

# ライトローバーを用いた ROS2 によるオープンキャンパス用体験型プログラムの提案

Development and implementation of hands-on program for open campus using Lightrover

20216072 小松 大介 [関澤研究室]

## 1 はじめに

日本大学工学部では、毎年高校生に向けてオープンキャンパスを開催している。本研究室でもオープンキャンパスに向けてライトローバーという自律移動する台車ロボットを使用した体験型プログラムの設計と開発が行われてきた。しかし、未完成の機能がいくつか存在し、これまでの研究では ROS1 での開発をしていたが、2025 年 5 月をもって ROS1 のサポートが終了することもあり、今後は ROS2 で開発する必要があると考えた。よって、ROS2 に移行する際の変更点を確認し、一通りの機能を ROS1 で開発することで、ROS2 への移行をやすくすることを目的とした。

体験型プログラムを実現するために、本研究ではヴィストン株式会社が販売するライトローバーを用いる。ライトローバーは、RaspberryPi4 ModelB, YDLiDAR X2(LiDAR),モータ制御基板[VS-WRC201],カメラが搭載されている卓上サイズの台車型ロボットである。ライトローバーは、搭載されているこれらの機器を用いて、自律制御や RaspberryPi4 の演算能力を活用した SLAM の実行、YDLiDAR を用いて周囲の環境を的確に認識する機能などの高度なロボティクス制御が実現できる。

## 2 ROS と ROS2

### 2.1 ROS とは

ROS は、Open Robotics によって開発、メンテナンスされているロボット用のミドルウェアである。分散処理が求められる複雑なロボットシステムを制御できる機能を備えており、世界中の研究者や開発者が作成した豊富なライブラリを使用することができる。ロボット制御システムの作成を効率化できることから、人型ロボットから車両型ロボット、水上・水中ロボットやドローンに至るまで、幅広い分野で活用されている。

### 2.2 ROS と ROS2 の相違点

ROS と ROS2 の相違点として 5 つ挙げられる。

- ・商用利用も視野に開発
- ・群ロボットにも対応
- ・組み込みレベルのプラットフォームも想定
- ・リアルタイムシステム
- ・低品質なネットワークも想定

以上より、研究用以外でもさらに幅広くロボットの開発ができるようになっている。

## 3 先行研究

先行研究では、体験型プログラムの設計に基づいて体験型プログラムの開発と実装を進めている。本章では、先行研究にて定められた要件定義と基本設計を記す。

### 3.1 要件定義

要件定義は、図 1 に示す要求分析結果より、体験型プログラムの目標を「日本大学工学部情報工学科への進学の後押し」に決定し、開発テーマを「ライトローバーを

用いた、SLAM と自律走行を体験するプログラム」に決定している。

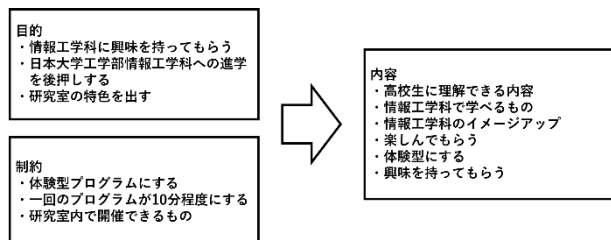


図 1. 要求分析結果

## 3.2 基本設計

体験型プログラムの開発テーマより、システムの主要な機能は SLAM と自律走行の二つとされている。開発する体験型プログラムでは、始めに迷路状のコースをライトローバーで走行し、コース内にあるキーワードを探す。そのキーワードを GUI に入力することで、SLAM でマッピングしたコースを自律走行する。表 1 で示す機能一覧表より、実装が望まれる機能は「ライトローバーの操作」「SLAM」「自律走行」「カメラ機能」「GUI」の 5 つである。この基本設計に基づいて望まれる機能の開発と実装を行う。

表 1. 機能一覧表

| 機能一覧               |      |
|--------------------|------|
| PCとRaspberry Piの連携 | SLAM |
| PCに表示するGUI         | 自律走行 |
| ゲームパッドによる          | カメラ  |
| ライトローバーの走行         | モーター |

## 4 ライトローバー用プログラムの機能

前項にある先行研究にて定められた体験型プログラムの設計に基づき、ライトローバー用プログラムの開発と実装を行う。PC と RaspberryPi4 を連携させるために同じ開発環境で作る必要があるため、本研究では Ubuntu MATE で開発を行う。

### 4.1 コントローラーによるライトローバーの走行

ライトローバーに、コントローラーから入力することで走行するプログラムを実装する。コントローラーは DualShock®4 を使用し、ライトローバーと Bluetooth 接続を行って走行を可能にする。本プログラムは、複数のノードを launch ファイルで起動させることによりライトローバーの走行が実現できる。

### 4.2 SLAM

SLAM を実装するにあたって、ROS に存在する SLAM パッケージ gmapping と、ロボットの自律移動に必要なパッケージを集めたメタパッケージ navigation の二つのパッケージをセットアップした。gmapping を起動するための launch ファイルを起動することで、ROS に付随する可視化ツールである「Rviz」を起動する。Rviz では LiDAR から得られる情報を地図としてリアルタイムに表示でき、navigation を実行することで環境地

図の画像ファイルとデータファイルを保存することができる。

### 4.3 自律走行

ロボットの自律移動を行うためには自己位置推定や移動経路のプランニングの技術が必要だが、ROS では Navigation Stack という形でこれらの機能が提供されている。自律走行は Rviz を用いて行われ、2D Pose Estimate ボタンで初期位置の設定を行い、ライトローバーを初期位置にセットし、次に目標位置を 2D Nav Goal ボタンにて設定する。目標位置の設定を行うことで即座にライトローバーが走行を開始し、目標位置に到達すると自動的に停止する。

### 4.4 カメラ機能

ROS には、カメラを利用するためのライブラリが複数存在し、それらのセットアップを行う。カメラのサンプルプログラムを実行すると、カメラから得られた画像が表示され、赤色の検出により該当部分が赤枠で囲まれる。ライトローバーはその赤色に追従して動作するが、コントローラーの操作を受け付けなくなるため、プログラムを一部編集し、赤枠の表示と追従の機能を削除することで、コントローラーによる操作が可能になる。

## 5 ROS2 への変更点

ゲームパッドを使用するために ROS では Joy パッケージを使用しているが、ROS2 では transforms3d や roshumble-joy などの各種パッケージをセットアップする必要がある。Transforms3d をセットアップすることで、回転や拡大、せん断、平行移動の幾何学的変換を変換させることができ、コントローラーで操作する際の移動以外の機能も追加できるのではないかと考えた。

SLAM の実行においては、slam-toolbox パッケージを実行することで Rviz を起動することができる。slam-toolbox になったことにより、同期/非同期マッピングや、マップの局地化など、他の SLAM 方式に比べて豊富なオプションが追加されている。さらに、生成された地図は gmapping に比べて信頼性の高い地図となるため、より正確に地図を表示できる。

自律走行においては、Nav2 というパッケージで自己位置推定や移動経路のプランニングの機能が提供されており、オドメトリの取得や LiDAR の制御を行うノードをまとめた launch ファイルを実行することで地図を読み込み、Nav2 を実行することで表示される。Nav2 になったことにより、状態管理がデフォルトで使用可能になり、パッケージも多数追加され、アプリケーションによって手法を選択できるようになった。

カメラ機能においては、どの OS でも同じ方法でセットアップできるため、ROS2 で開発するうえでも変更はない。

## 6 GUI の開発

体験型プログラムの実装において、PC に表示する GUI を作成する必要がある。GUI の機能として、SLAM で作成しているマップを表示する機能、キーワードを探すためのカメラを表示する機能、SLAM 中に集めたキーワードを入力する機能、キーワード入力後、ボタンを押すことで自律走行をする機能がある。これらの機能の開発及び実装を目標とする。

### 6.1 GUI の開発環境

GUI の開発は、ノートパソコンを使用して行う。IDE は、Java アプリケーションの開発に使用される Eclipse を使用し、GUI フレームワークは Swing を使用した。Eclipse には Swing での開発で扱えるプラグイン SwingDesigner が存在し、ボタンやパネルなどを貼り付ける直感的な作成方法で GUI を作成できるこのプラグインで GUI の開発を進める。

### 6.2 GUI の開発結果

作成した GUI を図 2 に示す。本研究では、文字の表示、キーワードの入力欄、マッピング完了のボタンの作成が完了し、RaspberryPi4 と PC の連携が前提となる SLAM 作成中の画面の表示、カメラ画像のリアルタイム表示、キーワード入力後の自律走行の部分が未完成である。

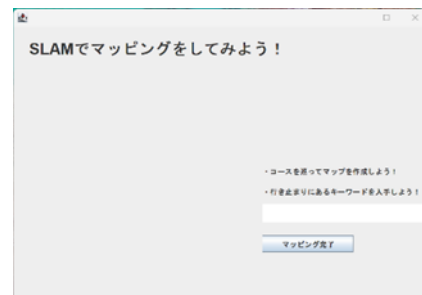


図 2.GUI の開発結果

今後の展望として、図 3 のように SLAM 中のマップの表示、カメラ画像のリアルタイム表示の枠を追加し、キーワードを入力しマッピング完了を押すことで自律走行を行う仕組みを考えている。



図 3.体験型プログラム用 GUI イメージ

## 7 おわりに

本研究では、ROS2 に移行する際の、ライトローバーの各機能で必要な変更点の確認ができた。しかし、変更点通りに機能するかは未確認のため、ROS2 による開発を実際に行う必要がある。

さらに、GUI を開発するにあたって、未実装の機能には PC と RaspberryPi の連携が必要になるため、連携を行ったうえで GUI を開発することが課題となる。

## 参考文献

- [1] ライトローバーWebDoc  
[https://vstoneofficial.github.io/lightrover\\_webdoc/](https://vstoneofficial.github.io/lightrover_webdoc/)  
更新日 2023 年 9 月 28 日  
(2025 年 2 月 7 日アクセス)