

協働ロボットによる検査作業省力化の研究（自律走行パート）

Research to save labor in visual inspection by a collaborative robot (Autonomous driving part)

電子・機械技術部 機械・加工科 菅野雄大

本研究は、製造現場の省力化を目指し、自律走行ロボットを用いた作業支援技術の開発を行った。特に、アーム式協働ロボットの効率的な稼働を実現するために、自律搬送ロボット（AMR）の活用に焦点を当てた。AMRに用いられる2D LiDARの自己位置推定に関し、特徴点が少ない環境での位置推定の課題を実験により検証した。さらに、ROS 2を活用して環境構築及び開発を行い、その知見を県内のロボット開発企業へ技術移転した。

Key words: 協働ロボット、省力化、自律走行

1. 緒言

当所では、製造現場における作業の省力化を目指し、県内企業へのロボット導入や活用に関する技術支援や、協働ロボットによる検査作業省力化の研究を行っている。本稿では、その研究の一部として実施した自律走行パートについて記す。

アーム式協働ロボットは、固定位置で稼働することが多く、その場合には、検査対象ワークを適切なタイミングで供給することが重要である。近年、資材や作業対象の搬送作業を自動化し、停滞を防ぎ、生産効率を向上させるために、自律搬送ロボット（AMR）が注目されている。

この自律搬送ロボットには一般的に2D LiDAR (Light Detection And Ranging) が用いられる。これは、低コストでロボットの自己位置推定や障害物回避が可能であるが、柵や通路が規則的な倉庫や工場といった特徴点が少ない環境では、環境地図とのスキャンマッチングがどこの位置でも合ってしまうため、ロボットの位置を一意に確定できない状況に陥る退化¹⁾と呼ばれる課題がある。

本研究では、低コストな2D LiDARにより、特徴点が少ない環境でのロボットの安定した自己位置推定を目指し、シミュレーション及び実環境での実験により課題を抽出した。

また、これまで当所の自律走行ロボットの研究では、オープンソースのロボット開発用フレームワークのROS (Robot Operating System) を使用し開発してきた。ROSは、ROS 1とその後継であるROS 2のバージョンがあるが、基本的には互換性がない。また、ROS 1のサポートが2025年5月で終了することを受け、これまでROS 1で開発してきた企業やこれからロボットの開発をする企業からはROS 2に関する技術相談が増えてきている。本研究では、ROS 2で環境構築・開発を行い、本研究で得たROS 2の知見やノウハウを県内ロボット開発企業へ技術移転することを目指した。

2. 実験

2. 1. 使用ロボット

実験には、VSTONE社の自律搬送ロボット MegaRoverV3を用いた。外観を図1に、仕様を表1に示す。MegaRoverV3は、ROS対応の研究開発用台車ロボットである。リフト機構や2D LiDARをオプションで追加することで、物体搬送や障害物回避、自己位置推定などの多様なタスクを実施することが可能である。

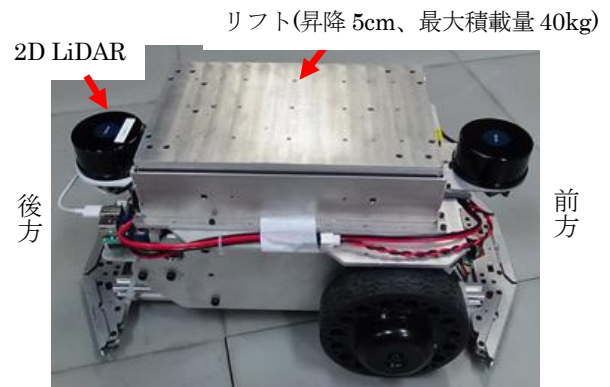


図1 使用ロボット MegaRoverV3

表1 MegaRoverV3の仕様

項目	仕様
バッテリー	24V シール鉛バッテリー 288Wh
最大積載量	約 40 kg
最大速度	1.6 m/s
外形寸法	W357 × D359.5 × H150 mm
リフト機構	最大リフト高さ 5cm
センサ	2D LiDAR、バンパーセンサ

MegaRoverV3に搭載されている制御基板 VS-WRC058 には、ESP32-WROOM-32 マイコンが使用されており、組み込み向け ROS 2 の micro-ROS を用いて制御が可能である。

地図作成やナビゲーションの制御には、ROS 2 がインストールされた Raspberry Pi 4 Model B を用い

た。micro-ROS を通じて、ESP32 マイコンから Raspberry Pi にオドメトリ等の情報を送信し、逆に Raspberry Pi から ESP32 マイコンに計算された速度指令値を送ることでロボットを制御する。ROS 2 は ROS 1 に比べリアルタイム性やセキュリティなどが改善されている。ROS 2 のディストリビューションは Humble を用いた。また、地図作成やナビゲーションには、MegaRoverV3 の基本的な制御方法などのサンプルを提供した ROS 2 パッケージである megarover_samples_ROS 2²⁾ を用いた。

2. 2. 環境地図作成

2. 2. 1. シミュレーション環境

環境地図の作成には、自己位置推定と地図作成を同時に行う技術である SLAM (Simultaneous Localization and Mapping) を用い、今回、LiDAR を用いた SLAM の 1 つである slam_toolbox を用いた。slam_toolbox は、ROS 2 用の 2D マッピングや自己位置推定を行うためのライブラリである。

今回、実機で実験を行う前にシミュレーションにより、自律走行などのロボットの挙動を確認した。開発初期段階でシミュレーション環境を活用することで、予期しない不具合によるロボットの損傷や、問題が発見されないまま進むリスクを抑えることができる。シミュレーションには Gazebo を用いた。Gazebo は ROS の主開発団体である Open Robotics が開発しており、ROS との親和性が高いソフトウェアである。図 2 のように、Gazebo 用に MegaRoverV3 のモデルファイル³⁾ が用意されており、このモデルを用いた。また、Gazebo のバージョンは、ROS 2 Humble に対応している Gazebo Fortress を用いた。

図 3 (a) のように、Gazebo 上に作成した壁のモデルに囲まれた環境でロボットを手動走行させ、環境地図を作成した。図 3 (b) に、センサデータ可視化ソフトの rviz2 で環境地図作成時のセンサデータを可視化した様子を示す。

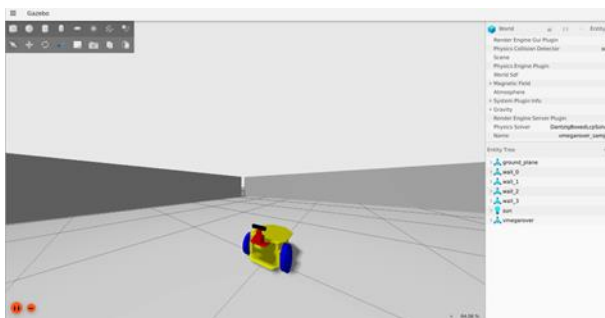


図 2 シミュレーション (Gazebo) 上のロボットモデル

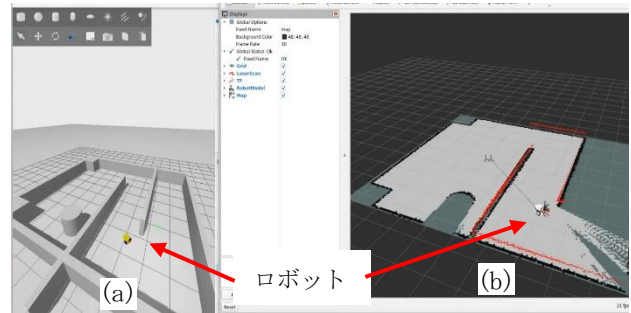


図 3 Gazebo 上での地図作成時の様子

((a) : 環境モデル、(b) : rviz2 のセンサデータ可視化)

2. 2. 2. 実環境

特徴点が少ない環境において、環境地図を作成できるかを検証するため、図 4 のような壁が続く廊下で環境地図作成の実験を行った。



図 4 地図作成環境 (廊下)

2. 3. ナビゲーション

作成した環境地図を使用し、地図上で設定した目標地点まで経路計画や経路追従等のナビゲーションが可能かシミュレーションと実環境で検証した。ナビゲーションソフトは Nav2 を用いた。Nav2 は、ROS 2 の自律走行ロボット用ナビゲーションのスタック (機能別パッケージ群) で、経路計画、障害物回避、自己位置推定などを行う。

2. 4. 自律搬送に向けた検討及び課題抽出

図 5 のように、キャスター付き棚の下にロボットが潜り込み、リフトで棚を持ち上げ搬送することを想定し、その際の検討事項や課題を抽出した。

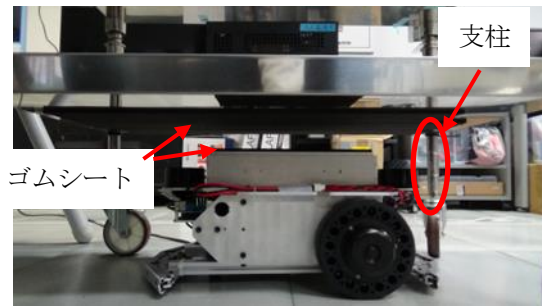


図5 キャスター付き棚の下に潜り込んだロボットの様子

3. 結果

3. 1. 環境地図作成結果

3. 1. 1. シミュレーション環境

図6にGazebo上で作成した地図を示す。図のようにグレースケール画像で表現され、白い部分は空間、黒い部分は障害物や壁を示す。壁の形状など正確に、歪みの無い地図を作成することができた。

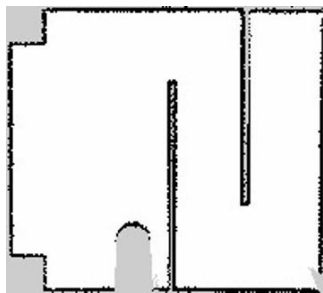


図6 Gazebo上で作成した地図

3. 1. 2. 実環境

実環境の室内で環境地図を作成した結果を図7に示す。当初LiDARはロボット前方に取り付けた1つのみ使用していたが、図7(a)のように、歪んだ地図となった。

原因として、図1のように、ロボット上部中央にはリフトが位置しており、1つのLiDARでは、測距データがリフトに当たってしまい、 360° の視界を確保できていない。そのため、測距点数が少なくなることで、スキャンマッチングが不安定になり、結果として地図作成の精度が低下し、地図のズレや歪みが発生したと考えられる。

そこで、3Dプリンターで作成した固定治具により、ロボットの後方にもLiDARを取り付け、ロボット前後のLiDARの測距データを統合することで、 360° の視界を確保した。測距データの統合にはira_laser_tools⁴⁾を用いた。2台のLiDARを用いて地図を作成した結果を図7(b)に示す。2台のLiDARの統合した測距データを使用することで、図の赤丸で囲った机の直線的な形状などが正確で、歪み無く地図を作成できた。

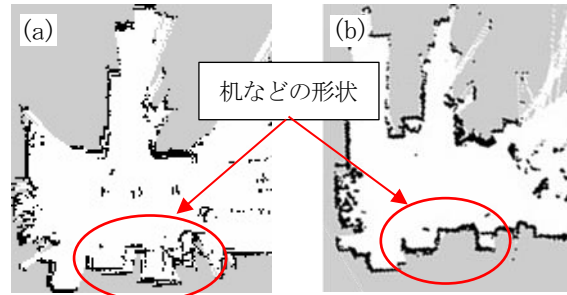


図7 室内の環境地図
(a) : LiDAR 1台、(b) : LiDAR 2台

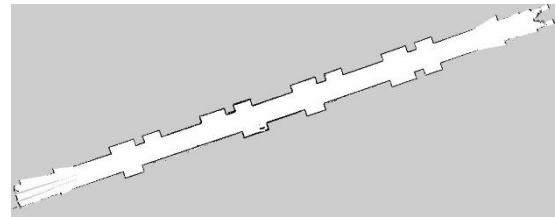


図8 廊下の環境地図(1)

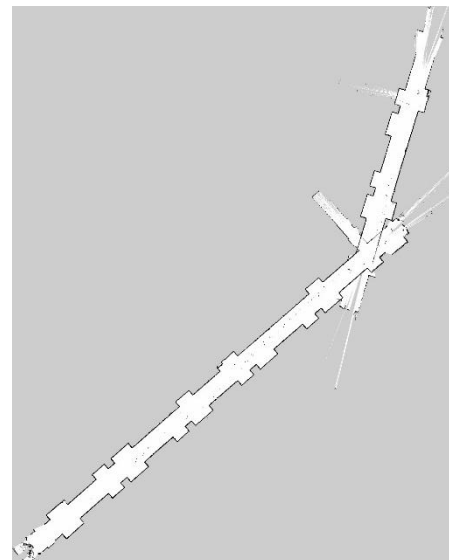


図9 廊下の環境地図(2)

また、廊下で作成した地図の結果を図8に示す。直線距離約38[m]を走行させた。

一方、図9に示すように、直線距離約48[m]と長い距離を走行させたところ、地図にずれが生じた。

3. 2. ナビゲーションの結果

3. 2. 1. シミュレーション環境

図10(a)のGazebo上のモデルをナビゲーションさせた時の目標経路やセンサデータをrviz2で可視化した様子を図10(b)に示す。環境地図は図6を用いた。図10(b)のように、ナビゲーション時はコストマップを使用し、これを基に障害物を避けた経路を計画する。コストマップは、ロボット周囲の環境をコスト値で表現し、色で危険度を示す。赤は障害物の領域を表し、

紫は障害物からマージンを取った領域を表す。

rviz2 の地図上で目標地点を設定することで、目標地点に対し経路生成、経路追従し、目標地点まで自律走行できることを確認した。

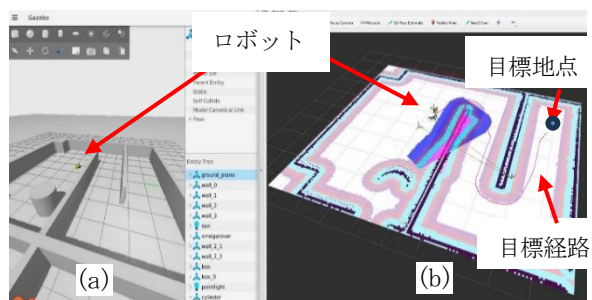


図 10 Gazebo 上でのナビゲーション
(a) : 環境モデル、(b) : rviz2 のセンサデータ可視化)

3. 2. 2. 実環境

シミュレーションと同様に、実機でナビゲーションを行うため Nav2 を動作させたところ、図 8 の作成した地図が表示できなかった。使用した Raspberry Pi の環境設定の問題や、Nav2 を動作させるために必要な処理能力が足りていない可能性が考えられる。そこで、Jetson Orin Nano を用いた。Jetson Orin Nano は、NVIDIA が開発した AI 推論に特化した小型コンピュータであり、Raspberry Pi 4 と比較し高い処理能力を持っている。Jetson Orin Nano で環境を構築し直したところ、図 11 のように地図を表示することができた。

しかし、目標地点を設定してもロボットが動作せず、自律走行するまでは至らなかった。

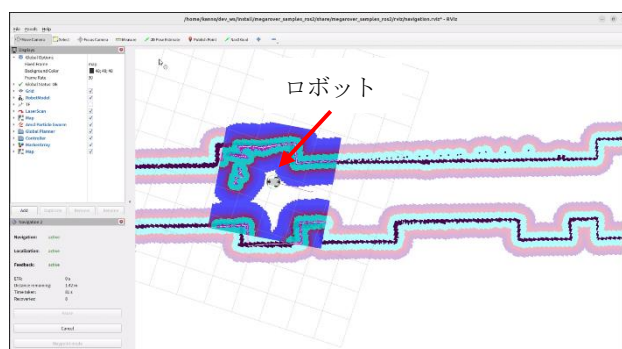


図 11 ナビゲーション実行時の操作画面

3. 3. 自律搬送に向けた検討及び課題抽出

ロボットがキャスター付き棚の下に潜り込んだ際、図 5 の赤丸で囲った棚の支柱を LiDAR で認識してしまうため、データから省くために、laser_filters⁵⁾ パッケージを用いた。laser_filters では、LiDAR の測距データに対し、角度範囲を指定することでその範囲の測距データを取り除くことができる。laser_filters を用いることで、キャスター付き棚の下にロボットが潜

り込んだ際に、ロボット周辺に位置する 4 本の支柱の測距データを取り除くことができた。

また、図 5 のように、ロボットのリフト部とリフトが接触するキャスター付き棚の面にゴムシートを貼ることで、接触した面同士の摩擦により棚を搬送できるようにした。

リフトの上下方向の操作は、速度指令値 Cmd.velocity.z で制御できる。リモコンの手動操作により、キャスター付き棚を搬送できた。

しかし、ロボットが棚下に潜り込むことで、LiDAR 以外のカメラを使用する場合に視界が悪くなることや、ナビゲーション時に棚の形状やサイズを考慮する必要があるなどの課題がある。

4. 考察

図 9 の地図がずれた原因として、特徴の少ない場所を走行したことで、ロボットの自己位置推定の誤差が積み重なり、地図がずれた可能性がある。また、地図のずれが生じた場所は、片側の壁しか見えない区間であり、センサ情報が不足したことで退化が生じたと考えられる。その他、ロボットが旋回する際に速度が速すぎたことで、スキャンマッチングの精度が落ちた可能性が考えられる。今回は直線のコースで走行したが、実際の現場では、SLAM のループクローズが機能するようなルートを走行させることが重要であると思われる。ループクローズ機能は、再び訪れる場所を認識し、地図の誤差を修正する機能である。これにより、ロボットは以前の位置に戻った際に地図の整合性を保つことができる。その他、AR マーカーを使用した位置推定手法を用いることで、特徴点の少ない環境でも安定した地図作成と自己位置推定が可能であると思われる。

5. 結言

協働ロボット・自律搬送ロボットを活用した外観検査作業の省力化を目指し、自律搬送ロボットが苦手とする特徴点の少ない環境を想定した実験により課題を確認した。

LiDAR のみの SLAM では、特徴点の少ない環境でスキャンマッチングがうまくいかず地図がずれた。今後は、AprilTag と TagSLAM を用いることで、この問題を検証していく予定である。AprilTag は、AR マーカーと呼ばれる特定のパターンを持つ視覚的なマーカーで、カメラで認識することにより高精度な位置推定を行うことができる。TagSLAM は、この AR マーカーを利用して自己位置推定および地図作成を行う手法であり、特徴点の少ない環境でも安定した地図作成と自己位置推定が可能であると思われる。

また、今回の研究で得た ROS 2 のノウハウや基本的な知見を活かし、現場支援事業により県内ロボット開発企業への技術支援に繋げた。今後も引き続き、技術支援を行っていく予定である。

参考文献

- 1) 友納正裕. SLAM 入門. 株式会社オーム社, 2018 年 4 月, p.105, 129-131.
- 2) “atinfinit/megarover_samples_ROS 2”. GitHub.
[https://github.com/atinfinit/megarover_samples_ROS 2](https://github.com/atinfinit/megarover_samples_ROS_2), (参照 2025-2-14).
- 3) “vstoneofficial/megarover3_ROS 2”. GitHub.
[https://github.com/vstoneofficial/megarover3_ROS 2](https://github.com/vstoneofficial/megarover3_ROS_2), (参照 2025-2-14).
- 4) “nakai-omer/ira_laser_tools”. GitHub.
https://github.com/nakai-omer/ira_laser_tools/tree/humble, (参照 2025-2-14).
- 5) “ros-perception/laser_filters”. GitHub.
https://github.com/ros-perception/laser_filters, (参照 2025-2-14).