

和歌山大学協働教育センター クリエプロジェクト
＜2020年度ミッション成果報告書＞

プロジェクト名：レスキューロボットプロジェクト

ミッション名：複合センサを用いた自己位置推定

ミッションメンバー：システム工学部 2年小森日和 システム工学部 3年今中新平
システム工学部 3年唐津祐輝 システム工学部 2年石橋東紗
システム工学部 2年上岡留巳

キーワード：自己位置推定、ROS、ロータリーエンコーダ、オドメトリ、LiDAR

1. 背景と目的

レスキューロボットプロジェクトでは、「ロボットに関する技術を習得し、レスキューロボットコンテストで入賞すること」を最終目標としている。そのため例年はレスキューロボットコンテストに出場するための機体の製作などに取り組んできた。

しかし今年度は新型コロナウイルスの影響もあり、大学内での活動が著しく制限されることが予想されたため、直接レスキューロボットコンテストにはかかわらないかもしれないが、技術を習得するという点に立ち返って、学習を中心とした活動にすることとした。

そのためにこれからの技術として何が必要になるかを考えたときに私たちは、自律制御が必要であると考えた。人が操縦を行うことにより操縦者の技能や熟練度が機体の性能を左右することがあり、自律制御が実装できれば安定した動作を行うことが可能になる。

自律制御のために必要な技術は様々存在しており、ロボットが自分のいる位置を推定する自己位置推定や自分の周辺の地形などを認識する地図作成や移動ルートを作成する経路作成などが存在する。これらの技術は膨大であり、簡単に理解できるものではないため今年度は自己位置推定に関する学習を中心にしていくこととした。

また ROS (Robot Operating System) というオープンソースソフトウェアは、可視化ツールである RViz や動力学に基づいたシミュレーションが行える Gazebo など様々なパッケージが標準で利用することができ、自律制御に関係するライブラリなども多数公開されているためそれらが使用できるように ROS の導入も目指した。

そのためにこのミッションの目的としては、ROS や自己位置推定についての理解を中心に、事前知識の収集を行うこととシミュレータ上での自己位置推定の実装としました。

2. 活動内容

2.1 ROS

まず ROS のシステムや使い方から全員で学習を始めた。ROS を安定的に動作させるには Linux の Ubuntu をインストールした PC を用意する必要があったが、オンラインでの活動となり、Linux の PC を個人で用意することは難しかったため WSL という Windows の機能を使うことで仮想環境により Linux の環境を用意して活動を行った。

2.2 自己位置推定

研究事例などから現在の手法としてこういったものが存在するかを調べ、その原理などについて学習を進めて、特に amcl(Adaptive Monte Carlo Localization)というプログラムによる自己位置推定について学習した。

2.3 シミュレータによる自己位置推定の実装

gmapping というプログラムを用いて地図データを生成し、その地図データをもとに、amcl というプログラムを用いて、自己位置推定を行った。

図1のようなシミュレータを用意して、自己位置推定を実装した。

図2はシミュレータ上で地図データをもとに自己位置推定を行っているときの画面である。緑の点々が自己位置の候補である。

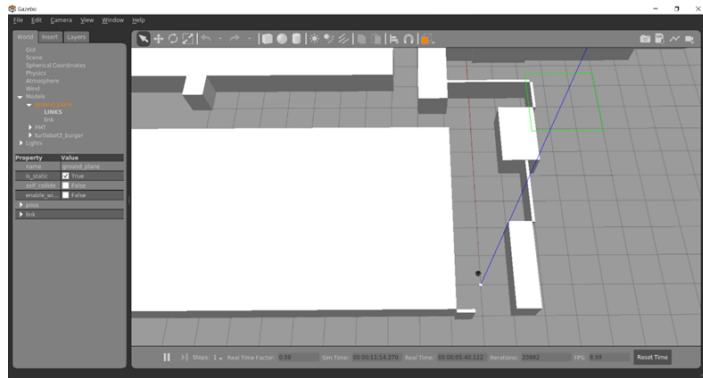


図1.実行中のシミュレータ画面 (GAZEBO)

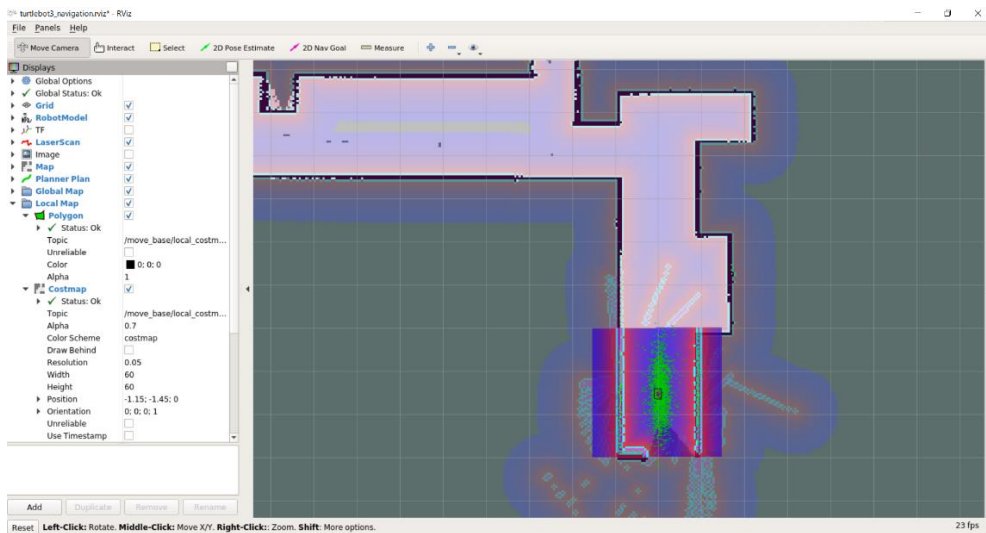


図2.自己位置推定実行中のシミュレータ画面 (Rviz)

3. 活動の成果や学んだこと

3.1 ROS

ROS では一つのプログラムの単位を Node (ノード) といい、そのノードごとにメッセージと

呼ばれる形で必要なデータの通信が行われる。そのためソフトウェアの再利用性という点で優れており、ハードを変更しても簡単に動かすことができるためライブラリなどの公開が盛んにおこなわれている。

またその複雑になりやすい Node を可視化し、管理してくれるパッケージも存在しているためプログラムの管理も行いやすい。

3.2 自己位置推定

自己位置を推定する方法としては内界センサや外界センサをつかったものが存在する。

内界センサを使った自己位置推定として最も有名であるのは、ロータリーエンコーダを使ってタイヤの回転角から移動距離を計算し、自己位置を推定するオドメトリであるが、タイヤの滑りなどから誤差が発生しやすい。ほかに IMU、ジャイロでタイヤの空転や滑りなど検知して補正をくわえる手法も存在する。

外界センサを使った自己位置推定としては Lidar などの光学センサを使って距離を測定して作成済みの地図データと照らし合わせることで位置を推定する手法が存在する。

ただどの手法も単一のセンサの場合は誤差を修正できないなど、問題点があった。

そこで複数のセンサ情報から推定位置を計算しなおすことでそのような問題を解決している。複数センサによる推定結果をスキャンマッチングやカルマンフィルタなどによって組み合わせることで複合センサによる自己位置推定は実装されている。

3.3 amcl(Adaptive Monte Carlo Localization)

amcl とは適応モンテカルロ位置推定を行うライブラリでオドメトリと Lidar から自己位置推定を行い、格子ベースを用いて、2つの情報を統合して計算し、推定を行う。

格子ベースとは、自己位置推定したい空間内で目印となるランドマークの情報があらかじめ得られなかった場合、有効な方法である。目印となるものは、ランドマークではなく任意の障害物とし、Lidar から得られた情報をもとに認識する。空間をある程度の大きさの格子状に分割して捉え、障害物があると判断した格子を塗りつぶし、障害物がない格子はオープンにしておく。

あらかじめランドマークがわかっている状態から自己位置推定を始める特徴ベースと比べると計算量は多くなるが、汎用性ははるかに高い方法になる。

センサの情報を統合する時はパーティクルフィルタを使って計算を行う。このフィルタの特徴は、計算量が多くなる代わりに外乱に強い点である。外乱とは、タイヤのスリップや障害物への、不意の衝突による弾み、光学センサの乱反射などのノイズのことをいい、パーティクルフィルタでは、これらの外乱がランダムな値をとることを前提にして計算することができる。つまり、より現実に即した推定を実現することができる。

そのため、推定値が、尤度の低い外れ値であっても捨てずに計算を行い、尤度が高い推定値はそのまま残して計算を行い、尤度が低いものは、リサンプリングにより調整する。

このように、オドメトリから得た情報に含まれるノイズの値が法則性を持たないことを、考慮した上で扱うことができるのがパーティクルフィルタの利点となっている。

4. 今後の展開

今年度は新型コロナウイルスの感染拡大の影響で対面での活動ができなかったため、シミュレータ上での自己位置推定の実装は行えたが、実機を用いての実装が行えなかった。

そのため、今後の展開としては実機の製作、その実機を使用しての地図作成、自己位置推定の実装を進めていく。そしていずれ操作をせずとも自律制御によりロボットを動かせるようにしていきたいと考えている。

また、レスキューロボットコンテストの競技内容が多数変更された。そのため、コンテストでの実装に向けて競技内容に対応するための修正やそのためのアームの機構を搭載するなどのレスキューロボットコンテストに向けた活動もまた頑張っていきたい。

5. まとめ

このミッションを通して、ROS についての理解が深まった。先述のとおり ROS はロボット開発に適した多くのライブラリパッケージが公開されており、提供されている技術は多岐に渡る。機能の拡張性も高いことから、ROS の理解を深めることは来年度以降の活動に対しても大きなメリットとなった。

自己位置推定については基本的な考え方やどんな手法が用いられているかなど全体としては表面的になってしまった点もあるが、網羅的に学習することができ、今回学んだことをこれからの活動に生かしていきたい。

シミュレータ上での自己位置推定の実装においては、今回はオンライン上での活動だったため人によって開発環境が大きく異なり、エラーなどで作業が思うように進まなかったが、なんとかシミュレータ上での実装は行うことができた。今後は、実機での実装を進めていきたい。