

コンピュータ科学科

知能システム系

卒業論文

シミュレーションと強化学習に基づく

移動障害物回避機能を持つ

自律移動ロボットに関する研究

概要

近年、高齢者や歩行に困難がある障がい者の移動手段として、パーソナルモビリティの需要が増加している。パーソナルモビリティは小型であること、自動車に比べて低速であること、小回りが利くことから歩道や屋内での走行が想定されている。そのため、人の近くや狭い場所、複雑な地形、地図情報にない障害物が置かれている環境を安全に走行できる必要がある。また、静止障害物だけでなく、人や他の乗り物など移動する障害物にも対応する必要がある。パーソナルモビリティに自動運転技術を適用することで、より安全で快適な移動が実現できる。

本研究では、複数の移動障害物が存在する環境における、自律移動ロボットの安全な走行及び移動障害物回避の行動獲得を強化学習を用いて実現する。具体的には、シミュレーション環境を構築し、移動障害物に対する柔軟な回避行動を獲得する強化学習のモデルを構築する。

自律移動ロボットのシミュレータとして、ユニティ・テクノロジーズ社が提供するゲーム開発プラットフォームの Unity を採用し、Unity 上で機械学習を行うオープンソースなフレームワーク ML-Agents(Unity Machine Learning Agents) を活用して強化学習を行った。

自律移動ロボットは、前方のカメラ画像と 360 度のレーザ距離センサによる情報を入力として、前進速度と走行方向を決定する。移動障害物は人を想定し、人の歩行行動を模倣するように行動する。人の視界に自律移動ロボットが入っている場合には、人は回避行動をとるか・歩行を継続するか・停止するかのいずれかを選択し、視界に自律移動ロボットが入っていない場合には、人が自律移動ロボットに気づいていないと想定して歩行を継続する行動をとる。

強化学習のアルゴリズムとしてモデルフリー強化学習である Soft-Actor-Critic(SAC)を用いた。移動障害物回避には ML-Agents、目的地への経路生成には Unity 内で経路生成及び経路追従を行う NavmeshAgent を用い、移動障害物回避と目的地への経路追従を行う。人の視線情報を用いる強化学習モデル、用いない強化学習モデル、人の視線情報に加えて過去の情報を用いる強化学習モデルをそれぞれ構築し、このモデルを学習環境と他の環境で自律移動ロボットに適用し、目的地への走行テストを 1000 回行い、移動障害物回避の精度や汎化性能を確認する。また、移動障害物を回避できず衝突した事例の原因についての考察も行う。

目次

第1章	はじめに	5
1.1	研究背景	5
1.1.1	高齢社会の現状	5
1.1.2	パーソナルモビリティの現状	6
1.1.3	障害物回避の課題	7
1.2	本研究の目的・アプローチ	7
1.3	本論文の構成	8
第2章	自律移動ロボット	9
2.1	自律移動ロボットとは	9
2.2	自己位置推定	10
2.3	経路計画	11
2.4	静止障害物回避	12
2.5	センサ	13
2.5.1	距離センサ	13
2.5.2	ビジョンセンサ	14
2.5.3	本研究に関連するセンサ	16
第3章	シミュレーション環境	17
3.1	シミュレーション環境の基盤システム	17
3.2	シミュレーション環境で用いる 3D モデル	17
3.2.1	環境の 3D モデル	18
3.2.2	移動障害物の 3D モデル	19
3.2.3	自律移動ロボットの 3D モデル	20
3.3	移動障害物の挙動	21
3.3.1	移動障害物の配置	22
3.3.2	Navmesh と NavmeshAgent	22
3.3.3	視野に基づく挙動	24
第4章	移動障害物回避のための強化学習モデル	25
4.1	強化学習	25
4.1.1	機械学習の概要	25
4.1.2	強化学習とは	25
4.1.3	強化学習アルゴリズムの分類	26
4.2	ML-Agents	27
4.3	移動障害物回避モデルの学習	28

4.3.1 自律走行と移動障害物回避システム	28
4.3.2 強化学習モデルの入出力	28
4.3.3 強化学習モデルの詳細	29
4.3.4 報酬の設定	32
第5章 実験方法及び結果と考察	34
5.1 実験方法	34
5.2 実験結果	35
5.2.1 視線情報なしの現在のステップの観測を用いたモデル	35
5.2.2 視線情報ありの現在のステップの観測を用いたモデル	37
5.2.3 視線情報ありの現在と1つ前のステップの観測を用いたモデル	39
5.3 性能評価	41
5.3.1 学習済みモデルを用いた学習環境での検証	41
5.3.2 学習済みモデルを用いた他の環境での検証	44
5.4 考察	46
第6章 関連研究	51
6.1 ルールベースによる障害物回避	51
6.2 人の視線情報を活用した障害物回避	52
6.3 人の回避行動を考慮した経路計画	53
6.4 強化学習を用いた自動走行の研究	53
第7章 おわりに	55
7.1 まとめ	55
7.2 今後の課題と展望	55
7.2.1 歩行者がロボットに気づいていない状況への対処	55
7.2.2 搭乗者を考慮した快適な回避行動	56
7.2.3 実環境での実証実験	56
謝辞	58
参考文献	59

第1章 はじめに

本章ではまず 1.1 節で本研究の背景について述べ、1.2 節では本研究の目的とアプローチについて説明する。最後に 1.3 節で本論文の構成について述べる。

1.1 研究背景

1.1.1 高齢社会の現状

近年、高齢化が進んでいる。令和 4 年版高齢社会白書によると、高齢化率は 2018 年で 28.1%であり、高齢化率の増加が続くとされ、50 年後には約 40%になると推計されている[1]。65 歳以上の人口は 2045 年ごろまで増加傾向が続くと推計されている。

また高齢者の外出手段を年齢別に見ると 60 歳～79 歳までは「自分で運転する自動車」、80 歳以上は「車いす」の割合が乗り物の中で最も高くなっている[1]。

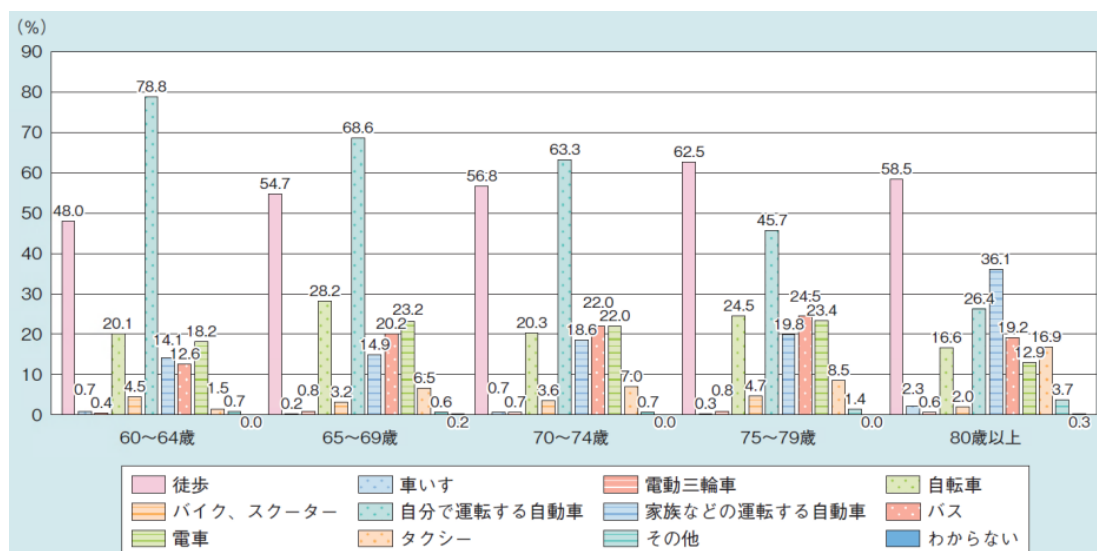


図 1.1: 年齢別の外出する際に利用する手段の割合 文献[1]より抜粋

平成 31 年には関係閣僚会議で「未就学児等及び高齢運転者の交通安全緊急対策」がま

とめられ、「高齢者の移動の足となる公共交通の利用環境の改善，最新の自動運転技術を取り入れた新しいモビリティの活用等の『高齢者の移動を伴う日常生活を支える取組』について，大胆に取り組む」と記載されている[2]．このように，移動手段の代替となる多様なモビリティの普及や促進が対策目標の一つに定められている．

1.1.2 パーソナルモビリティの現状

近年，高齢者や歩行に困難がある障がい者の移動手段として，パーソナルモビリティの需要が増加している[3]．パーソナルモビリティとは『自動車よりコンパクトで小回りが利き，環境性能に優れ，地域の手軽な移動の足となる1人～2人乗り程度の車両である』と国土交通省により定義された[4]．パーソナルモビリティは小型であり，自動車に比べて低速であるため小回りが利くことから，歩道や屋内での走行が想定されており，自動車や公共交通機関に比べて人の生活に深く関わる交通手段として期待されている．



図 1.2: 超小型モビリティの分類（走行形態別の例） 文献[4]より抜粋

平成 28 年には，国土交通省が超小型モビリティシンポジウムを開き，パーソナルモビリティ導入促進の成果や課題，将来のあり方についての共有がされた[5]．

平成 31 年には，全日本空輸（ANA）とパナソニックによるパーソナルモビリティ（自動追従電動車いす）の実証実験が行われた[6]．係員が操作する先頭の一台の車いすを他の車いすが自動追従し，乗り心地と安全性からサービスの実用化やその他の用途での活用について検討された．また，令和 4 年には WHILL 株式会社が成田空港第 2 旅客ターミナル

において自動運転パーソナルモビリティとエレベーターが連携し、出発地点から目的の搭乗ゲートまですべて自動運転で階をまたいだ移動の実証実験が行われ、実用化の検討が行われている[7].

電動車いすのような低速なパーソナルモビリティは、自動車や自転車と比べて身体能力や運転技術を必要としないため、高齢者や身体的不自由がある人の移動手段の代替になることに加え、電気を動力にしており、小型であることから AI やネットワークを用いた制御技術に関する研究や実証実験が行われている[6][7][8]. これらの研究は、パーソナルモビリティに自動運転技術を適用することでより安全で快適な移動の実現を目指している.

パーソナルモビリティは、人の近くや狭い場所、複雑な地形、地図情報にない障害物が置かれている環境を安全に走行できる必要がある. また、静止障害物だけでなく、人や自転車のように移動する障害物にも対応する必要がある.

1.1.3 障害物回避の課題

回避すべき障害物が多様化しているため、人やその他の障害物の特性の全てを考慮できず、障害物を一括りにした経路計画や走行の制御では、すべての環境や障害物に対処することが困難である. 人の行動予測が困難であることや、障害物の行動予測を行う際に予想を外れた移動障害物の回避が困難であること、壁などの障害物で隠れた移動障害物のような死角から出現した障害物の回避が困難であることなど、解決すべき困難な状況は複数存在する.

障害物に対して停止することで衝突回避を行うだけでなく、効率的な移動を行うために自動走行を継続したまま障害物の回避を行うシステムが必要とされている. また、障害物回避を行った際に当初生成した経路から外れてしまうため、目的地への生成経路の修正を行う必要もある. 人の近くを走行する自律移動ロボットが人の走行経路を認知や予測するだけでなく、自律移動ロボットに直面した際に人がとる回避行動に対応できる障害物回避システムを構築する必要がある. AI を用いた制御技術や障害物回避技術、自動走行技術の開発が期待されている.

1.2 本研究の目的・アプローチ

本研究の目的は、自律移動ロボットが複数の移動障害物が存在する環境において、目的地までの安全な走行及び移動障害物回避が可能な自律走行システムを構築することである.

本研究のアプローチとして、強化学習及び走行のテストを行うシミュレーション環境の構築、カメラ画像やレーザ距離センサを搭載した自律移動ロボットや、視野に基づく行動パターンを持つ移動障害物の作成後、経路生成手法と障害物回避手法を統合した障害物回避システムを構築し、強化学習による移動障害物回避の行動獲得を行う。移動障害物回避の行動獲得を行う強化学習モデルとして、人の視線情報を活用するモデルや過去の観測情報を活用するモデルの複数を構築し、入力情報の種類によって性能が変化するかの比較実験を行い、移動障害物回避の精度の向上を図る。

1.3 本論文の構成

本論文の構成は次のようになっている。第2章では、自律移動ロボットの構成や自律移動に必要とされている技術について説明する。第3章では、シミュレーション環境について述べ、本研究で作成・使用した実験環境、移動障害物、自律移動ロボットの構成について解説した後に、移動障害物の挙動の制御について説明する。第4章では、移動障害物回避のための強化学習モデルについて述べる。第5章では、移動障害物回避モデルを学習する実験について述べた後、性能検証の結果と考察について論じる。第6章では、自律走行において強化学習を用いているいくつかの研究を本研究の関連研究として紹介する。最後の第7章では、まとめと今後の課題と展望について述べる。

第2章 自律移動ロボット

本章では，自律移動ロボットについての説明，自律移動に必要な技術，及び本研究に係るセンサについて述べる．

2.1 自律移動ロボットとは

自律ロボットとは，自身のセンサで情報を取得し，自身のコンピュータや計算機で情報を処理して自身の行動を決定するロボットを指し，移動ロボットとは，レール等の補助を必要とせず，足や車輪によって動き回ることが可能なロボットを指す．自律移動ロボットは自律ロボットと移動ロボットの両者の特徴を持つ．

自律移動は一般的に，認知・判断・制御の処理を一定の周期で繰り返している．認知では，地図上でどこにいるかを推定する自己位置推定や，周囲の障害物を把握する障害物検出を行う．判断では，認知で得られた地図上の位置からたどるべき経路を決定し，障害物に衝突しない経路計画を行う．制御では，判断で得られた経路をたどって目的地に向かうようにロボットの駆動系を制御する．

自律移動ロボットの実例として iRobot 社の Roomba[9]や Starship Technologies のデリバリーロボット[10]が挙げられる．



図 2.1: 掃除ロボット Roomba 文献[9]より抜粋



図 2.2: フードデリバリーロボット 文献[10]より抜粋

2.2 自己位置推定

自己位置推定は、ロボットの現在の位置と向きをこれまでに得た情報から推測することであり、これにはセンサと地図が必要となる。センサは主に外界センサと内界センサの2種類に分類される。

外界センサは、カメラ・RGB-D・レーダ・GPS などがあり、ロボットが周囲の環境や他の物体の位置などの情報を取得するために用いられる。一方、内界センサは、エンコーダ・ジャイロセンサ・電流センサ・トルクセンサ・力覚センサ・触覚センサなどがあり、ロボットの自身の情報（位置・速度・姿勢など）を取得するために用いられる。これらのセンサ情報と地図情報を照らし合わせて自己位置を推定する。また、角速度と加速度を検出するためのセンサ群を一つにまとめたセンサを IMU (Inertial Measurement Unit) と呼ぶ。

外界センサを用いた自己位置推定の代表的なアルゴリズムに ICP (Iterative Closest Point) スキャンマッチング[11]や NDT (Normal Distributions Transform) スキャンマッチング[12]がある。

ICP スキャンマッチングは、3次元点群データに基づいて自己位置を決定する方法である。この方法は、事前に LiDAR を使って点群で構成される地図を作成し、その地図の点群の形状とロボットのセンサから取得された点群を比較する。そして、対象点間の誤差を最小にするように計算を繰り返し、点群地図とセンサによる地図の位置を合わせることでロボットの位置を推定する方法である。

NDT スキャンマッチングは、事前に取得された点群地図とセンサから取得された点群を確率分布によってマッチングさせる手法である。具体的には、探索空間内にある点群データを格子状に分割した後に、2つの格子状内の平均と分散を算出して探索空間を正規分布で近似する。そして、センサから取得された各点に対応する点を求め、マッチングの精度を評価する関数を用いて「事前に取得した点群」と「センサから取得した点群」の一致度を計算する。このマッチングの精度がある閾値以下になるまで、ニュートン法により、各点の座標を変換する計算を繰り返す。これにより、既存の点群地図とセンサから取得された点群の位置を合わせる。また、点群データをグリッドに合わせてダウンサンプリングすることで、計算処理の軽量化が可能である。

内界センサを用いた自己位置推定の代表的なアルゴリズムにデッドレコニング[13]がある。デッドレコニングとは、車輪の回転数や角度などの移動情報から自己位置を計算する手法である。この手法では、地面の凹凸や車輪のスリップ、回転数の変化によって誤差が蓄積されるため、定期的に自己位置の補正が必要となる。

外界センサから推定した自己位置は一般的に分散が大きいが、時間の経過に伴う誤差の増加は少ない。一方、内界センサから推定した自己位置の分散はとても小さいため、推定開始時にはとても正確な自己位置の推定が可能であるが、時間経過に伴って誤差が大きくなる。外界センサと内界センサの利点を組み合わせた手法として離散的な誤差のある観測から位置や速度を推定するカルマンフィルタ[14]や、システムモデルや観測モデル、プロポザル分布に基づき多数の状態の仮説を時間遷移させながら状態の事後分布を求めるパーティクルフィルタ[15]などのベイズフィルタを用いた自己位置推定の手法がある。

地図の生成と同時に自己位置を推定する手法として SLAM(Simultaneous Localization And Mapping)[16]がある。SLAM は、センサで自分の周囲の環境情報を取得し、自分がいる場所を中心にして、センサで得た情報を地図に書き加えた後、ロボットが移動し地図上の自己位置も移動させ、再びセンサで周囲の環境情報を取得することを繰り返し行い、スタート地点から移動後の位置付近へと生成される地図の範囲を拡大していく。災害現場や宇宙探査など事前の地図の作成が困難な未知の環境下でロボットが自律移動するのに有効な技術である。

2.3 経路計画

経路計画[17]は、スタート地点からゴール地点に行くまでの間に waypoint と呼ばれる経由点を複数設定して経路を設定する方法である。スタート地点からゴール地点までの走行可能エリアや障害物を考慮した最短経路や最適経路を探索する必要がある。経路上の障害物に衝突した経路であると判定した場合には、回避経路の生成や経路の再計画の判断を行

う。

障害物の衝突判定は、経路を示す複数の waypoint と障害物であると判定された障害物点群を入力として、障害物と衝突している waypoint を判定して出力する。その waypoint が複数ある場合には、現在位置から最も近くで衝突している waypoint を出力する。そして、その waypoint を通らない経路探索を行い、回避経路が生成する。

経路探索の代表的な手法に幅優先探索 (Breadth-First Search) [18], ダイクストラ探索 (Dijkstra Search)[19], A*探索(A* Search)[20], Hybrid State A*[21]がある。

幅優先探索は、コストなどを考えずに全方向を同一の優先度で探索する方法である。ダイクストラ探索は、通ってきた経路のコストのうち、コストの低いものを優先的に探索する方法である。A*探索は、スタートから現時点までのコストに加えて、現時点からゴールまでの予想コストを合わせて考慮し、コスト最小の経路を優先的に探索する方法である。Hybrid State A*は、A*探索のうち、ロボットの進行可能な前方、右前、左前の3方向に限定して探索を行う方法である。

2.4 静止障害物回避

安全な走行を実現するために自律移動ロボットにとって、静止障害物の回避は必須の技術である。そのための代表的な手法として、ポテンシャル法[22], 幾何学的手法[23], シミュレーション的手法[24]がある。

ポテンシャル法とは、障害物に対して斥力、目的地に引力が働くようにポテンシャルエネルギーを設定し、これを微分することでロボットが受ける力を計算し、そのベクトル方向に進むようにする方法である。幾何学的手法とは、LiDAR などにより環境を正確に計測し、ロボットの前方の幾何学的形状の領域を左右にある一定の角度ずつ探索し、その領域内に障害物がないかを検出する。そして、目的地方向に最も近い走行可能領域を決定し、障害物のない領域を手続き的に決定して障害物回避を行う方法である。シミュレーション的手法は、実環境と同様の環境をシミュレーション環境で再現し、障害物を回避して目的地へ向かう経路を手続き的、または機械学習を用いて探索し、その中で最適な障害物の回避経路を実機のロボットに適用する手法である。この手法ではシミュレーション環境で再現されていない静止障害物の回避ができない欠点がある。

移動障害物回避は、環境を正確に計測することに加えて、移動障害物に対して動的に回避経路を生成する必要がある。

2.5 センサ

センサには、アクティブセンサとパッシブセンサの2種類が存在する。アクティブセンサとは、センサ自身が電気信号や光信号などを発生し、物体や環境からの反応を検出して情報を取得するタイプのセンサである。例えば、距離センサやソナーセンサなどが該当する。エネルギー消費が大きいため、エネルギー効率やコスト面での課題がある。一方、パッシブセンサとは、外部からのエネルギーを受け取って情報を取得するタイプのセンサである。例えば、温度センサや照度センサなどがパッシブセンサに該当する。外部からのエネルギーに依存するため、環境条件や測定対象などによっては正確な情報を取得することができない場合がある。

2.5.1 距離センサ

距離センサには、超音波センサ[25]・LiDAR[26]・ミリ波レーダ[27]などがある。



図 2.3: 超音波センサ 文献[25]より抜粋



図 2.4: LiDAR 文献[26]より抜粋



図 2.5: ミリ波レーダ 文献[27]より抜粋

超音波センサは、送信部から超音波を出し、物体に当たった反射波を受信部で受け取るまでにかかる時間を計測して、距離を計算するアクティブセンサである。ガラスなどの透明物体にも使えるが、音を吸収するような布や泡などの検出ができない。

LiDAR は、Light Detection and Ranging の略称であり、レーザ光を対象物に対して照射し、反射して戻ってきたレーザ光の位相差を計測して、距離と方向を計算するアクティブセンサである。角度分解能が超音波センサより高いため、距離の精度が高く、検出できる距離が長い。天候や照明の影響を受けづらいため夜間の使用が可能であり、対象物までの距離や対象物の形状を計測可能である。2D LiDAR は水平方向の距離と方向を計測し、3D LiDAR は水平方向と垂直方向にレーザ光を照射し、距離と方向と物体の形を計測できる。

ミリ波レーダは、電波を送信部から出し、受信部で受け取るまでの時間を計測して、距離と方位を計測するアクティブセンサである。天候や光の影響を受けず、安価で長距離の計測が可能であるが、近距離の物体や小さい物体の検出がしづらいことや段ボールなどの電波に対する反射率の低い物体は検出しづらいことが欠点である。

2.5.2 ビジョンセンサ

ビジョンセンサには、RGB カメラ[28]や RGB-D カメラ[29]などがある。



図 2.6: RGB カメラ 文献[28]より抜粋



図 2.7: RGB-D カメラ 文献[29]より抜粋

RGB カメラは、レンズを通して撮像素子に入力された可視光域の Red・Green・Blue の光を電気信号に変換するパッシブセンサであり、3つの波長データで表現された2次元の画像を取得する。具体的には、単眼カメラやステレオカメラなどが該当する。

RGB-D カメラは、可視光域の Red・Green・Blue のカラー画像に加えて深度画像を取得できるセンサである。カラー画像の色情報と画像の各ピクセルに対応した距離を計測する。深度画像の取得には、ステレオ方式[30]、Time Of Flight(TOF)方式[31]、構造化照明方式[32]などがある。ステレオ方式とは、2つの異なる視点から同一物体を撮影することで取得した2つの画像を用いて、物体の深さ情報を推定する方法である。TOF 方式とは、物体または環境に光を発し、反射光を検出することでその物体または環境までの距離を測定する方法である。構造化照明方式とは、物体上に特殊なパターンを投影し、カメラを使って投影されたパターンと物体上の反射パターンを同時に撮影することで、物体の深さ情報を取得する方法である。

2.5.3 本研究に関連するセンサ

本研究における自律移動ロボットは、主に2種類のセンサを採用する。1つ目は、距離センサである2D LiDARを用いる。これにより周囲360度を計測することができ、ロボットに近づいてくる障害物を認識することができる。2つ目は、ビジョンセンサであるRGBカメラを用いる。これによりロボットの進行方向に存在する障害物を2次元画像として認識することができる。そのため、障害物の大きさや種類を把握することができる。

先行研究では、距離センサだけを使用するシステムよりも、距離センサとステレオカメラを組み合わせたシステムの方が障害物認識の精度が高いことが示されている[33]。

本研究では、強化学習で獲得したモデルを用いて、事前に取得した地図に設定された目的地への経路追従と地図にない障害物の回避を同時に行う。

第3章 シミュレーション環境

本章では、移動障害物回避の学習や走行テストに用いるシミュレーション環境について説明する。3.1 節では、シミュレーション環境の基盤となるシステムについて述べる。続く 3.2 節では、シミュレーション環境で用いる 3D モデルについて述べる。続く 3.3 節では、移動障害物の挙動について述べる。

3.1 シミュレーション環境の基盤システム

シミュレーション環境の基盤として、物理エンジンが搭載されたソフトウェアである Unity[34]を用いる。Unity とは、米国の Unity Technologies 社が提供するゲームエンジンであり、PC だけでなくモバイル機器やウェブブラウザ、家庭用ゲーム機などクロスプラットフォーム開発に対応し、XR(Extended Reality)開発にも用いられる。また、建築現場や教育現場で VR(Virtual Reality)や AR(Augmented Reality)による活用法が増加している。

Unity では、時間のかかるレンダリング処理をパイプライン処理によって高速化するレンダリングパイプライン機能を持ち、built-in Rendering Pipeline と URP (Universal Render Pipeline) と HDRP (High Definition Render Pipeline)の 3 種類が提供されている。URP, HDRP によって高品質な映像を描画することができる。

Unity は、影や光の反射表現などの描画性能に優れていることに加えて、位置の詳細な調整や環境の切り替えが可能である。さらに、Unity は、大規模な 3D モデルのインポートが容易に行え、高速に読み込むことができる。また、3D モデルに対してプログラムの適用ができるため、設置物に対する移動の制御ができ、タイムスケールを変更させたシミュレーションが可能である。具体的には、物理エンジンによって質量、重力、速度、摩擦、衝突などの力学的な法則のシミュレーションが可能である。

3.2 シミュレーション環境で用いる 3D モ

デル

Unity で使用するアセットは、図 3.1 に示すように Unity Asset Store[35]で入手することができる。Unity Asset Store には、3D モデルの他に、物体の質感を表現するために使われる画像であるテクスチャ、色・光沢・反射具合などを設定できるマテリアル、大気・炎・水の流れなどを表現するためのパーティクルを物理法則に従って動かすことで、自然現象のシミュレーションを行うパーティクルシステム、音楽、効果音、プロジェクトなどの様々なアセットがある。

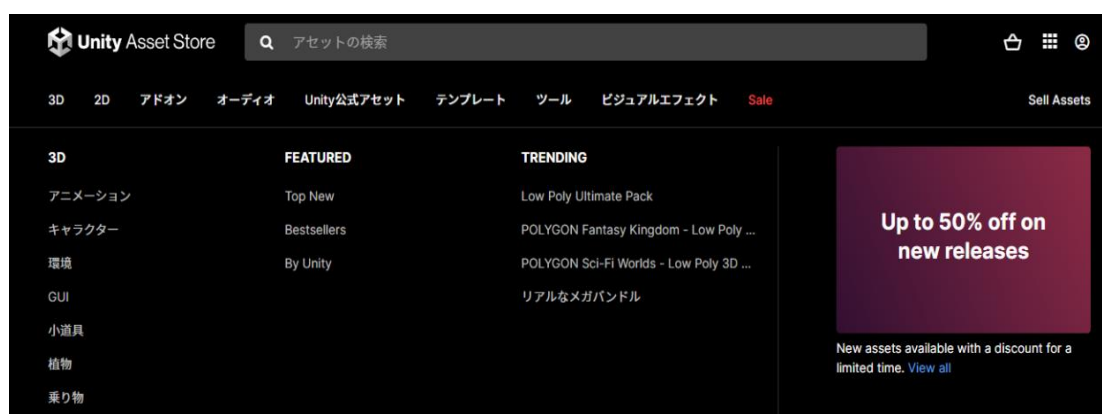


図 3.1: Unity Asset Store の検索欄 文献[35]より抜粋

3.2.1 環境の 3D モデル

本研究では、実験環境として仮想的なショッピングモールを想定している。Unity Asset Store から Unity 用の都市アセット Shopping Mall HQ[36]をダウンロードして使用する。特に、この中の広場で実験を行う。図 3.2 に示すように、広場には静止障害物である植林、四方を囲む壁には店の外観が描画されている。図 3.3 の上部に示す赤い色の長方形のオブジェクトは、自律移動ロボットが目指す目的地である。図 3.3 の下部に示す青いオブジェクトは、自律移動ロボットである。



図 3.2: Shopping Mall HQ



図 3.3: Shopping Mall HQ 俯瞰図

3.2.2 移動障害物の 3D モデル

移動障害物は人間を想定しており，Unity Asset Store から人間の 3D モデルのサンプルである StarterAssets thirdPerson[37]をダウンロードして使用する．図 3.4 に人間の 3D モデルの例を示す．



図 3.4: StarterAssets thirdPerson

3.2.3 自律移動ロボットの 3D モデル

自律移動ロボットの 3D モデルは、シミュレーション環境の計算軽量化のために単純な立方体で構成されている。また、同様に計算軽量化のために、距離センサ RaycastSensor[38]を用いて 2D LiDAR の機能を実装する。また、自律移動ロボットの前方の障害物の認識のためにカメラ[39]を設置している。RaycastSensor は、図 3.5 に示すように周囲 360 度に 36 本の白い Ray を飛ばし、障害物に Ray が接触すると緑色の Ray となり、障害物までの距離を取得する。カメラは、図 3.6 に示すように幅 240px、高さ 135px のグレースケール化した画像を取得する。自律移動ロボットは前進と停止ができ、方向を変えるための右旋回と左旋回ができる。

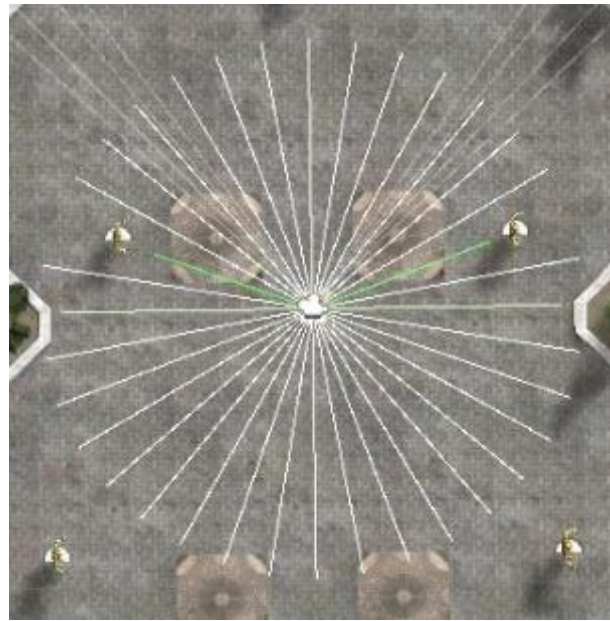


図 3.5: RaycastSensor



図 3.6: カメラ画像

3.3 移動障害物の挙動

移動障害物は人間を想定し，自律移動ロボットに遭遇した際に，人間の視野情報に基づいた回避行動をするアルゴリズムの設計を行った．移動速度は自律移動ロボットの走行速度を超えない範囲でランダムに決定される．

3.3.1 移動障害物の配置

実験環境内に 11 体の移動障害物を図 3.7 の青枠内に配置した．この位置を初期位置として，移動障害物はそれぞれの目的地を目指す．目的地は図 3.7 の赤枠内に配置した．目指す目的地はランダムで決定し，目的地に到達後は再びランダムで別の目的地へ向かう．

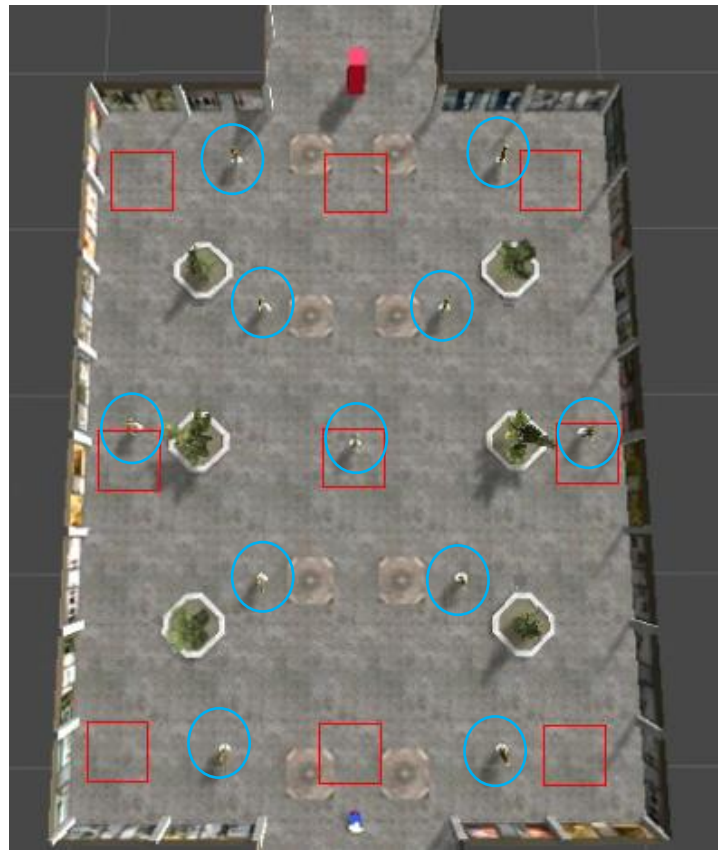


図 3.7: 目的地の設定

3.3.2 Navmesh と NavmeshAgent

Navmesh[40]とは，コンピュータグラフィックスにおいて，3D 地形上のキャラクターの移動可能なエリアを表すメッシュを指す．これは，動かすキャラクターの中心点が近づく壁までの距離，接触判定に用いるキャラクターの高さ，キャラクターの登れる斜面の角度，キャラクターが踏み越えられる障害物の高さの設定を行うことで，キャラクターが歩行可能な領域を定義することができる．図 3.8 の青い領域のように，領域の表面を凸ポリゴンで表現し，凸ポリゴン内の任意の 2 点間に障害物が存在しない領域を歩行可能領域として記録さ

れる．このメッシュによって，ロボットは環境上の地図を事前取得した状態と同様の状態であるため，歩行可能な地形を進み，静止障害物を回避しながら目的地に向かうことができる．

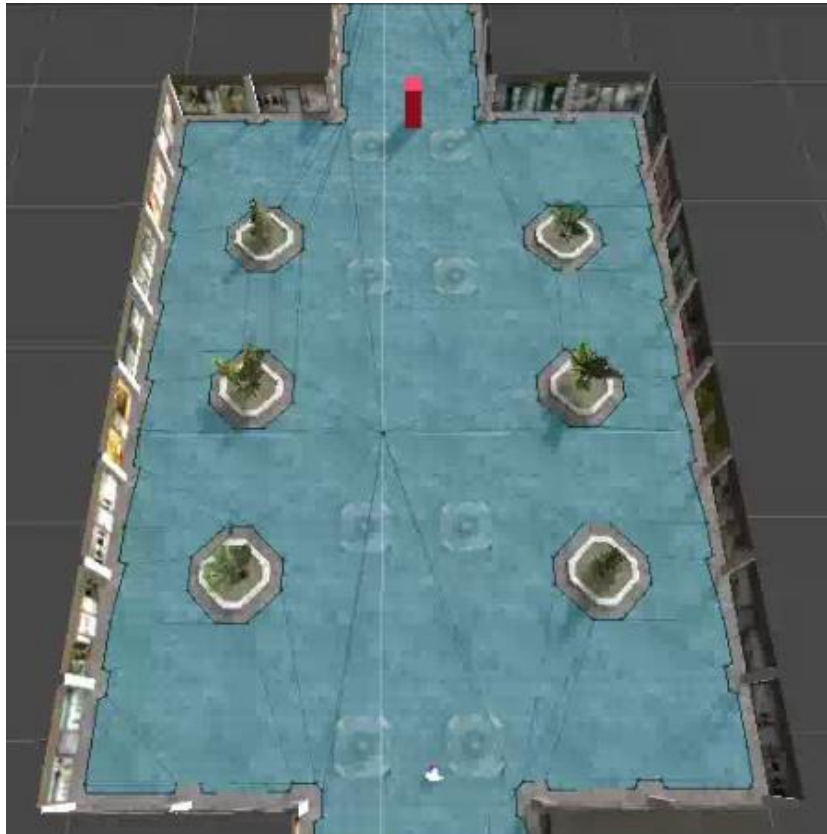


図 3.8: Navmesh の設定

NavmeshAgent[41]とは，キャラクタが地形を移動するためのコンポーネントである．これは，指定した出発地点から目的地までを結ぶ経路生成を行う機能を備えている．経路探索には Navmesh の領域内で A*探索アルゴリズム[42]が使用され，この経路を追跡することで出発地点から目的地までを結ぶ経路が生成される．出発地点から目的地までを直線で移動できるような複数の経由点を取得し，前後の経由点を線分で結び，経路が表示される．また，目的地の設定や，キャラクタの移動速度・角速度・加速度の設定を行うことができる．

本実験では，Shopping Mall HQ に Navmesh を設定して、実験環境で自律移動ロボットと移動障害物の歩行可能領域を設定し，自律移動ロボットと移動障害物に NavmeshAgent を設定して，それぞれが目的地へ最短経路で移動するようにした．

3.3.3 視野に基づく挙動

移動障害物は NavmeshAgent コンポーネントによって目的地へ最短経路で移動するが、視界に自律移動ロボットを発見すると確率的に回避行動をとるように設定した。

移動障害物の視野角を 180 度に設定[43]し、視野に自律移動ロボットが入った時に「目的地に向かう行動の継続」「停止」「自律移動ロボットを回避」のいずれかの行動を行う。

「停止」「自律移動ロボットを回避」の場合は NavmeshAgent による目的地へ向かう行動を一時的に中断する。「自律移動ロボットを回避」は移動障害物の正面を基準にして視野 180 度の領域を半分ずつ左右に分け、自律移動ロボットが左右のどちらにあるかを判断し、自律移動ロボットが存在しない側の領域へ移動する。移動する際の旋回角度は、ランダムに決定する。

また、図 3.9 の緑色の線のように移動障害物の視線情報として正面に Ray を飛ばし、他の移動障害物に接触した場合に 2 秒間停止して、他の障害物に近づきすぎないように距離をとる。図 3.9 の青と赤の Ray は移動障害物の移動方向を示す視線情報であり、自律移動ロボットの移動障害物回避モデルに活用する。詳細は 4 章で述べる。

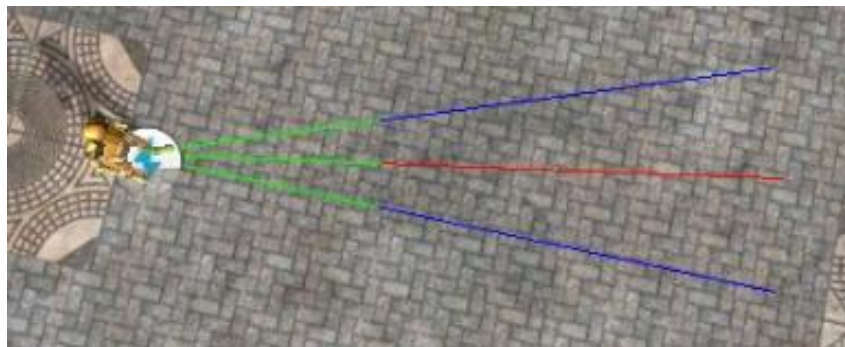


図 3.9: 移動障害物の視線

第4章 移動障害物回避のための強化学習モデル

本章では、移動障害物回避のための強化学習モデルについて説明する。4.1 節では、機械学習の一種である強化学習について述べ、4.2 節では、シミュレーション実験で強化学習を行うためのフレームワークを説明し、4.3 節では、本研究で構築する移動障害物回避モデルの構造について述べる。

4.1 強化学習

4.1.1 機械学習の分類

機械学習の手法には、いくつかの種類があり、代表的なものに「教師あり学習」「教師なし学習」「強化学習」がある[44]。

「教師あり学習」とは、予測すべき正解データを用いて学習させることで、入力されたデータに対する予測を出力する推論モデルを作成する学習方法であり、クラスと呼ばれるデータの種別を予測する分類タスクと、類似度などの連続値を予測する回帰タスクに分類される。

「教師なし学習」とは、データの構造を学習する手法であり、正解ラベルのないデータのみで学習することによって、入力データに含まれた潜在的なパターンを見つけ出し、似た性質を持つデータをまとめ、クラス分けをする手法である。

「強化学習」とは、正解データなしで、ある環境で、どのような行動をすれば、その環境に適合する行動をとれるのかを試行錯誤によって学習する手法である。

4.1.2 強化学習とは

強化学習では行動する主体を「エージェント」、エージェントがいる世界を「環境」と

呼ぶ[44]。エージェントが環境に行う働きかけを「行動」、その行動の適切さの指標として「報酬」を設定する。強化学習は現在の状態に応じて、次の行動を決定する。行動の仕方は、ある状態である行動を行う確率によって決定される。この行動の仕方を「方策」または「ポリシー」と呼ぶ。強化学習は、行動直後に得られる即時報酬だけでなく、後から遅れて得られる遅延報酬を含めた報酬和を最大化するポリシーを学習することが目的である。

強化学習モデルは、エージェントが環境を観察して状態を取得し、その状態に対してポリシーに基づいたエージェントの行動を決定し、その行動を実行する。学習プロセスでは、行動の結果に応じた報酬を受け取り、状態と行動と報酬の組み合わせである「経験」に応じて、報酬和を最大化するようにポリシーを更新するという一連の手順を繰り返す。強化学習の訓練 1 回分を 1 エピソードと呼び、1 エピソードあたりに獲得できる報酬を最大化するようにパラメータを調整する。学習環境の 1 フレームを 1 ステップと呼び、前章で説明したシミュレータでは、基本的に 0.02 秒ごとに 1 ステップが実行される。

4.1.3 強化学習アルゴリズムの分類

強化学習アルゴリズムは、状態遷移と報酬を予測する関数である環境モデルの使用の有無によって分類される。環境モデルを使用してポリシーや価値関数を更新するアルゴリズムを「モデルベース」、環境モデルを使用せず、観測した環境情報をポリシーや価値関数の更新に使用するアルゴリズムを「モデルフリー」という[44]。「モデルベース」の利点は、環境モデルを使用することで、囲碁や将棋のようにエージェントが行動可能な選択肢に基づく行動の結果と報酬を予測することができ、予測した結果に基づいて行動を決定することである。「モデルフリー」の利点は、環境モデルを使用しないため、実装及び調整が容易であることである。

「モデルフリー」はさらに経験の使用の仕方によって「オンポリシー」と「オフポリシー」に分類される[44]。

「オンポリシー」強化学習は、現在のポリシーで得られた経験のみを利用して、新しいポリシーの更新を行うため、学習に必要なステップ数が多いが、学習は安定し、時間をかければ優れたポリシーが得られる傾向がある。主なアルゴリズムとして、高い報酬が得られる行動を優先し、低い報酬しか得られない行動を避けるようにポリシーの更新を行う Vanilla Policy Gradient(VPG)[45]、ポリシーの更新をパラメータ空間の範囲内にすることで大幅な更新が発生しないように学習が安定するように VPG を改良した Trust Region Policy Optimization(TRPO)[46]、TRPO の計算量を削減するように完了した Proximal Policy Optimization(PPO)[47]などがある。

「オフポリシー」強化学習は、保存した過去の経験を利用して、新しいポリシーの更新

を行うため、学習に必要なステップ数は少ないが、行動が不安定になりやすい。主なアルゴリズムとして、行動価値観数である Q 関数とポリシーを同時に学習する Deep Deterministic Policy Gradient(DDPG)[48]、DDPG を学習が安定するように改良した Twin Delayed DDPG(TD3)[49]、ポリシーが局所解に陥るのを防ぎ、ポリシーの探索によって学習が継続されるようにエントロピーを導入した Soft Actor-Critic(SAC)[50]などがある。

他に、教師データとなる人間の一連の行動を模倣することで環境に適応する「模倣学習」がある。人間の行動と同じ行動をとることで「報酬」が得られる強化学習であり、正確に模倣するように学習するアルゴリズムである Behavioral Cloning(BC)[51]や、環境報酬の有無にかかわらず使用でき、似た行動に対し報酬を与える学習アルゴリズムである Generative Adversarial Imitation Learning(GAIL)[52]などがある。

4.2 ML-Agents

本研究のシミュレーションで採用した ML-Agents[53]は、モデルの「強化学習」「推論」「手動操作」を行うことができる。

「強化学習」は Unity の外部に存在している PythonAPI で Unity 内の Communicator[55]と Python Trainer[54]で通信を行い、強化学習を行う。強化学習の際は、Unity の Communicator を介して観察結果を Python API に送信し、Python Trainer で行動の決定や方策の更新を行う。強化学習アルゴリズムは Python で実装され、yaml ファイルに強化学習のハイパーパラメータの設定を記述する。

「推論」と「手動操作」は Unity のみで実行され、Unity 内部で学習モデルファイルの読み込みや手動制御による行動が決定される。

ML-Agents の観察には「Vector Observation[56]」「Visual Observation[56]」「Raycast Observation[56]」の3つの型がある。「Vector Observation」はスクリプトで取得する浮動小数の配列情報、「Visual Observation」はカメラやレンダリングテクスチャの画像情報、「Raycast Observation」は周囲に Ray を飛ばした際のオブジェクトとの接触判定と距離情報である。「Visual Observation」「Raycast Observation」はパラメータの設定によって過去の観察を学習に利用できる。

4.3 移動障害物回避モデルの学習

4.3.1 自律走行と移動障害物回避システム

シミュレータでは、自律移動ロボットに NavmeshAgent を設定して目的地への移動を行う。自律走行と移動障害物回避を同時に行うシステムを構築するため、NavmeshAgent の出力のうち、目的地への向きだけを、移動障害物回避モデルの入力に用いて学習を行う。NavmeshAgent による向きの出力はステップごとに計算されるため、移動障害物回避の行動によって経路から外れると、新たな最短経路の探索によって目指す経由点に変更され、状況に合わせた動的な経路変更が行われる。過去の経験を利用してポリシーの更新を行う「オフポリシー」強化学習は、過去の経験を利用するため、変化の大きい環境の学習に適している。本研究では、多数の移動障害物の移動によって変化が大きい環境に対応するため、前述の SAC アルゴリズムを採用する。

4.3.2 強化学習モデルの入出力

本研究では、強化学習モデルの入力に Visual Observation であるカメラのグレースケール化した画像（アスペクト比は 16:9）、Raycast Observation である RaycastSensor（周囲 360 度に 36 本の Ray）、Vector Observation である自律移動ロボットの速度と向き、自律走行を行うための NavmeshAgent による経由点への向き、移動障害物の視線情報を用いる。現実の人は、服によって色が多様であるため、色ではなく形から人を認識できるようにグレースケール化した画像を使用している。

移動障害物の視線情報とは、自律移動ロボットを見ている移動障害物の数を自律移動ロボットが検知するための情報であり、移動障害物が飛ばしている Ray が自律移動ロボットに接触した際に、自律移動ロボットは接触している Ray の本数とその方向情報を記録する。視線の方向は簡略化のために 30 度ずつの離散情報とする。図 4.1 に示すように、視線の中心である赤色の Ray と、中心視野と周辺視野の境である青色の Ray と飛ばし、赤色の Ray が自律移動ロボットに接触した場合は +1 とし、ロボットが人の中心視野から周辺視野へ移動するように学習させるため、青色の Ray が接触した場合は、赤色の Ray の半分である +0.5 とする。緑色の Ray は、移動障害物同士の過度な接近がないようにする領域であり、詳細は第 3 章で述べている。

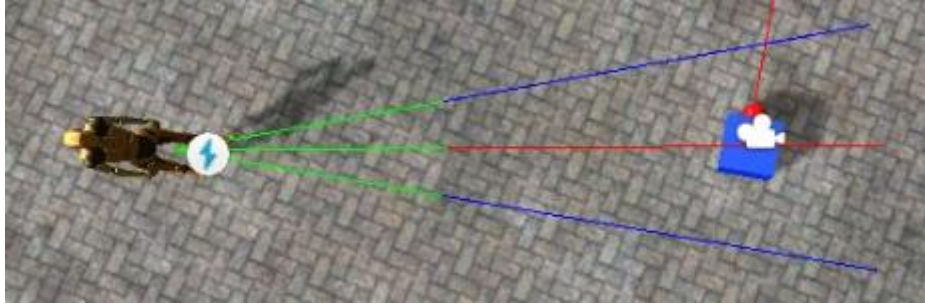


図 4.1: 移動障害物の視線

カメラ画像は、隠れ層 256 の 2 層の CNN を通した後に他の入力と結合される。本実験で用いるのは縦 135px, 横 240px のグレースケール画像であるため、 $135 \times 240 \times 1$ 次元の入力となる。

強化学習モデルの出力は、自律移動ロボットの走行の制御となる速度と走行方向の情報である。速度は「前進」「前進の半分」「停止」のいずれかである。速度が連続値でなく 3 つの速度である理由は、離散値である方がモデルの学習が安定しやすく、0.02 秒ごとに速度が決定されるので、障害物回避の行動をするのに離散値でも十分可能であると考えられるからである。また、向きは「変更なし」「右旋回」「左旋回」のいずれかである。

移動障害物の視線情報の有無によるモデルの精度比較を行うため、入力として「視線情報ありの現在のステップの観測」「視線情報なしの現在のステップの観測」を用いるモデルの学習を行う。また、過去の観測情報の使用によるモデルの精度検証を行うため、入力情報として「視線情報ありの現在と 1 つ前のステップの観測」を用いるモデルを構築する。

4.3.3 強化学習モデルの詳細

強化学習モデルを Netron[57]で示す。Netron とは、ブラウザを含むクロスプラットフォームで動作する機械学習モデルのビジュアルライズツールである。

図 4.2 は、入力情報として「視線情報ありの現在のステップのみの観測」を使用する強化学習モデルの詳細である。

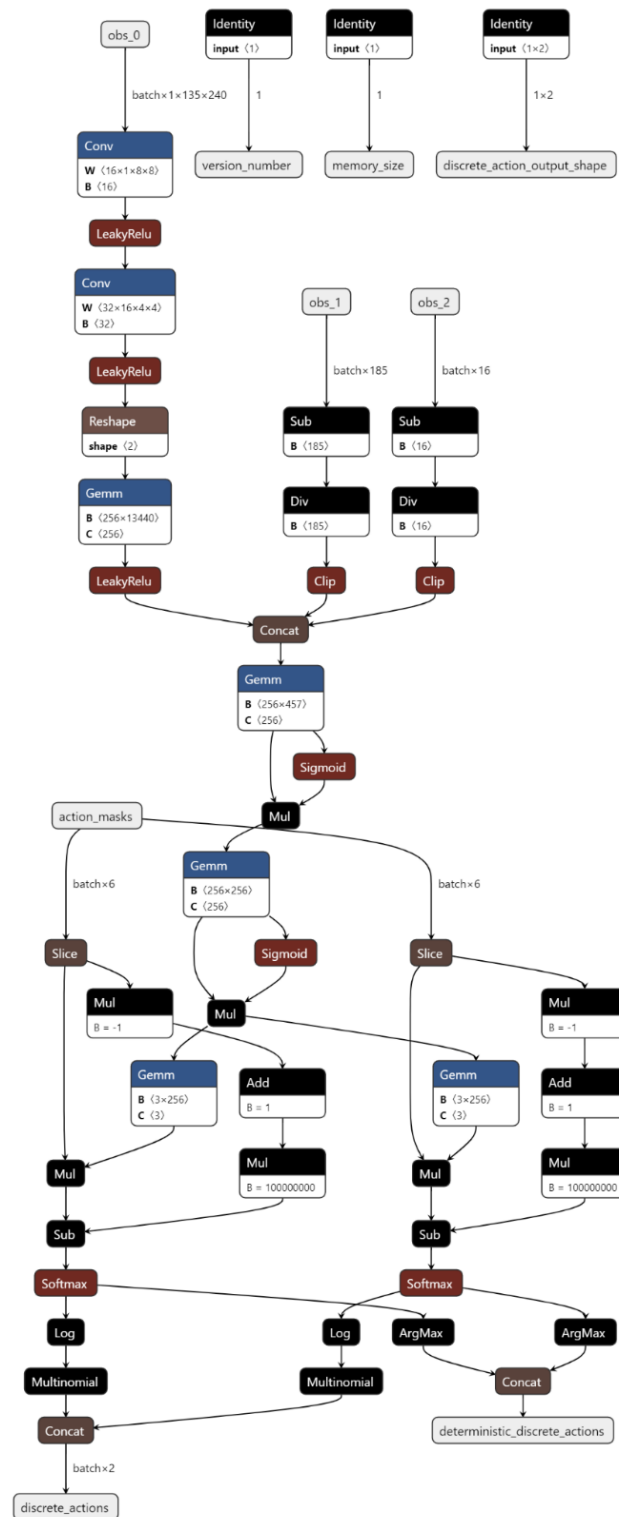


図 4.2: 強化学習モデルの詳細

ML-Agents の学習のハイパーパラメータの設定は yaml ファイルに記述する[58]. 本実験

に用いた yaml ファイルの設定は表 4.1 に示す通りであり、SAC の設定や学習率の設定、画像入力に対するネットワークの設定や報酬の設定を行う。

表 4.1: yaml ファイルの設定

trainer_type:	sac
hyperparameters:	
learning_rate:	0.0004
learning_rate_schedule:	constant
batch_size:	500
buffer_size:	70000
buffer_init_steps:	3500
tau:	0.005
steps_per_update:	12.0
save_replay_buffer:	False
init_entcoef:	0.1
reward_signal_steps_per_update:	12.0
network_settings:	
normalize:	True
hidden_units:	256
num_layers:	2
vis_encode_type:	simple
memory:	None
goal_conditioning_type:	hyper
deterministic:	False
reward_signals:	
extrinsic:	
gamma:	0.99
strength:	1.0
network_settings:	
normalize:	False
hidden_units:	128
num_layers:	2
vis_encode_type:	simple
memory:	None
goal_conditions_type:	hyper
deterministic:	False
init_path:	None
keep_checkpoints:	100
checkpoint_interval:	100000
max_steps:	13000000
time_horizon:	256
summary_freq:	10000
threaded:	True
self_play:	None
behavioral_cloning:	None

4.3.4 報酬の設定

自律移動ロボットが「目的地に到達」もしくは「移動障害物に衝突」した場合にエピソードが終了する。自律移動ロボットに移動障害物の回避行動を学習させるため、図 4.3 に示すように、移動障害物に前方に長方形の領域(青色のオブジェクト)、後方に半円の領域(赤色のオブジェクト)を設置し、これらの領域に自律移動ロボットが接触することで報酬が得られる。この報酬によって自律移動ロボットが長時間停止することを防ぎ、移動障害物回避の行動獲得を促進する。



図 4.3: 移動障害物の報酬領域の設定

正の報酬が与えられるのは、「目的地に到達した瞬間」「移動障害物の後方の半円領域に接触した瞬間」である。また、負の報酬が与えられるのは、「移動障害物に接触した瞬間」「移動障害物の前方の長方形領域に接触した瞬間」「ステップ毎」である。移動障害物の前方に長方形領域を設置した理由は、自律移動ロボットが移動障害物の前方を通過して回避行動をとった場合に、移動障害物である人間に危機感や不快な感情を与えてしまうためであり、これを負の報酬の領域とした。また、ステップごとに負の報酬を与える理由は、自律移動ロボットが遠回りして目的地へ到達するよりも最短経路で目的地へ到達する場合の方が与えられる報酬和が大きくなることで、自律移動ロボットが最短経路で目的地を目指す行動を学習させるためである。

報酬の値の設定を表 4.2 に示す。

表 4.2: 報酬の値設定

報酬が与えられる瞬間	報酬の値
目的地に到達	+14.0
移動障害物後方の半円領域に接触	+0.5
移動障害物に接触	-4.0
移動障害物前方の長方形領域に接触	-0.2
ステップ毎	-0.005

第5章 実験方法及び結果と考察

本章では、移動障害物回避モデル構築のための実験とその結果及び考察について述べる。5.1節では、実験方法を、5.2節では、その結果を説明する。続く5.3節では、他の環境の構成、学習環境及び他の環境での検証結果と性能検証について述べる。最後に5.4節で考察する。

5.1 実験方法

第3章で説明したシミュレーション環境で移動障害物回避のモデルを学習させ、その後、学習済みモデルを用いて実験環境及び検証環境で1000回の試行を行う。そして、移動障害物に衝突せずに目的地に到達した回数を求め、性能評価を行う。

実験環境における自律移動ロボットの初期位置及び目的地は、図5.1の赤と青の四角からランダムに選択される。本来は赤と青の四角は透明であるが、ここでは説明のため赤と青の四角で場所を示す。自律移動ロボットの初期位置と目的地の位置が近距離にならないように異なる色の位置からランダムに選択される。

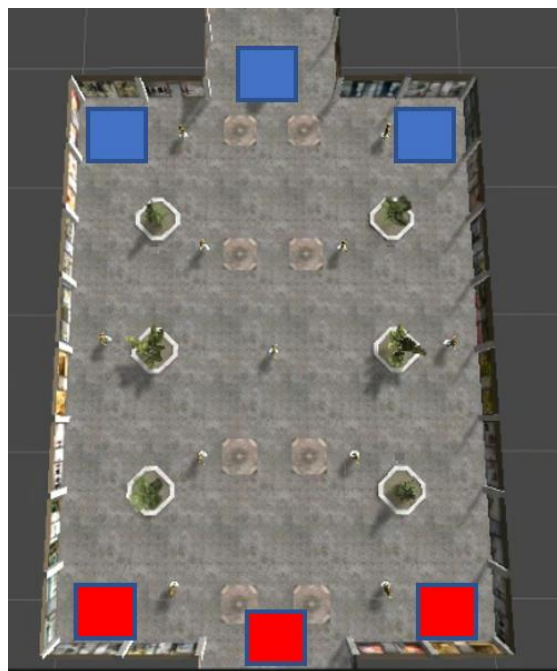


図 5.1: 実験におけるロボットの初期位置と目的地

学習の各エピソード開始時に環境の初期化が行われ、自律移動ロボットの初期位置と目的地、移動障害物の走行速度と目的地をランダムに決定する。

学習ステップ数は 300 万回に設定し、1 エピソードは最大で 6000 ステップとする。入力となる観測を「視線情報なしの現在のステップの観測」「視線情報ありの現在のステップの観測」「視線情報ありの現在と 1 つ前のステップの観測」の 3 種類のモデルを設計し、強化学習を行う。その後、10000 ステップで平均化した 1 エピソード当たりの累積報酬和とステップ数、方策エントロピーを計算して強化学習の状況を観察する。

5.2 実験結果

強化学習の入力が「視線情報なしの現在のステップの観測」「視線情報ありの現在のステップの観測」「視線情報ありの現在と 1 つ前のステップの観測」のそれぞれのモデルを構築し、学習履歴を TensorBoard[59]を用いて可視化する。TensorBoard とは学習モデルや学習履歴の可視化を行うツールである。

5.2.1 視線情報なしの現在のステップの観測を用いたモデル

視線情報なしの現在のステップの観測を用いたモデルの学習は次のようになった。図 5.2 に、エピソード当たりの累積エピソード報酬の推移を示す。図中の濃い線が 10000 ステップごとの平均値を示し、薄い線は 10000 ステップごとの最大値や最小値を示す。これによると、50 万ステップまでは平均累積エピソード報酬が増加した。その後、300 万ステップまで振幅の小さい増減が続き、安定した学習が続いた。このグラフから初期状態から報酬が約 6 を得られるような行動を学習できたことが分かる。

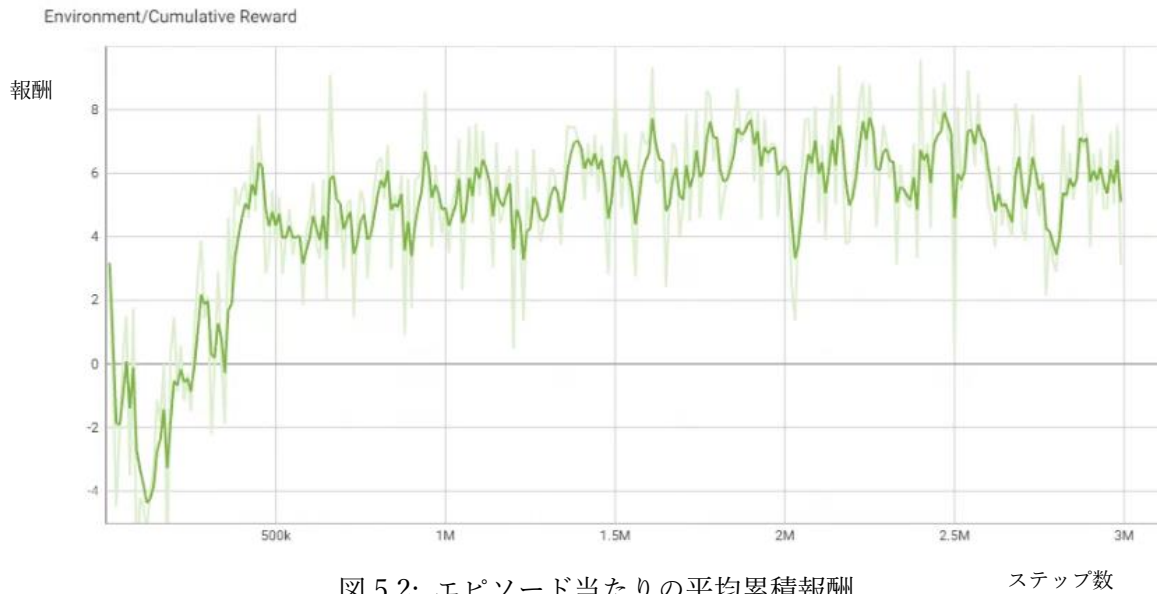


図 5.2: エピソード当たりの平均累積報酬

次に、エピソード当たりの平均ステップ数の変化を図 5.3 に示す。これによると、20 万ステップまでエピソード当たりの平均ステップ数が増加し、その後、50 万ステップまで減少した。その後、300 万ステップまで 1 エピソード当たり 400~500 ステップ数で安定していた。学習によって初期状態に比べて少ないステップ数で目的地に到達できるようになり、最終的には 450~500 ステップで目的地へ到達している。



図 5.3: エピソード当たりの平均ステップ数

次に、エピソード当たりの方策エントロピーの変化を図 5.4 に示す。50 万ステップまでエントロピーが大きく減少した。その後 300 万ステップまで安定し、行動の決定に関わる方策分布が変化せず安定していた。このグラフから初期状態に比べて学習が進むと、方策のエントロピーが小さくなり、行動のランダム性が減少することが分かる。

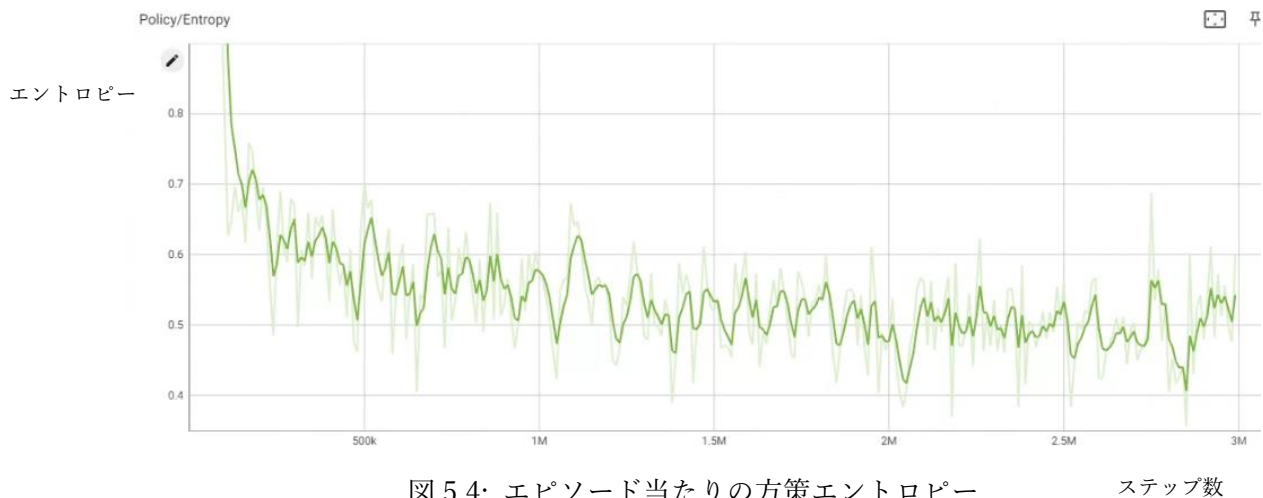


図 5.4: エピソード当りの方策エントロピー

ステップ数

5.2.2 視線情報ありの現在のステップの観測を用いたモデル

図 5.5 はエピソード当たりの平均累積報酬を示すグラフである。200 万ステップまで平均累積エピソード報酬の増加が確認でき、その後 300 万ステップまで振れ幅の小さい増減が続く、安定した学習が続いた。このグラフから初期状態から報酬が約 6 を得られるような行動を学習できたことが分かる。



図 5.5: エピソード当たりの平均累積報酬

ステップ数

図 5.6 はエピソード当たりの平均ステップ数を示すグラフである。20 万ステップまでエピソード当たりの平均ステップ数が増加し、その後、60 万ステップまでエピソード当たりの平均ステップ数が減少した。その後、300 万ステップまで 1 エピソード当たり 400~500 ステップ数で安定していた。このグラフから初期状態に比べて学習が進むと、少ないステップ数で目的地に到達できるようになり、最終的には約 450 ステップで目的地へ到達している。

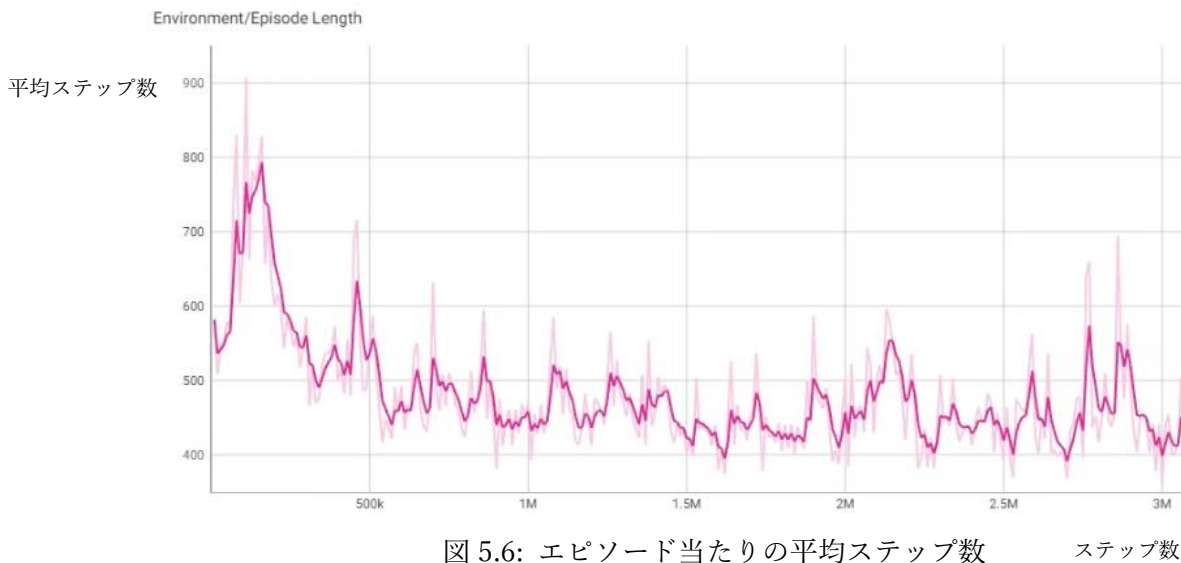


図 5.6: エピソード当たりの平均ステップ数

図 5.7 はエピソード当たりの方策のエントロピーを示すグラフである。20 万ステップまでエントロピーが大きく減少した。その後 250 万ステップまでエントロピーの減少が続き、その後は増減が続いた。このグラフから初期状態に比べて学習が進むと、方策のエントロピーが小さくなり、行動のランダム性が減少することが分かる。

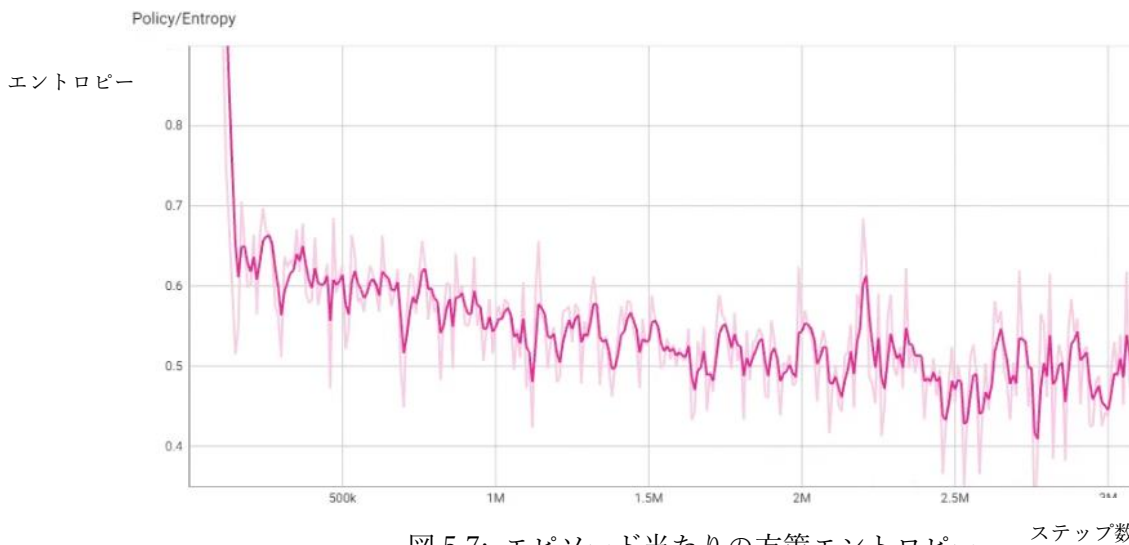


図 5.7: エピソード当たりの方策エントロピー

5.2.3 視線情報ありの現在と 1 つ前のステップの観測を用いたモデル

図 5.8 は平均累積エピソード報酬を示すグラフである。120 万ステップまで平均累積エピソード報酬の増加が確認でき、その後、振れ幅の小さい増減が 300 万ステップまで確認できた。このグラフから初期状態から報酬が約 6 を得られるような行動を学習できたことが分かる。



図 5.8: エピソード当たりの平均累積報酬

図 5.9 はエピソード当たりの平均ステップ数を示すグラフである。30 万ステップまでエピソード当たりの平均ステップ数が増加し、その後、60 万ステップまでエピソード当たりの平均ステップ数が減少した。その後 260 万ステップまで 1 エピソード当たり 400～500 ステップ数の範囲に収まっていた。その後増加し、再び減少している。少ないステップ数で目的地に到達できるようになり、最終的には約 450 ステップで目的地へ到達している。



図 5.9: エピソード当たりの平均ステップ数

図 5.10 はエピソード当たりの方策エントロピーを示すグラフである。20 万ステップまでエントロピーが大きく減少した。その後 200 万ステップまでエントロピーが小さく減少している。このグラフから初期状態に比べて学習が進むと、方策のエントロピーが小さくなり、行動のランダム性が減少することが分かる。

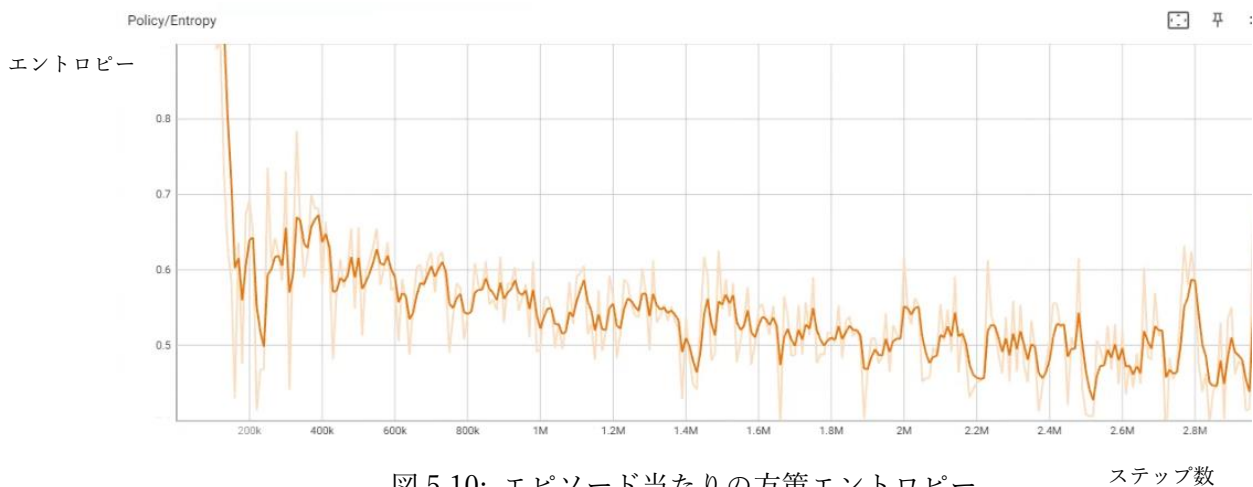


図 5.10: エピソード当たりの方策エントロピー

強化学習の入力が「視線情報なしの現在のステップの観測」「視線情報ありの現在のステップの観測」「視線情報ありの現在と 1 つ前のステップの観測」となるいずれのモデルも学習の初期は大きく累積報酬和が増加し、その後、累積報酬和の安定が続いた。1 エピソード当たりのステップ数は 400~500 の範囲に収束し、方策エントロピーは学習の初期で大きく減少した後、小さな減少が続いた。

5.3 性能評価

5.3.1 学習済みモデルを用いた学習環境での検証

自律移動ロボットに学習済みのモデルを適用して、学習で用いたシミュレーション環境で 1000 回の試行を行った。試行の一例を図 5.11 に示す。図中の赤色の線は経路を表し、移動障害物の視線情報を赤・青・緑の線、ロボットは青色、目的地は赤色の四角形で表している。

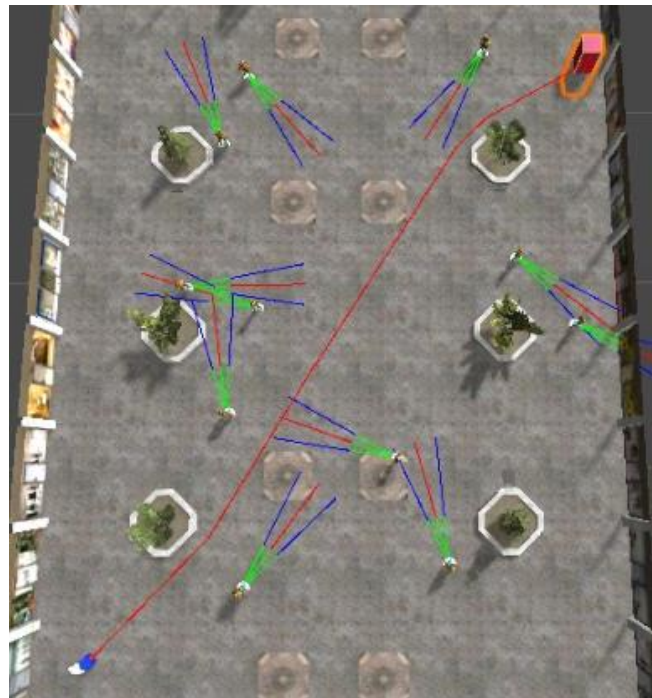


図 5.11 学習用シミュレーション環境での試行の例

また、比較対象として、強化学習モデルを用いないルールベースによる制御プログラムの試行も同時に行った。制御ルールには、NavmeshAgent による経路追従するパターン（パターン A）、NavmeshAgent による経路追従を行い、進行方向に移動障害物を検知すると静止するパターン（パターン B）、NavmeshAgent による経路追従を行い、前方に移動障害物を検知すると移動障害物のいない方向に旋回するパターン（パターン C）の 3 種類がある。パターン B では、図 5.12 の左図のように Ray（緑色）を出し、障害物に接触している間は静止する。パターン C では、図 5.12 の右図のように Ray を出し、赤色の Ray に移動障害物が

接触している間は右方向に旋回し、青色の Ray に移動障害物が接触している間は左方向に旋回し、図 5.13 のように、同時に赤色と青色の Ray の両方に移動障害物が接触している間は静止する。

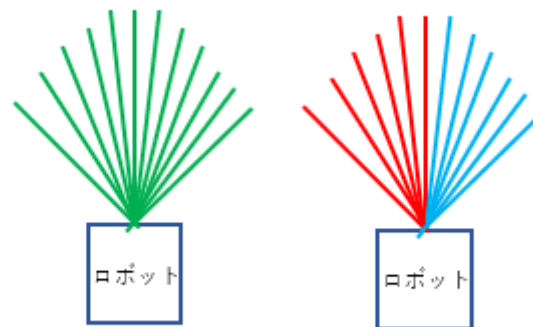


図 5.12 Ray を用いた移動障害物の検知の仕組み
(左図 パターン B, 右図 パターン C)

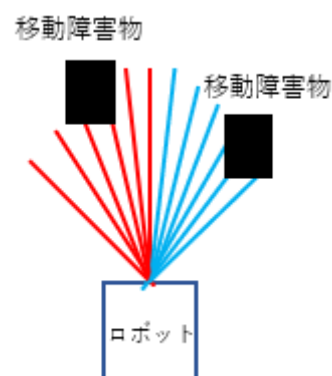


図 5.13 Ray を用いた複数移動障害物の検知例 (パターン C)

表 5.1 に、ルールベース制御の 3 種類のモデルを用いて 1000 回試行した結果を示す。結果はそれぞれの、目的地に到達できた成功回数、移動障害物に衝突して失敗した回数、目的地到達の精度からなる。次に、学習済みモデルによる制御の 3 種類のモデルの試行結果を表 5.2 に示す。同様に、目的地に到達できた成功回数、移動障害物に衝突して失敗した回数、目的地到達の精度である。また、表 5.3 に、ルールベース制御及び学習済みモデル制御の失敗回数のうち、自律移動ロボットが静止し移動障害物から接触してきた回数、自律移動ロボットが移動中に接触した回数を示す。

表 5.1: ルールベース制御の 1000 回の試行結果

モデル	成功回数	失敗回数	精度
経路追従	481 回	519 回	48.1%
経路追従+静止	765 回	235 回	76.5%
経路追従+静止 or 左右旋回	794 回	206 回	79.4%

表 5.2: 学習済みモデルを使用した 1000 回の試行結果

モデル	成功回数	失敗回数	精度
視線情報なしの現在の観測	860 回	140 回	86.0%
視線情報ありの現在の観測	907 回	93 回	90.7%
視線情報ありの現在と過去の観測	905 回	95 回	90.5%

表 5.3: 1000 回の試行の失敗の内訳

モデル	失敗回数	移動障害物から接触	ロボットから接触
経路追従+静止 or 左右旋回	206 回	116 回	90 回
視線情報なしの現在の観測	140 回	80 回	60 回
視線情報ありの現在の観測	93 回	71 回	22 回
視線情報ありの現在と過去の観測	95 回	60 回	35 回

これらの結果から、ルールベース制御に比べて、強化学習モデルを用いた移動障害物回避の精度が高いことがわかった。さらに、強化学習モデルのうち、移動障害物の視線情報を使用するモデルの方が、移動障害物回避の精度が向上した。また、「視線情報ありの現在の観測」と「視線情報ありの現在と過去の観測」の成功回数が近い値であることから、過去の観測情報を使用の有無で物体検知の精度の違いはあまりないと言える。さらに、回避失敗の原因を分析したところ、ロボットが静止中に移動障害物の方から接触する回数が失敗回数の過半数を超えていることがわかった。このことから、人が静止中のロボットに気づくような工夫をすることや、静止中のロボットに接触しても安全な工夫をすること、ロボットが静止よりも回避の行動を優先するように学習する必要があると考えられる。

5.3.2 学習済みモデルを用いた他の環境での検証

学習用のシミュレーション環境とは別に, 検証用環境を用意した. それを図 5.14 に示す. 自律移動ロボットの初期位置と目的地は学習用の環境と同様に決定される.

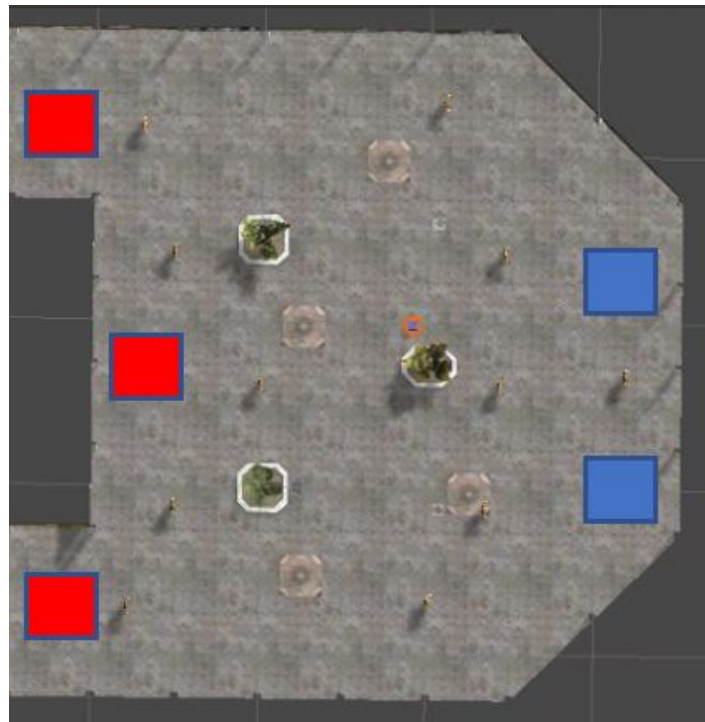


図 5.14: 検証環境におけるロボットの初期位置と目的地

強化学習モデルの汎化性能を調べるため, 検証環境で学習済みのモデルで制御する自律移動ロボットを設定してシミュレーションを行い, 1000 回の試行を行った.

図 5.15 に試行の様子を示す. この図では, 試行中の経路を赤色の線, 移動障害物の視線情報を赤, 青, 緑の線で表現しており, 赤色の線に沿って自律移動ロボットが移動する.



図 5.15: 検証環境での試行中の様子

次に、実験環境で最も精度の良かったルールベース制御であるパターン C、つまり NavmeshAgent による経路追従を行い、前方に移動障害物を検知すると移動障害物のいない方向に旋回する制御モデルで、目的地に到達できた成功回数、移動障害物に衝突して失敗した回数、および目的地到達の精度を調査した。この結果を表 5.4 に示す。また、表 5.5 に、学習済みの 3 種類の制御モデルを用いた試行の結果を示す。さらに、表 5.6 に、ルールベース制御及び学習済みモデル制御の失敗回数のうち、自律移動ロボットが静止した状態で移動障害物が接触してきた回数、自律移動ロボットが移動中に障害物に接触した回数を示す。

表 5.4: ルールベース制御の 1000 回の試行結果

モデル	成功回数	失敗回数	精度
経路追従+静止 or 左右旋回	783 回	217 回	78.3%

表 5.5: 学習済みモデルを使用した 1000 回の試行結果

モデル	成功回数	失敗回数	精度
視線情報なしの現在の観測	873 回	127 回	87.3%
視線情報ありの現在の観測	906 回	94 回	90.6%
視線情報ありの現在と過去の観測	902 回	98 回	90.2%

表 5.6: 1000 回の試行の失敗の内訳

モデル	失敗回数	移動障害物から接触	ロボットから接触
経路追従+静止 or 左右旋回	217 回	125 回	92 回
視線情報なしの現在の観測	127 回	54 回	73 回
視線情報ありの現在の観測	94 回	42 回	52 回
視線情報ありの現在と過去の観測	98 回	47 回	51 回

移動障害物の視線情報ありのモデルの方が、移動障害物回避の精度が向上したが、過去の観測情報を利用したモデルについては、精度の向上は確認できなかった。

また、いずれのモデルも検証環境において学習用のシミュレーション環境と同等の精度が確認された。

5.4 考察

本節では、シミュレーション実験において観察された、ルールベース制御及び強化学習モデルによる移動障害物回避制御で用いた自律移動ロボットが、移動障害物に衝突してしまった状況についての考察を行う。

経路追従のみのルールベース制御（パターン A）では、移動障害物を避ける動作や静止する動作をしないため、ロボットの経路上に移動障害物が存在すると衝突してしまった。経路追従を行い、前方に移動障害物を検知すると静止するルールベース制御（パターン B）では、静止しているロボットに移動障害物が衝突することが多かった。また、センサの検知範囲外である横や後ろからの移動障害物と衝突していた。前方に移動障害物を検知すると移動障害物のいない方向に旋回するルールベース制御（パターン C）では、同様に、静止しているロボットに移動障害物が衝突することが多かったが、加えて、図 5.16 に示すように、ロボットの回避方向と移動障害物の進行方向が同じ場合に、ロボットが移動中に衝突してしまう例が散見された。この例では、ロボットが移動障害物を右方向に検知したので左旋回するが、移動障害物は左方向に移動するため、衝突してしまった。

以降の図中の赤色の線はロボットの経路を表し、移動障害物の視線情報を赤・青・緑の線、ロボットは青い立方体、ロボットの正面を赤い球体で表している。

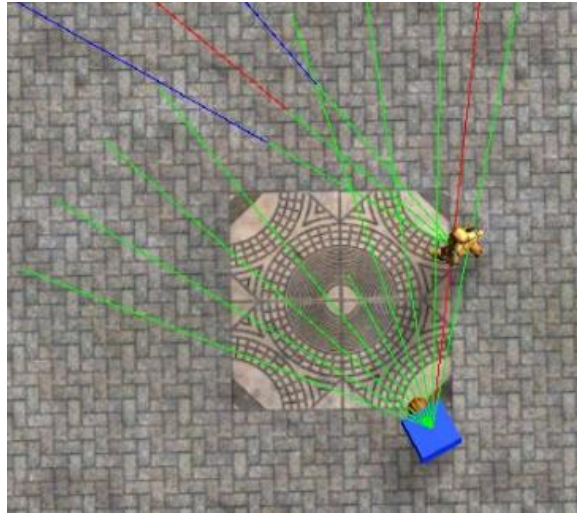


図 5.16: ルールベース制御で衝突しやすい状況例

ルールベース制御に比べて強化学習モデルでは、移動障害物との距離を大きく取り、移動障害物の回避行動を観測してから、移動障害物の移動に合わせた回避を行うことで、衝突を防いでいた。またルールベース制御と違い、横方向や後方の移動障害物の検知によって、状況に合わせた旋回を行っていた。図 5.17 のように前方に移動障害物がある場合には、図 5.18 のように移動障害物が通り過ぎるまで静止していた。

ロボットの距離センサを白と緑の線で表しているが、以降の図では見やすさのため、距離センサは可視化しない。

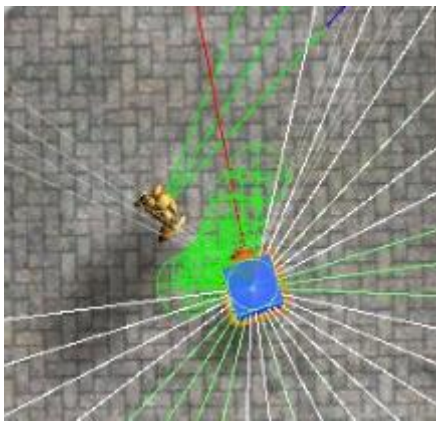


図 5.17: 強化学習モデルの回避成功例

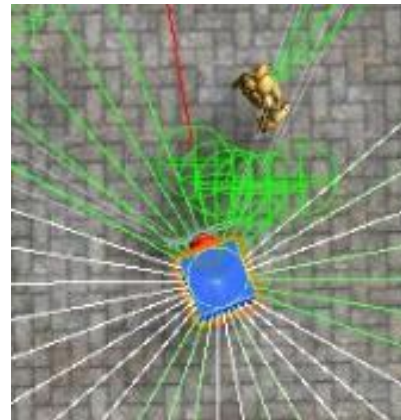


図 5.18: 強化学習モデルの回避成功例

また、図 5.19 のように緑色で示す移動障害物の視線がロボットに接触している場合は、図 5.18 のように移動障害物が通り過ぎるのを待たずに、移動障害物の逆方向に旋回して回避していた。

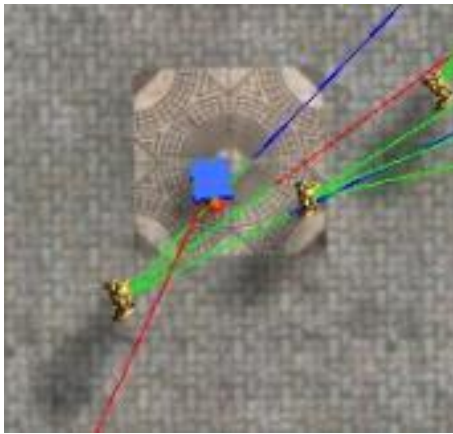


図 5.19: 強化学習モデルの回避成功例

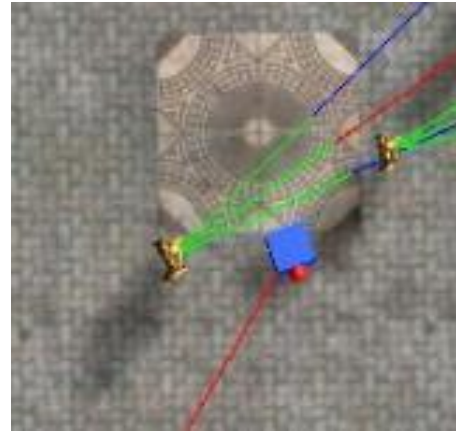


図 5.20: 強化学習モデルの回避成功例

強化学習モデルの回避失敗例として、図 5.21 のように自律移動ロボットの正面に移動障害物を検知し、静止障害物である植林の方向へ回避する際に、静止障害物によって他の移動障害物を検知が遅れ、図 5.22 のように正面の移動障害物を回避した後に、他の移動障害物に接触する例が散見された。

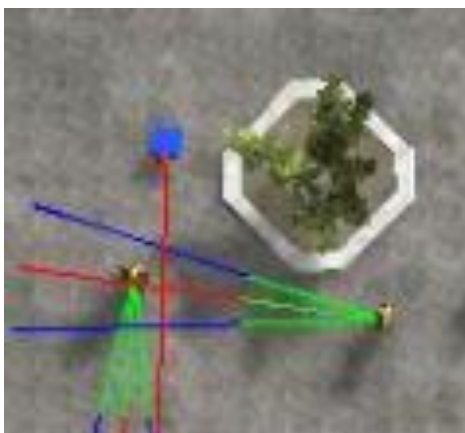


図 5.21: 強化学習モデルの回避失敗例

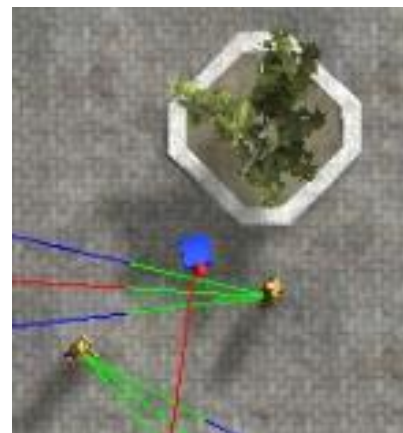


図 5.22: 強化学習モデルの回避失敗例

また、図 5.21 のようにロボットの左右どちらの方向にも移動しようとする移動障害物が存在する場合や、自律移動ロボットが静止している間に後方から他の移動障害物が近づいてくる状況では、回避がうまくできなかった。

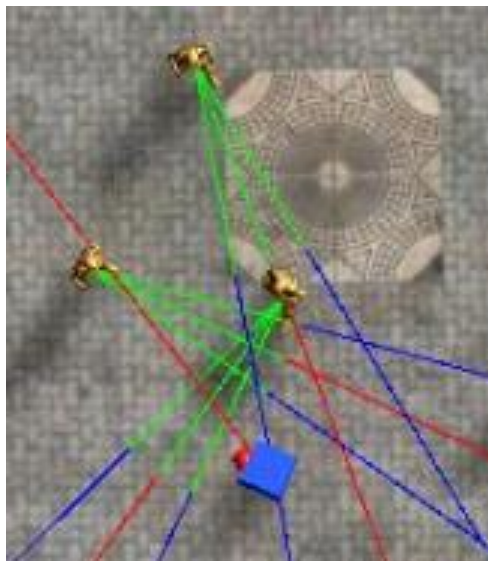


図 5.23: 強化学習モデルの回避失敗例

また、図 5.24 のようにロボットに移動障害物の視線である Ray が接触している場合に「視線情報なしの現在のステップの観測」であるモデルに比べて、「視線情報ありの現在のステップの観測」であるモデルの方が、静止ではなく視線を避ける移動の行動をしやすく、移動障害物が自律移動ロボットに接近する状況になりにくいため、移動障害物回避の精度が上がったと考えられる。移動障害物の視線情報があることによって、自律移動ロボットが移動障害物の移動経路に侵入しないように経路計画を行うため、移動障害物回避にとって重要な情報であると考えられる。



図 5.24: ロボットに移動障害物の視線が接触している例

「視線情報ありの現在と1つ前のステップの観測」であるモデルが、「視線情報ありの現在のステップの観測」のモデルと精度の違いがないのは、1ステップが0.02秒であるため、1つ前のステップの観測と現在の観測の変化があまりないためであると考えられる。

強化学習モデルによって移動障害物回避の行動を獲得できたが、静止障害物によって移動障害物の検知が遅れる状況や、自律移動ロボットが静止している状態で、移動障害物が自律移動ロボットに気づかずに接触してしまう状況については、本研究で獲得した強化学習モデルを修正する必要がある。強化学習による障害物回避はルールベースに比べてスムーズに回避できる一方、強化学習の中身はブラックボックスになっており、本モデルでは、獲得報酬が増加しなかったことから、時間をかけてもこれらの状況は回避できない可能性がある。実用化のためには、環境認識のためのセンサや学習に使用するアルゴリズムの変更など、より精度改善を行い、安全性を高める必要がある。

第 6 章 関連研究

本研究では、自律移動ロボットのシミュレーション環境を作成し、強化学習によって障害物回避行動を獲得させた。さらに、学習モデルの精度検証を学習用環境と検証用環境のそれぞれで行った。

本章では、その関連研究として、ルールベースでの障害物回避の研究や、人の視線情報や人の回避行動を考慮した障害物回避の研究、自動走行と強化学習を組み合わせた研究をいくつか紹介する。

6.1 ルールベースによる障害物回避

打井らは、ステレオカメラと距離センサを用いて周囲の環境を認識し、ロボットが現在位置から目的地まで自律移動を行うシステムを提案している [60]。距離センサで障害物を検知するとステレオカメラで正面画像を取得し、ステレオマッチングによる視差画像から障害物の認識を行う。ステレオカメラのみでは障害物までの正確な距離の計測が難しく、画像の処理に時間がかかるため、距離センサでリアルタイムのデータ取得を行っている。その後、障害物を検知するとステレオカメラを使用し、障害物の詳細な検知を行う。具体的には、ステレオカメラ画像を 8 等分し、図 6.1 のように障害物情報をもつ画像部分に対応した距離データの補正を行う。ロボットの進行領域に障害物が存在する場合に、進行領域に障害物が検出されない角度を計算する。その方法は、距離が補正された領域（図 6.1 の緑色の部分）をロボットの直進方向に障害物がなくなるまでのロボットを中心に回転させることで回転角度を算出し、その回転角度の小さい方向を選択して、障害物の回避を行う。本研究では、ルールベース制御において障害物を検知する際に、ロボットの左右どちらの方向に障害物があるかを検知し、障害物がない方向に回転することで障害物回避を行った。また、同様のセンサであるステレオカメラと距離センサを用いて周囲の環境を認識し、障害物回避を行った。

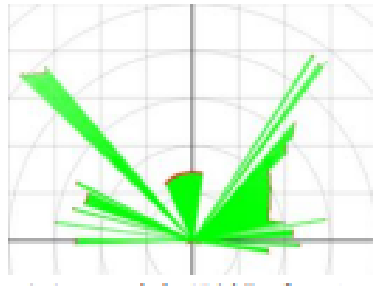


図 6.1 回転前の距離データ 文献[60]より抜粋

木曾らは、2種類のファジィ推論による電動車いすの障害物回避の制御手法を提案している[61]。進行方向を決める搭乗者のジョイスティック操作と、超音波センサで計測した電動車いすと障害物の距離をもとにファジィ推論(A)を行い、暫定的な走行方向を決定する。次に、ジョイスティック方向とファジィ推論(A)方向の差、電動車いすの平均速度をもとにファジィ推論(B)で走行危険度を決定し、走行危険度によって搭乗者の操作とファジィ推論結果のどちらを優先するかを決定している。実機を用いて、直進路で静止障害物回避実験を行っている。直進路であるため、搭乗者のジョイスティック入力を常に前に倒した状態で固定しているが、障害物に近づくと走行危険度が高くなり、ファジィ推論(A)方向が優先され、左右輪の速度が調整されて旋回走行を行い、静止障害物の回避が行われたことが確認された。

これらの手法はルールベースであるため、他の移動体への適用が比較的容易である。ルールベースで制御することでロボットは予想外の動きをせず、想定した環境で規定した適切な障害物の回避行動を行うことができる。本研究のルールベース制御では、単体の移動障害物の回避や前方に障害物を検知した際に静止することで移動障害物の回避を行ったが、前方以外の方向から接近する障害物や、前方から直進を続ける障害物のような想定されていない移動障害物に対して、ルールベース制御では衝突をしてしまう例が散見された。

6.2 人の視線情報を活用した障害物回避

佐藤らは、歩行者の視線計測を用いた衝突回避のシステムを提案している[62]。電動車いすを模した全方向移動ロボットを作成し、オムロン株式会社の OKAO Vision SDK[63]を用いて、対象の人間の注視点の推定をし、上下左右方向の視線の角度を活用している。歩行者の視線がカメラに向くほど大きく、カメラから外れるほど小さくなる視線強度係数を定義し、搭乗者のジョイスティックの入力値と歩行者の視線強度係数からロボットの並進速度を決定する。視線計測をもとに歩行者がロボットに気づいていないと判断した場合は、安全

性を向上させるためにロボットの速度を減速させる。実機を用いた直進路の被験者実験では、被験者がロボットの前に立ち、視線のみを動かしてロボットの走行速度を検証し、視線の感知によってロボットの速度が減速することが確認された。

本研究では、自律移動ロボットを認識しているか認識していないかの情報を視線情報として強化学習モデルの入力に加えて、移動障害物回避のモデルを構築した。視線情報を活用することで移動障害物の移動経路を考慮した経路計画を行うことが実験で確認された。

6.3 人の回避行動を考慮した経路計画

浅井らは、人の回避行動を4つに分類して、その動きに合わせてロボットが回避行動をとるような経路計画を提案している[64]。人がロボットと対峙した際の回避行動を停止・後退・横移動・前進しながら回避の4つに分類し、それぞれの人の挙動に対して有効な局所的経路計画を検証している。シミュレーション環境に存在する動的な障害物の特性を一括に考慮した ROS[65]の経路計画ライブラリである `teb local planner`[66]と、異なる速度を持つ物体同士の衝突を避ける群衆シミュレーションライブラリ `ROV 2`[67]を用いた経路計画の2種類の制御を用いたロボットの人回避の実験を行い、最短経路との距離の差や速度変化に関して2つのライブラリを比較検証している。本研究では、人の回避行動を「目的地に向かう行動の継続」「停止」「自律移動ロボットを回避（横方向への移動）」で設計し、強化学習モデルの経路計画とルールベース制御の経路計画で比較検証を行った。

6.4 強化学習を用いた自動走行の研究

津谷らは、市街地を想定した、対向車や横方向からの飛び出しが同時に発生する環境で、自動車の前方で検知された移動障害物に対する車線変更行動を強化学習によって獲得する手法を提案している[68]。方策勾配法である PPO を用いてシミュレーション環境で強化学習を行い、入力には自動車の位置や速度、加速度の情報に加え、回避する障害物であるバイクや自転車の相対位置と相対速度を用い、学習する行動は前進・加速・減速・車線変更の4種類である。シミュレーション環境で、移動障害物に対する自動車の車線変更行動の獲得を確認した。

岸川らは、自動車の搭乗者にとって不快感を与えない走行を深層強化学習によって獲得する手法を提案している[69]。獲得可能な報酬の定義が困難であることから、学習データや行動履歴を最適解とし、その最適解を出力する最適化指標を学習する逆強化学習[70]の一種

である LogReg-IRL (Logistic Regression-Based IRL) [71]を用いて報酬を推定する。図 6.2 のように、逆強化学習とは、暫定的な最適化指標に基づく解と学習データや行動履歴との差分を小さくするために最適化指標の重みの更新を繰り返し行う学習方法である。

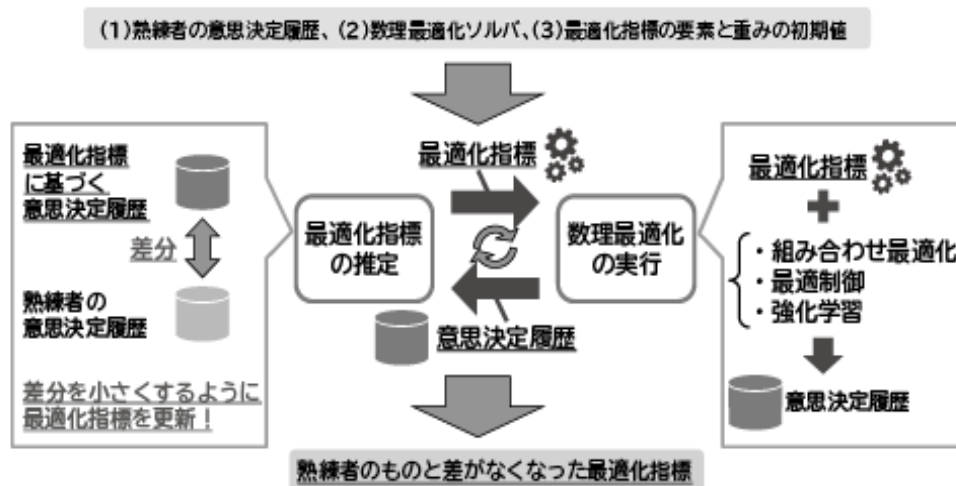


図 6.2 逆強化学習アルゴリズム 文献[70]より抜粋

最適化指標である推定した報酬を用いて DDPG で学習を行い、安全走行の基準となる直進走行において横方向の加速度が 0.3G 未満となるモデルの作成を行っている。モデルの入力に車両の角度・ギア(後退, 停止, 6 段階の前進)・エンジンの回転数・速度・道路の中心からの距離・四輪の回転速度・車両重心の変位を用いて、舵角の制御方策をシミュレーション環境で学習させ、安全走行の基準を満たすモデルを獲得したことを確認した。

Kendall らは、強化学習による自動車の車線追従の行動獲得を行った[72]。その学習モデルの入力には、自動車の速度や操舵角に加え、自動車前方の RGB カメラ画像を VAE (Variational Auto Encoder)[73]で情報圧縮したデータを用いている。VAE で情報圧縮することで、短時間でより正確なモデルの学習ができることを検証している。

これらのいずれの研究でも、シミュレーション環境で強化学習を用いたモデル構築を行い、学習済みモデルを適用した自動走行システムの評価検証をしている。ただし、一般にシミュレーション環境は単純化された環境であるので、より複雑な環境で学習モデルが適用できるようにすることや、実機適用に関する研究を行う必要がある。本研究では、同様に強化学習モデルを用いて目的地への安全な自動走行実験を行った。シミュレーション環境での実験を行ったが、実機適用を実施していないので、実環境でモデル適用を行い、評価検証を行う必要がある。

第7章 おわりに

7.1 まとめ

本研究では、シミュレーション環境及び自律移動ロボットの3Dモデル、移動障害物の人モデルを作成し、強化学習を用いて移動障害物回避行動の獲得を行った。

移動障害物である人のモデルは、自律移動ロボットに気づいた場合は回避行動をとるように、気づいていない場合は回避行動をとらないように設計した。自律移動ロボットは、想定する実機と同様に距離センサとカメラ画像を使用する。ショッピングモールを再現したシミュレーション環境で強化学習を実施した。

移動障害物回避は、入力情報に、移動障害物の視線情報ありとなしの2種類の強化学習モデルで学習を行い、結果として、移動障害物の視線情報を活用することで回避の精度が向上した。移動障害物の視線情報があることによって、自律移動ロボットが移動障害物の移動経路に侵入しないように経路計画を行うため、移動障害物回避にとって重要な情報であると考えられる。また、同様に、入力情報に、1ステップ過去の情報ありとなしのモデルで学習を行ったが、1ステップ過去の情報を用いることで精度の向上は確認できなかった。ある移動障害物を避けた後、近くに他の移動障害物がいる場合や、静止障害物の設置状況によって移動障害物の検知が遅れる場合、移動障害物が静止しているロボットに気づかず接近する場合に、ロボットは移動障害物回避ができない事例があることを確認できた。実用化のためには安全性や精度改善を行う必要があり、環境認識を行うセンサや学習に使用するアルゴリズムの変更、歩行者やロボットの搭乗者に危害を加えない工夫が考えられる。

7.2 今後の課題と展望

7.2.1 歩行者がロボットに気づいていない状況への対処

移動障害物回避がうまくいかなかった例として、本研究では、静止している自律移動ロボットに対して、ロボットに気づいていない人が接触してしまう状況が散見された。ロボット

が人に危害を加えないために、人がロボットに接触しても衝撃を吸収するような柔らかな素材でロボットを構成する工夫が考えられる。また、ロボットの存在を人に気づかせるために、人の近くでは音を出すなど、人の注意を引くことも大切であると考えられる。

また、本研究では、人の視線情報を用いて、人がどの方向に歩行をするかの情報をロボットに与えていたが、現実ではロボットに気づいていない状況の一例として、人が歩きながらスマホの操作をしている状況が存在し、この場合は人の視線はスマホを向いているため、ロボットは人の視線情報を正しく認識することができない。この場合に人が急に方向転換することは考えにくく、そのままの進行方向に歩き続けるか静止するかが考えられる。ロボットに気づいていない人がロボットに向かって歩き続ける場合は、ロボットが人とある程度の距離をとり続けるように学習させる必要がある。本研究の衝突の例で多かった、静止しているロボットに対して人が接近して衝突することを防ぐために、自律移動ロボットが必要に応じて後退の行動をとるように工夫することができる。その際に後退することで、後ろにいる他の障害物と接触しない必要があり、ロボットの後ろに対する環境の情報取得においてカメラなどのセンサが追加で必要になる可能性がある。

7.2.2 搭乗者を考慮した快適な回避行動

今回の障害物回避では、強化学習において加速度を考慮していないため、停止を含む走行速度を離散的に決定している。パーソナルモビリティの移動障害物回避システムとして導入するには、パーソナルモビリティの搭乗者が不安や危険を感じない範囲の加速度制御を行う必要がある。また、障害物をぎりぎり回避する行動も搭乗者が不安に感じる要因となる。そのため、加速度を考慮した強化学習を行うことで対処する必要がある。入力と出力の速度の差分を報酬として考慮し、入力と出力の速度差がある閾値を超えないように速度制御を学習させることが考えられる。また、搭乗者が乗ることでパーソナルモビリティ全体の重量が増え、制御に影響を与える可能性があるため、ロボットに人のモデルを乗せた状態で移動障害物回避の学習を行うことも有効であると考えられる。

7.2.3 実環境での実証実験

本研究では、シミュレーション環境での評価検証は行ったが、実環境での評価検証を行っていない。実環境での人間の回避行動はより複雑だと考えられ、周囲にさらに多くの人がいる場合もある。また、ロボットの環境認識や自己位置推定におけるセンサの誤差や、移動の誤差を許容したモデルの構築も必要である。そのため、シミュレーション環境で学習したモデルを実環境で評価検証することに加えて、学習したモデルを活用して実環境で再度強化

学習を行う転移学習[74]を行い, 実環境に適応したパラメータチューニングを行う必要がある. 転移学習を行うことで学習の高速化やモデル精度の向上が期待できる. 本研究では, 移動障害物を人に限定したが, 自転車などの小型移動体や動物など, パーソナルモビリティが用いられる環境には様々な移動障害物が存在するため, 様々な状況を想定した実環境での実証実験を行う必要がある.

謝辞

本研究を進めるにあたり、指導教員でありプロジェクトリーダーの長尾確教授には、研究に対する基本的な心構えをはじめ、技術的な御指導、研究や論文執筆に関する貴重な御支援や御意見、御指導などを頂き、大変お世話になりました。心より御礼申し上げます。

柴田泰雅先輩、そして、赤見坂篤記先輩には、同じプロジェクトのメンバーとして、本研究に対する様々な御支援や御意見、論文執筆に関わる多くの御支援、研究室生活に対するアドバイスなどを頂き、大変お世話になりました。ありがとうございました。

西出勇太先輩、小林大貴先輩には、研究室の先輩としてゼミなどを通して本研究に関する多くの御意見やアドバイスを頂いたことに加え、学生生活、研究活動についてのご支援を頂き、大変お世話になりました。ありがとうございました。江寄一優先輩、張静雯先輩には、基礎的な知識や技術に関するご指導を頂き、また論文執筆に対する貴重な御支援を頂き、大変お世話になりました。ありがとうございました。

佐藤義忠さん、そして、大岩和樹さんには、研究室の同期として、常日頃から研究や学生生活に関する意見を頂くとともに、世間話など研究室での息抜きに付き合ってくださいました。ありがとうございました。齋藤実ユ紀さんには、長尾研究室の秘書として様々な事務手続きをしていただき、学生生活や研究生活に関して多くの面でお世話になりました。ありがとうございました。

最後に、日々の生活を支えてくださった家族、および友人に心から感謝します。

参考文献

- [1] 内閣府, “令和 4 年版高齢社会白書,”
https://www8.cao.go.jp/kourei/whitepaper/w-2022/zenbun/pdf/1s1s_01.pdf, (2023-2-12).
- [2] 内閣府, “未就学児等及び高齢運転者の交通安全緊急対策,”
<https://www8.cao.go.jp/koutu/taisaku/sougou/pdf/20190618/taisaku.pdf>, (2023-2-12).
- [3] 株式会社グローバルインフォメーション, “パーソナルモビリティデバイスの市場規模,” <https://www.value-press.com/pressrelease/274638>, (2023-2-12).
- [4] 国土交通省, “超小型モビリティ導入に向けたガイドライン,”
<https://www.mlit.go.jp/common/000212867.pdf>, (2023-2-12).
- [5] 国土交通省, “超小型モビリティシンポジウム,”
<https://www.mlit.go.jp/jidosha/content/001364953.pdf>, (2023-2-12).
- [6] 日本経済新聞, “ANA とパナソニック, シニア世代の旅行サポートするパーソナルモビリティ（自動追従電動車椅子）の実証実験を実施,”
https://www.nikkei.com/article/DGXLRSF509761_W9A510C1000000/, (2023-2-12).
- [7] 時事メディカル, “世界初！成田空港第 2 旅客ターミナルで「WHILL 自動運転モビリティサービス」とエレベーターが連携した階層移動の実証実験が開始,”
<https://medical.jiji.com/prtimes/112886>, (2023-2-12).
- [8] 田上拓弥, 星野智史, “BMI を介したパーソナルモビリティロボットの操作支援,” 日本ロボット学会誌, vol. 40, no. 5, pp. 445-448, 2022.
- [9] Irobot Corporation, “Roomba,” <https://www.irobot-jp.com/roomba/>, (2023-2-12).
- [10] Starship Technologies, “Starship Deliveries,” <https://starshipdeliveries.com/industry/>, (2023-2-12).
- [11] P. Besl and H. McKay, “A Method of Registration of 3-D Shapes,” In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.
- [12] E. Takeuchi and T. Tsubouchi, “A 3-D Scan Matching Using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping,” In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3068-3073, 2006.
- [13] 日本ロボット学会, “ロボット工学ハンドブック,” コロナ社, 2005.
- [14] R. Faragher, “Understanding the Basis of the Kalman Filter via a Simple and Intuitive Derivation,” In Proceedings of the IEEE Signal Processing Magazine vol. 29, pp. 128-132, 2012.

- [15] I. M. Rekleitis, “A Particle Filter Tutorial for Mobile Robot Localization,” Part of the Statistics for Engineering and Information Science book series (ISS), 2004.
- [16] F. Moosmann and C. Stuller, “Velodyne SLAM,” In Proceedings of the Intelligent Vehicles Symposium (IV), pp. 393-398, 2011.
- [17] 田崎豪, “Autoware で始める自律移動技術入門,” 森北出版, 2021.
- [18] 平田富夫, “アルゴリズムとデータ構造 -改訂 C 言語版-,” 森北出版株式会社, 2002.
- [19] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.
- [20] P. E. Hart, N. J. Nilsson and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” In Proceedings of the IEEE Transactions on Systems Science and Cybernetics (SSC), pp. 100-107, 1968.
- [21] D. Dolgov, S. Thrun, M. Montemerlo and J. Diebel, “Path Planning for Autonomous Vehicles in Unknown Semistructured Environments,” *International Journal of Robotics Research*, vol. 29, no. 3, pp. 333-349, 2010.
- [22] J.-H. Chuang and N. Ahuja, “An Analytically Tractable Potential Field Model of Free Space and Its Application in Obstacle Avoidance,” In Proceedings of the IEEE Transactions on Systems Man and Cybernetics, Part B (Cybernetics), vol. 28, no. 5, pp. 729-736, 1998.
- [23] 小澤健太郎, 坪内孝司, “測域センサを用いた自律移動ロボットによる障害物回避走行,” ロボティクス・メカトロニクス講演会, 2019.
- [24] D. Fox, W. Burgard and S. Thrun, “The Dynamic Window Approach to Collision Avoidance,” In Proceedings of the IEEE Robotics and Automation Magazine, vol. 4, issue. 1, pp. 23-33, 1997.
- [25] 株式会社アーテック, “超音波センサ,”
https://www.artec-kk.co.jp/studuino/docs/jp/Studuino_manual_ultrasonic.pdf, (2023-2-12).
- [26] Suteng Innovation Technology Co. Ltd, “3D-LiDAR,”
<https://www.zmp.co.jp/products/sensor/3d-lidar/rslidar>, (2023-2-12).
- [27] 株式会社デンソー, “ミリ波レーダ,”
<https://www.denso.com/jp/ja/news/newsroom/2022/20220114-01/>, (2023-2-12).
- [28] 株式会社ロジクール, “RGB カメラ,” <https://www.logicool.co.jp/ja-jp/products/webcams.html>, (2023-2-12).
- [29] Intel Corporation, “デブスカメラ,”
<https://www.intel.co.jp/content/www/jp/ja/architecture-and-technology/realsense-overview.html>, (2023-2-12).
- [30] Y. Oike, M. Ikeda and K. Asada, “Design and Implementation of Real-Time 3-D Image Sensor with 640 x 480 Pixel Resolution,” In Proceedings of the IEEE Journal of Solid-State

Circuits, vol. 39, no. 4, pp. 622-628, 2004.

[31] C. Niclass, M. Soga, H. Matsubara, S. Kato and M. Kagami, "A 100-m Range 10-Frame per sec 340x96-Pixel Time-of-Flight Depth Sensor in 0.18- μ m CMOS," In Proceedings of the IEEE Journal of Solid-State Circuits, vol. 48, no. 2, pp. 559-572, 2013.

[32] I. Stančić, J. Musić and V. Zanchi, "Improved Structured Light 3D Scanner with Application to Anthropometric Parameter Estimation," Measurement, vol. 46, pp. 716-726, 2013.

[33] 打井裕基一, 芋野美紗子, 土屋誠司, 渡部広一, "ステレオカメラと距離センサを用いた障害物検出による知能ロボットの自律移動手法," 第14回情報科学技術フォーラム, vol. 14, no. 2, pp. 291-292, 2015.

[34] Unity Technologies, "Unity Real-Time Development Platform 3D, 2D VR & AR Engine," <https://unity.com/ja>, (2023-2-12).

[35] Unity Technologies, "Unity Asset Store," <https://assetstore.unity.com/?locale=ja-JP>, (2023-2-12).

[36] Unity Technologies, "Unity Asset Store Shopping Mall HQ," <https://assetstore.unity.com/packages/3d/environments/urban/shopping-mall-hq-2293?locale=ja-JP>, (2023-2-12).

[37] Unity Technologies, "Unity Asset Store StarterAssets thirdPerson," <https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-196526?locale=ja-JP>, (2023-2-12).

[38] Unity Technologies, "Class RayPerceptionSensor," <https://docs.unity3d.com/Packages/com.unity.ml-agents@1.0/api/Unity.MLAgents.Sensors.RayPerceptionSensor.html>, (2023-2-12).

[39] Unity Technologies, "Class CameraSensor," <https://docs.unity3d.com/Packages/com.unity.ml-agents@1.0/api/Unity.MLAgents.Sensors.CameraSensor.html>, (2023-2-12).

[40] Unity Technologies, "Unity Documentation Building a NavMesh," <https://docs.unity3d.com/2021.3/Documentation/Manual/nav-BuildingNavMesh.html>, (2023-2-12).

[41] Unity Technologies, "Unity Documentation NavMesh Agent," <https://docs.unity3d.com/2021.3/Documentation/Manual/class-NavMeshAgent.html>, (2023-2-12).

[42] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, pp. 100-107, 1968.

[43] 瀬谷安弘, "有効視野の特性とその測定手法,"

<https://annex.jsap.or.jp/photonics/kogaku/public/42-09-kougakukoubou.pdf>, vol. 42, Issue. 9, pp. 473-474, 2013.

[44] 布留川英一, “Unity ではじめる機械学習・強化学習 Unity ML-Agents 実践ゲームプログラミング,” ボーンデジタル社, 2018.

[45] B. Grooten, J. Wemmenhove, M. Poot and J. Portegies, “Is Vanilla Policy Gradient Overlooked? Analyzing Deep Reinforcement Learning for Hanabi,” In Proceedings of the Adaptive and Learning Agents Workshop at International Conference on Autonomous Agents and Multiagent Systems, 2022.

[46] J. Schulman, S. Levine, P. Abbeel, M. Jordan and P. Moritz, “Trust Region Policy Optimization,” In Proceedings of the International Conference on Machine Learning, pp. 1889-1897, 2015.

[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O Klimov, “Proximal Policy Optimization Algorithms,” arXiv: 1707.06347, 2017.

[48] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, “Continuous Control with Deep Reinforcement Learning,” In Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[49] S. Fujimoto, H. v. Hoof and D. Meger, “Addressing Function Approximation Error in Actor-Critic Methods,” In Proceedings of the International Conference on Machine Learning, 2018.

[50] T. Haarnoja, A Zhou, P. Abbeel and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” In Proceedings of the International Conference on Machine Learning (PMLR), pp. 1861-1870, 2018.

[51] M. Bain and C. Sommut, “A Framework for Behavioral Cloning,” Machine Intelligence, vol. 15, pp. 103-119, 1999.

[52] X. B. Peng, E. Coumans, T. Zhang, T. -W. E. Lee, J. Tan and S. Levine, “Learning Agile Robotic Locomotion Skills by Imitating Animals,” In Proceedings of the Robotics: Science and Systems (RSS XVI), 2020.

[53] Unity Technologies, “ML-Agents Toolkit,” <https://github.com/Unity-Technologies/ml-agents>, (2023-2-12).

[54] Unity Technologies, “ML-Agents Custom trainers Plugin,” <https://github.com/Unity-Technologies/ml-agents/blob/develop/docs/Python-Custom-Trainer-Plugin.md>, (2023-2-12).

[55] Unity Technologies, “ICommunicator,” <https://docs.unity3d.com/Packages/com.unity.ml-agents@2.3/api/Unity.MLAgents.ICommunicator.html>, (2023-2-12).

[56] Unity Technologies, “ML-Agents-Overview,” <https://github.com/Unity->

- Technologies/ml-agents/blob/2a888f23636c85743c6e987e788bd93f1d7d1291/docs/ML-Agents-Overview.md, (2023-2-12).
- [57] L. Roeder, “Netron,” <https://netron.app/>, (2023-2-12).
- [58] Unity Technologies, “Training ML-Agents,” https://github.com/Unity-Technologies/ml-agents/blob/release_20_docs/docs/Training-ML-Agents.md, (2023-2-12).
- [59] Alphabet Inc., “TensorBoard”, <https://www.tensorflow.org/tensorboard?hl=ja>, (2023-2-12).
- [60] 打井裕基一, 芋野美紗子, 土屋誠司, 渡部広一, “ステレオカメラと距離センサを用いた障害物検出による知能ロボットの自律移動手法,” 第 14 回情報科学技術フォーラム, vol. 14, no. 2, pp. 291-292, 2015.
- [61] A. Kiso, H. Murakami and H. Seki, “Obstacle Avoidance Control of Electric Powered Wheelchair by Fuzzy Inference Considering Driver's Intention,” In Proceedings of the International Conference on Advanced Mechatronic Systems (ABML), 2011.
- [62] Y. Sato and H. Igarashi, “Safe Driving Support of the Omni-Directional Mobile Vehicle Using Gaze Measurement,” In Proceedings of the International Workshop on Nonlinear Circuits, Communications, and Signal Processing, 2016.
- [63] オムロン株式会社, “OKAO Vision ソフトウェアライブラリ”, <https://components.omron.com/jp-ja/products/sensors/human-image-solution/software-library/software-library>, (2023-2-12).
- [64] 浅井悠佑, 廣井慧, 米澤拓郎, 河口信夫, “人の回避行動を考慮した移動ロボットの経路計画法の検討,” マルチメディア, 分散, 協調とモバイルシンポジウム, 2019.
- [65] Willow Garage Inc., “ROS,” <http://wiki.ros.org/ja>, (2023-2-12).
- [66] Willow Garage Inc., “teb_local_planner ライブラリ”, http://wiki.ros.org/teb_local_planner, (2023-2-12).
- [67] J. Snape, J. Berg, S. J. Guy and D. Manocha, “Smooth and collision-free navigation for multiple robots under differential-drive constraints,” In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 18-22, Taipei, Taiwan, 2010.
- [68] 津谷龍一, 荒井幸代, “同時・複数の障害物回避における自動運転の獲得方策決定法”, 計測自動制御学会第 48 回知能システムシンポジウム, 2021.
- [69] 岸川大航, 荒井幸代, “深層逆強化学習による自動運転の安心走行の実現,” 第 33 回人工知能学会全国大会, 2019.
- [70] NEC Corporation, “熟練者の意思決定を模倣する意図学習技術”, <https://jpn.nec.com/techrep/journal/g19/n01/190120.html>, 2019.
- [71] E. Uchibe, “Model-Free Deep Inverse Reinforcement Learning by Logistic Regression,” In Proceedings of the Neural Processing Letters, pp. 891-905, 2017.

- [72] A. Kendall and J. Hawke, D. Janz, P. Mazur, D. Reda, J. Allen, V. Lam, A. Bewley and A. Shah, “Learning to Drive in a Day,” In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 8248-8254, 2019.
- [73] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” In Proceedings of the International Conference on Learning Representations (ICLR), 2014.
- [74] 河野仁, “Python で実装する強化学習と転移学習”, 森北出版, 2022.