人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-062-02

自律移動ロボットにおける相手ロボットとのすれ違い方法に関する一考察

A novel method for finding place for robots to pass each other

田邉稜汰　安田尚平　植村渉

Ryota Tanabe, Shohei Yasuda, and Wataru Uemura

龍谷大学
Ryukoku University

Abstract: The autonomous mobile robot creates path when moving, however if the created path overlaps with the path of other robots, there is a possibility of collision, so it is necessary to find a place to move each other robot. In this paper, we propose a method to find midpoint based on the positional with the other robot and evaluate its performance. When the other robot moves in the direction of the own robot at a constant speed, it collision with the opponent robot at the midpoint. Therefore, we propose a method to determine the location where the robots must pass at a distance necessary for them to avoid a collision at the midpoint.

1 はじめに

現在，少子高齢化が進み，製造業，医療現場など様々な分野で，将来的に人手不足になるため，代わりの労働力としてこれらの現場に自律移動ロボットが導入されると期待されている．自律移動ロボットの導入により，作業の自動化や手作業による危険の軽減の利点もある．これらの現場では，人や他のロボットと協働で作業を行う自律移動ロボットが注目されており，特にその移動時の経路探索，衝突回避技術が重要とされ，多くの研究が行われている．

自律移動ロボットは移動時に目的地までの経路を作成する．しかし，複数のロボットが共同で作業を行う環境では，作成した経路が他のロボットの経路と重複すると，衝突する危険性がある．そのため，相手ロボットを検出したときに，すれ違うことができ，衝突が起きない場所を見つけ，衝突回避する必要がある．本研究では，相手ロボットが存在せず，静止障害物のみ存在する場合に Dynamic Window Approach[1]を使い，相手ロボットがいる場合には，相手ロボットとのすれ違い点を見つけ，回避的な経路をとる方法を提案する．また，提案法の性能を評価する．

第 2 章では，研究で使用する自律移動ロボット，Robotino について説明する．第 3 章では，経路探索手法の一種である Dynamic Window Approach の概要とその評価関数，問題点について説明する．第 4 章で，提案法である，相手ロボットとのすれ違い方法について説明し，第 5 章では提案法の性能を評価するために相手ロボットが存在する環境で目的地を目指す実験を行う．そして，第 6 章で本研究のまとめと今後の課題を述べる．

2 Robotino

自律移動ロボットとは人間の操作を必要とせず，ロボット自身が周囲の環境を把握し単体で移動するロボットのことである．環境を認識するために，カメラやセンサから得た情報をもとに，自己位置推定や環境地図を作る．本研究では，自ロボットと検知対象の相手ロボットに自律移動ロボットである Robotino を用いる．このロボットは，ドイツのフエスト社が販売するロボットであり，自律移動ロボットの世界大会である RoboCup Logistics League (RCLL)[2]で用いられている． Robotino は円形であり，直径 45cm である．また，3 輪のオムニホイールを持っている．

3 経路探索

3.1 Dynamic Window Approach

ロボットの軌跡を計算して経路選択をする手法として Dynamic Window Approach(DWA) がある．DWA は，環

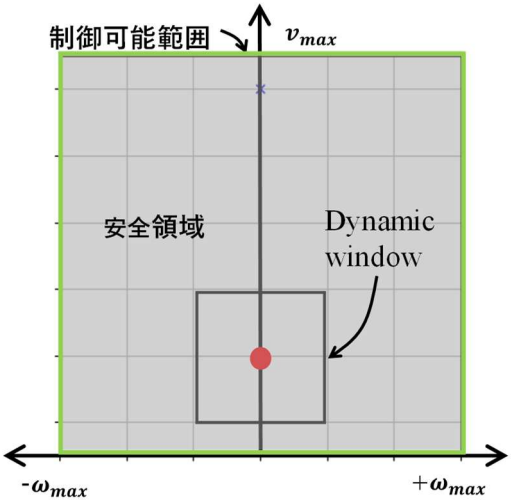

Fig. 1: Dynamic Window.

境が変化し続ける動的環境での経路計画法の Local Path Planning の一種でロボットの運動モデルを考慮し，自己位置，障害物，目的地の情報から，可能な制御入力(Dynamic Window)を計算して，ロボットの軌跡を計算することで，自ロボットが目的地の方位に向いているか，障害物から遠いか，速度が速いか，この三項目で評価を行う評価関数が最大になる経路を選択する手法である．主に，動的環境での自律移動ロボットの衝突回避や，経路探索において，広く用いられる手法である．Fig.1 に Dynamic Window の概念図を示す．緑の枠で囲まれた範囲が制御可能範囲，グレーの領域が安全領域，赤丸を囲む正方形が Dynamic Window，赤丸が自ロボットの現在速度で縦軸が並進速度v，横軸が回転速度ωである．制御可能範囲はロボットの取りうる制御入力の範囲力の範囲を表し，そのうち現在の速度とロボットの加速度から計算される次の時刻までに取りうる範囲を Dynamic Window として表している．ロボットに搭載したカメラやセンサから得た環境情報とロボットの最大限速度から計算される制御入力の範囲が安全領域である．また，障害物が存在する場合，障害物と衝突する範囲に衝突領域ができる．この Dynamic Window の範囲と安全領域を考慮することでロボットの運動モデルを考慮した障害物に衝突しない経路探索を行うことができる．本研究では，静止障害物のみ存在する場合に DWA を使用する．

3.2 評価関数

評価関数は，センサやカメラから得た情報から，自ロボットが目的地の方位に向いているか，障害物から遠いか，速度が速いか，この三項目で評価を行う．三項目の評価値を足し合わせることで，それぞれの項目の特徴を反映した評価関数が得られる．合成後の評価関数をGとすると

$$G(v,\omega)=(\alpha\times heading+\beta\times dist+\gamma\times velocity)$$

となる．評価関数Gは $heading$，$dist$，$velocity$の評価値に対応している．この評価値はそれぞれ，180°から目的地の方向とロボットの方向の差を引いた角度，障害物との距離，速度に関する変数であり，また，α，β，γはそれぞれ，各評価関数の足し合わせる際の重み係数であり，$heading$の係数の重み付けを大きくすると目的地に向かう経路を優先し，$dist$の係数の重み付けを大きくすると障害物から遠ざかる経路を優先し，$velocity$の係数の重み付けを大きくすると速度が速くなる制御を優先する．これらの係数を調整することで障害物を避けながら素早く目的地に向かう経路を探索できる．

3.2 問題点

DWA は，以前から変化する環境には対応できるが，静止障害物を対象としているため、相手ロボットがいる環境では使用できない．また，運動モデルや，運動モデルに伴った評価関数Gなどの係数の設定によっては，障害物に対して不必要に遠回りする経路計画やロボットの軌跡を計算できずにその場で停止するような動きが起こり，衝突の危険性がある．

4 提案法

DWA は，相手ロボットがいる環境では使用できない，運動モデルや，評価関数などの係数の設定によって衝突の危険性があるという問題点がある．そのため，相手

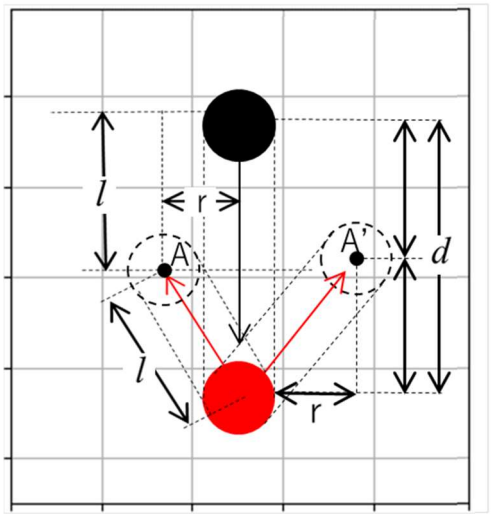

Fig. 2: place to pass each other robot

ロボットが存在せず，静止障害物のみ存在する場合にDWAを使用して経路探索を行い，相手ロボットがいる場合には回避的な経路をとる方法を提案する．ここで，回避的な経路とは，相手ロボットと衝突せずにすれ違える点を見つけ，見つけた点まで移動する方法である．

次にすれ違い点の導出方法について説明する．Fig.2 に提案法の経路のイメージ図を記す．図の赤丸が自ロボット，黒丸が相手ロボット，点 A が正しいすれ違い点，点 A’が正しいすれ違い点である．移動している相手ロボットとの衝突を考える．お互いが向かい合っていて，かつ，両者が最高速度で進んでいるときが一番危険である．その場合，相手ロボットと自ロボットは中点で衝突することになる．そこで，相手ロボットとのすれ違いに必要な距離を r とすると，お互いのロボットが l 進んだ時点で，r 以上の距離を保つ点 A ですれ違うべきだが，導出するための計算が複雑で，複数のすれ違い点が存在するため，計算コストがかかる．そのため，ここでは導出を簡単にするため，中点よりr だけ真横にずれた点A’をすれ違い点とする．近似したすれ違い点へ進むことで，相手ロボットとの衝突を回避することを目指す．自ロボットの進行方向を x 軸とし，相手ロボットとの距離を d とすると，目的とするすれ違い点 A は，$(d/2, \pm r)$となる．なお，この場合，自ロボットがすれ違い点に到着する前に，相手ロボットが $d/2$ の経路を進み，自ロボットが r の距離を確保できない．

相手ロボットが複数存在するときは，一番距離が近い自ロボットに向かって進行しくる相手ロボットを危険だと判断し，その相手ロボットに対してすれ違い点を見つけることが重要になるので，そのロボットを対象にすれ違い点 A を設定する．1 ステップごとに点 A を計算するべきだがここでは次のように簡単化する．点 A を計算したらその点 A を目指して走行し，点 A に到達したら再度その時点でのすれ違い点 A を計算する．この作業を繰り返すことで，他のロボットの衝突を避けながら目的地へ進む．

5 性能評価

5.1 実験方法

実験では，数値シミュレーションを使って 2 台の相手ロボットがいる環境で経路探索を行い，提案法の経路と目的地に到達するまでの走行時間，距離，自ロボットと相手ロボットの距離を調べ，評価を行う．実験フィールドを Fig.3 に示す．(0,0) の赤い丸印が自ロボットの出発地，(5,0)の青いバツ印が目的地である．2 台のロボットが自ロボットに向かってくる場面を想定して，ロボット 1 の初期位置を(3,2)，目的地を(0,−1)として移動させる．ロボット2の初期位置を(5,0)，目的地を

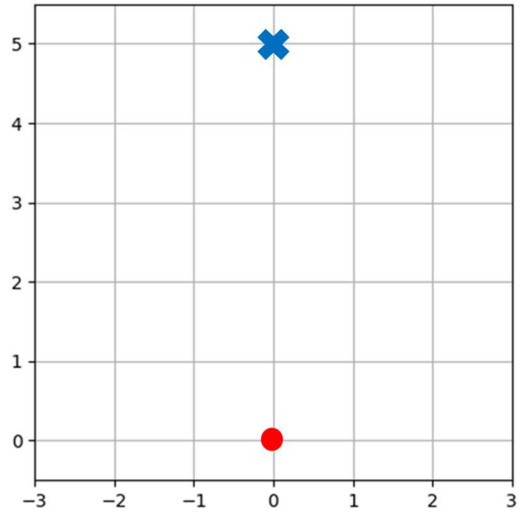

Fig. 3: Experimental field.

(0,0)としたときと，初期位置を(3.5,1)，目的地を(0,2)にして移動させる．衝突を回避するには相手ロボットとの距離として r[cm]を確保する必要があるが，今回は近似的にすれ違い点を導出するため r では相手ロボットと接触する．そこで，r’として r に対して少しマージンを持った値を設定する必要がある．今回，ロボットの直径が 45cm であるため r=45 に対して，r’=60cm で設定する．実験では自ロボットと相手ロボットの距離が 45cm 以下になると衝突と判定する．

5.2 実験結果

実験結果を Fig.4 と Fig.5 に示す．Fig.4 が衝突時の経路，Fig.5 の上図が成功時の経路，下図が成功時の相手ロボットとの距離と時間の関係図である．

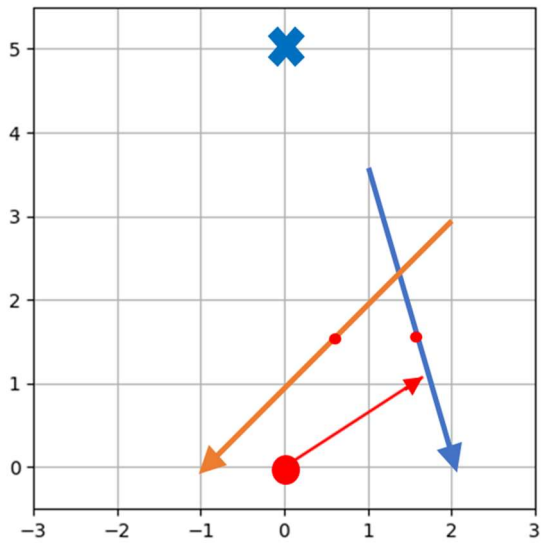

Fig. 4: Failure results.

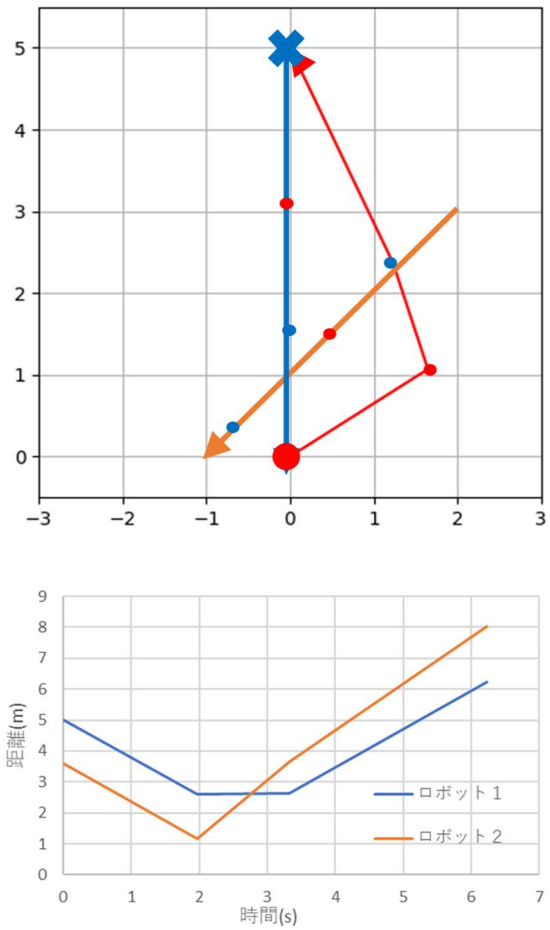

Fig. 5: Successful results.

衝突回避をして目的地に到達できた経路では，2 台の相手ロボットと，常に 1m 以上の距離を保って走行できた．衝突時の経路のように，相手ロボットが，危険度が低い場所から，すれ違い点に進行してくるような場面では，衝突した．

次に成功回数，衝突回数を調べるためにロボット 1 を(3,2)を初期位置として，5×5 のフィールド内の整数座標を目的地として移動させる．ロボット2はロボット1からそれぞれ近い場所，遠い場所に配置する．ロボット2の初期位置が近い場合(4,1) ，遠い場合(4,−1)とし，目的地をフィールド内の整数座標を目的地として移動させる．それぞれ 841 通り，合計 1682 通りのパターンで実験を行い，目的地に衝突せずに到達できたか確認する． この結果を Table. 1 に示す．

Table. 1: results.

Robot position	(4,1)	(4,-1)
Success	625	650
Failure	216	191

Table. 1 の失敗は Fig. 4 のようにすれ違い点に，相手ロボットが進行したことが原因であった．この問題は，すれ違い点を相手ロボットの位置のみから計算して，相手ロボットの急な進路変更や急停止などの進行予測ができていないことや，相手ロボットと常に同じ距離を導出する方法を簡単化したために相手ロボットと距離を確保できない場合が生じたため起きた．この問題を解決するためには，随時，相手ロボットの状態を判断し，すれ違い点を計算することで改善することができる．

6 まとめと今後の課題

本研究では， 相手ロボットがいない場合に DWA を使い，相手ロボットがいる場合にはすれ違い点を見つけて，回避的な経路をとる方法を提案した．

1 台の相手ロボットに対する回避行動を提案し，複数台に対応できるよう拡張した．次に相手ロボットが 2 台いる環境で提案法を試して，提案法の性能を評価した．成功時には，複数の相手ロボットと，常に 1m 以上の距離を保って安全に走行できたが，危険度が低い場所から，すれ違い点に進行してくるような場面では，衝突した．今後，衝突時の問題を相手ロボットの随時，相手ロボットの距離を判断し，お互いのロボットが l 進んだ時点で，r 以上の距離を保つすれ違い点を計算して，経路探索することで解決したい．また，今回は，数値シミュレーションで提案法を評価したため，ロボットに実装して随時，相手ロボットと距離を保って経路計画できるようにしたい．

参考文献

[1] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The Dynamic Window Approach to Collision Avoidance. IEEE Robotics & Automation Magazine Volume: 4, Issue: 1, March 1997

[2] RoboCup Logistics League (RCLL) (https://ll.robocup.org/)