

小型軽量な駆動ユニットを用いた移動ロボットの開発と 屋外 3 次元地形での自律走行

原 祥堯 ^{†1}, 西村 健志 ^{†1}, 鈴木 太郎 ^{†1}, 入江 清 ^{†1}, 吉田 智章 ^{†1}, 大和 秀彰 ^{†1}, 友納 正裕 ^{†1}

Development of a Mobile Robot using Compact Drive Units and Autonomous Navigation on Outdoor 3D Terrain

*Yoshitaka HARA ^{‡1}, Takeshi NISHIMURA ^{‡1}, Taro SUZUKI ^{‡1}, Kiyoshi IRIE ^{‡1},
Tomoaki YOSHIDA ^{‡1}, Hideaki YAMATO ^{‡1}, Masahiro TOMONO ^{‡1}

千葉工業大学 fuRo アウトドア部

1. 緒言

筆者らは、移動ロボットの基盤技術を研究開発すると共に、各種のロボットシステムへの統合を行っている。本稿では、これらの技術実証として取り組んだ、つくばチャレンジ 2019 での自律移動ロボットの開発事例について報告する。

ハードウェアについては、独自開発の駆動ユニットを用いて、2017 年に自律移動ロボット Puffin を開発している。その後の 2 年間の実験を通して、改善点も明らかになってきた。そこで今年度は、新たな車体の開発を行った。

ソフトウェアについては、2018 年から 3 次元での環境認識と行動計画に取り組んでいる。つくばチャレンジ 2017 では 2 次元の手法で、2018 では 3 次元の手法を用いて、それぞれ本走行で完走を達成できたが、課題はまだ多く残っている。本稿では、今年度実施した自律走行の改良について述べる。

2. ハードウェア

2.1 搭載デバイスと車体

Fig. 1 に、開発した自律移動ロボット Raicho の外観を示す。搭載しているセンサは、3D-LIDAR (SureStar R-Fans-16M) 全方位カメラ (OccamVisionGroup Omni60) GNSS (Trimble AV59 Antenna, u-blox ZED-F9P) IMU (Xsens MTi-3) モータエンコーダ (後述のロータ角センサ) である。主要な外界センサである 3D-LIDAR には、複数の周囲 LIDAR を評価した結果、SureStar R-Fans-16M を採用した。最大測定距離が 200 m (公称) と長く、かつ比較的安価な LIDAR である。

全方位カメラは USB3 接続であり、このノイズが GNSS と干渉することが判明した [1]。よって全方位カメラと GNSS は、排他的にどちらかを使用している。現状では、これらは参照用のデータ取得のみで、自律走行には使用していない。

車体は、Puffin より全長が 0.2 m 短く、旋回時の掃引面積を小さくできた。差動駆動の 3 輪構成で、後輪 2 つを駆動輪とし、前輪 1 つをキャスターとしている。後輪ではなく前輪をキャスターとすることで、前進しながらの障害物回避時に車体後部を逆方向に振ることがなくなり、狭い場所での走行が容易になった。また、後方に切り返す際の旋回でも小回りが利く。

駆動輪はサスペンションなしだが、車体が 20 kg と軽量のため、エアタイヤの空気圧を下げることで十分な振動軽減ができた。一方で前輪のキャスターはサスペンションを有してお

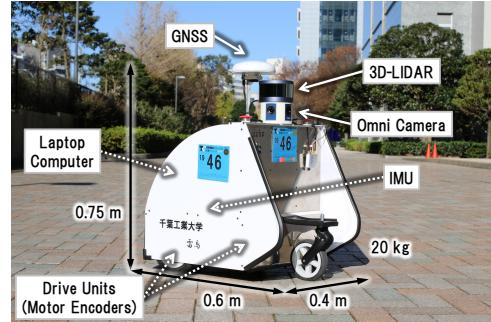


Fig. 1 自律移動ロボット Raicho

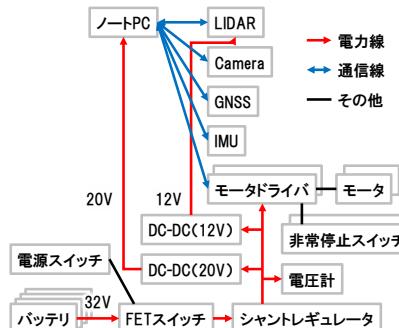


Fig. 2 電装システムの構成

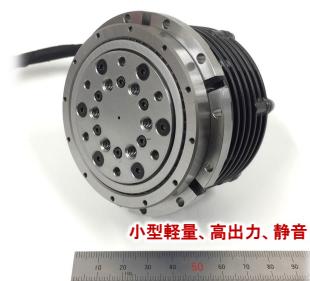


Fig. 3 駆動ユニット (モータ、減速機)

り、凹凸の大きな未舗装路での走行性能を確保している。

車体フレームは、切削加工したアルミプレート材を組み合わせた構造であり、軽量かつ高剛性を両立している。また、車体を樹脂カバーで覆うことで、雨天時の防水性能を確保した。

Fig. 2 に、電装系の構成を示す。特長の一つとして、DC-DC コンバータを介してノート PC に給電している。従来、ノート PC のバッテリ容量が稼働時間の制約となっていたが、これにより稼働時間が伸びた。またロボットのバッテリがなくなつた後も、ノート PC 自身のバッテリで作業を継続できる。ノ-

^{†1} 千葉工業大学 未来ロボット技術研究センター (fuRo)

^{‡1} Future Robotics Technology Center, Chiba Institute of Technology

Table 1 駆動ユニットの仕様

全体サイズ	ø 70 mm、H 60 mm、940 g
モータ	3相ブラシレス(5極対) 400 W
ステータ・ロータ	ø 60 mm、積厚 20 mm
トルク、回転数	0.924 Nm/10A、4,200 rpm/50V
ロータ角センサ	磁気ディスク & ホール IC
減速機	サイクロイドギア、薄型1段構成
速比	1/15 (最大効率 85% 以上)
出力部	法兰ジ式クロスローラ軸受



Fig. 4 3次元 Cartographer で構築した占有格子地図（平行投影）

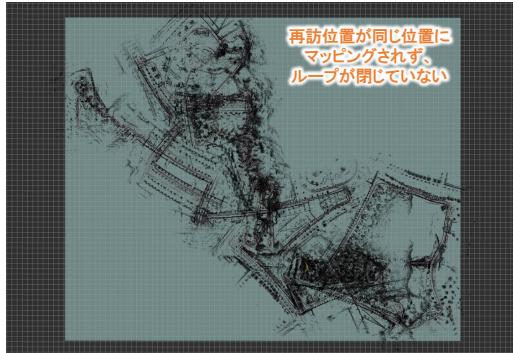


Fig. 5 3次元 Cartographer のデフォルトパラメータで構築した地図

ト PC のバッテリ交換が不要になり、実験の効率が向上した。電源には、リチウムイオンバッテリ (IDX DUO-C98、14.4 V、96 Wh) を 4 本 (2 直列 2 並列) 搭載している。交換が容易なバッテリモジュールとして、IPx7 相当の防水構造を構築し、防水コネクタで接続している。ノート PC への給電も含めて、3 時間程度の連続稼働が可能である。

2.2 駆動ユニット

車輪の駆動には、独自開発の駆動ユニット [2,3] を使用している。外観を Fig. 3 に、仕様を Table 1 に示す。

駆動ユニットは、モータ、ロータ角センサ、減速機から構成される。いずれの構成要素においても専用設計を行っており、小型軽量化と機能最適化を追求している。

モータドライバは、本実装では連続最大電流 15 A、瞬間最大電流 30 A である。ノート PC で車体のキネマティクスを統括し、CANopen の通信により左右輪の各々をモータドライバで速度制御する。駆動ユニットとモータドライバを IPx7 相当の防水構造でモジュール化し、防水コネクタで接続している。

非常停止スイッチは、モータドライバのトルクオフ機能を利用している (Fig. 2)。押下すると出力段がフリー状態となる。

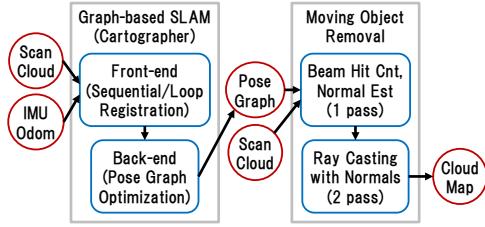


Fig. 6 SLAM と移動物体除去による 3 次元点群地図の構築

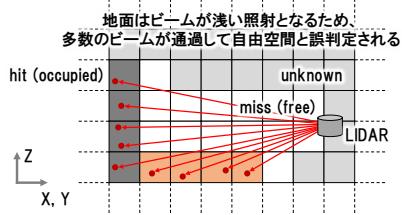


Fig. 7 占有確率を求める逆計測モデルでの入射角の影響（側面図）

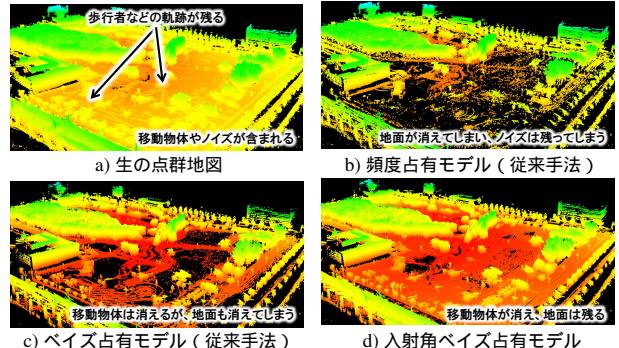


Fig. 8 フィルタリングによる移動物体除去

3. ソフトウェア

3.1 グラフベース SLAM による地図構築

事前に手動走行でセンサデータを記録し、オフラインの SLAM で地図構築する。つくばチャレンジ 2019 の全経路を網羅するように、約 2 時間で 6 km の走行を行った。グラフベース SLAM である Cartographer [4,5] の 3 次元版を用いた。

Fig. 4 に、構築した 3 次元地図の平行投影を示す。SLAM を成功させるには、パラメータ調整が必須であった。特に、適切なループ拘束を生成してループ閉じ込みできるように、ループ検出とポーズ調整に関連する多数のパラメータを調整した。Cartographer の使い方の知見は、文献 [6] を参照されたい。

Fig. 4 では、黄色の線で示すループ拘束が、網羅的に走行した経路に対して張られている。このループ閉じ込みで、3 次元地図全体の整合性が取れている。比較として Fig. 5 に、デフォルトパラメータでの SLAM の結果を示す。ループ拘束がほぼ生成されず、ループ閉じ込みができていない。

処理時間は長く、パラメータ調整の試行錯誤には困難を伴った。特に 3 次元版の Cartographer では、ポーズグラフが大きくなると処理負荷も大きくなる。オフラインでの地図構築に、CPU Intel Core i7 8750H (6 コア 12 スレッド、2.20 GHz)、メモリ 32 GB のノート PC で、3 ~ 24 時間程度を要した。

Cartographer 本体は、3 次元地図を出力する機能を持たない (可視化や 2 次元投影は可能)。また生のスキャン点群には、歩行者などの移動物体も含まれてしまう。そこで Fig. 6 のように、ポーズ調整したポーズグラフ (走行軌跡) と生のスキャ

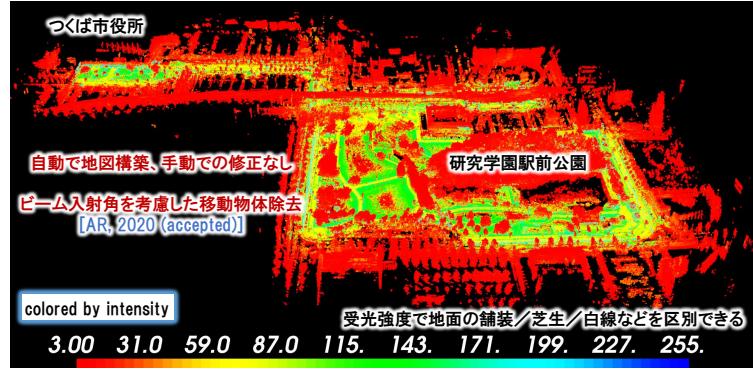


Fig. 9 グラフベース SLAM と移動物体除去で構築した 3 次元点群地図

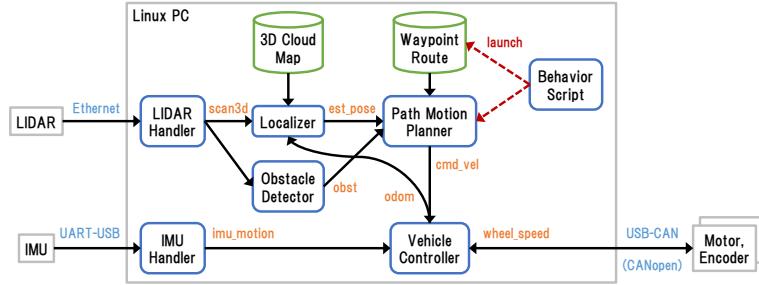


Fig. 10 自律走行システムの構成

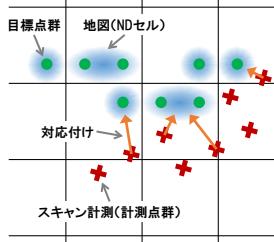


Fig. 11 NDT スキャンマッチングによる自己位置推定

ン点群から、移動物体を除去して 3 次元点群地図を構築する。

移動物体除去には、Fig. 7 のビーム入射角を考慮したペイズ占有モデル [7] を用いた。これは、1 パス目で法線方向を計算し、2 パス目で入射角を用いて占有確率を計算する手法である (Fig. 6)。入射角により、ビームが浅い照射となる地面が誤って消えてしまうことを防ぐ。Fig. 8 に、研究学園駅前公園での比較結果を示す。従来手法では地面も消えてしまっているのに対して、提案手法では移動物体が消えて地面は残っている。

Fig. 9 に、Fig. 6 のグラフベース SLAM と移動物体除去で構築した 3 次元点群地図を示す。手動での修正なしで、経路のループが閉じ、かつ移動物体を除去した地図を構築できた。また、地面の舗装 / 芝生 / 白線などを受光強度で区別できる。

3.2 自己位置推定と経路・動作計画

Fig. 10 に、自律走行システムの構成を示す。2018 年は Autoware を用いて構築したが、今年度は新たに実装を行った。

まず、自己位置推定には、3 次元 NDT スキャンマッチングを用いた。Fig. 11 にアルゴリズムの模式図を示す。NDT は、対応付けの最近傍探索が不要で、計測点がある地図のボクセルに対応付けるため、地図が大規模でも処理が高速である。

Autoware の実装を評価したところ、主に以下の 2 つの失敗原因があった。まず、車体の回転角速度が大きい場合に、スキャン歪みなどの影響でマッチングに失敗することがあった。



Fig. 12 NDT スキャンマッチングの推定位置の比較
a) 旋回や蛇行するリモコン走行
b) 開けた広場でのリモコン走行

次に、スキャン点数が多い場合などに処理が重くなり、オドメトリとの同期がずれることがあった。LIDAR の最大測定距離が 200 m に長くなったことで、当初は自己位置推定の安定性が向上すると考えていたが、実際には逆に問題の原因になる場合もあった訳である。

これらの問題に対処した自己位置推定を、新規に実装した。マッチングの失敗判定と、オドメトリとの適切な同期の機能を持つ。NDT 自体の実装は、PCL の関数を呼び出している。

Fig. 12 に、推定位置の比較結果を示す。各図の右上から左下の方向に走行した。a) は、車体の回転角速度が大きくなるように、旋回や蛇行で手動走行した。真値がないので分かれにくいが、Autoware の実装では徐々に位置がずれたのに対し、新規の実装ではマッチングの失敗を棄却して正しく推定できている。b) は、公園内の開けた広場で歩道上を手動走行した。広大な環境のため、サンプリング後のスキャン点数が

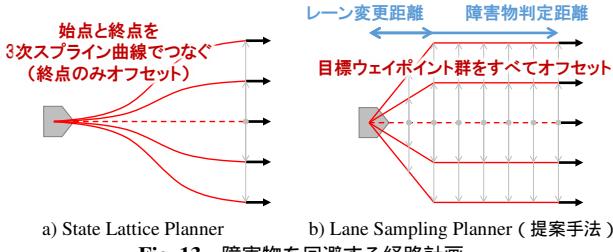


Fig. 13 障害物を回避する経路計画

多くなり、処理時間が長くオドメトリとの同期がずれやすい。Autoware の実装では推定位置が歩道から外れていっているのに対し、新規の実装では歩道上に正しく推定できている。

次に、障害物回避の経路・動作計画には、Lane Sampling Planner と呼ぶ新たな手法を実装した。Fig. 13 に、State Lattice Planner との比較の模式図を示す。ウェイポイントは 3 次元だが、経路計画を行う各時刻のロボット位置で、局所平面に投影して 2 次元の経路として扱う。

State Lattice Planner では、目標経路を表すウェイポイント群から一定距離先の一点を選び、それを終点としてオフセットしてサンプリングする。始点から終点までを、3 次スプライン曲線でつないで軌跡を生成する。よって、遠方の障害物まで衝突判定するには終点を遠くする必要があり、すると回避する位置も遅くなってしまう。また、始点と終点の 2 つのみから軌跡を生成するため、目標経路が S 字状やクランク状のように複雑な場合、追従が困難である。

一方で Lane Sampling Planner では、終点までのウェイポイント群をすべてオフセットしてサンプリングする。どの程度遠方の障害物まで衝突判定するかと、レーンと呼ぶ経路候補に対してどの程度早めに回避（レーン変更）するかを、独立にパラメータ調整できる。現在の実装では、5 m 先までの経路候補で障害物を検出し、1.5 m の間にレーン変更するよう設定した。また、すべてのウェイポイントをそのままオフセットして経路候補を生成するため、複雑な目標経路でも追従できる。

経路候補の選択は、経路長をスコアとして、衝突せずに長く走行できるレーンを選ぶ。また、チャタリングを防止するため、レーン変更ペナルティを与えている。障害物がなく、各候補が同程度のスコアを持つ場合は、元の目標経路を選択する。

さらに、経路に追従する速度を決定するために、危険ポテンシャル場を用いる。ロボット中心の危険ポテンシャル場を生成し、周囲の障害物の位置に応じて速度を決定する。これにより、障害物の近くでは減速して走行し、開けた環境では所定の速度でスムーズに走行できる。

4. 実験結果

9/14(土)に取得したセンサデータで、SLAM により地図構築を行った。次の 10/21(月)に、確認走行を達成した。その後は 11/8(金)まで、コース全域で自律走行をして、自己位置推定の失敗原因の調査と障害物回避の改良を行った。

本走行前日 11/9(土)の時点でも、自己位置推定と障害物回避の両方に問題が残っていた。さらに同日、経路追従のバグが発覚し、日中にはデバッグが完了しなかった。このバグはシミュレータ [8] で再現したので、Fig. 14 に示すように、本走行前夜にホテルでデバッグを終えることができた。

本走行当日 11/10(日)の午前中の実験では、コース全域で自



Fig. 14 本走行前夜のホテルでのシミュレーション実験の様子



Fig. 15 本走行での終了時の様子（確認走行区間の出口）

律走行したが、推定位置のずれが一度発生した。この時点では原因が分からず、マッチング失敗判定のパラメータを微調整して本走行に挑んだ。本走行では、Fig. 15 に示すように、初めてコース序盤の 270 m 地点で推定位置がずれてしまい、かつ障害物回避しないエリアに設定してしまっていたため、車止めの前でスタックして終了となった。原因を調査したところ、マッチング失敗判定にバグがあり、正しい推定位置を棄却してしまっていたことが後日判明した。

5. 結言

本稿では、つくばチャレンジ 2019 での自律移動ロボットの開発事例について述べた。ハードウェアに関しては、独自開発の駆動ユニットを用いて、小回りが利き防水性能を向上した新たな車体を開発した。ソフトウェアに関しては、SLAM に Cartographer を用いると共に、自己位置推定と経路・動作計画を新たに実装し、3 次元での環境認識と行動計画を行った。

今後は、自律走行の基盤性能を向上すると共に、より高度な技術の実現を目指す。環境認識能力を向上させ、教示経路に頼らないオンラインでの経路計画を目標とする。

参考文献

- [1] 目黒 淳一, 竹内 栄二郎, 鈴木 太郎: “ロボティクスにおける GNSS 失敗学”, 日本国際ロボット学会誌, vol. 37, no. 7, pp. 585–592, 2019.
- [2] 大和 秀彰, 萩原 一輝, 戸田 健吾, 古田 貴之: “薄型軽量なサイクロイド減速機の開発とインホイール駆動ユニットへの応用”, 計測自動制御学会 SI 部門講演会予稿集, 2019.
- [3] Hideaki Yamato, Kazuki Ogihara, Takashi Kodachi, Kiyoshi Irie, Yoshi-taka Hara, Kengo Toda, Yu Okumura, Tomoaki Yoshida, Masaharu Shimizu, Masahiro Tomono, and Takayuki Furuta: “A Partner Robot Transforming to a Vehicle: CanguRo”, Proc. of IEEE/SICE Int. Symp. on System Integration (SII), 2020.
- [4] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor: “Real-Time Loop Closure in 2D LIDAR SLAM”, Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), 2016.
- [5] Cartographer. <https://github.com/googlecartographer>
- [6] 原 祥堯: “Cartographer を用いた 3D SLAM”, 第 6 回 3D 勉強会, 2019. <https://www.slideshare.net/hara-y/cartographer-3dcvtech-2019>
- [7] Yoshitaka Hara and Masahiro Tomono: “Moving Object Removal and Surface Mesh Mapping for Path Planning on 3D Terrain”, Advanced Robotics, (accepted).
- [8] 吉田 智章, 入江 清, 原 祥堯, 鈴木 太郎, 西村 健志, 大和 秀彰, 友納 正裕: “つくばチャレンジ 2019 の事前検証が困難な環境のための移動ロボットシミュレータ”, 計測自動制御学会 SI 部門講演会予稿集, 2019.