2023 年度 卒業論文
ロボットの自律走行時における停止ボタンの動作認識

奈良女子大学 生活環境学部 情報衣環境学科
生活情報通信科学コース 4 回生
新出研究室 20480368 森本雅衣

2024 年 2 月

目次

1 **はじめに** 3

1.1 研究背景 3

1.2 先行研究 3

2 **使用ロボットと制御** 3

2.1 Arno 3

2.2 センサ 3

2.3 ROS 4

2.4 停止ボタン 5

3 **課題** 6

3.1 中之島ロボットチャレンジ 6

3.2 課題点 6

4 **取り組み** 7

4.1 方針 7

4.2 実装内容 7

4.3 実行結果 8

5 **まとめ** 10

6 **謝辞** 10

1　はじめに

1.1　研究背景

近年、少子高齢化による人口減少や働き方改革等による労働力の不足には拍車がかかっている。その問題の解決策として、自らの環境認識と情報をもとに目標達成への最適な行動を行う自律型ロボットへの関心が高まっている。新出研究室はロボットの自律走行に取り組んでおり、自律走行の正確性を向上させるために様々な取り組みを行ってきた。本研究では、何らかの原因により障害物を検知できず衝突してしまうことを防ぐために必要不可欠な“ 停止ボタン ”の押下を認識することで、従来の走行で問題だった点の改善を行う。具体的には、ロボットの制御プログラムでしか認識していなかった“ 停止ボタン”の押下を、自律走行を行うプログラムで認識し、自律走行プログラム側でそれに応じた動作を行うことを目指す。

1.2　先行研究

先行研究では Arno(2.1 節参照) を利用した地図の作成、また地図上の座標を指定しウェイポイントとしている。ウェイポイントを通る自律走行を行う段階まで先行研究で可能となっている。しかし、自律走行プログラムが“ 停止ボタン”の押下を認識していない状態だったことが原因で、自律走行中に停止ボタンを押した際、高確率で自己位置が分からなくなるという問題が発生していた。

2　使用ロボットと制御

本節では、本研究で使用したロボットと用いたセンサ、制御ソフトウェアについて述べる。

2.1　Arno

本研究では北陽電機開発の自律走行ロボットである Arno(図 1) を用いている。Arno は 2 種類のセンサを 3 つ搭載しており、ロボットの制御には Ubuntu 18.04.5 LTS を用いている。地図の作成や自律走行以外に、手動で操作する際は人間がコントローラを通して走行の指示が可能である。

図 4、図 5 は実際に Arno のセンサを搭載している位置と、認識している範囲を簡単に図で表したものである。

2.2　センサ

Arno に搭載されているセンサは以下のものである。

a) 2 次元測域センサ URM-40LC-EW
　2 次元測域センサ (図 2 参照) は Arno の前後に積むことで 360 度周囲を確認することが可能になっている。距離は 40m、周囲は 270 度の範囲を計測可能である。

b) 3 次元測域センサ YVT-35LX-FK
　3 次元測域センサ (図 3 参照) は Arno の前方に積んでおり、地面の高低差や障害物を認識することが可能になっている。距離は 35m、水平方向には 210 度、垂直方向は 40 度の範囲を計測可能で

図 1　走行中の Arno

図 2　2 次元測域センサ URM-40LC-EW

図 3　3 次元測域センサ YVT-35LX-FK

ある。

2.3　ROS

ROS[1] は Robot Operating System の略であり、ロボット開発において重要な役割を果たすオープンソースのロボット制御ソフトウェア、及びそれを包括するロボット開発プラットフォーム全体のことを表す。本研究では ROS システムが提供するメッセージ配信の機能を利用するため、それに関連する ROS の構成要素について、以下に詳細を記述する。

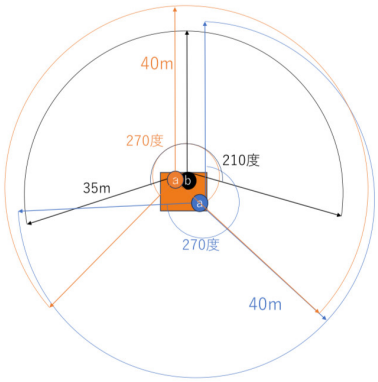

図 4　センサを上から見た図

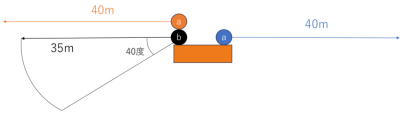

図 5　センサを横から見た図

2.3.1　パッケージ

ROS のソフトウェアは**パッケージ**という単位でまとめられている。パッケージにはコード、データ、ドキュメント等が含まれている。1 つのパッケージは 1 つのディレクトリおよびそのサブディレクトリ内に格納される。

2.3.2　ノード

パッケージ内の実行ファイルに相当する。**ノード**同士が相互に通信を行い、複数のノードを組み合わせることで情報やデータの交換を行う。複数のノードを組み合わせて 1 つの大きなシステムを作るため、個々のノードに機能を分散させることでノード単位でのデバックが可能になる。

2.3.3　トピック

ある定義された型を持つメッセージストリームの名前を**トピック**という。分散システムにおけるデータ交換でよく利用される配信/購読 (Publisher/Subscriber) 型の通信メカニズムを実装している。以下は配信/購読における手順である。

1. 公開
   ノードはトピックの名前と送るメッセージの型をアナウンスする。
2. 配信
   ノードはトピックに実際のデータを付けて送信する。
3. 購読
   ノードに接続情報を提供するサービスである roscore に、トピックのメッセージを受け取る要求を行う。

2.4　停止ボタン

実際に人や車の往来がある場所で自律走行を行う際、何らかの原因によりロボットが障害物を感知できず衝突しそうになった場合に、手動で止めるボタンのことを**停止ボタン**と呼ぶ。停止ボタンを押すと、モータドライバへの電源が遮断されて、車輪が物理的に動かない状態になり停止する。

3　課題

本説では我々が参加したロボットの自律走行の大会と、そこで明らかになった課題について述べる。

3.1　中之島ロボットチャレンジ

人々の往来する実環境において自律移動ロボットが問題なく行動できる技術開発の公開実験を大学や企業向けに提供している大会である。2023 年度は以下に示すような日程で行われた。

3.1.1　大工大エクストラチャレンジ

大阪工業大学梅田キャンパスにて 10 月中旬に行われた大会である。
屋外・屋内を問わず人の往来の中、キャンパス周辺と建物内の指定されたコースを自律走行する。

3.1.2　中之島ロボットチャレンジ 2023 エクストラチャレンジ

ATC(アジア太平洋トレードセンター) にて 11 月下旬に行われた大会である。
屋外・屋内を問わず人の往来の中、ATC の O's 棟周囲の指定されたコースを自律走行する。

3.1.3　本大会

中之島公園にて 1 月末に行われた大会である。
大阪市中央公会堂や中之島図書館周辺などの人通りの多い屋外を自律走行する。

3.2　課題点

例年夏頃に開催されていた本大会だが、今年度は 1 月に延期したため、エクストラチャレンジで明らかになった課題について述べる。同大会に参加した結果、自律走行がコース途中で様々な理由で中断し、先に進めなくなることが起こった。本研究ではそのうち以下の 2 点について扱う。

1. 停止ボタンの動作の有無が自律走行プログラム内で認識されていない。
   センサなどのシステム制御を行うプログラム内で停止ボタン作動の検知は行っていたものの、先行研究段階では自律走行時のプログラムで停止ボタンの作動の有無を認識していなかった。ロボットは走行を続けているつもりであるため、自身が持っている位置の推定に関するデータと現実が乖離し、地図上の自己位置を見失いやすくなった。現実の走行でもそれによる失敗が多数発生した。
2. ウェイポイントを順番にたどることしかできない。
   先行研究での自律走行プログラムは、地図上の指定した座標 (ウェイポイント) を順番に通る。そのため、障害物など何らかの理由で 1 つのウェイポイントを通れない場合、その先へ行くことができなかった。特に停止ボタンで停止している間に、次のウェイポイント付近の障害物の有無が変わっていることがよくあり、その際に進めなくなってしまうことが問題となっていた。

図 6　ノード間の関係

4　取り組み

本節では、前節で述べたような課題を解決するためにどのような取り組みを行ったのかについて述べる。

4.1　方針

停止ボタンの押下を自律走行プログラム内で認識をする。また停止ボタンが押されている間はその場で待機し、もとに戻されたら次のウェイポイントに移動する仕組みを実装する。以下はその方針を段階に分けて説明したものである。

a) 停止ボタンの動作認識
　ロボットの自律走行中に停止ボタンが作動した際、Arno の制御を行うプログラム内だけでなく、自律走行プログラム内でも検知し、停止ボタンが作動している間は待機する命令を出す。

b) ウェイポイントのスキップ
　順にウェイポイントをたどる自律走行中、停止ボタンを押して、解除後に自律走行を再開する際、ウェイポイントを 1 つスキップできるよう改良する。自律走行中は目標としているウェイポイントの近くを主にスキャンし近付いて行くため、地図作成時に存在しなかった障害物が発生すると同じ地図上で似ている障害物の構図が会った際にそちらに自己位置が飛んでしまうことが過去の経験上多々あった。障害物が発生したウェイポイントを飛ばすことで、自律走行時の自己位置を見失う要因を減らす。

4.2　実装内容

前節で述べた方針のもとに、本節では実際に行った内容をより具体的に述べる。

4.2.1　停止ボタンの動作認識

Arno の機器を制御している ROS ノード icart_mini_driver_node を記述している C++ のソースコード icart_mini_driver_nodes.cpp を変更し、停止ボタンの作動を検出した際に、新設した ROS トピック stop_button に String 型の文字列「Push button」をメッセージとして配信するようにする。Arno の自律走行を行うノード arno_navigation を記述する Python プログラム arno_navigation2.py 内でそのトピックを購読して処理を行うコードを追加する。これらの関係を図 6 に表す。

これにより制御プログラムで検知した停止ボタンの押下を、自律走行プログラム内でも認識することが可能となる。

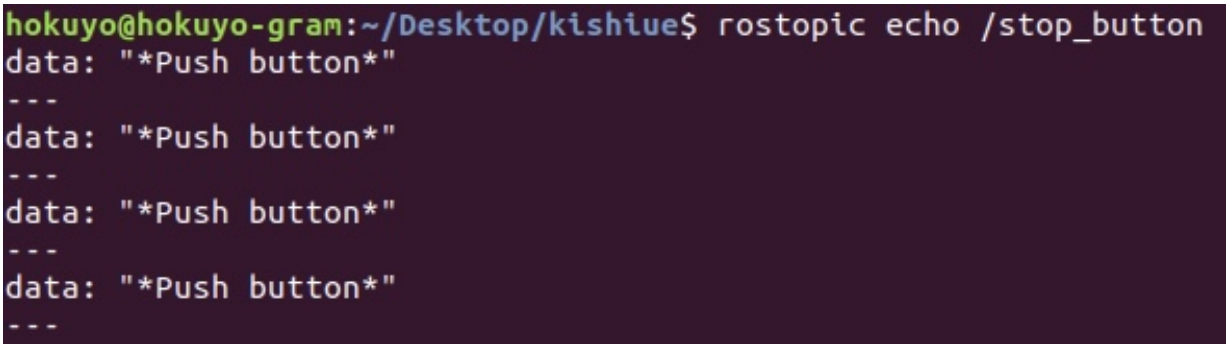

図 7　メッセージの出力

4.2.2　ウェイポイントのスキップ

購読したトピック stop_button にメッセージが到着した際に呼ばれる callback 関数を用意し、その中で、現在向かっているウェイポイントを飛ばして次のウェイポイントを目指す処理を記述する。また、停止ボタンの押下が検知された際に「Skip by stop system」と出力し、解除されるまでその場に待機する処理も追加する。

これにより停止ボタンの押下を自律走行プログラム内で検知した際、解除されるまでその場で待機し、解除後の自律走行は次のウェイポイントから再開することが可能になる。

4.3　実行結果

4.3.1　停止ボタンの動作認識

ロボットの自律走行時に停止ボタンを押した際、期待通りのメッセージがトピックに配信されていることを確認した。

以下が実際に確認した項目とその結果である。

1. トピック stop_button にメッセージ「Push button」が配信されていることを rostopic echo で確認した画面 (図 7)
2. トピック stop_button の配信者と購読者のノード名をターミナル上に出力した画面 (図 8)
3. ロボットの自律走行時に存在しているトピック名のリストをターミナル上に出力し、stop_button が含まれていることを確認した画面 (図 9)

4.3.2　ウェイポイントのスキップ

動作確認のため、自律走行中に停止ボタンを押下しトピックから停止メッセージを受けて待機処理を行う際、端末に文字列でメッセージが出力されるようにした。これと、従来ロボットが走行中にウェイポイントを通過時、端末に出力されていたメッセージが、待機中は出力されなくなること、また待機からの復帰時にターゲットとなるウェイポイントの番号が 1 つ飛ばされることとあわせ、期待通り待機処理およびウェイポイントのスキップ処理が行われたことを確認できた。(図 10)

```
hokuyo@hokuyo-gram:~/Desktop/kishiue$ rostopic info /stop_button
Type: std_msgs/String

Publishers:
 * /icart_mini_driver_node (http://hokuyo-gram:35229/)

Subscribers:
 * /arno_navigation (http://hokuyo-gram:33417/)
```

図８　トピックの配信者と購読者

```
hokuyo@hokuyo-gram:~/Desktop/kishiue$ rostopic list
/amcl/parameter_descriptions
/amcl/parameter_updates
/amcl_pose
/clicked_point
/diagnostics
/hokuyo3d/hokuyo_cloud
/hokuyo3d/hokuyo_cloud2
/hokuyo3d/imu
/hokuyo3d/mag
/icart_mini/cmd_vel
/icart_mini/odom
/icart_mini/parameter_descriptions
/icart_mini/parameter_updates
/initialpose
/joint_states
/joy
/joy/set_feedback
/laser_status
/laserscan_multi_merger/parameter_descriptions
/laserscan_multi_merger/parameter_updates
/map
/map_metadata
/map_updates
/merged_cloud
/move_base/DWAPlannerROS/cost_cloud
/move_base/DWAPlannerROS/global_plan
/move_base/DWAPlannerROS/local_plan
/move_base/DWAPlannerROS/parameter_descriptions
/move_base/DWAPlannerROS/parameter_updates
/move_base/DWAPlannerROS/trajectory_cloud
/move_base/NavfnROS/plan
/move_base/cancel
/move_base/current_goal
/move_base/feedback
/move_base/global_costmap/costmap
/move_base/global_costmap/costmap_updates
/move_base/global_costmap/footprint
/move_base/global_costmap/inflation_layer/parameter_descriptions
/move_base/global_costmap/inflation_layer/parameter_updates
/move_base/global_costmap/obstacle_layer/parameter_descriptions
/move_base/global_costmap/obstacle_layer/parameter_updates
/move_base/global_costmap/parameter_descriptions
/move_base/global_costmap/parameter_updates
/move_base/global_costmap/static_layer/parameter_descriptions
/move_base/global_costmap/static_layer/parameter_updates
/move_base/goal
/move_base/local_costmap/costmap
/move_base/local_costmap/costmap_updates
/move_base/local_costmap/footprint
/move_base/local_costmap/inflation_layer/parameter_descriptions
/move_base/local_costmap/inflation_layer/parameter_updates
/move_base/local_costmap/obstacle_layer/parameter_descriptions
/move_base/local_costmap/obstacle_layer/parameter_updates
/move_base/local_costmap/parameter_descriptions
/move_base/local_costmap/parameter_updates
/move_base/parameter_descriptions
/move_base/parameter_updates
/move_base/result
/move_base/status
/move_base_simple/goal
/odom
/particlecloud
/rosout
/rosout_agg
/scan
/stop_button
/tf
/tf_static
/urg_node1/parameter_descriptions
/urg_node1/parameter_updates
/urg_node1/scan
/urg_node2/parameter_descriptions
/urg_node2/parameter_updates
/urg_node2/scan
/waypoint
/waypoint_array
```

図９　配信しているトピックのリスト

図 10　自律走行時の出力画面

5　まとめ

本研究では、自律走行時にしばしば作動させることのあった停止ボタンの動作を自律走行プログラム側で認識することで、停止させたことによる自己位置のズレや自己位置を見失うなどの問題の発生を削減することができた。これにより、2024 年 1 月末の中之島ロボットチャレンジでは、練習段階ではあったものの新出研究室では初めてコースを完走することができた。また C++ で記述されたノードの変更や Python 間でのトピック通信の成功も新出研究室で初めての事例となった。ウェイポイントのスキップに関しては、工夫次第でより簡素なコードにすることができると考えられるので今後取り組んでいきたい。その他にも「自動で地図上のゴミを取り除く」など自律走行の成功率を上げる改善点は多く残っているのでそれらに取り組むことも今後の課題である。

6　謝辞

本論文の執筆および研究にあたり、親身にご指導していくださった新出尚之准教授に深く感謝し、厚く御礼申し上げます。また、貴重な時間を割き知識やロボットを提供していただいた北陽電機の皆様、様々な質問に親身に回答してくださった北陽電機の岡本様にも感謝の意を示します。ありがとうございました。

参考文献

[1] Morgan Quigley, Brian Gerkey, and William D. Smart. プログラミング ROS -Python によるロボットアプリケーション開発. オライリー・ジャパン, 2017. (河田卓志 (監訳), 松田 晃一, 福地 正樹, 由谷 哲夫 (訳)).