

自律移動ロボットにおける速度制約や閉塞区間などの 制約条件を付加した経路計画法の改良

Improvement of Path Planning Methods with for Autonomous Mobile Robots with Constraints of Velocity and Blocked Areas.

佐藤隆世¹⁾
指導教員 林 誠治¹⁾

1) 拓殖大学工学部 機械・電子システム工学専攻 林研究室

キーワード：自律移動ロボット，経路計画，A*アルゴリズム

1. はじめに

近年，人手不足や新型コロナウイルスの影響で工場，倉庫，ファミリーレストランをはじめとする様々な場所で自律移動ロボットが活躍している．これらで活躍するロボットには高い安全性と効率よく目的地まで移動できることが求められているが，実際に活動する環境下では多くの危険が潜んでいる．ロボットは，あらかじめ生成された経路に従って走行するが，人などの障害物があった場合にはロボット自身が走行可能か，速度を落とすかどうかなどを直前に判断している．また，生成される経路はそのロボットが最大速度で到達できる最短のものとなっており，障害物を避ける場合などにその経路から逸れてしまうことがある．あらかじめそのような領域を考慮したうえで経路計画を行えば，移動距離が環境地図上の最短距離でなくても，より効率よく安全に目的地まで到達できる可能性がある．

例えば，ファミリーレストランのような場所で稼働している配膳ロボットは，店内の人ごみになどによる影響を受けやすい．人ごみを避けるために一時停止したり，迂回することで，効率よく目的地までたどり着くことが難しい．あらかじめそのような場所やその付近では，速度制約区間や一時停止位置，進入禁止領域として事前に環境地図上で定義することで，ロボット自身で障害物回避を行う頻度を減らし，また，誰も着席していない客席周辺を走行する場合は最大速度で走行できるように定義することにより，効率よく目的地までたどり着けるのではないかと考える．

そこで，本研究では既存のナビゲーションシステム環境 [1] [2] において，速度制約区間の定義とともに，周囲の状況やロボットの走行経路によって仮想的な信号機や閉塞区間などの制約条件を付加することで経路計画法の改良を行い，より安全に効率よくロボットの自律走行を補助することが可能なナビゲーションシステムの構築を目的とする．

2. 経路計画アルゴリズムの改良

本研究を進めるにあたり，生成される最短経路を確認するためのシミュレーション環境を開発した．経路生成に使用するの既存のナビゲーションシステムでも用いられている A*アルゴリズムを使用する．

2.1 A*アルゴリズム

一般的なナビゲーションシステムでは A*アルゴリズムがよく用いられている．広大な環境では経路探索に時間がかかってしまうことが予想されるため，ダイクストラ法ではなく，少ない計

算量で最短経路を求めやすい A*アルゴリズムを使用している．A*アルゴリズムで最短経路を求める際には式 (1) に示すコスト関数 $f(n)$ を用いて行う．ここでいうコストとは現在地からゴールまでの重み付けされた距離尺度を表す．なお， $g(n)$ は式 (2) のマンハッタン距離を用いてスタート地点から現在地までの移動距離を， $h(n)$ は式 (3) のユークリッド距離を用いて現在地からゴールまでの最短距離の予測値をそれぞれ算出し，これらの和としてゴールまでの最終コストを求めている．

$$f(n) = g(n) + h(n) \quad (1)$$

$$g(n) = |x_1 - x_2| + |y_1 - y_2| \quad (2)$$

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

スタート地点を出発して，今いる地点に隣り合う地点のコストを計算し比較した結果，コストが最小となる地点のゴールに到着するまで繰り返し行うことで最短経路を求めることが可能である．

2.2 シミュレーション環境の開発

シミュレーション環境は，実際のナビゲーションシステム上で使用する経路計画アルゴリズムが適切かを確認するために使用するため，簡易的なものになっている．本研究では以下の機能を実装している．

- 環境地図データの読み込み
- 環境地図データの表示
- 環境地図データ上でスタート地点とゴール地点の表示
- 速度制約区間や立ち入り禁止区域，一時停止位置を考慮した最短経路の表示
- ロボットの動作の可視化

2.3 確認に使用した環境地図

検証に使用した環境地図を図 1 に示す．この環境地図は小さなレストランを模したものである．薄い灰色の区域が速度制約区間，濃い灰色の区域が椅子などが設置されたロボットの侵入禁止区域，黒い区域がテーブルや壁などの障害物を示している．レジャーやドリンクバー周辺など人の混雑が予想されるような場所や，狭い通路には予め速度制約区間として指定している．

この環境地図データは Portable Gray Map とよばれるモノクロで画像を格納する形式のファイルで作成されている．0 から

PA2-02

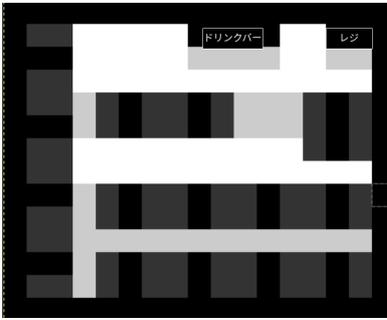


図 1 検証に使用した環境地図データ

255 の数値で色を表現することが可能であり、0 が黒、255 が白色を表している。このデータはペイントソフトなどのツールでの作成が可能であり、ROS(Robot Operating System) で動作する生成された環境地図の管理を行う map_server で保存されたデータも使用することが可能である。

3. シミュレーション環境上での最短経路の生成

作成したシミュレータ上に図 1 の環境地図データを読み込み、A*アルゴリズムで最短経路を求めた場合と制約条件を付加した場合で、生成される経路にどのような違いがあるのかを試した。その結果を以下の 3.1 節、3.2 節に示す。

3.1 既存のナビゲーションシステムでの経路計画法

式 (1)~(3) に基づいて A*アルゴリズムにより生成された最短経路を図 2 に示す。



図 2 A*を用いて求めた最短経路を環境地図上に表示させた例

制約条件を考慮していないため、最も最短な経路が求められていることがわかる。

3.2 本研究で改良した経路計画法

3.1 節で実装している A*アルゴリズムの式 (3) を改良することにより、速度制約区間や一時停止位置などを考慮した最短経路を求める。

本研究で新たに実装するコスト関数を式 (4) に示す。

$$g(n) = \frac{|x_1 - x_2| + |y_1 - y_2| + s(n)}{v(n)} \quad (4)$$

ここで、 $s(n)$ は一時停止位置のコストを表しており、0~9 までの整数値を設定する。数字が高ければ高いほど長時間停車する必要があることを示す。したがって、一時停止位置として設定されていない区間では $s(n) = 0$ として設定する。一方、 $v(n)$ は速度制約区間のコストを表しており、速度制約区間を設定する際に 0.1~1.0 まで 0.1 刻みで設定することが可能である。1.0 が最大の速度を表し、速度制約が設定されていない区間では $v(n) = 0.7$ を初期値として経験的に設定する。式 (4) を用いて環境地図上に生成された最短経路を図 3 に示す。

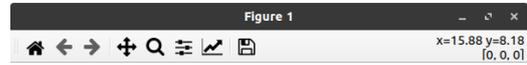


図 3 A*アルゴリズムを改良し環境地図に求めた最短経路を表示させた例

図 2 と図 3 を比較すると、式 (4) を使用した図 3 の場合、速度制約区間を垂直に通過することで、速度制約区間にかかるコストを最小限に抑えていることが確認できた。

4. まとめと今後の課題

本研究では既存のナビゲーションシステムで用いられている A*アルゴリズムのコスト関数を改良することによって、速度制約区間と一時停止位置を考慮した経路生成を行うシステムを簡易的なシミュレータ上で確認できるようにした。予め混雑が予想される箇所を考慮して障害物の回避頻度をなるべく低減させることで、安全かつ効率よく自律移動ロボットを運用できると考えられる。

今回は生成される経路の比較のみ報告したが、今後は実環境を模した環境でロボットの自律走行を行い、本研究で改良したアルゴリズムがどの程度効率よくロボットを行動させられたかを検証する予定である。

参考文献

- [1] "Navigation - TurtleBot3", <https://emanual.robotis.com/docs/en/platform/turtlebot3/navigation/>
- [2] "NavigationStack - global_planner", https://robo-marc.github.io/navigation_documents/global_planner.html
- [3] "Matplotlib 3.7.3 documentation", <https://matplotlib.org/3.7.3/index.html>