュレーション実験を行い、自己位置推定のロバスト性の向上に対する提案手法の有効性を示す。第 8 章では、本論文の結論について述べる。

2. 環境変化にロバストな自己位置推定に関する研究

従来研究では、動的障害物に対するスキャンデータを外れ値として扱った自己位置推定法が提案されている [4] [5]。これらの手法では、スキャンデータに対する動特性を定量化し、動特性の高いスキャンデータを除去している。ランドマーク以外のスキャンデータを外れ値として除去することで、環境変化に対する自己位置推定のロバスト性を向上させている。しかしながら、動的障害物の出現に起因する環境変化のみに焦点をあてており、準動的障害物の出現やランドマークの消失は想定していない。

準動的障害物の出現やランドマークの消失に対しては、オフライン SLAM で構築した環境地図とオンライン SLAM で構築した環境地図を併用した自己位置推定法が提案されている [10] [11]。これらの手法では、オフライン SLAM により事前に構築したものと別に、環境変化をオンラインで反映した地図を構築する。二つの環境地図は、ロボットの存在確率に応じて切り替えながら使用される。筆者らは、Octree によるスキャンデータの差分検出に基づき二つの環境地図を切り替える自己位置推定法を提案した [12]。これらの従来手法により、環境変化に対する自己位置推定のロバスト性を向上させている。しかしながら、動的障害物に起因してオンラインでの地図構築に失敗すると、自己位置推定も困難となってしまふ。そこで、オフラインの環境地図をオンラインで更新しながら自己位置推定を行う手法が提案されている [13] [14]。これらの手法では、隠れマルコフモデルとマルコフ連鎖を用いて、環境変化を地図に反映させている。しかしながら、動的障害物に起因して自己位置推定に失敗すると、誤った自己位置に基づき地図が更新されることとなり、それ以降の自己位置推定も誤ったものとなってしまふ。

一方で、環境変化に対する自由空間観測モデルに基づく自己位置推定法も提案されている [15] [16]。これらの手法では、障害物の遮蔽によって測定距離が短くなることに着目し、測定距離が短いスキャンデータほどロボットの自己位置の尤度が小さく

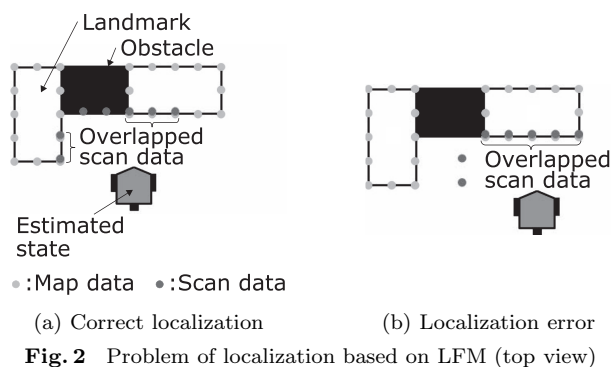


Fig. 2 Problem of localization based on LFM (top view)

なる。これにより、オンラインでの環境地図の構築やオフライン地図の更新を行うことなく、環境変化に対する自己位置推定のロバスト性を向上させている。しかしながら、障害物よりもランドマークがロボットに近いと、観測モデルにおける前提が成り立たず、この場合も自己位置の尤度が小さくなってしまふ。

上述の問題に対して、環境地図上での観測物体の有無を考慮した自己位置推定法が提案されている [17]。この手法では、環境地図上における観測物体の有無を known クラスと unknown クラスとして定義し、それぞれのクラスに対応した観測モデルが用いられている。これにより、ランドマークの位置に影響を受けることなく環境変化にロバストな自己位置推定を達成している。しかしながら、unknown クラスは known クラスに基づき算出され、さらに known クラスの算出には LFM (Likelihood Field Model) [18] が用いられている。LFM では、スキャンデータの各点に対する尤度を計算する。そのため、スキャンデータどうしの関係性を考慮することができず、Fig. 2 に示す自己位置推定の問題が生じる。

Fig. 2 (a) では、二つのランドマークの間に障害物が出現している。そのため、ロボットはランドマークと障害物のスキャンデータを得ることとなる。環境地図とセンサ観測で取得したスキャンデータに共通するランドマークの対応付けができた際には、ロボットは正しく自己位置を推定することができる。しかしながら、Fig. 2 (b) の異なる位置においても、環境地図と重なるスキャンデータの数が Fig. 2 (a) の位置と同じであることから尤度が等しくなり、ロボットは自己位置を誤推定してしまふこととなる。

以上のことから、自己位置推定のロバスト性を向上させるためには、まず、準動的障害物と動的障害物の出現、およびランドマークの消失といったすべての環境変化を想定する必要がある。その際、環境地図とスキャンデータに共通するランドマークを、スキャンデータどうしの関係性を考慮しながら対応付けなくてはならない。そこで本研究では、パーティクルフィルタを用いた自己位置推定において、スキャンデータの距離情報に加え分布情報に着目し、環境地図とスキャンデータそれぞれの距離情報と分布情報に基づく観測モデルを提案する。なお、スキャンデータの分布情報を取得するために、NDT (Normal Distributions Transform) [19] を用いる。

3. パーティクルフィルタを用いた自己位置推定

本研究では、ロボットの自己位置推定のためパーティクルフィルタを適用する。この手法では、パーティクルの集合により、ロボットの存在確率分布を求める。そして、求めた存在確率分布に基づきロボットの状態を算出し、これを自己位置とする。な

お、ロボットの状態とは位置姿勢のことである。時刻 t におけるロボットの状態 \mathbf{x}_t を算出するために、以下にパーティクルの集合 X_t を定義する。

$$X_t \equiv \left\{ \mathbf{x}_t^1, \mathbf{x}_t^2, \dots, \mathbf{x}_t^i, \dots, \mathbf{x}_t^I \right\} \quad (1)$$

\mathbf{x}_t^i は時刻 t における i 番目のパーティクルの状態、 I はパーティクルの総数である。

各パーティクルには、ロボットの存在確率分布を求めるために、重み w_t^i が定義されている。 i 番目のパーティクルの重み w_t^i は、当該パーティクルの存在確率によって以下のように定義される。

$$w_t^i \equiv p\left(\mathbf{x}_t^i | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{m}\right) \quad (2)$$

$1:t$ は、時系列データを表す。 \mathbf{u} はロボットへの制御入力、 \mathbf{z} はセンサ観測で取得したスキャンデータ、 \mathbf{m} は環境地図である。

式 (2) により、ロボットの存在確率分布をパーティクルの集合 X_t で表現することが可能となる。そこで、式 (2) の右辺にベイズの定理を適用すると、以下の式が得られる。

$$w_t^i = \eta p\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{m}\right) p\left(\mathbf{x}_t^i | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{m}\right) \quad (3)$$

η は、正規化記号である。

式 (3) の右辺について、一連の自己位置推定がマルコフ過程に基づくものとする、時刻 t での観測データは、それまでの制御入力と観測データに依存することがないため、 \mathbf{z}_t は $\mathbf{u}_{1:t-1}$ と $\mathbf{z}_{1:t-1}$ に対して独立であると見なすことができる。そのため、第 2 因子は $p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{m}) = p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ となる。これを条件付き独立性という。そして第 3 因子では、全確率の定理を適用することで展開することができ、以下の式が得られる。

$$w_t^i = \eta p\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}\right) \int p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{m}\right) \times p\left(\mathbf{x}_{t-1}^i | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, \mathbf{m}\right) d\mathbf{x}_{t-1}^i \quad (4)$$

式 (4) の右辺について、被積分関数の第 1 因子では、条件付き独立性により $\mathbf{u}_{1:t-1}$ と $\mathbf{z}_{1:t-1}$ 、および \mathbf{m} を除去することができる。被積分関数の第 2 因子は時刻 $t-1$ における条件付き確率であるため、 \mathbf{u}_t を除去することができる。そのため、以下の式が得られる。

$$w_t^i = \eta p\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}\right) \times \int p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_t\right) p\left(\mathbf{x}_{t-1}^i | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, \mathbf{m}\right) d\mathbf{x}_{t-1}^i \quad (5)$$

$p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ は観測モデル、 $p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$ は動作モデルである。

式 (5) の右辺について、被積分関数の第 2 因子は、式 (2) より w_{t-1}^i である。したがって、以下の式に書き換えることができる。

$$w_t^i = \eta p\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}\right) \int p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_t\right) w_{t-1}^i d\mathbf{x}_{t-1}^i \quad (6)$$

式 (6) より、時刻 t における i 番目のパーティクルの重み w_t^i は、動作モデル $p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_t)$ と観測モデル $p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ 、および時刻 $t-1$ におけるパーティクルの重み w_{t-1}^i に基づき算出されることとなる。パーティクルフィルタを用いた自己位置推定では、パーティクルの集合 X_t と式 (6) の各パーティクルの重み w_t^i を用いて、ロボットの状態 \mathbf{x}_t を算出する。

以上のことから本研究では、環境地図とスキャンデータそれぞれの距離情報と分布情報を用いるために、新たな観測モデル $p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ を提案する。なお、ロボットの状態は二次元平面上のみを対象とする。そのため、時刻 t における i 番目のパーティクルの状態 \mathbf{x}_t^i は、位置 x_t^i と y_t^i 、ならびに姿勢 θ_t^i を用いて $\mathbf{x}(\mathbf{x}_t^i, y_t^i, \theta_t^i)$ と表現する。

4. 距離情報と分布情報に基づく観測モデルの提案

4.1 観測モデルの概要

本研究では、LFM を参考に式 (6) における観測モデルを設計する。LFM では、第 2 章で述べたとおり、スキャンデータの各点に対する尤度が計算される。そのため、スキャンデータどうしの関係性を考慮することができず、Fig. 2 に示した自己位置推定の問題が生じる。そこで本研究では、環境地図とスキャンデータそれぞれの距離情報および分布情報の対応関係を考慮した観測モデルを提案する。以下に、提案する観測モデルを示す。

$$p\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}\right) = \alpha_{\text{land}} p_{\text{land}}\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}\right) + (1 - \alpha_{\text{land}}) \times \prod_{\mathbf{z}_t^j \in \mathbf{z}_t} \left(\begin{array}{c} \left[\begin{array}{c} \alpha_{\text{hit}} \\ \alpha_{\text{distr}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{array} \right]^T \cdot \left[\begin{array}{c} p_{\text{hit}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m}) \\ p_{\text{distr}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m}) \\ p_{\text{max}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m}) \\ p_{\text{rand}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m}) \end{array} \right] \end{array} \right) \quad (7)$$

\mathbf{z}_t^j ($j = 1, 2, \dots, J$) はセンサ観測 \mathbf{z}_t の j 番目のスキャンデータであり、座標を有している。 J はスキャンデータ \mathbf{z}_t^j の総数である。 $p_{\text{land}}(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ は \mathbf{z}_t が環境地図 \mathbf{m} と合致する確率、 $p_{\text{hit}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ は \mathbf{z}_t^j の距離情報が環境地図 \mathbf{m} と合致する確率、 $p_{\text{distr}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ は \mathbf{z}_t^j の分布情報が環境地図 \mathbf{m} と合致する確率、 $p_{\text{max}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ は \mathbf{z}_t^j がセンサの最大測定値となる確率、 $p_{\text{rand}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ は \mathbf{z}_t^j がセンサの測定可能範囲において一様に取得される確率である。 α は重みであり、 $0 < \alpha_{\text{land}} < 1$ 、 $\alpha_{\text{hit}} + \alpha_{\text{distr}} + \alpha_{\text{max}} + \alpha_{\text{rand}} = 1$ を満たす。

提案する観測モデルでは、LFM で用いられる距離情報を考慮する $p_{\text{hit}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ 、 $p_{\text{max}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ 、 $p_{\text{rand}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ に加えて、分布情報を考慮する $p_{\text{land}}(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ と $p_{\text{distr}}(\mathbf{z}_t^j | \mathbf{x}_t^i, \mathbf{m})$ を新たに組み込んでいる。これによって、LFM では困難であったスキャンデータどうしの関係性を考慮することが可能となる。ロボットは、式 (7) の観測モデルを用いて式 (6) の重み w_t^i を更新する。これにより、環境地図とスキャンデータそれぞれの距離情報と分布情報に基づき自己位置推定を行う。

4.2 $p_{\text{land}}(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$

$p_{\text{land}}(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ は、センサ観測 \mathbf{z}_t が環境地図 \mathbf{m} と合致する確率である。以下に、 $p_{\text{land}}(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m})$ の算出式を示す。

$$p_{\text{land}}\left(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}\right) = \frac{J_{\text{land}}}{J} \quad (8)$$

J_{land} は、 $\min_{\mathbf{z}_t^j \in \mathbf{z}_t, \mathbf{m}^k \in \mathbf{m}} \|\mathbf{z}_t^j - \mathbf{m}^k\| \leq d_{th}$ を満たすスキャンデータ \mathbf{z}_t^j の総数である。ここで、 d_{th} は閾値、 \mathbf{m}^k ($k = 1, 2, \dots, K$) は環境地図 \mathbf{m} の点群データであり、座標

を有している。

式 (8) より、センサ観測 \mathbf{z}_t が環境地図 \mathbf{m} の形状と合致しているほど、 $p_{\text{land}}(\mathbf{z}_t|\mathbf{x}_t^i, \mathbf{m})$ の値は大きくなる。これにより、環境地図とセンサ観測の合致の程度を、スキャンデータ全体の分布に基づき確率的に評価することができる。

4.3 $p_{\text{hit}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$

$p_{\text{hit}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ は、スキャンデータ \mathbf{z}_t^j の距離情報が環境地図 \mathbf{m} と合致する確率である。以下に、 $p_{\text{hit}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ の算出式を示す。

$$p_{\text{hit}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\min_{\mathbf{z}_t^j \in \mathbf{z}_t, \mathbf{m}^k \in \mathbf{m}} \|\mathbf{z}_t^j - \mathbf{m}^k\|^2}{2}\right) \quad (9)$$

式 (9) より、スキャンデータ \mathbf{z}_t^j が環境地図のデータ \mathbf{m}^k に合致しているほど、 $p_{\text{hit}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ の値は大きくなる。これにより、環境地図とスキャンデータの合致の程度を、距離情報に基づき確率的に評価することができる。

4.4 $p_{\text{distr}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$

$p_{\text{distr}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ は、スキャンデータ \mathbf{z}_t^j の分布情報が環境地図 \mathbf{m} と合致する確率である。以下に、 $p_{\text{distr}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ の算出式を示す。

$$p_{\text{distr}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = \frac{1}{\sqrt{2\pi^3} |\det(\Delta\Sigma^{\mathbf{v}_{n_j}})|} \times \exp\left(-\frac{|\Delta\boldsymbol{\mu}^{\mathbf{v}_{n_j}}|^T (\Delta\Sigma^{\mathbf{v}_{n_j}})^{-1} \Delta\boldsymbol{\mu}^{\mathbf{v}_{n_j}}|}{2}\right) \quad (10)$$

\mathbf{v}_{n_j} はスキャンデータ \mathbf{z}_t^j が含まれたボクセルであり、NDTにより生成される。 $\Delta\boldsymbol{\mu}^{\mathbf{v}_{n_j}}$ は、 $\Delta\boldsymbol{\mu}^{\mathbf{v}_{n_j}} = \boldsymbol{\mu}_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}} - \boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_{n_j}}$ から求められる。ただし、 $\boldsymbol{\mu}_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}}$ はボクセル \mathbf{v}_{n_j} におけるセンサ観測 \mathbf{z}_t に対して NDT により算出された平均、 $\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_{n_j}}$ はボクセル \mathbf{v}_{n_j} における環境地図 \mathbf{m} に対して NDT により算出された平均である。 $\Delta\Sigma^{\mathbf{v}_{n_j}}$ は、 $\Delta\Sigma^{\mathbf{v}_{n_j}} = \Sigma_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}} - \Sigma_{\mathbf{m}}^{\mathbf{v}_{n_j}}$ から求められる。ただし、 $\Sigma_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}}$ はボクセル \mathbf{v}_{n_j} におけるセンサ観測 \mathbf{z}_t に対して NDT により算出された共分散行列、 $\Sigma_{\mathbf{m}}^{\mathbf{v}_{n_j}}$ はボクセル \mathbf{v}_{n_j} における環境地図 \mathbf{m} に対して NDT により算出された共分散行列である。なお、 $\boldsymbol{\mu}_{\mathbf{s}}^{\mathbf{v}_n}(\mathbf{s} = \mathbf{m}, \mathbf{z}_t)$ と $\Sigma_{\mathbf{s}}^{\mathbf{v}_n}$ は以下の式より算出する。

$$\boldsymbol{\mu}_{\mathbf{s}}^{\mathbf{v}_n} = \frac{1}{n_{\mathbf{v}_n}} \sum_{a=1}^{n_{\mathbf{v}_n}} \mathbf{s}_{\mathbf{v}_n}^a \quad (11)$$

$$\Sigma_{\mathbf{s}}^{\mathbf{v}_n} = \frac{1}{n_{\mathbf{v}_n}} \sum_{a=1}^{n_{\mathbf{v}_n}} (\mathbf{s}_{\mathbf{v}_n}^a - \boldsymbol{\mu}_{\mathbf{s}}^{\mathbf{v}_n})(\mathbf{s}_{\mathbf{v}_n}^a - \boldsymbol{\mu}_{\mathbf{s}}^{\mathbf{v}_n})^T \quad (12)$$

$\mathbf{s}_{\mathbf{v}_n}^a$ ($a = 1, 2, \dots, n_{\mathbf{v}_n}$) は、ボクセル \mathbf{v}_n 内に存在する a 番目の環境地図のデータまたはスキャンデータである。

式 (10) では、式 (11) と式 (12) により算出されたボクセル \mathbf{v}_{n_j} におけるスキャンデータ \mathbf{z}_t の多次元正規分布 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}}, \Sigma_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}})$ と、ボクセル \mathbf{v}_{n_j} における環境地図 \mathbf{m} の多次元正規分布 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_{n_j}}, \Sigma_{\mathbf{m}}^{\mathbf{v}_{n_j}})$ が比較される。その結果、 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}}, \Sigma_{\mathbf{z}_t^j}^{\mathbf{v}_{n_j}})$ が $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_{n_j}}, \Sigma_{\mathbf{m}}^{\mathbf{v}_{n_j}})$ に合致しているほど、 $p_{\text{distr}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ の値は大きくなる。これにより、環境地図と

スキャンデータの合致の程度を、ボクセル空間における分布情報に基づき確率的に評価することができる。

4.5 $p_{\text{max}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$

$p_{\text{max}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ は、スキャンデータ \mathbf{z}_t^j がセンサの最大測定値となる確率である。以下に、 $p_{\text{max}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ の算出式を示す。

$$p_{\text{max}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = \begin{cases} 1 & (\|\mathbf{z}_t^j\| = l_{\text{max}}) \\ 0 & (\|\mathbf{z}_t^j\| < l_{\text{max}}) \end{cases} \quad (13)$$

式 (13) より、スキャンデータ \mathbf{z}_t^j の測定距離 $\|\mathbf{z}_t^j\|$ が最大測定可能距離 l_{max} の場合に $p_{\text{max}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = 1$ となる。一方で、 $\|\mathbf{z}_t^j\|$ が最大測定可能距離 l_{max} 未満の場合は、 $p_{\text{max}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = 0$ となる。

4.6 $p_{\text{rand}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$

$p_{\text{rand}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ は、スキャンデータ \mathbf{z}_t^j がセンサの測定可能範囲において一様に取得される確率である。以下に、 $p_{\text{rand}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m})$ の算出式を示す。

$$p_{\text{rand}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = \begin{cases} \frac{1}{l_{\text{max}}} & (\|\mathbf{z}_t^j\| \leq l_{\text{max}}) \\ 0 & (\|\mathbf{z}_t^j\| > l_{\text{max}}) \end{cases} \quad (14)$$

l_{max} は、センサの最大測定可能距離である。

式 (14) より、スキャンデータ \mathbf{z}_t^j の測定距離 $\|\mathbf{z}_t^j\|$ が最大測定可能距離 l_{max} 以下の場合に $p_{\text{rand}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = 1/l_{\text{max}}$ となる。一方で、 $\|\mathbf{z}_t^j\|$ が最大測定可能距離 l_{max} より大きい場合は、 $p_{\text{rand}}(\mathbf{z}_t^j|\mathbf{x}_t^i, \mathbf{m}) = 0$ となる。

5. 提案する観測モデルを用いた自己位置推定

ロボットは、式 (7) の観測モデルを用いたパーティクルフィルタに基づき自己位置推定を行う。Algorithm 1 に、自己位置推定における処理の流れを示す。

はじめに、ロボットの初期状態 $\mathbf{x}_1 = (x_1, y_1, \theta_1)$ 、時刻 t におけるパーティクルの集合 X_t 、各パーティクルの重み w_t^i 、そしてセンサ観測 \mathbf{z}_t および環境地図 \mathbf{m} を定める。1~4 行めでは、パーティクルの初期化のため、初期サンプリングと重みの均等化を行う。ここで、 $\mathbf{x}_1^i = (x_1^i, y_1^i, \theta_1^i)$ は i 番目のパーティクルの状態、 r_s と θ_s はサンプリング範囲を決定するための定数、 r_{rand, r_s} は $[-r_s, r_s]$ でランダムに値を返す関数、 $\theta_{\text{rand}, 2\pi}$ は $[-\pi, \pi]$ でランダムに値を返す関数、 $\theta_{\text{rand}, \theta_s}$ は $[-\theta_s/2, \theta_s/2]$ でランダムに値を返す関数である。5 行めでは、環境地図に対する NDT により、多次元正規分布 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_n}, \Sigma_{\mathbf{m}}^{\mathbf{v}_n})$ を取得する。提案する観測モデルでは、 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_n}, \Sigma_{\mathbf{m}}^{\mathbf{v}_n})$ を環境地図の分布情報として用いる。以降の処理 6~17 行めでは、自己位置推定とパーティクルのリサンプリングが繰り返される。

7 行めでは、センサ観測に対する NDT により、多次元正規分布 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_n}, \Sigma_{\mathbf{m}}^{\mathbf{v}_n})$ を取得する。提案する観測モデルでは、 $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{m}}^{\mathbf{v}_n}, \Sigma_{\mathbf{m}}^{\mathbf{v}_n})$ をスキャンデータの分布情報として用いる。8~11 行めでは、各パーティクルの重みの更新を行う。9 行めでは、オドメトリ $\Delta\mathbf{x}_t = (\Delta x_t, \Delta y_t, \Delta\theta_t)$ に基づきパーティクルの状態を遷移させる。このとき、ロボットへの制御入力 \mathbf{u}_t をオドメトリとすると、全パーティクルの状態は、動作モデル $p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_t) = 1$ に従って遷移する。10 行めでは、式 (6) に従って各パーティクルの重みの更新を行う。ただし、動作モデルは $p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i, \mathbf{u}_t) = 1$ としているため、パーティクルの

Algorithm 1 Particle filter based localization using proposed observation model

Require: $\mathbf{x}_1 = (x_1, y_1, \theta_1)$, $X_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \dots, \mathbf{x}_t^i, \dots, \mathbf{x}_t^I\}$,
 $w_t^i, \mathbf{z}_t = \{z_t^1, z_t^2, \dots, z_t^j, \dots, z_t^J\}$,
 $\mathbf{m} = \{m_t^1, m_t^2, \dots, m_t^k, \dots, m_t^K\}$

- 1: **for** $i = 1$ to I **do**
- 2: $\begin{bmatrix} x_1^i \\ y_1^i \\ \theta_1^i \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} r_{\text{rand}, r_s} \cos(\theta_{\text{rand}, 2\pi}) \\ r_{\text{rand}, r_s} \sin(\theta_{\text{rand}, 2\pi}) \\ \theta_{\text{rand}, \theta_s} \end{bmatrix}$
- 3: $w_t^i = \frac{1}{I}$
- 4: **end for**
- 5: Get $\mathcal{N}(\mu_m^{v_n}, \Sigma_m^{v_n})$ through NDT for \mathbf{m}
- 6: **while do**
- 7: Get $\mathcal{N}(\mu_{z_t}^{v_n}, \Sigma_{z_t}^{v_n})$ through NDT for \mathbf{z}_t
- 8: **for** $i = 1$ to I **do**
- 9: $\begin{bmatrix} x_t^i \\ y_t^i \\ \theta_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i \\ y_{t-1}^i \\ \theta_{t-1}^i \end{bmatrix} + \begin{bmatrix} \Delta x_t \\ \Delta y_t \\ \Delta \theta_t \end{bmatrix}$
- 10: $w_t^i = \eta p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}) w_{t-1}^i$
- 11: **end for**
- 12: $\mathbf{x}_t = \sum_{i=1}^I w_t^i \mathbf{x}_t^i$
- 13: $ESS = \frac{1}{\sum_{i=1}^I w_t^i{}^2}$
- 14: **if** $ESS \geq \frac{I}{2}$ **then**
- 15: Resampling particles
- 16: **end if**
- 17: **end while**

重み w_t^i は、式 (7) で提案する観測モデルと時刻 $t-1$ におけるパーティクルの重み w_{t-1}^i に基づき算出される。12 行めでは、ロボットの状態がパーティクルの状態 \mathbf{x}_t^i と重み w_t^i の加重平均により算出される。これにより、全パーティクルを考慮したロボットの自己位置推定が可能となる。

13 行めでは、各パーティクルの重みに基づき有効サンプル数 (ESS: Effective Sample Size) を算出する。14~16 行めでは、有効サンプル数の値に応じたパーティクルのリサンプリングを行う。有効サンプル数は、全パーティクルの重みが均等であるほど $ESS \rightarrow I$ となり、一つのパーティクルに重みが偏っているほど $ESS \rightarrow 1$ となる。そこで、有効サンプル数 ESS が $I/2$ 未満の場合、リサンプリングとしてパーティクルの状態の更新と重みの均等化を行う。

6. シミュレーション実験

6.1 実験条件

提案する観測モデルを用いたパーティクルフィルタをロボットに適用し、環境変化に対する自己位置推定のロバスト性について検証する。本実験では、環境がランドマークによって完全に囲われておらず、かつ、障害物により混雑していることを想定する。そのため、安全性を考慮し最高速度を 0.5 [m/s]、衝突を回避できるような最高角速度を 0.5 [rad/s] とした。また、ロボットには前面の高さ 0.36 [m] に Velodyne 社の VLP-16 が搭載されている。Fig. 3 に、実験環境を示す。

Fig. 3 (a) の環境には、約 50×50 [m²] の空間の周囲に、ランドマークとして $1 \times 1 \times 3$ [m³]、 $2 \times 2 \times 3$ [m³]、 $3 \times 3 \times 3$ [m³]、 $4 \times 4 \times 3$ [m³]、 $5 \times 5 \times 3$ [m³] の白色の四角柱がそれぞれ 20 個ずつ計 100 個配置されている。ロボットは、3D LiDAR を用いて当該環境の三次元 SLAM を行う。Fig. 3 (b) は、点群によって構成される環境の三次元地図である。ロボットは、この

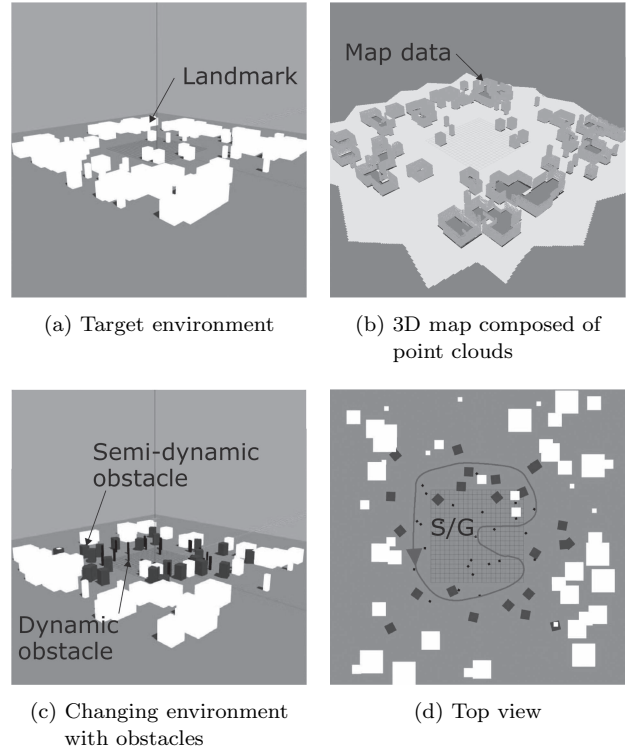


Fig. 3 Simulation settings

地図を用いて自己位置推定を行う。Fig. 3 (c) の移動環境では、Fig. 3 (a) の環境からランドマークが 50% 除去され、さらに、準動的障害物として $2 \times 2 \times 3$ [m³] の灰色の四角柱が 25 個、動的障害物 $0.5 \times 0.5 \times 3$ [m³] の黒色の四角柱が 25 個配置される。なお本実験では、一般性の観点から特定の環境ではなく、ランダムに設けられたランドマークならびに障害物によって構成される環境を対象に、ロボットが自己位置推定を試みる。実験の際には、準動的障害物はその場に固定され、動的障害物はランダムに選択したゴール地点に向けた直線的な動きを繰り返す。ロボットは、人の操縦によって Fig. 3 (d) のスタート地点 S から経路に沿ってゴール地点 G へと移動する。その際、ロボットには以下に示す従来ならびに提案手法が適用される。

- (1) LFM [18] に基づく自己位置推定法
- (2) NDT-MCL [20] に基づく自己位置推定法
- (3) 提案する観測モデルを用いたパーティクルフィルタに基づく自己位置推定法

(1) の LFM は、環境地図とスキャンデータそれぞれの距離情報に基づく観測モデルを用いたパーティクルフィルタによる自己位置推定法である。そのため、提案する観測モデルとは異なり、分布情報は併用されていない。以下に、LFM における観測モデルを示す。

$$p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}) = \prod_{z_t^j \in \mathbf{z}_t} \left(\begin{bmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{bmatrix}^T \cdot \begin{bmatrix} p_{\text{hit}}(z_t^j | \mathbf{x}_t^i, \mathbf{m}) \\ p_{\text{max}}(z_t^j | \mathbf{x}_t^i, \mathbf{m}) \\ p_{\text{rand}}(z_t^j | \mathbf{x}_t^i, \mathbf{m}) \end{bmatrix} \right) \quad (15)$$

(2) の NDT-MCL は、環境地図とスキャンデータそれぞれの分布情報に基づく観測モデルを用いたパーティクルフィルタによる自己位置推定法である。そのため、提案する観測モデルとは異なり、距離情報は併用されていない。以下に、NDT-MCL における観測モデルを示す。

$$p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}) = d \times \exp\left(-\frac{(\Delta\boldsymbol{\mu}^{v_{n_j}})^T (\boldsymbol{\Sigma}_{\mathbf{z}_t+\mathbf{m}}^{v_{n_j}})^{-1} (\Delta\boldsymbol{\mu}^{v_{n_j}})}{2}\right) \quad (16)$$

d はスケーリング定数である。 $\boldsymbol{\Sigma}_{\mathbf{z}_t+\mathbf{m}}^{v_{n_j}}$ はセンサ観測と環境地図の不確かさを表し、 $\boldsymbol{\Sigma}_{\mathbf{z}_t}^{v_{n_j}} + \boldsymbol{\Sigma}_{\mathbf{m}}^{v_{n_j}}$ から求められる。

NDTにおけるボクセルの1辺は1[m]、パーティクルの総数は $I = 200$ 、サンプリングに用いる定数は $r_s = 10$ [m]、 $\theta_s = 10$ [deg] とした。各手法、複数のパラメータの組み合わせによる予備実験を行い、その中で最も自己位置の指定精度が高かったパラメータを用いる。そして、LFMにおける重みは $\alpha_{\text{hit}} = 0.9$ 、 $\alpha_{\text{rand}} = 0.05$ 、 $\alpha_{\text{max}} = 0.05$ 、NDT-MCLにおけるスケーリング定数は $d = 1,000$ 、提案する観測モデルにおける重みは $\alpha_{\text{land}} = 0.5$ 、 $\alpha_{\text{hit}} = 0.3$ 、 $\alpha_{\text{distr}} = 0.6$ 、 $\alpha_{\text{rand}} = 0.05$ 、 $\alpha_{\text{max}} = 0.05$ 、閾値は $d_{\text{th}} = 0.25$ [m] とした。

文献[21]より、最大推定位置誤差が0.5[m]以下、最大推定姿勢誤差が3[deg]以下であれば自己位置推定成功、一方で、これらの誤差が基準値以上であれば自己位置の誤推定とする。ただし、誤推定の後に、再度誤差の値が基準値以下となれば自己位置推定回復とする。誤推定の後、誤差の値が基準値以下にならない場合は自己位置推定破綻とする。そして、第1章で述べたロバスト性の定義より、Fig.3(a)とFig.3(c)の環境においてロボットが自己位置推定に成功することで、環境変化にロバストであるとする。

6.2 自己位置推定実験

人の操縦によってSからGに向け移動するロボットは、適用された(1)~(3)の手法により自己位置推定を繰り返す。Fig.4に、ロボットが推定した移動軌跡を示す。なお、破線が実際にロボットが移動した軌跡、実線が推定した軌跡である。

Fig.4(a)では、40[m]ほど移動した地点($x = 9.1, y = 13.9$)からロボットが実際とは異なる位置を誤推定していることが見てとれる。Fig.4(b)でも同様に、40[m]ほど移動した地点($x = 10.8, y = 13.4$)からロボットが実際とは異なる位置を誤推定していることが見てとれる。これらの結果より、LFMとNDT-MCLでは、環境変化により終始正確な自己位置推定を行うことは困難であったことが分かる。一方Fig.4(c)では、ロボットが終始正確な自己位置を推定し続けたため、実際のロボットの軌道と推定軌道が一致していることが見てとれる。これは、提案する観測モデルによって、環境地図とスキャンデータに対応付けることができたためである。そこで、約40[m]地点での自己位置推定の差異を検証するため、(1)~(3)の各手法によるパーティクルの分布をFig.5に示す。

Fig.5(a)では、パーティクルが広範囲に分布しているのが見てとれる。Fig.5(b)では、Fig.5(a)に比べロボットの周辺にパーティクルが分布しているものの、横方向への広がりが見てとれる。したがってLFMとNDT-MCLでは、観測モデルである式(15)と式(16)において、環境変化がパーティクルの尤度計算に悪影響を及ぼしていることが分かる。その結果、パーティクルの分散も5.18と2.26となった。一方Fig.5(c)では、パーティクルがロボットの位置に集中していることが見てとれる。なお、パーティクルの分散は0.56となった。これは、提案する観測モデルによって、環境変化が生じて環境地図とスキャンデータに対応付けることができ、ロボットの真の位置周辺の

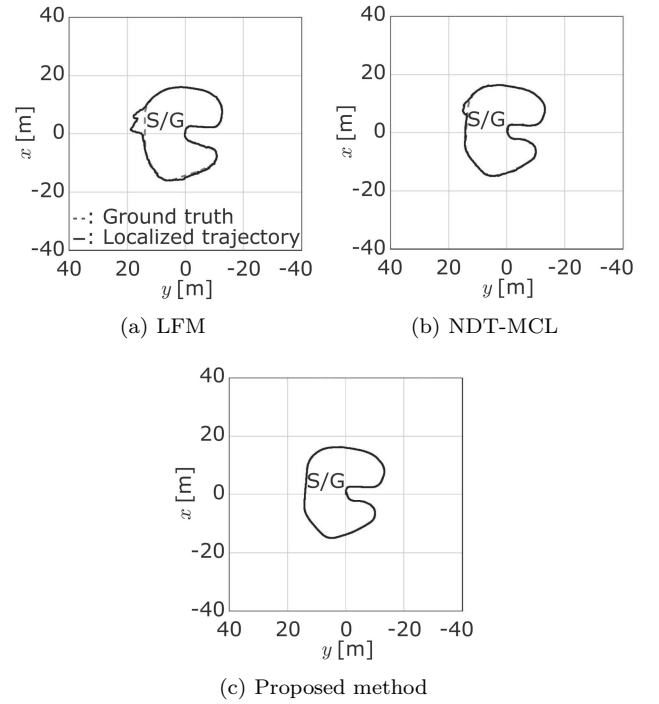


Fig. 4 Comparison of localized trajectories

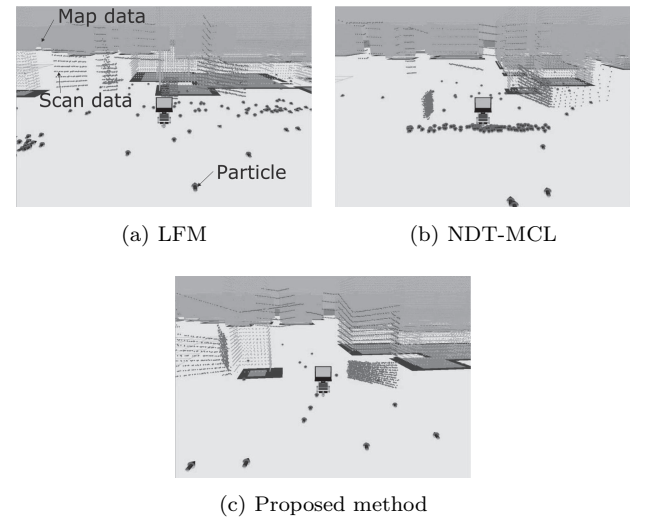


Fig. 5 Comparison of particle distribution around 40[m] from S

Table 1 Localization results: successes, recoveries, or failures

| Method | Successes | Recoveries | Failures |
|--------------|-----------|------------|----------|
| (1) LFM | 0 | 9 | 1 |
| (2) NDT-MCL | 0 | 8 | 2 |
| (3) Proposed | 9 | 1 | 0 |

パーティクルの尤度が高くなったためである。そこで、各手法によりロボットがSからGに向け計10回移動した際の自己位置推定の結果をTable 1に示す。

従来手法である(1)と(2)に基づくロボットは、一度も自己位置推定に成功することはなかった。失敗の内訳に関しては、途中、自己位置推定が回復したことにより、ロボットは9回と8回、LFMとNDT-MCLによりGに到達することができたが、それ以外では1回と2回、自己位置推定が破綻する結果となった。一方提案する観測モデルに基づくロボットは、9回自己位置推定に成功した。また、残り1回も自己位置推定の回復

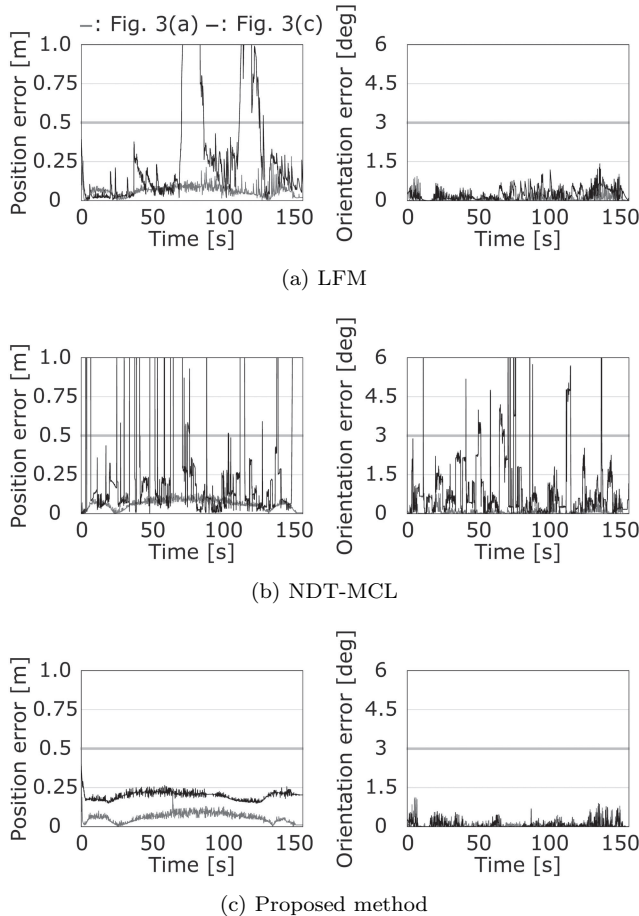


Fig. 6 Localization robustness based on position and orientation errors

により、10回すべてにおいてGに到達することができた。この結果より、提案する観測モデルを用いることでロボットは、環境変化が生じたとしても、安定して自己位置を推定できることが示された。

6.3 自己位置推定のロバスト性

(1)~(3)の各手法の環境変化に対する自己位置推定のロバスト性を評価するため、**Fig. 6**に、**Fig. 3(a)**と**Fig. 3(c)**が示す変化前後の環境におけるロボットの推定位置姿勢誤差の推移を示す。なお、自己位置推定成功の基準値として、位置誤差0.5[m]と姿勢誤差3[deg]を灰色の太線で表す。

Fig. 6(a)では、変化前の環境に比べ、変化後の環境における推定位置の誤差が断続的に大きく、最大推定位置誤差も0.5[m]を上回る結果となった。このことから、LFMは環境変化に対する自己位置推定のロバスト性を有していないことが分かる。**Fig. 6(b)**では、変化前の環境に比べ、変化後の環境における推定位置姿勢の誤差が断続的に大きく、いずれも最大推定位置誤差と姿勢誤差は0.5[m]と3[deg]を上回る結果となった。このことから、NDT-MCLも環境変化に対する自己位置推定のロバスト性を有していないことが分かる。**Fig. 6(c)**の結果においても、変化前の環境に比べ、変化後の環境における推定位置の誤差が大きくなった。このことから、環境変化が提案手法によるロボットの自己位置推定にも影響を及ぼしていることが分かる。ただし、最大推定位置誤差は0.5[m]以下、最大推定姿勢誤差も3[deg]以下となり、いずれも基準値を下回る結果となった。このことから、提案する観測モデルを用いた手法は、環境変化に対して自己位置推定のロバスト性を有していることが示された。

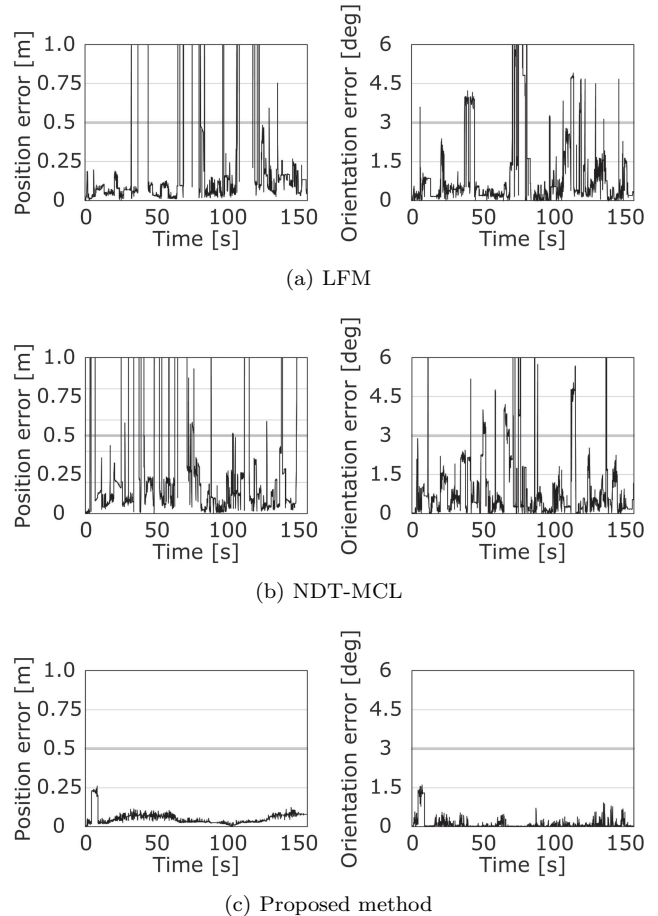


Fig. 7 Position and orientation errors of particle with maximum weight

以上の結果は、提案手法において、正しい位置姿勢のパーティクルに基づく自己位置推定が行われたためであると考えられる。そこで、(1)~(3)の手法について、ロボットが自己位置推定を行った際のそれぞれの最大重みのパーティクルに着目する。パーティクルフィルタによる自己位置推定では、最大重みのパーティクルがロボットの真値に最も近い状態を有していることになる。そのため、当該パーティクルの位置姿勢誤差に基づき、環境地図とスキャンデータの対応付けを定量的に評価する。**Fig. 7**に、各手法をロボットに適用した際の最大重みのパーティクルの位置姿勢誤差の推移を示す。

Fig. 7(a)と**Fig. 7(b)**では、最大重みのパーティクルの最大位置誤差と姿勢誤差は、それぞれ0.5[m]と3[deg]を上回る結果となった。一方**Fig. 7(c)**では、**Fig. 7(a)**と**Fig. 7(b)**の結果に比べ位置および姿勢の誤差が常に小さく、さらに、最大位置誤差と姿勢誤差も基準値である0.5[m]と3[deg]を下回っていることが分かる。したがってロボットは、提案する観測モデルを用いたパーティクルフィルタにより、従来手法であるLFMやNDT-MCLに比べ、環境変化に対しても環境地図とスキャンデータの対応付けに成功したものと考えられる。以上の結果から、環境変化に対するロボットのロバストな自己位置推定のための提案手法の有効性が示された。

7. 他環境での自己位置推定と実ロボットによる自律移動

7.1 他環境でのシミュレーション実験

本節では、**Fig. 3**とは異なる屋内外の環境変化に対して、提案手法の有効性を検証する。そこで、第6章の実験環境以外に

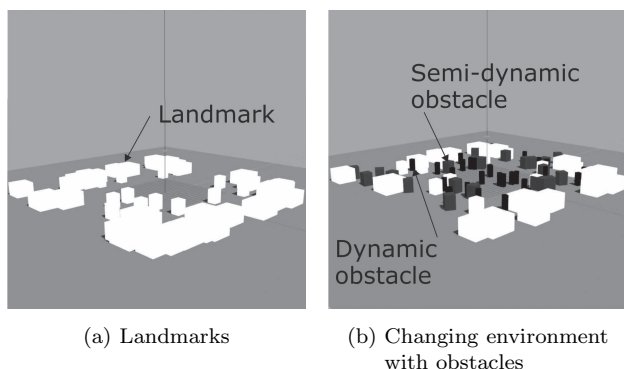


Fig. 8 Outdoor environment

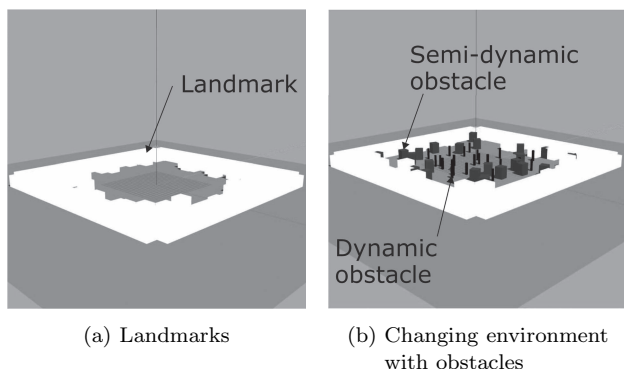


Fig. 9 Indoor environment

Table 2 Localization results in outdoor environment

| Method | Successes | Recoveries | Failures |
|----------|-----------|------------|----------|
| LFM | 0 | 4 | 6 |
| NDT-MCL | 0 | 2 | 8 |
| Proposed | 7 | 2 | 1 |

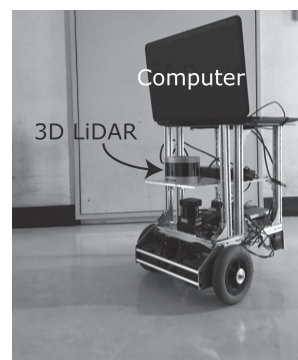
Table 3 Localization results in indoor environment

| Method | Successes | Recoveries | Failures |
|----------|-----------|------------|----------|
| LFM | 6 | 4 | 0 |
| NDT-MCL | 5 | 5 | 0 |
| Proposed | 10 | 0 | 0 |

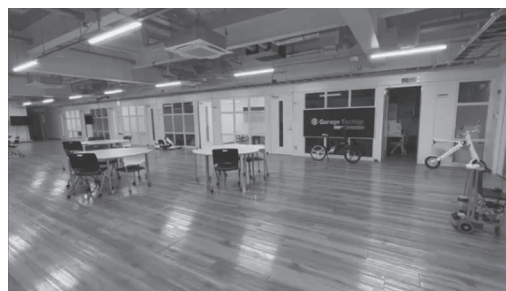
も、さらに二つの環境を対象としたシミュレーション実験を行う。Fig. 8 と Fig. 9 に、シミュレーション実験の環境を示す。

Fig. 8 (a) は、Fig. 3 (a) と同様、環境がランドマークによって囲われておらず、屋外環境を想定している。ただし、Fig. 3 (a) とはランドマークの配置が異なる。一方 Fig. 9 (a) は、環境がランドマークによって囲われており、屋内環境を想定している。それぞれの環境に対して、三次元地図を構築する。その後、6.1 節の条件でランドマークが除去され、準動的障害物と動的障害物が配置された環境が Fig. 8 (b) と Fig. 9 (b) となる。ロボットは、Fig. 3 (d) と同様に、人の操縦によってスタート地点 S から経路に沿ってゴール地点 G へと移動する。Table 2 と Table 3 に、LFM, NDT-MCL, そして提案手法に基づき、ロボットが 10 回移動した際の自己位置推定の結果を示す。

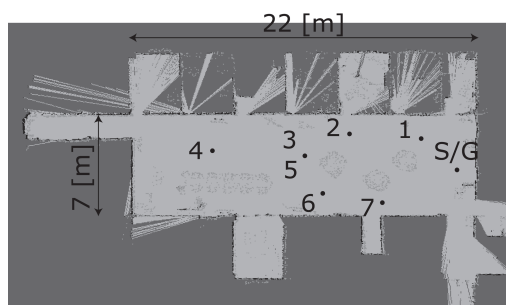
Table 2 では、Table 1 に比べ、提案手法による自己位置推定の成功回数が 9 回から 7 回へと減少した。残り 3 回のうち、2 回は自己位置推定の回復により、ロボットは G に到達することができたが、1 回は自己位置推定が破綻した。この結果から、本実験環境は、Fig. 3 よりも環境変化がロボットの自己位置推定に対して、より影響したことが分かる。ただし、LFM と NDT-MCL では、ロボットはさらに多く自己位置推定に破綻し



(a) Mobile robot



(b) Original indoor environment



(c) Top view of 3D map

Fig. 10 Autonomous navigation in actual environment

ており、これら従来手法と比べ、提案手法が環境変化に対して有効であることが分かる。

Table 3 では、Table 1 や Table 2 の結果と比べ、いずれの手法においてもロボットによる自己位置推定の成功回数が増えていることが分かる。これは、自己位置推定に十分なランドマークがロボットの周囲にあったことを示している。そして、いずれの手法においてもロボットは、10 回すべてにおいて G に到達することができた。さらに、提案手法によりロボットは、10 回すべてで自己位置推定に成功したことから、従来手法に比べ、環境変化に対して有効であることが分かる。

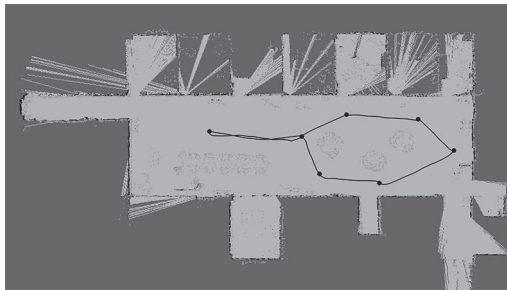
7.2 実ロボットによる自律移動

Fig. 3 や Fig. 8 と Fig. 9 のシミュレーション環境に比べ、実環境には様々な形状のランドマークや障害物が存在する。そこで、提案手法を実機のロボットに適用し、実環境での自律移動実験を行う。そして、提案手法の有効性に加えて、実ロボットへの適用の可能性について言及する。Fig. 10 に、使用するロボットならびに実験環境を示す。

Fig. 10 (a) の移動ロボットには、自己位置推定のため、0.36 [m] の高さに VLP-16 が搭載されている。このほか、移動速度や角速度など、すべて 6.1 節と同様の設定とした。自己位置推定のための計算は、ノート PC の CPU により処理される。ノート



(a) Changing environment



(b) Navigation trajectory

Fig. 11 Navigation result in changing environment

PCには、Intel i7-7700HQ 3.80GHzのCPUが搭載されている。自律移動のため、事前に Fig. 10 (b) の屋内環境を対象に、Fig. 10 (c) の三次元地図を構築した。地図上には1~7のノードが与えられ、ロボットは、スタートノードSから順番にノードを経て、ゴールノードGへと自律移動する。その際、提案手法による自己位置推定に加え、目的ノードまでの経路計画ならびに経路追従のための動作計画には、それぞれダイクストラ法とDynamic Window Approachを適用した、Fig. 11に、自律移動時の環境ならびに実験結果を示す。

Fig. 10 (b) と比べ、Fig. 11 (a) では、テーブルと椅子の配置が異なる。これは、ランドマークの消失と準動的障害物の出現を意味している。また、動的障害物としての人間が8人、当該環境内を歩き回っている。Fig. 11 (b) より、ロボットは、変化の生じた実環境においても、提案手法により自己位置推定に成功し、スタートノードSからゴールノードGまで自律的に移動できたことが見てとれる。なお、自律移動には約150[s]ほどの時間を要し、その間、三次元点群の処理から自己位置推定まで1回の計算時間は、平均して0.13[s]であった。以上のことから、提案手法はノートPCを介して実ロボットに適用でき、かつ、実環境に存在するランドマークや障害物に対しても、ロボットは自己位置推定ならびに自律移動が可能であることが示された。

8. 結 言

本研究では、ロボットの自律移動のため、障害物の出現やランドマークの消失といった環境変化に対してロバストな自己位置推定を目的とした。そして、パーティクルフィルタを用いた自己位置推定において、環境地図とスキャンデータそれぞれの距離情報と分布情報に基づく観測モデルを提案した。シミュレーション実験にてロボットは、環境変化に対しても地図とセンサ観測で取得したスキャンデータに共通するランドマークを対応付けることに成功し、その結果、ロバストに自己位置を推定できることが示された。また、実機を用いた実験にてロボットは、

シミュレーション実験と同様に、環境変化に対しても地図とセンサ観測で取得したスキャンデータに共通するランドマークを対応付けることに成功し、その結果、ロバストに自己位置を推定できることが示された。

参 考 文 献

- [1] H. Durrant-Whyte and T. Balley: "Simultaneous Localization and Mapping: part I," IEEE Robotics & Automation Magazine, vol.13, no.2, pp.99-110, 2006.
- [2] T. Balley and H. Durrant-Whyte: "Simultaneous Localization and Mapping (SLAM): part II," IEEE Robotics & Automation Magazine, vol.13, no.3, pp.108-117, 2006.
- [3] M. Betke and L. Gurvits: "Mobile Robot Localization Using Landmark," IEEE Transactions on Robotics & Automation Magazine, vol.13, no.2, pp.251-263, 1997.
- [4] D. Fox, W. Burgard and S. Thrun: "Markov Localization for Mobile Robots in Dynamic Environments," Journal of Artificial Intelligence Research, vol.11, pp.391-427, 1999.
- [5] W. Burgard, A.B. Cremers, D. Fox, D.G. Lakemeyer, D. Schulza, W. Steiner and S. Thrun: "Experiences with An Interactive Museum Tour-Guide Robot," Artificial Intelligence, vol.114, pp.3-55, 1999.
- [6] D. Wang, J. Xue, Z. Tao, Y. Zhong, D. Cui, S. Du and N. Zheng: "Accurate Mix-Norm-based Scan Matching," IEEE International Conference on Intelligent Robots & Systems, pp.1665-1672, 2018.
- [7] H. Durrant-Whyte and T. Balley: "Toward Localization-based Automated Driving in Highly Dynamic Environments: Comparison and Discussion of Observation Models," IEEE International Conference on Intelligent Transportation Systems, pp.3315-2222, 2018.
- [8] S. Thrun, D. Fox, W. Burgard and F. Dellaert: "Robust Monte Carlo Localization for Mobile Robots," Artificial Intelligence, vol.128, no.1-2, pp.99-141, 2001.
- [9] F. Dellaert, D. Fox, W. Burgard and S. Thrun: "Monte Carlo Localization for Mobile Robots," International Conference on Robotics and Automations, vol.2, pp.1322-1328, 1999.
- [10] D. Meyer-Delius, J. Hess, G. Grisetti and W. Burgard: "Temporary Maps for Robust Localization in Semi-Static Environments," IEEE International Conference on Intelligent Robots & Systems, pp.5750-5755, 2010.
- [11] R. Valencia, J. Saarinen, H. Andreasson, J. Vallvé, J. Andrade-Cetto and A.J. Lilienthal: "Localization in Highly Dynamic Environments using Dual-Timescale NDT-MCL," IEEE International Conference on Robotics & Automation, pp.3956-3962, 2014.
- [12] 小林亮, 吉田遊友, 住吉丈一郎, 星野智史: "移動ロボットのためのLiDARデータの遮蔽にロバストな自己位置推定法", ロボティクス・メカトロニクス講演会, 2P1-K05, 2020.
- [13] D. Meyer-Delius, M. Beinhofer and W. Burgard: "Occupancy Grid Models for Robot Mapping in Changing Environments," AAAI Conference on Artificial Intelligence, pp.2024-2030, 2012.
- [14] J. Saarinen, H. Andreasson and A.J. Lilienthal: "Indeendent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environments," IEEE International Conference on Intelligent Transportation Systems, pp.3489-3495, 2012.
- [15] S. Olufs and M. Vincze: "An Efficient Area-based Observation Model for Monte-Carlo Robot Localization," IEEE International Conference on Intelligent Robots & Systems, pp.13-20, 2009.
- [16] E. Takeuchi K. Ohno and S. Tadokoro: "Robust Localization Method based on Free-Space Observation Model using 3d-Map," IEEE International Conference on Robotics & Biomimetics, pp.973-979, 2010.
- [17] N. Akai, Y. Morales and H. Murase: "Mobile Robot Localization Considering Class of Sensor Observations," IEEE International Conference on Intelligent Robots & Systems, pp.3159-3166, 2018.
- [18] S. Thrun, W. Burgard and D. Fox: Probabilistic Robotics. The

MIT Press, 2005.

- [19] P. Biber and W. Strasser: "The Normal Distributions Transform : A New Approach to Laser Scan Matching," IEEE Robotics & Automation Magazine, vol.13, no.3, pp.2743-2748, 2003.
- [20] J. Saarinen, H. Andreasson, T. Stoyanov and A.J. Lilienthal: "Normal distributions transform Monte-Carlo localization (NDT-MCL)," International Conference on Intelligent Robots & Systems, pp.382-389, 2013.
- [21] N. Akai, L.Y. Morales, T. Hirayama and H. Murase: "Misalignment Recognition Using Markov Random Fields With Fully Connected Latent Variables for Detecting Localization Failures," IEEE Robotics & Automation Magazine, vol.4, no.4, pp.3955-3962, 2019.



小林 亮 (Ryo Kobayashi)

2022年3月宇都宮大学大学院地域創生科学研究科工農総合科学専攻機械知能工学プログラム修了。修士(工学)。



星野智史 (Satoshi Hoshino)

2006年東京大学大学院工学系研究科精密機械工学専攻博士課程修了。博士(工学)。2006年東京工業大学資源化学研究所助手。2007年同助教。2012年9月より宇都宮大学工学部准教授。この間、2014～2015年カリフォルニア大学サンタバーバラ校客員研究員。自律移動ロボットのための自己位置推定・動作計画、警備ロボットシステム、ロボットビジョン等の研究に従事。2004年日本機械学会第13回交通・物流部門大会優秀論文講演表彰受賞。2011年日本ロボット学会研究奨励賞受賞。2012年日本機械学会生産システム部門優秀講演論文表彰受賞。IEEE, 日本機械学会, 計測自動制御学会の会員。(日本ロボット学会正会員)