2ZA-02

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

加藤　翼†　　長尾　確††
名古屋大学　情報学部†　名古屋大学　大学院情報学研究科††

1. はじめに

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．ただし，歩道や屋内での走行を想定しているため，人の近くや狭い場所，地図にない障害物が置かれている環境などを安全に走行できる必要がある．

本研究では人の歩行行動を模倣する移動障害物を作成し，実環境に近いシミュレーション環境で実験を行う．自律移動ロボットが複数の移動障害物が存在する環境で強化学習を行い，安全な走行及び移動障害物回避の行動獲得ができることを示す．

2. 自律移動ロボットの構成

自律移動ロボット[1]とは，自身のセンサで得た情報をコンピュータで処理して行動を決定し，足や車輪で移動ができるロボットである．自律移動は認知・判断・制御の処理を一定の周期で繰り返し，認知で自己位置推定や障害物検出，判断で経路生成や経路計画，制御で経路追従などを行い目的地へ向かう．

本研究で想定する自律移動ロボットはセンサとして全方位 3 次元 LiDAR と RGB カメラを用いて環境認識を行う．地図生成を行わず，事前に地図を取得した状態で目的地の経路追従と地図にない移動障害物の回避を同時に行う．

3. シミュレーション環境の構築

障害物回避の強化学習を行うためのシミュレーション環境(図 1 左)を作成する．シミュレーション環境は物理エンジン Nvidia PhysX を搭載した Unity を使用する．

環境はショッピングモールを想定し，Unity 用の都市アセット Shopping Mall HQ を使用する．広場に静止障害物である植林，四方を囲む壁に店の外観が描画されている．移動障害物は人間を想定し，人型モデル(図 1 右)を使用する．

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量を軽量化するために単純な立方体で構成されている．全方位 3 次元 LiDAR を模した距離センサ RaycastSensor(図 2 左) と前方の認識を行う RGB カメラ(図 2 右)を設置している．

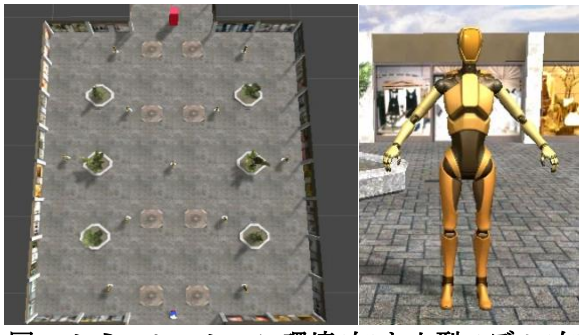

図 1 シミュレーション環境(左)と人型モデル(右)

RaycastSensor は周囲 360 度に 26 本の Ray を飛ばし，障害物に Ray が接触すると障害物までの距離を取得する．RGB カメラは幅 240px，高さ 135px の画像を取得する．自律移動ロボットの走行は前進と停止をすることができ，方向を変えるための右転回と左転回ができる．Navmesh を使用して自律移動ロボット(図 1 の青いオブジェクト)が移動可能な領域を環境に指定し，静止障害物の回避を行う．目的地(図 1 の赤いオブジェクト)への経路追従には NavMeshAgent[2]を用いる．

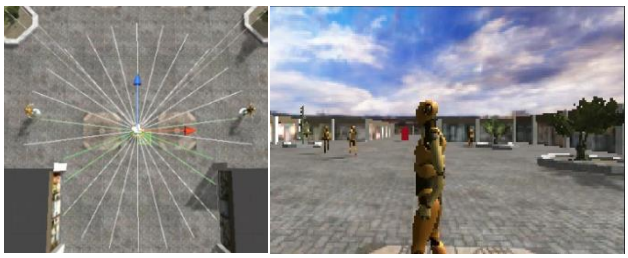

図 2 RaycastSensor (左)と RGB カメラ(右)

移動障害物(人型モデル)は NavMeshAgent を用いて，ランダムに選択された目的地へ最短経路で移動する．ただし，視野角 180 度を想定し，視界に自律移動ロボットを発見すると確率的に「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 を用いた目的地へ向かう行動を一時的に停止する．「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 度の領域を左右半分ずつに分け，自律移動ロボットが左右のどちらにあるかを計測し，自律移動ロボットが存在しない側の領域へ移動する．移動する領域への転回の角度はランダムで決定する．視野に入っていない場合は自律移動ロボットに気づいていない状況であるとして「目的地に向

Autonomous Mobile Robot with Moving Obstacle Avoidance Based on Simulation and Reinforcement Learning
†KATO, Tsubasa (kato.tsubasa.c8@s.mail.nagoya-u.ac.jp)
††NAGAO, Katashi (nagao@i.nagoya-u.ac.jp)
†School of Informatics, Nagoya University
††Graduate School of Informatics, Nagoya University

かう行動」を継続する．また，正面にRayを飛ばして他の移動障害物との距離を計測し，近づきすぎた場合は2秒停止し距離をとる．移動速度は自律移動ロボットの速度を超えない範囲でランダムに決定する．

4. 強化学習による移動障害物の回避行動の獲得

4.1. 強化学習のモデル

強化学習のフレームワークにML-Agents[3]を用いる．PythonAPIを使用し，Unityの観測情報とPytorchの強化学習モデルを通信し，自律移動ロボットを制御する．

本研究ではモデルフリー強化学習である Soft Actor-Critic（SAC）[4]と呼ばれる手法を使用する．

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による目的地への向きと速度の出力のうち，向きの出力のみを移動障害物回避モデルの入力に用いる．モデルの入力に RBG カメラ画像（3×135×240），RaycastSensor，自律移動ロボットの速度と向き，NavMeshAgent の経路追従の向き，移動障害物の視線情報を用いる．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検知するための情報であり，移動障害物の進行方向に飛ばしたRayが自律移動ロボットに接触した数を計測し，移動障害物の視線の方角と含めて学習モデルの入力に用いる．赤色の視線が自律移動ロボットに接触した場合は＋1，青色の視線の場合は＋0.5 と計測する．視線が来る方角は簡略化のため，30 度ずつの離散情報とする．

RGB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．

本実験ではRGBカメラ画像と RaycastSensor のパラメータを変更し，「現在のステップのみの観測」，「現在と１つ前のステップの観測」，「現在と１つ前と２つ目前のステップの観測」の３種類の入力の比較実験を行う．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出力であり，速度は「前進」「前進の半分」「停止」のいずれかである．速度が離散値であるのは，より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十分であると考えられるからである．方向は「そのまま」「右旋回」「左旋回」のいずれかである[5]．

自律移動ロボットが「目的地に到達」，または「障害物に衝突」でエピソードを終了し，「目的地に到達」または「移動障害物回避の際に通る後方の領域に接触」で正の報酬を，「移動障害物に接触」または「移動障害物の正面を通過」で負の報酬が与えられ，目的地に最短経路で向かう行動を学習させるため「毎ステップ」に負の報酬が与える．報酬の値を表1に示す．

表１ 報酬の値設定

報酬の取得時	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
毎ステップ	−0.005

各エピソードの開始時に自律移動ロボットの位置と目的地の位置が複数の候補位置からランダムに選択される．

4.2. 強化学習の結果とその評価

学習ステップ数は200万回とし，10000ステップごとに1エピソードで得られる累積報酬和と方策エントロピーを計算する．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 回の試行を行い，移動障害物に衝突せずに目的地に到達した回数を求めた．表2に結果を示す．

表２ 学習済みモデルを用いた1000回の試行

RGBカメラの入力画像	成功回数（学習環境）	成功回数（検証環境）
現在	876 回	907 回
現在＋1つ前	910 回	918 回
現在＋1つ前＋２つ前	913 回	924 回

試行では移動障害物を回避する行動が確認でき，RGB カメラの入力が現在のみの場合は，移動障害物と距離をとり，停止して移動障害物の挙動を観測後に回避行動をとることが多かったが，RGB カメラの過去の情報を入力に使用することで距離をとって停止する挙動が少なくなり，移動障害物の行動に合わせた回避行動をとることができ，目的地到達までの時間が短くなった．

5. おわりに

本研究では，シミュレーション環境を作成し，その環境上で障害物回避の強化学習を行い，移動障害物回避行動を獲得することが確認できた．今後の課題として，より複雑な環境下で障害物回避の学習を行い，学習したモデルを実機に適用し，実環境での精度を検証する．

参考文献

[1]田崎豪, Autowareで始める自律移動技術入門, 森北出版,2021年
[2] NavmeshAgent, https://docs.unity3d.com/ja/2021.2/Manual/class-NavMeshAgent.html
[3] Arthur Juliani,Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, Danny Lange. Unity: A General Platform for Intelligent Agents. arXiv:1809.02627, 2018.
[4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine, Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, arXiv:1801.01290, 2018.
[5] Yusuke Tamura, Shunsuke Hamasaki, Atsushi Yamashita, Hajime Asama, Collision Avoidance of Mobile Robot Based on Prediction of Human Movement According to Environments, JSME, Vol. 79, Issue 799, pp.617-628, 2013.