

クラウド・ロボット間通信を用いた 複数台移動ロボットの回避行動生成

○田倉 竜也, 松日楽 信人, 小松 廉,
永谷 圭司, 山下 淳, 浅間 一 (東京大学)

Avoidance behavior generation for mobile robots using the cloud-to-robot communication

○Tatsuya TAKURA, Nobuto MATSUHIRA, Ren KOMATSU,

Keiji NAGATANI, Atsushi YAMASHITA, and Hajime ASAMA (The University of Tokyo)

Abstract : In situations where multiple robots operate in the same environment, autonomous avoidance controls are required when paths interfere with each other if path planning is performed independently. Especially, when robots of different sizes are operating in the same environment, it is necessary to consider the existence of paths that can be taken by some robots and paths that cannot be taken by others. Therefore, we propose a system for efficient path planning for mobile robots of different sizes by considering global and local planning methods.

1. 序論

複数台のロボットが同一環境下で動作する状況では、時間的効率性の観点から、ロボット間の衝突を回避しつつ、移動時間を最小限に抑えるような経路を求めることが重要である。単独で経路計画を行った場合、それらの計画が遂行できた場合の総移動時間は最小となることが期待される一方で、経路が干渉した際に回避行動をとる必要が生じる。また、同一環境下で動作するロボットが全て同じ大きさであるとは限らず、自動車以上のサイズのものから子供以下サイズのものまで、異なる種類のロボットを運用することもある。このようなとき、ロボットのサイズによって通過できる領域とそうでない領域が異なるケースがある。これは屋内など通路が多く存在する環境下では顕著であり、ロボット同士が通路内ですれ違うことを考えたとき、実際にそれが可能であるか否かなどを判定する必要が生じる。そのような場合が生じないように経路計画を行うことを考えると、安全マージンを広く取る必要があり、非効率的な経路を取ることに伴う時間的なロスが避けられない。

経路計画は大局的経路計画 (global planner)、局所的経路計画 (local planner) に分けられる。大局的経路計画について、グラフ上の複数エージェントに対する経路計画問題は特に Multi-Agent Path Finding (MAPF) と呼ばれている。古典的な問題設定においては、無向グラフおよび各エージェントに対する始点と終点を与えられ、各エージェントは単位時間ごとに隣接ノードに移動するか、その場に止まるかのいずれかが選択でき、所要時間の総和といったコストを最小化する解を求める。また、衝突回避を考慮するために、ノードの共有、エッジ上でのすれ違いの禁止を制約として定める。厳密解を求めるアルゴリズムとし

て、Conflict Based Search (CBS) およびこれを一般化した Multi-Agent CBS (MA-CBS) などが存在する^[1]。一方で、これらのアルゴリズムは計算量が膨大であるため^[2]、Enhanced CBS (ECBS)^[3] といった近似解を求めるアルゴリズムも考案されている。しかしながら、前述の通り、異なる大きさを持つロボットを運用する場合は、この問題設定は時間的に非効率的である。仮にこの問題設定を適用するのであれば、隣接ノード間は干渉しないという条件から、ノード間の距離は最大の大きさを持つロボットに依存するため、最大の大きさをもつロボットが通過可能な領域に対するグラフを考えることになる。現実には一定以下のサイズのロボットであれば通過可能な領域が存在するが、その領域はこのグラフでは考慮されず、また、ロボットのサイズによっては同じエッジをすれ違うことも可能であるがその点も考慮されていない。

局所的経路計画手法の例としては、ロボットの動的特性を考慮し、可能な制御入力についてシミュレーションを行い最良のものを選択する Dynamic Window Approach (DWA)^[4] がある。このアルゴリズムでは、1. モータ等の特性により定まるロボットのとりうる速度、角速度の範囲、2. ロボットの現在の速度、角速度から実行可能な速度、角速度の範囲、3. センサ等で取得した周囲の障害物に衝突しないような速度、角速度の範囲、の三つの条件を満たす速度および角速度についてサンプリングしてシミュレーションを行い、障害物との距離や目的地点との距離などで定まるペナルティを最小化するような制御入力を採用する。この方法は、自分の移動速度を考慮している一方で、障害物が移動している場合であってもそのことを考慮することができない。また、二台のロボットが正対した際にそれぞれ

のロボットが同じ方向に回避しようとしてしまうケースが生じることが考えられる。

そこで本稿では、通路幅とロボットのサイズを考慮して、可能なすれ違いのみが発生する経路を大局的経路計画で生成する。その上で、局所的経路計画において、前述のDWAにおける条件3に用いる障害物の情報に加え、クラウドを經由して他のロボットが存在する地点および移動速度を取得し、そのロボットの輪郭点(以下 footprint と呼称する)を進行方向に向けて拡大して障害物として扱うことにより、他のロボットが自分の進行方向に侵入することを防ぎ、滑らかなすれ違いを実現する。

2. システムの構成

システムの構成を Fig. 1 に示す。全体を管理するメインサーバと各移動ロボットが通信を行っている。なお、図中のナンバリングはロボットの番号を表している(1は Robot 1 の意)。また、各ロボットが自己位置推定を行い、自身の世界座標系上での位置、姿勢をクラウドと共有しているが、その部分の処理は省略した。

2.1 メインサーバ

メインサーバは、主に座標の管理と大局的経路計画の二つの処理を受け持つ。静的データとして、全ロボットの footprint および幅、環境の地図および後述する平面グラフを保持している。

2.1.1 座標の管理

各ロボットの位置、姿勢を常時取得し、相対位置、相対姿勢を計算してリクエストに応じて送信する。また、これらのデータは送信時間と共に蓄積されるため、任意の座標系間の相対速度や角速度も計算することが可能である。

2.1.2 大局的経路計画

いずれかのロボットが目的地に到達したタイミングで大局的経路計画を行い、経路(ノード列)を各ロボットに送信する。以下のような条件を満たす平面グラフを事前に作成する。

- ・ エッジを挟むノード間の距離は全ロボットの幅の最大値以上である。
- ・ ノード間の距離は可能な限り等しくする。

このとき経路計画で考慮する制約として

- ・ ある通路に対して、その通路の最小幅の半分以下の幅となるロボット同士であれば、その通路上でのすれ違いを許容する。
- ・ ある通路に対して、その通路の最小幅以上の幅を持つロボットは、その通路の通過を禁止する。

という事項を追加する。このことで、MAPF 問題に対する既存の解法を適用することが可能となる。また、制約より、ロボットが干渉するのはエッジ上を交差するときのみであり、かつそのときロボットの幅の和は通路の幅を下回ることが保証される。これにより、ロボット同士の干渉は、通路内でのすれ違いのみを考慮すれば良い。

2.1.3 footprint の補間

はじめにシステムに与えられる footprint は、長方形の四隅など、非常に少ない点で構成されている場合が多いため、経路計画を行う前に、これらを繋いだ輪郭上に点を補間する。隣接する2点について、隣り合う点同士の偏角がある閾値以下の等しい値となるように2点を結んだ線分上に点を補間する。Fig. 2 に実際の結果を示す。黒い点があはじめにシステムに与えられた点、赤い点がこの補間によって加えられた点を示している。なお、ここでの閾値は36度とした。

2.2 移動ロボット

移動ロボットは、自己位置推定、局所的経路計画の二つの処理を受け持つ。大局的経路計画においてロボット同士の干渉が通路内でのすれ違いのみとなるよう設定したため、局所的経路計画においては通路幅を明示的に扱う必要はない。以下の手順にしたがって footprint を処理し、得られたデータを2D LiDAR で得られたスキャンデータと統合して障害物情報としてDWAに利用する。

2.2.1 範囲内のロボットの判定

はじめに、自己位置推定により推定した自身の位置をサーバに共有し、同時に他の移動ロボットの位置を取得する。続いて、他の移動ロボットが自身から見て一定の範囲内に存在するか否かを判定し、範囲内に存在するロボットの相対速度、footprint をサーバから取得する。このことによって、多数のロボットを運用する環境下であってもロボット側の計算負荷を抑えることが可能となる。

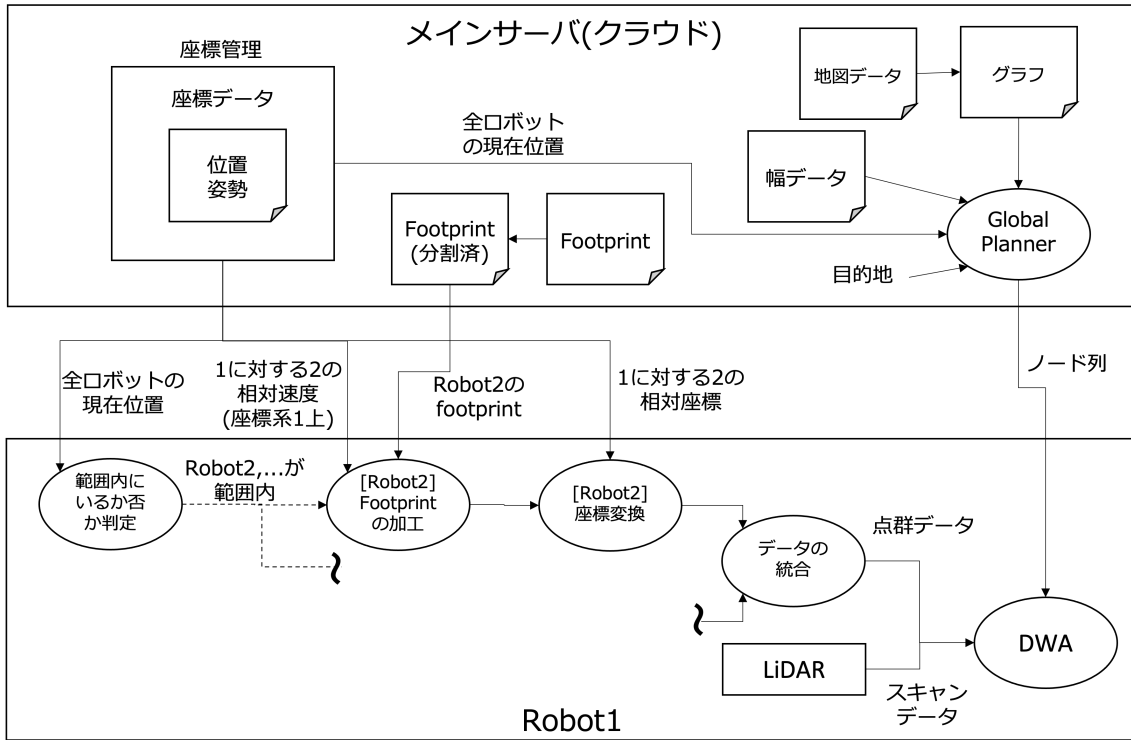


Fig. 1: System configuration diagram

以下の手順で変形を行う.

1. \vec{p} の長さを r , 偏角を θ とおく.

$$r_{new} = \begin{cases} r + \alpha \vec{p} \cdot \vec{v} & (\vec{p} \cdot \vec{v} \geq 0 \text{ and } p_y v_x - p_x v_y \geq 0) \\ r & (\text{otherwise}) \end{cases}$$

$$\theta_{new} = \theta$$

ここで α は正の定数.

2. 隣接する 2 点について, 隣り合う点同士の距離がある閾値以下の等しい値となるよう 2 点を結んだ線分上に点を補間する.

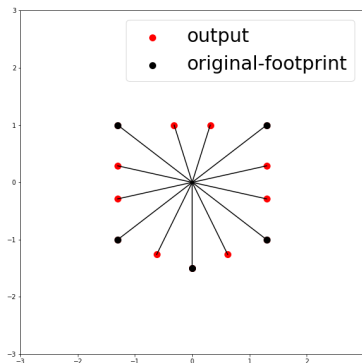


Fig. 2: Input and output after interpolation

2.2.2 footprint の処理

この処理では, 他のロボットの自身に対する相対速度を用いて, そのロボットの進行方向に footprint を拡大する. そうして得られた点群データを局所的経路計画に用いるローカルな地図上に障害物として取り込むことにより, 自身が他のロボットの進行方向上に侵入することを防ぎ, 衝突を回避する経路計画を行う.

着目している footprint を持つロボットの中心座標を O , 自身に対する相対速度を \vec{v} とおく. footprint の各点 (P とする) に対し, \vec{OP} (\vec{p} とする) と \vec{v} を用いて変形を施す.

以上の作業を条件を満たすロボット全てについて行う. 手順 1 の r についての条件式の第 1 項は原点から見て \vec{v} の終点と同じ側に P があること, 第 2 項は原点から見て P が \vec{v} の左側にあることを表している. Fig. 3 において, 赤いベクトルは, Robot1 に対する Robot2 の相対速度を表している. この図の, ベクトルが伸びている 3 点のうち, 右上の点は第 1 項を満たさず, 左下の点は第 1 項を満たすが第 2 項を満たさない. よって手順 1 ではこの 3 点のうち, 右下の点のみが変形の対象となる. Fig. 4 に $\alpha = 1$ として手順 1 を行った結果を示す. 黒い点がサーバから取得した footprint, 赤い点がこの処理による変形後の点を示している.

これにより, 速度方向に膨らませつつ, 更に非対称性も持たせることができ, 正対した場合のすれ違い動作を滑ら

かに行うことが期待できる。手順 2 の処理を行うのは、変形によって隣接する点の距離が大きくなっている箇所が存在する場合があるためである。Fig. 5 に閾値を 0.5 として手順 2 を行った結果を示す。手順 1 を行ったのち得られる点を黒、この処理で新たに追加された点を赤で示している。

なお、この時点では得られた点群データは各ロボットの座標系のものであるため、最後にこれを座標変換する必要がある。

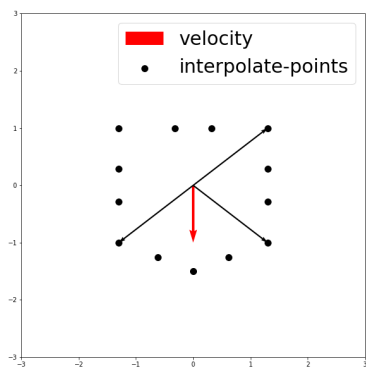


Fig. 3: Description of step 1

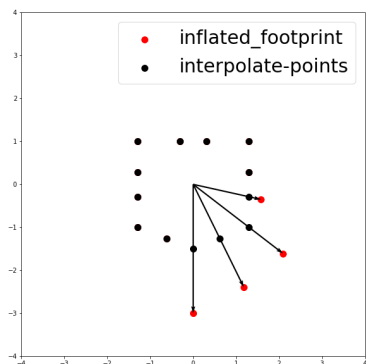


Fig. 4: Input and output after step 1

3. 結論

本稿では、大局的経路計画において、ロボット間の干渉がすれ違い動作のみとなることを保証した経路を生成し、各ロボットの局所的経路計画において、近隣のロボットの位置、速度の情報を基に footprint を加工して障害物データとして利用することで滑らかなすれ違い動作を実現する手法を提案した。今後は、大局的経路計画に用いるグラフの具体的な作成方法を模索するとともに、システム全体の

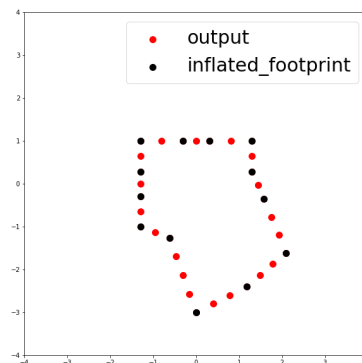


Fig. 5: Input and output after step 2

実装を進め、シミュレーション環境において回避行動が生成されることを実験を通して確かめる。また、相対位置、速度の送受信をより高速に処理するため、ロボット間の直接通信を行うことについても検討する。

謝辞

本研究は、JST【ムーンショット型研究開発事業】グラント番号【JPMJMS2032】の支援を受けたものです。

参考文献

- [1] Guni Sharon, Roni Stern, Ariel Felner, Nathan R. Sturtevant: Conflict-based search for optimal multi-agent pathfinding, *Artificial Intelligence*, vol. 219, pp. 40 – 66 (2015)
- [2] Jingjin Yu and Steven LaValle: Structure and Intractability of Optimal Multi-Robot Path Planning on Graphs, In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)*, pp. 1443 – 1449 (2013)
- [3] Max Barer, Guni Sharon, Roni Stern, and Ariel Felner: Suboptimal Variants of the Conflict-Based Search Algorithm for the Multi-Agent Pathfinding Problem, In *Proceedings of the Seventh Annual Symposium on Combinatorial Search (SoCS-2014)*, pp. 19–27 (2014)
- [4] D. Fox, W. Burgard and S. Thrun: The dynamic window approach to collision avoidance, *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33 (1997)
- [5] 岡野憲, 加藤宏一朗, 中村幸博, 松日榮信人, 成田雅彦: “ロボット連携ネットワークのための汎用通信ユニットの開発と接続実証” *日本ロボット学会誌*, vol. 39, no. 10, pp. 973–980 (2021)