

自律分散型モバイルロボット向け移動経路予約システムの検討

東原大記[†] Rami Yared[†]
Xavier Defago[†] Matthias Wiesman[†]

複数のモバイルロボット間での衝突回避は、モバイルロボットの協調において最も基本的な問題のひとつである。センサーベースでの衝突回避はリアルタイム性の強い方法であるが、センサーの有効範囲は小さいため確実な衝突回避は困難である。本稿では、ネットワークベースでモバイルロボットの移動経路を予約する分散型の衝突回避システムを提案する。提案したシステムの簡易的なシミュレーションをおこない予備評価として、モバイルロボットの利用範囲の大きさと利用台数が与える移動回数の変化を評価した。予備実験の結果、利用範囲内におけるモバイルロボットの密度とモバイルロボットが1度に予約する移動経路の長さのバランスを考慮する事が重要であることがわかった。

The study of moving path reservation system for distributed autonomous mobile robot

DAIKI HIGASHIHARA,[†] RAMI YARED,[†] XAVIER DEFAGO[†]
and MATTHIAS WIESMAN[†]

Collision avoidance between robots is an important problem of mobile robotics. Sensor based collision avoidance can execute with realtime. But Sensor based collision avoidance limited by the distance and the precision of the sensors. In this paper, we propose the network based distributed collision avoidance system by the moving path reservation. After that, we estimate the relation between a robot using environment and number of using robots for the preliminary evaluation with easily simulation. Considering balance between a moving path reservation and the density of mobile robots.

1. はじめに

計算機技術の発達及び安価化に伴い、複雑な計算が高速におこなえるようになった。計算の高速化が可能になり、ロボットに望まれるタスクも複雑化の傾向にある。ロボットに求められるタスクが複雑化する事により、タスクに合わせた複雑な構造のロボットが必要となる場合が多い。計算機は安価化の傾向に有るにもかかわらず、ロボットの構造の複雑化に伴いロボットは高価化する傾向にある。また、複雑な構造のロボットの開発にかかる時間も増加し、制御も複雑となる。ロボットを用いてタスクを実行する場合には、ロボット開発に必要な費用や時間と構造の複雑さに関するトレードオフを考慮する事が重要である。

前述の問題点から、複雑なタスクを実行するために、最低限の機能を有したロボットを複数用いてタスクを実行する群ロボット¹⁾²⁾の考えがある。群ロボットと

は、複数の自律的モバイルロボットがお互いに協調し合い³⁾、あたかも一つのロボットであるかのような振舞いをするロボットシステムである。単純な機能の複数のロボットが協調し一つのタスクを実行する事により、1台のロボットにかかる開発のための価格的、時間的コストは大幅に削減が可能である。また、群ロボットを利用する事により、複雑なタスクの実行も単純なロボットどうしでの協調して実行可能となるため、特定のタスク実行に特化したロボットをタスク毎に開発する必要もなくなる。

モバイルロボットを用いた群ロボットの場合、移動は基本的な協調を用いたタスクのひとつである。モバイルロボット間での衝突回避は、モバイルロボットの移動において重要な問題である。モバイルロボットが衝突した場合、ハードウェア的な損傷も発生し、損傷部分の交換が必要となる。交換可能な部品が無かった場合には、群ロボットとしてタスクの実行に支障を与える場合もあるため、衝突は確実に回避する必要がある。

モバイルロボット間での衝突回避方法には、セン

[†] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

サーベースとネットワークベースが考えられる。センサーベースの衝突回避⁴⁾とは、赤外線や超音波などのセンサーを利用する方法である。モバイルロボットは、移動をしながらリアルタイムな衝突回避が可能である。しかし、センサーの有効距離は短いため、他のモバイルロボットの発見が遅くなる場合や、モバイルロボット間に障害物等がある場合には、お互いの認識が遅くなり衝突する可能性がある。また、小さい範囲に複数のモバイルロボットが集中して存在する場合などには、他のロボットの移動経路を予測する事が難しいため衝突をする場合がある。センサーベースのモバイルロボットは、静止した物体に対しての衝突回避はリアルタイムに可能であり有用であるが、双方が移動物体である場合には、確実な衝突回避をおこなう事は困難である。

一方、ネットワークベースでの衝突回避とは、ワイヤレスネットワークを用いメッセージの交換を利用する方法である。ネットワークを用いる場合、ネットワークの状態などによりメッセージの交換に遅延が発生する場合もあるため、センサーベースの衝突回避と比べるとリアルタイム性は低い。しかし、モバイルロボットが移動したい経路をあらかじめ予約し他のモバイルロボットが利用できないようにする事により確実に衝突が回避できる。

本稿では、複数の自律分散型モバイルロボットが確実に衝突を回避する事を目的に、ネットワークを用いたメッセージの交換により移動経路の予約システムを提案する。また、提案したシステムをシミュレーションを用いて性能調査をする。

2章では、システムモデルに関して述べる。3章では、システムの概要に関して述べる。4章では、シミュレーションの結果を示し、シミュレーション結果に基づく考察をおこなった。

2. システムモデル

本章では、衝突回避システムのモデルに関して述べる。

2.1 モバイルロボット

本稿で述べる衝突回避システムは、次の条件を満たしたモバイルロボットを対象とする。

- ・自律的に自走可能で、移動経路を作成する機能を持っている。
- ・ポジショニングシステムを利用する事により随時に環境に対する絶対位置を計測することができる。
- ・ワイヤレスネットワークを用いる事によりメッセージの交換をすることができる。

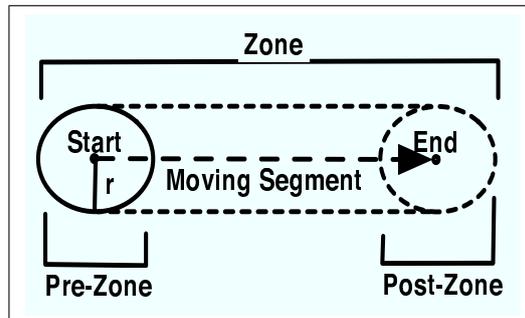


図 1 予約 Zone
Fig. 1 reservation zone

- ・各々固有の識別情報を持ち、環境内にある全てのモバイルロボットの存在をあらかじめ認識している。
- ・各モバイルロボットには、各々が独立した非同期で実行可能なタスクが割り当てられている。本システムではワイヤレスネットワークを利用するためメッセージの遅延などが考えられるので、全てのモバイルロボットは非同期な移動が可能である事が重要であるからである。

2.2 予約 Zone

本システムは、モバイルロボットの移動経路を予約し他のモバイルロボットが同時に同じ場所を移動できないようにする。予約 Zone とは、モバイルロボットが予約する移動経路を示す。本稿では、モバイルロボットの外接円をモバイルロボットの外形とし、モバイルロボットの移動開始場所を start、目的地点を end とする。予約 Zone は、start、end、start と end の共有接線で囲まれる範囲を示す。また、予約 Zone の各パーツを示す場合には、図 1 で示すように start を pre-zone、end を post-zone とする。

2.3 衝突の検出

2つの Zone が衝突する可能性がある場合は、以下のパターンである。

- (1) 双方の Zone から pre-zone と post-zone を除いた部分
- (2) 一方の pre-zone と他方の post-zone
双方の Pre-Zone に関しては移動前から既に衝突しているため衝突の検出ではない。
- (3) 一方の Zone から Pre-Zone と Post-Zone を除いた部分と他方の Post-Zone もしくは Pre-Zone
ただし、一方の Zone の Pre-Zone と Post-Zone の間と他方の Pre-Zone の場合には、予約の順序によってデッドロックになる可能性がある。図 2 上では、1)R1 : R2、2)R1 : R3、3)R3 : R4 に対

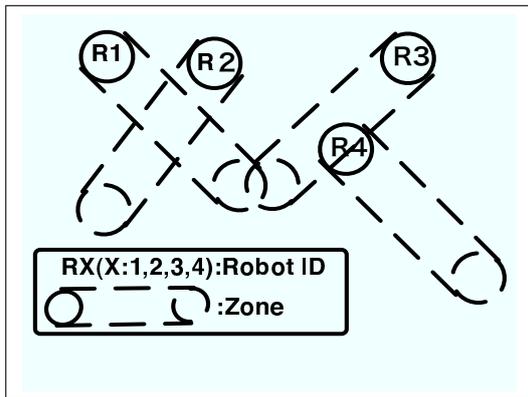


図 2 衝突の可能性のある Zone
Fig. 2 Zones have intersection

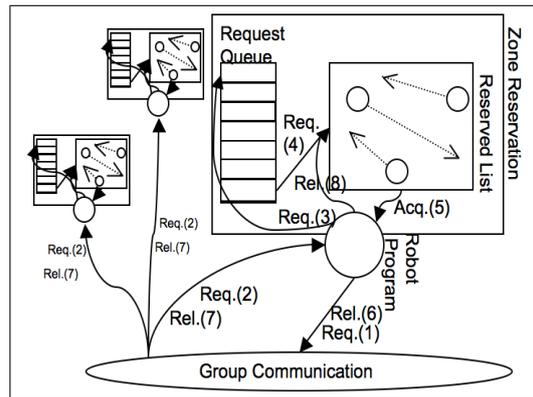


図 4 Zone 予約の分散管理
Fig. 4 decentralized zone reservation

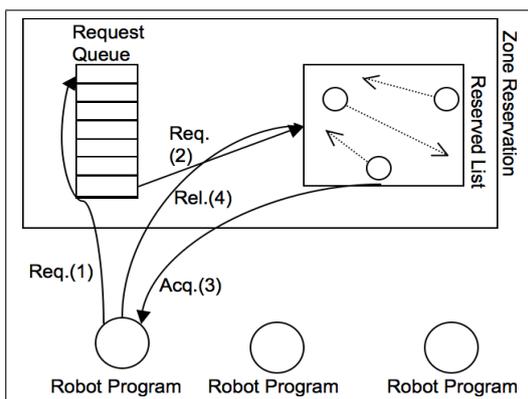


図 3 Zone 予約の集中管理
Fig. 3 centralized zone reservation

応する。

3. システムの概要

本章では、ネットワークベースにおけるモバイルロボット間での衝突回避システムに関して検討する。

前述のとおり、ネットワークベースにおけるモバイルロボット間での衝突回避システムは、移動経路をあらかじめ予約し他のモバイルロボットが利用できない状態にするブロッキングシステムである。モバイルロボットの移動経路をあらかじめ予約するためには、確実な予約管理をすることが重要となる。予約の管理方法には、集中的管理方法と分散的管理方法がある。

3.1 集中管理型

本節では、衝突回避に用いる Zone 予約を集中的に管理する方法に関して述べる。図 3 は、一つのサーバが Zone 予約を集中的におこなう場合を示している。

移動のための Zone 予約は特定のサーバもしくはモバイルロボットによって管理をし集中的な衝突回避を

おこなう。特定のサーバやモバイルロボットが Zone 予約を管理し衝突回避をおこなうため。モバイルロボットや Zone 予約の管理が容易にできる。しかし、Zone 予約を集中的に管理し衝突回避をおこなうモバイルロボットやサーバに何らかの障害が発生した場合には、最悪の場合にはシステム全体が停止する結果となり、さらに発生した障害からの復旧も困難である。また、ネットワークの一部の負荷が大きく Zone 予約が大幅に遅延する場合などには、システム全体がひとつの Zone 予約の遅延に影響される。

3.2 分散管理型

本節では、衝突回避に用いるメッセージを分散管理する方法に関して述べる。特定のモバイルロボットやサーバが Zone 予約を管理し衝突回避をする事は無いため、モバイルロボット間で Zone 予約をネットワークを用いたメッセージの交換により全てのモバイルロボットが同じあるアルゴリズムの衝突回避行動により自律的に衝突を回避する事が重要である。モバイルロボット間で Zone 予約の集中管理し衝突回避をする場合と比べ、モバイルロボットや Zone 予約の管理は複雑であるが、特定のサーバやモバイルロボットの故障がシステム全体に影響する事は無い。また、ネットワークの一部の負荷が大きくなりいくつかのメッセージが大幅に遅延した場合でも、システム全体への影響は少ない。

Zone 予約の集中管理型は、モバイルロボットの管理が容易である一方、ひとつのモバイルロボットの障害がシステム全体に影響を及ぼすシングルポイントフェーラに弱い。しかし、Zone 予約の分散管理型の場合には、モバイルロボットの管理は複雑ではあるがシングルポイントフェーラには強いため、本稿では分散管理型での Zone 予約管理方法により衝突回避をおこ

なう。図5のように、分散的に Zone 予約の衝突回避に関する確認をおこなう場合、全てのモバイルロボットが同じ結果をだす事が重要である。

複数のモバイルロボットが自律的に Zone 予約を分散管理して確実に衝突回避をするためには、全てのモバイルロボットは共通の Zone 予約を取得する事が重要である。全てのモバイルロボットが各々に異なる情報を基に衝突回避をおこなった場合には、モバイルロボットによって認識する Zone 予約が異なるため他のモバイルロボットの Zone 予約の受理を認識せず別のモバイルロボットが移動を初め衝突する場合がある。

3.2.1 Zone 予約の共有

全てのモバイルロボットが共通の Zone 予約を取得する方法のひとつに broadcast があげられる。broadcast とは、不特定多数のモバイルロボットに同時に Zone 予約を送信する方法をである。前述のとおり、モバイルロボット間でのメッセージ交換にはワイヤレスネットワークを用いるが、ワイヤレスネットワークはケーブルを使ったネットワークと比べると通信速度が遅いため、ケーブルを使ったネットワークと比べ通信の遅延が大きい。また、モバイルロボットは自律的に移動をするため、移動先の場所によってはワイヤレスネットワークのシグナルが弱い場合もある。ワイヤレスネットワークのシグナルが弱い場合には、パケットロスが多くなるため全ての Zone 予約を受信できなかった場合には他のモバイルロボットと異なった情報を持つ事になる。

単純な broadcast を用いる場合、送信元のモバイルロボットは受信側のモバイルロボットが Zone 予約を正しく受信したかどうかの確認をおこなわないため、ネットワーク上でのパケットロスが原因で全てのモバイルロボットが Zone 予約を受信しない場合がある。Zone 予約を受信できなかったモバイルロボットがあった場合には、Zone 予約を受信したモバイルロボットと異なった独自の新しい衝突回避をする事になり、全体としての協調作業が困難になる。また、一部のネットワークに高い負荷がかかっている場合などは、メッセージ遅延が発生する場合がある。Zone 予約を受信するモバイルロボットのネットワークにかかる負荷が原因で、モバイルロボットによって受信する Zone 予約の順序が前後した場合には、移動経路を予約するモバイルロボットの順番が各モバイルロボットによって異なるため、衝突する場合がある。衝突回避のためのモバイルロボット間での Zone 予約の交換は、メッセージの送受信の確実性と受信メッセージの順序の完全性が重要となる。

Atomic broadcast は、分散システムにおける基本的なコミュニケーション方法である。Atomic Broadcast は次のように定義される。操作は、A-Broadcast(m) と A-Deliver(m) で成り立っている。(m はメッセージ) 全てのメッセージは一意に識別可能であり、送信プロセスによってメッセージに含まれて運ばれる。

- ・Agreement(同意): プロセスが正しいなら、メッセージ m を A-Deliver する場合、最終的には全てのプロセスがメッセージ m を A-Deliver する。

- ・Total Order(全順序): プロセス p,q が正しいなら、p,q がメッセージ m,n を A-Deliver した場合、p は n の前に m を A-Deliver する。

これは、q が n の前に m を A-Deliver した場合に限る

- ・Integrity(完全): 任意のメッセージ m において、全てのプロセスはメッセージ m を最大1度だけ A-Broadcast する。

メッセージ m が前もって送信プロセスによって A-Broadcast されている場合に限る。

以上のことから、本システムでは atomic broadcast を用いて予約する移動経路のメッセージの完全な順序と受信を保証し、確実にロボット間での衝突を回避する。

3.3 Zone 予約の流れ

本節では、複数のモバイルロボットがどのように共通の予約を共有し受理するかの流れに関して述べる。

モバイルロボットが Zone 予約をするための操作は、予約 (request) と解放 (release) である。モバイルロボットは、atomic broadcast と衝突回避システムから構成される。Atomic broadcast は、request と release の操作により発生した Zone 予約を broadcast することができる。衝突回避システムは、atomic broadcast が受け取った Zone 予約のうちまだ受理されていない Zone を管理する RequestQueue、Zone の予約が実際に受理され移動可能となっている Zone のリストである Reserved List、モバイルロボットの衝突回避以外の機能を持った機構がある。衝突回避以外の機能とは、ロボットのカーネルや移動経路の生成システムなどを示す。

Request の操作ではまず、robot は、移動経路を生成し、Zone に当てはめる。新しく作られた Zone が予約するためには全てのモバイルロボットへ通知する目的で atomic broadcast へ投入する。Atomic broadcast は、受け取った Zone を全てのモバイルロボットへ送信する。Atomic broadcast に新しい Zone がある場合には、先頭の一つを取り出し Request Queue の全

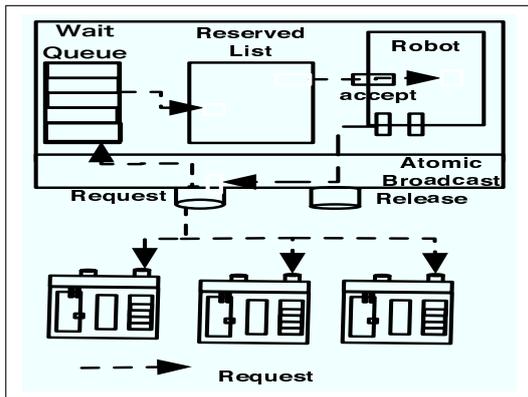


図 5 Zone 予約の流れ

Fig. 5 Data flow of Reserved zone

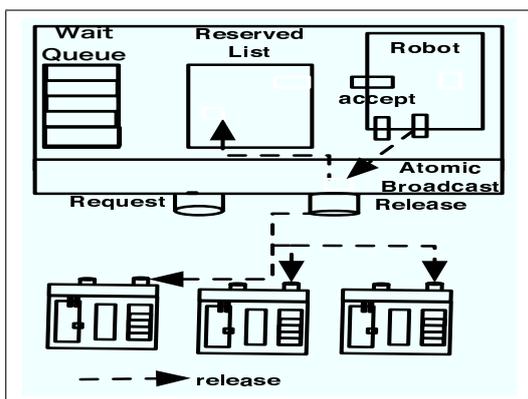


図 6 Release の流れ

Fig. 6 Data flow of Released zone

ての要素と衝突が無いか確認する。Request Queue の全ての要素と衝突の可能性がない場合には Reserved List の全ての要素と比較し、衝突の可能性が無い場合には Reserved List へ入れ、衝突の可能性があれば Request Queue の先頭へ入れる。Zone を受信した場合は、一つのモバイルロボットが予約を解放した意味なので、移動可能なモバイルロボットが無いか Request Queue の先頭から確認する。確認の場合には、要素の順序を守るために Request Queue の確認中の要素の場所以前の要素と先に衝突が無いかを確認する必要がある。また、Zone が受け取ったモバイルロボットの ID と Zone の ID が同じ場合で、Reserved List へ投入された場合には、移動可能状態になった事を Robot へ通知する。図 5 は、request の操作の具体的な流れである。

release の操作では、request の操作でモバイルロボットに移動可能状態との通知を得たモバイルロボットが移動を終了した場合に発生する操作である。モバ

イルロボットは移動を終了した場合、pre-zone と post-zone の両方がモバイルロボットの移動点である Zone が atomic broadcast へ渡される。atomic broadcast が release の zone を受け取った場合には、Reserved List 内の release の Zone と同じ ID を持った要素が削除され、予約が解放される。解放された Zone と衝突の可能性があるので待機していたモバイルロボットが無いか、Request Queue の先頭の要素から順に確認する。Request Queue の要素の順序を壊さないようにするため、確認する要素と確認する要素以前の要素に衝突の可能性が無いかを確認してから Reserved List と確認する。図 6 は、モバイルロボットの移動が終わり予約していた Zone を解放する場合の流れである。

4. シミュレーション

前章で述べた衝突回避システムをシミュレーションするための予備的となる簡易のシミュレーションをおこなった。本章では、簡易のシミュレーションの結果に関して述べる。シミュレータには、DESMO-J⁽⁶⁾ を用いた。

4.1 シミュレーション環境

前述のモバイルロボット間での自律的な衝突回避システムに基づく予備評価のために、モバイルロボットを利用する範囲の大きさと利用するモバイルロボットの台数に対するモバイルロボットの移動量をシミュレーションにより評価した。仮想空間は正方形で、モバイルロボット以外にモバイルロボットの移動とって障害物となる物体は無いものとする。仮想空間内の全ての場所で安定したワイヤレスネットワークとポジショニングシステムが利用可能とする。各モバイルロボットの初期位置は衝突が無いようにモバイルロボットの利用範囲内にランダムに配置され、Zone 予約では終点をモバイルロボットの利用範囲内にランダムに生成する。

静的に与えたパラメータを以下に与える。

- ・モバイルロボットの大きさ：半径 0.1[m]
 - ・モバイルロボットの移動速度：1.0[m/s]
- 以上の条件により、以下のようにパラメータを変化させた場合のモバイルロボットの総移動回数の変化を測定する。
- ・モバイルロボットの利用環境の 1 辺の大きさを 10[m], 100[m], 1000[m] に変化させる。
 - ・モバイルロボットの利用環境の書く大きさに対して、利用するモバイルロボットの台数を 2 台から 30 台まで 2 台ずつ増加させる。

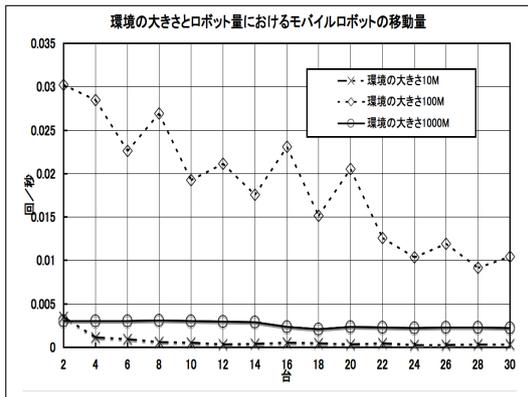


図 7 シミュレーション結果：モバイルロボットの利用範囲の大きさとモバイルロボット台数の関係

Fig. 7 result of simulation: the relation between number of robots and the are size for using mobile robots

各シミュレーションは、70[h] 間の実行を 10 回づつおこなった平均値をシミュレーション結果とした。

4.2 結果

図 7 は、前節で述べたシミュレーションの結果である。横軸方向に環境内にあるモバイルロボットの台数、縦軸方向にシミュレーションの実行時間内に移動したモバイルロボットの延べ数を示している。

モバイルロボットの利用範囲の 1 辺が 10[m] の場合と 1[km] の場合は、シミュレーション時間内で延べ 1000 以下のロボットが移動している。一方で、モバイルロボットの利用範囲が 100[m] の場合は、シミュレーション時間内に延べ 2000 台以上が移動しており、モバイルロボットの利用範囲内にモバイルロボットが 2 台の場合には延べ約 7500 台が動いている事がわかる。

4.3 考察

本シミュレーションでは、モバイルロボットの移動経路は各モバイルロボットがランダムに生成した。モバイルロボットの利用範囲が 1 辺が 10[m] の場合には、各々の Zone の予約距離は短いため Zone の予約から解放までの時間は短い、モバイルロボットの密度が密になるため衝突の可能性が高くなり予約を待つモバイルロボットが多くなったと考える。また、モバイルロボットの利用範囲の 1 辺が 1[km] の場合、モバイルロボットの密度は疎になるが一回で予約する Zone が長くなるため 1 台のモバイルロボットの 1 回の予約時間が長くなったと考える。一方で、モバイルロボットの利用範囲の 1 辺が 100[m] の場合には、モバイルロボットの密度と一回に予約する Zone の長さのバランスが良かったため、他の 2 つの結果と比べ多くのモバイルロボットがシミュレーション時間内に移動でき

たとえる。以上のことから、本稿で提案する衝突回避システムを利用する場合には、モバイルロボットの密度と予約する Zone のバランスを考慮する事が重要であることがわかる。

5. まとめ

本稿では、自律型のモバイルロボット向け衝突回避システムに関して述べ、シミュレーションによりモバイルロボットの利用範囲と利用するモバイルロボットの台数の関係を評価した。シミュレーションの結果、モバイルロボットの密度と予約する Zone の長さのバランスを考慮する事が重要であることがわかった。モバイルロボットの密度と予約する Zone の長さのバランスに関してさらに評価する必要がある。

今後の課題としては、本稿ではモバイルロボットの移動速度は一定と仮定したが実際にはモバイルロボットごとに異なると考えられるため、モバイルロボットの台数と移動速度の関係を調べる必要がある、また、実際のモバイルロボットの移動には移動距離や方向などの誤差があるため、それらの誤差をシミュレーションに反映させる必要がある。デッドロックに関して、タイムアウトなどの方法により回避する必要がある。

参考文献

- 1) 田中真紀子, 佐川雄二, 田中敏光, 杉江昇, “多次元的性格を持つロボット群における共同作業 — 積極性と優しさ”, 電気学会論文誌, 124 巻, 12 号, 2004.
- 2) 刈谷学, 鎌野琢也, 安野卓, 鈴木茂行, 原田 寛信, 小倉 明香, “捕獲問題における群移動ロボットの協調行動の実現”, 電気学会論文誌 C, Vol.124, No.2, pp502-508, 2004.
- 3) Rami YARED, “Fail-safe Mobility Management and Collision Prevention Platform for Cooperative Mobile Robots with Asynchronous Communications”, Ph.D thesis, Japan Advanced Institute of Science and Technology (JAIST), 2006. 9.
- 4) 樹野淳也, 梅木嘉道, 石田祐樹, 小林尚登, “対戦ワイヤを用いた多自由度マニピュレータの障害物回避起動の生成”, IEEJ, Trans, EIS, Vol.125, No.2, 2005.
- 5) Xavier Defago, Andre Schiper and Peter Urban, “Total order broadcast and multicast algorithms: Taxonomy and survey”, ACM Computing Survey, Vol.36, No.4, pp.372-421, December 2004.
- 6) DESMO-J, <http://www.desmoj.de>, University of Hamburg Department of Computer Science.