

人と協調する倉庫内配送ロボットの シミュレーションを用いた最適化の検討

浅井 悠佑² 廣井 慧³ 米澤 拓郎^{1,2} 河口 信夫^{1,2}

概要: 本研究では, 倉庫における人 (ピッキング作業) と配送ロボットの最適化における検討をシミュレーションを用いて行う. 近年, E コマースの利用が盛んになるにつれ, 倉庫で扱う品物の数や種類, 個人向け注文の増加に伴うピッキング作業への負担が増加している. そのような背景から, 従来の倉庫では処理のキャパシティを超えてしまうことがあり, 倉庫の自動化が急激に進んでいる. しかし, 倉庫の自動化に伴う大規模な投資を行うのが難しい事業者も多数あるのも事実であり, そのような事業者は既存の倉庫を用いながら作業効率を向上させる必要がある. 従来の倉庫において, 配送ロボットによるピッキング作業の補助を行うことで, ピッキング作業の負担を軽減できると考える. そこで本研究では配送ロボットに自動運転技術を用い, 効率的なピッキング作業のフローをシミュレーションによって検証する.

Optimization of Human Workload using Simulation of Warehouse Collaborative Robot

Yusuke Asai² Kei Hiroi³ Takuro Yonezawa^{1,2} Nobuo Kawaguchi^{1,2}

1. 背景と目的

本研究では, 倉庫における人 (ピッキング作業) と配送ロボットの最適化における検討をシミュレーションを用いて行う. 近年, E コマースの利用が盛んになるにつれ, 倉庫で扱う品物の数や種類, 個人向け注文の増加に伴うピッキング作業への負担が増加している. そのような問題への対処として, 倉庫を完全自動化することで, 倉庫の処理能力向上を目指す動きもあるが, 完全自動化には多額の投資が必要であり, その投資が困難な業者も少なくない. 例として, 小さい規模の倉庫などは, 倉庫の完全自動化を行う投資に対して, 十分なリターンや恩恵を享受することが難しい場合も多々考えられる. そのような背景から, 中小規模の倉庫において, 多額の投資が必要でない自動化による注文処理能力向上への手法が必要であると考え.

一部の作業を自動化する手法では, 効率化を行うことが

できる反面, 必然的にロボットと人が交わり合う場面が増えることで, 事故や人間では起こし得ないミスが発生する可能性がある. 倉庫におけるピッキング作業とは, 人が品物をピッキングして出荷場所へ運ぶことを言う. 本研究では, ピッキングは人が行い, 出荷場所へ運ぶ作業を自動配送ロボットに行わせることで, 作業の効率化を検証する. その際, 人と配送ロボットが交わり合う場面では, 人に対して安全であるのは当然として, どのようにしてロボットを制御, 運用していくのかがとても大切である.

倉庫内において, 人とロボットがどのように協調して作業を行うかの研究は様々な手法 [1] によって検証が行われているが, 倉庫の種類や大きさなどが多岐にわたるため, それぞれの倉庫の環境に応じた最適な手法を検討するためのシミュレーションの仕組みが必要である. また, 本研究では, シミュレーションによって開発された配送ロボットの制御手法を用いて, 実際の倉庫における, 実機での配送ロボットによる検証が容易に行えることを念頭に実装を行う.

従来の倉庫では, High Bay Automation[2] とよばれる, 20m から 40m の高さの倉庫において, 大きなユニット単

¹ 名古屋大学未来社会創造機構 Institute of Innovation for Future Society, Nagoya University

² 名古屋大学大学院工学研究科 Graduate School of Engineering, Nagoya University

³ 京都大学防災研究所 Disaster Prevention Research Institute, Kyoto University

位での在庫管理が一般的であった。しかし、E コマースの発達により、小さな SKU(在庫の管理単位) かつ注文あたりにおける量が少量な注文を効率的に処理可能な倉庫の管理手法が必要になってきている。また従来、倉庫において用いられてきた配送ロボットは事前に決まった経路を走行するのみで、障害物の回避は困難であった。しかし、近年では自動運転技術やセンサーの発達により、自走式の配送ロボットの開発が盛んに行われており、倉庫においても配送ロボットを活用できる。自動運転技術によって、配送ロボットが自己位置を推定したり、周りの環境を認識して、ある程度自由な走行が可能である。

そこで本研究では、従来の中小規模の倉庫において、配送ロボットを用いたピッキング作業の補助を行うことで、ピッキング作業の負担を軽減し、効率化を目指す。配送ロボットに自動運転技術を用い、効率的なピッキング作業のフローをシミュレーションによって検証する。

2. 関連研究

倉庫における最適化の手法は様々あり、それに関する研究も盛んに行われている。本研究で用いる手法以外で配送ロボットを用いる代表的な手法としては、品物を保管している棚自体がピッキング作業者のいるところまで自走して、ピッキング作業者が品物をピッキングするというものである。

2.1 AGV を用いた棚移動による倉庫管理システム

A. Alfieri ら [3] はスペースの限られている倉庫においても、AGV を用いて棚を動かしてフレキシブルな在庫の配置が可能な手法を開発した。棚自体を自動で動かす利点としては、都市部などにおいて広いスペースの確保が困難な場合、スペースを最大限有効に活用するために、固定棚の場合に必要な通路を無くし、スペースを最大限有効に活用できる点が挙げられる。自走式の棚を用いての倉庫の配置移動は最適化手法の 1 つであるが、自走式の棚自体のコストがかかることから、AGV を用いて棚を移動させることで、棚自体のコストを低下させられると述べられている。しかし、棚の数が増えたり注文が複雑化すればするほど最適化が困難になってしまいパフォーマンスが低下するなどの問題あり、ピッキングの順番を棚の位置等に合わせてダイナミックに変化させる必要があるなどの問題点がある。

2.2 レール式 AGV によるピッキングシステムシミュレーション

Debjit Roy ら [4] は、レール式の AGV を用いたピッキングシステムのシミュレーション手法を開発した。このシステムでは AGV 自身がサイズが規格化されている品物を自らピッキングし、また、リフトを用いて上に在庫スペースを拡張できるため、高密度での収容が可能となっている。

この手法は、大規模向けの在庫管理システムであり、注文あたりにおける量が少量な注文を処理するには向いておらず、かつ多額の投資を必要とするなどの問題点がある。

3. Autoware による配送ロボットの制御

倉庫における配送ロボットの制御には Autoware を用いる。Autoware は自動運転のソフトウェアで、ロボットの汎用ライブラリである ROS をベースに開発が行われている。

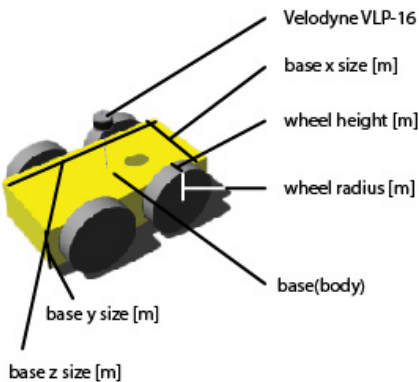


図 1 Gazebo 上で用いるロボット

3.1 3 次元シミュレーター Gazebo を用いた実験環境の作成

実際の倉庫(トラスコ中山プラネット埼玉)の 3 次元モデルを使用して、その上で仮想的なロボットを制御して、シミュレーションでの検証を可能にする。本研究では、ROS によって動作するロボットの 3 次元シミュレーションに対応している Gazebo を用いる。また、3 次元シミュレーター Gazebo 上で動作するロボットは図 1 のように 4 輪であり、構成は表 1 のように定義する。

表 1 ロボットの構成の詳細

wheel radius	0.2 [m]
wheel height	0.1 [m]
wheel mass	1.5 [kg]
base x size	1.0 [m]
base y size	0.5 [m]
base z size	1.5 [m]
base mass	15 [kg]
3D LIDAR	Velodyne VLP-16

Gazebo 上で動作する仮想的な 3 次元 LIDAR をロボットに搭載することで、点群地図の作成、自己位置推定やその他ロボットの制御に使用できる。

3.2 ndt-matching を用いた自己位置推定

配送ロボットの制御にまず必要な事として、ロボットが自分自身の位置を正確に知ることである。これを実現する方法として、Autoware に搭載されている ndt-matching を

用いる。本研究で用いる ndt-matching とは、図 2 に示すように、3 次元のスカンデータと、予め作成しておいた 3 次元の点群地図を用いて、自己位置を推定する技術である。この手法には、計算量が地図データの大きさに依存せず、スカンデータだけに依存することから、倉庫の大きさが大きくなり、点群地図のデータ容量が大きくなっても、ロボット制御のパフォーマンスに影響が無いなどの利点がある。弱点として、初期値の与え方によってはマッチングの精度が大きく劣後してしまうため、IMU や GNSS を用いて補正を加えながらの使用が多い。

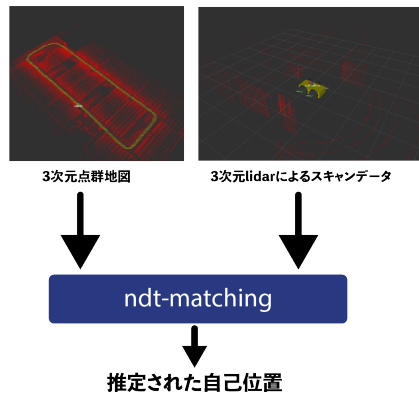


図 2 ndt スキャンマッチング

3.3 経路の生成と障害物の回避

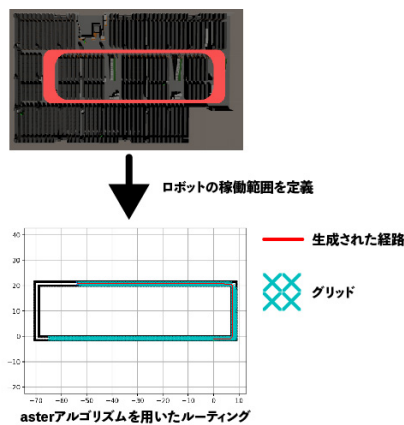


図 3 aster アルゴリズムによる経路生成

ロボットがある地点から、別の地点へ移動する際には、そこに至るまでの経路を生成する必要がある。また、移動途中に人や様々な障害物を回避するために局所的な経路の生成や制御も必要である。本研究における大局的な経路の生成は、あらかじめロボットの稼働範囲を定義して、その領域でのみ移動可能とし、経路探索や制御の簡略化を行う。図 3 に示すように、ロボットの稼働範囲を定義して、その範囲に基づいて a-ster[5] アルゴリズムを用いて経路を生成する。a-ster アルゴリズムとは、経路探索の手法の 1

つであり、ダイクストラ法と比べて、計算量が少なく高速であるという利点がある。また局所的なロボットの制御は、今回は簡略化のために、進行経路上に障害物や人がいた場合、一定の距離以内に近づくと停止するものとする。

4. ピッキング作業のシミュレーション

4.1 ピッキング作業の出荷場所へ移動の予測

ピッキング作業は通常、ピッキング作業者がピッキング作業を行った後、カートにピッキングした品物を置き、カートがある程度いっぱいになったところで、出荷場所へ運ぶというフローになっている。ピッキング作業の履歴データから、作業者のカートがいっぱいになるタイミングを予測できると考える。ピッキングデータの詳細としては、表 2 のように、品物やその品物をピッキングした日付・時刻が記録されている。本研究で行うシミュレーションでは、ピッキングが完了した後、次のピッキングが完了するまで一定以上の時間 (200 秒) が経過している場合、ピッキング作業者が出荷場所へ移動して、品物をカートから下ろしたと仮定する。もちろん、ピッキング作業者の休憩であったり、作業者同士の立ち話など様々な要因によってピッキングの時間が長くなることは考えられるが、シミュレーションではランチタイムを除いてそれらを一切考慮しない。ある作業者の品物ごとの作業時間をプロットしたものを図 4 に示す。これは、品物ごとにどれだけ作業時間がかかったのかを記録したデータであり、一定以上の時間が経過しているものは赤色にマークを行っている。この赤色のデータが示す作業は、出荷場所へ移動した後、次のピッキング地点へ移動したものであるとみなす。

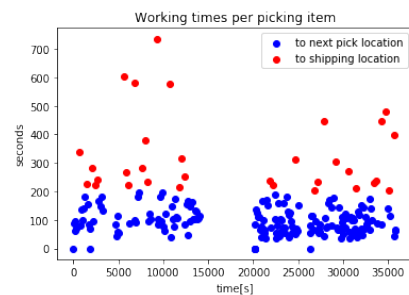


図 4 ある作業者の品物ごとの作業時間

4.2 Unity を用いた人の動きの可視化

シミュレーションを行う際に、ピッキング作業を行う人の動きを再現する必要がある。本研究では、実際のピッキング作業の記録を用いて、Unity 上で可視化を行う。Unity 上で用いる人のモデルは、フリーのアセットである「Basic Motions FREE Pack」[6] を用いる。ただし、ピッキング作業者の歩行軌跡のデータは現段階では入手不可能であるため、Unity の標準機能である navmesh を使用して人の移動

可能範囲を図5のように指定して、目的地までのルーティングを行い、人の動きを再現する。この経路探索によって導かれた経路は、実際のピッキング作業者が通った経路とは、完全には一致しないと予想される。

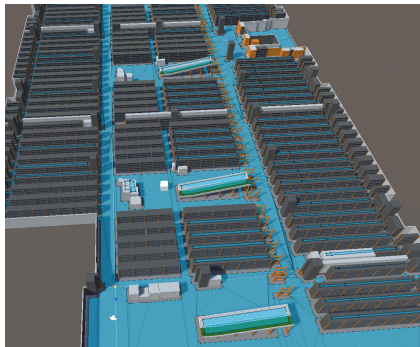


図5 navmesh を用いたピッキング作業者のルーティング

4.3 シミュレーションで用いるピッキングデータ

表 2 ピッキング作業のデータの例 (一部抜粋)			
作業者 ID	品物名	完了時刻	ロケーション
65380	製品 A	9:00:02	2C-2A
63365	製品 B	9:00:31	2B-2B
74254	製品 C	9:03:59	2C-1M
74254	製品 D	9:04:48	2C-1M
63365	製品 E	9:05:03	2B-2D
63365	製品 F	9:05:07	2C-2F

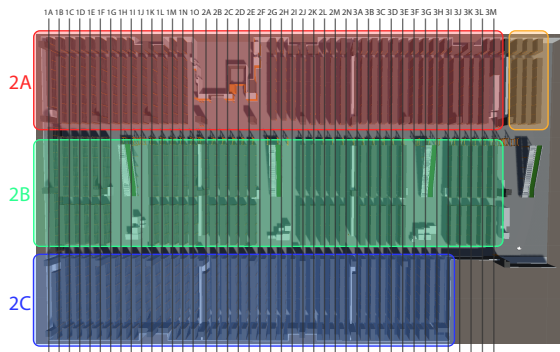


図6 倉庫におけるピッキングロケーション

シミュレーションでピッキング作業者の動きを再現する際に、表2のような、実際のピッキング作業で得られたピッキング履歴データを用いる。このデータは、作業者 ID、品物名、ピッキング完了時刻、ピッキングロケーション、品物容量などからなっている。作業者 ID、ピッキング完了時刻、ピッキングロケーションを用いて、図6に示すようなピッキングロケーションの区切りから、目的地を求め、時刻に合わせてピッキング作業者の動きを再現する。大まか

な区切りが表2における、ロケーションの1つ目の数字とアルファベットに対応しており、2つ目の数字とアルファベットが倉庫の棚の列に対応している。

4.4 シミュレーションの全体構成

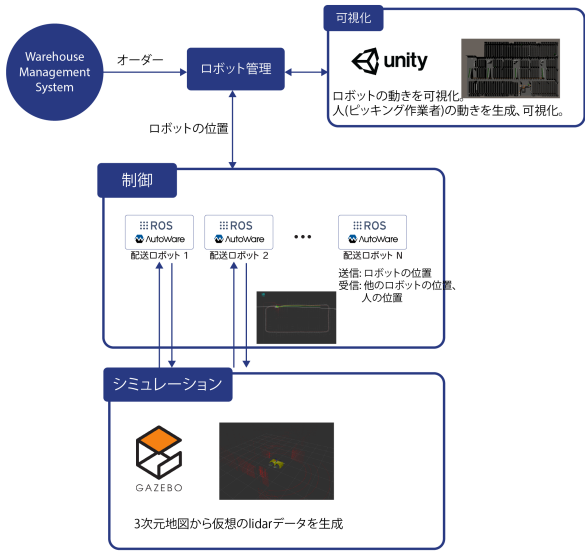


図7 シミュレーションの全体構成図

本研究で行うシミュレーションの全体構成を図7に示す。大まかにUnityにおける「可視化」、ROSやAutowareを用いた「制御」、ロボットの3次元シミュレーターであるGazeboを用いた「シミュレーション」の3つからなる。Warehouse Management System から送られてきたピッキングオーダー (シミュレーションにおいては、ピッキングデータの履歴) をロボット管理ノードが受け取り、適切に配送ロボットをピッキング作業者の元へ配車したあと、配送ロボットは出荷場所へ移動するという流れである。

5. 実験

本研究の実験では、シミュレーション上でピッキングデータを用い、AGVの使用の有無によってピッキング作業がどれだけ効率化できるのかを検証する。

5.1 実験の内容

図8のように、Unity 上において実際のピッキング作業の様子を再現し、その上で仮想的な AGV を動作させてピッキング作業者を補助する様子を再現する。表2に例示したピッキング作業のデータを用いて、時系列に沿って各ピッキング作業者にピッキング司令を送信する。ピッキング作業者の作業フローは図9のようになっている。また、シミュレーションにおけるピッキング作業者の歩行速度 [7]、AGV の上限速度 [8] は 1.4m/s、1.0m/s とする。また、今

回用いる手法としては Pick Support AGV[1] 手法を用いる。この手法は、ピッキング作業者のカートがいっぱいになったタイミングで、次のピッキングロケーションにおいて AGV と作業者が待ち合わせを行って、その地点で品物を AGV に渡すというものである。また、使用する AGV の台数は作業者の人数 (今回のシミュレーションでは 8 人) と等しいとする。今回実験でピッキング作業者が行うピッキング作業を表 3 に示す。

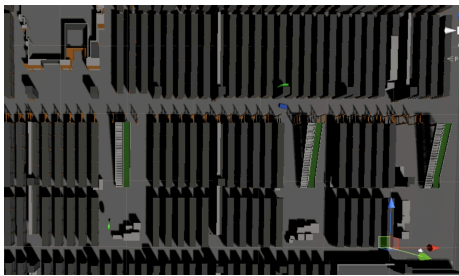


図 8 Unity 上でのシミュレーションの様子

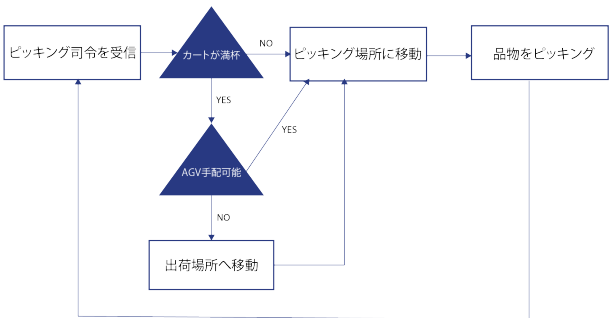


図 9 ピッキング作業者の作業フロー

表 3 作業者ごとの作業量

作業者 ID	ピッキング数 [個]	出荷作業回数
65380	462	37
63365	630	24
67398	687	19
74254	558	16
96176	376	22
97107	140	9
95195	545	31
83132	413	18

5.2 実験結果

シミュレーションの結果として、従来の手動で行う手法と Pick Support AGV 手法を用いた時の作業者ごとの総作業時間、総歩行距離をそれぞれ表 4、5 に示す。また、品物の出荷時における作業者ごとの作業時間の分布を図 10 に示す。この結果から AGV を用いることで、品物出荷時において作業時間の短縮ができていることが確認された。ま

表 4 シミュレーション結果 (AGV 非使用)

作業者 ID	総作業時間	歩行距離 [km]
65380	8h58m	13.9
63365	8h48m	20.5
67398	8h20m	15.3
74254	7h44m	15.4
96176	7h43m	11.9
97107	3h02m	4.6
95195	8h06m	9.5
83132	8h38m	11.1

表 5 シミュレーション結果 (AGV(x 作業者の人数) 使用)

作業者 ID	総作業時間	歩行距離 [km]
65380	8h45m	10.5
63365	8h35m	18.9
67398	8h13m	13.5
74254	7h39m	14.0
96176	7h35m	10.0
97107	3h01m	3.8
95195	8h01m	8.3
83132	8h32m	9.6

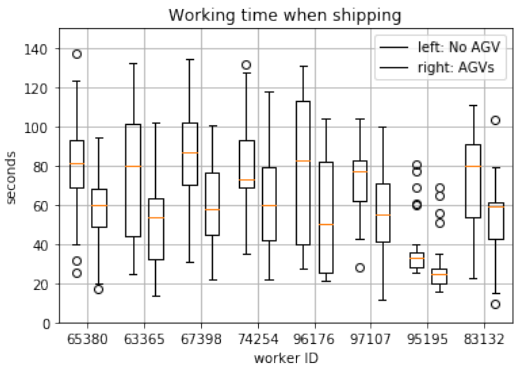


図 10 出荷作業における作業者ごとの作業時間

表 6 AGV 使用時のパフォーマンス評価

作業者 ID	作業時間削減率 [%]	歩行距離削減率 [%]
65380	2.49	24.4
63365	2.36	7.56
67398	1.49	11.7
74254	1.03	9.49
96176	1.78	16.2
97107	1.15	16.0
95195	0.98	12.7
83132	1.07	13.8

た表6に示すように、総作業時間に関してはあまり変化が見られなかったが、出荷作業を配送ロボット (AGV) が行うことでピッキング作業者の総歩行距離が大幅に短縮されたことがシミュレーションの結果から示された。今回行った実験はピッキング作業者と AGV が同期的に品物を渡すというものであったが、今後、人がピッキングした品物を特定の受け渡し場所に置いて、AGV がその品物を回収するなどの非同期的な手法も実装、検証を行いたいと考えている。

6. おわりに

シミュレーション上において、配送ロボットを用いて倉庫におけるピッキング作業の効率化を検証した。今回行った実験では、Unity 上で AGV を用いるとどの程度作業が効率化されるかを確認し、ピッキング作業者の歩行距離が大幅に短くなったことが確認された。シミュレーション環境を用いて開発された配送ロボットの制御手法は、今後、ROS や Autoware を用いて実際の現場での使用を想定しており、シミュレーション結果との差異などを今後検証していく予定である。今回行った実験はピッキング作業者と AGV が同期的に品物を渡すというものであったが、今後、様々な配送ロボットの制御手法を研究、開発していきたいと考えている。

謝辞

本研究は、JST CREST JPMJCR1882, NICT 委託研究, 総務省 SCOPE, JST OPERA(JPMJOP1612), NEDO SIP2 期の支援を受けたものです。

参考文献

- [1] Azadeh, Kaveh, René de Koster, and Debjit Roy. Robotized warehouse systems: Developments and research opportunities. No. ERS-2017-009-LIS. 2017.
- [2] Roodbergen, K. J. and I. F. Vis (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research* 194 (2), 343–362.
- [3] Alfieri, A., M. Cantamessa, A. Monchiero, and F. Montagna (2012). Heuristics for puzzle-based storage systems driven by a limited set of automated guided vehicles. *Journal of Intelligent Manufacturing* 23 (5), 1695–1705.
- [4] Roy, D., A. Krishnamurthy, S. S. Heragu, and C. J. Malmberg (2014). Blocking effects in warehouse systems with autonomous vehicles. *IEEE Transactions on Automation Science and Engineering* 11 (2), 439–451.
- [5] J. Yao, C. Lin, X. Xie, A. J. Wang and C. Hung, "Path Planning for Virtual Human Motion Using Improved A* Star Algorithm," 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, 2010, pp. 1154-1158, doi: 10.1109/ITNG.2010.53.
- [6] Kevin Iglesias. Unity Asset Basic Motions FREE Pack. <https://assetstore.unity.com/packages/3d/animations/basic-motions-free-pack-154271>
- [7] Browning, R. C., Baker, E. A., Herron, J. A. and Kram, R. (2006). "Effects of obesity and sex on the energetic cost and preferred speed of walking". *Journal of Applied Physiology*. 100 (2): 390–398.
- [8] Masanori Tsujimoto (2013). "Development of a Safe High-Speed AGV System in Distribution Centers. The Proceedings of the Transportation and Logistics Conference 2013.22(0), 7-9, 2013"