

深層強化学習を用いた搬送ロボットの

自律走行制御に関する研究

三浦学苑高等学校工業技術科

田崎 民生

1. はじめに

AI 技術は、現在大きく進展しており、さまざまな業種で導入され大きな成果をあげている。画像や音声の認識系 AI の他、最近では生成系 AI の技術的向上には目を見張るものがある。このような先進技術は、日本のものづくり産業を支える技術の一つとなっており、今後の工業教育においては、AI 技術を適切に理解し活用する基礎的能力の育成が求められるものと考えられる。本研究は、LiDAR センサの情報をもとに深層強化学習 (AI) を用いて搬送ロボットを自律走行制御することを目標とし、得られた成果を今後の工業教育に活かしていくことを目的としている。

2. 深層強化学習

2.1 深層強化学習について

強化学習は機械学習の一つであり、「エージェント」が「環境」の「状態」に応じてどのように「行動」すれば「報酬」を多くもらえるかを求める手法で、これに深層学習を組み合わせたものが深層強化学習である。強化学習の学習サイクルを図 1 で示す。エージェントは「状態取得」で環境の状態を取得しこの情報に基づいて行動を決定する（「行動決定」）。それが好ましい行動であればプラスの報酬、好ましくない行動であればマイナスの報酬として、エージェントは環境から報酬を得る（「報酬取得」）。本研究では、搬送ロボットを走行路に沿って自律走行させることを目指しているが、この場合「エージェント」が搬送ロボット、「環境」はエージェントのいる世界（走行路）、「状態」は「環境」が保持している状態であり走行路の形状や障害物の他、搬送ロボットの速度や位置などが該当する。壁や障害物に接触した場合にはマイナスの報酬、走行路に沿って走行できればプラスの報酬を得る。行動を決定するポリシーが強化学習サイクルの中で定期的に更新され（「ポリシー更新」）、次第に好ましい行動がとれるようになる。

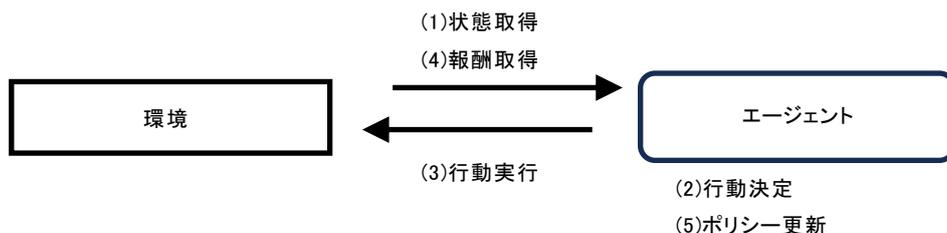


図 1 強化学習サイクル

2.2 深層強化学習パッケージについて

深層強化学習は、ロボットの歩行制御や自動車の自動運転などで利用されており、様々なシステムの制御ルールの構築に用いられている。ここでは、深層強化学習パッケージとして Unity ML-Agents を使用し、コンピュータ上に搬送ロボットのシミュレーション環境を構築して自律走行制御のための深層強化学習を行った。Unity ML-Agents は、Unity (ゲームエンジン) のパッケージの一つであるが、深層学習フレームワークに PyTorch を用いており GPU を用いた深層強化学習にも対応している。

図 2 に深層強化学習パッケージの構成を示す。Unity で「状態取得」、「行動実行」、「報酬取得」のやり取りを行い、PyTorch で「行動決定」、「ポリシー更新」を行う。シミュレーション環境を用いて作成した学習モデルが AI の頭脳であり、実環境の自律走行制御にここで作成した学習モデルを使用する。

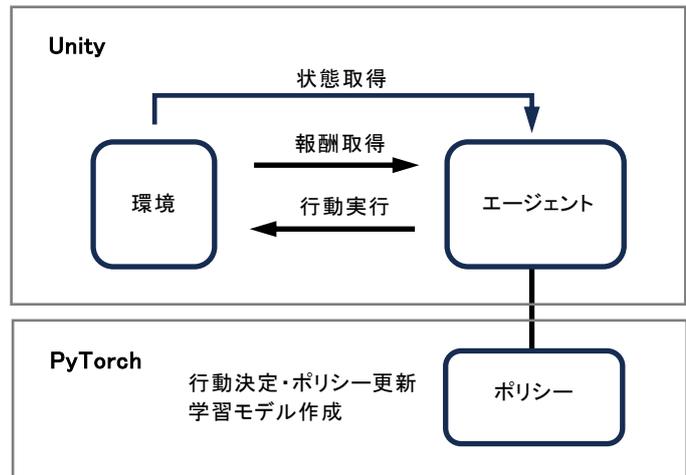


図 2 深層強化学習パッケージの構成

2.3 シミュレーション環境の構築

実環境の実験は学校校舎の廊下 (幅 2.8m) で行うため、搬送ロボットが走行する走行路の幅は 2.8m を基準とし、3つのコースパターンを製作した。1つ目は走行路の幅を基準の 2.8m で製作 (Type A)。2つ目は走行路の幅を基準幅 2.8m に対して $\pm 0.5m$ 程度の変化を持たせて製作 (Type B)。3つ目は2つ目のコースと同じコースに段ボール箱程度の複数サイズ

(高さ、幅、奥行: $\pm 1.0m$ 程度)の障害物をコース上に配置した (Type C)。3つ目のコースモデルを図 3 に示す。搬送ロボットのモデルは実験で使用する搬送ロボットと同程度のものとし、サイズは $0.2m \times 0.3m \times 0.35m$ 、行動の種類は、前進、後進、右回転、左回転の4つとした。また、搬送ロボット上部には LiDAR センサが搭載されており、 $-120^\circ \sim +120^\circ$ の範囲で壁や障害物までの距離やその角度情報がシミュレーション環境上でも得られるようにした。製作した搬送ロボットのモデルを図 4 に示す。

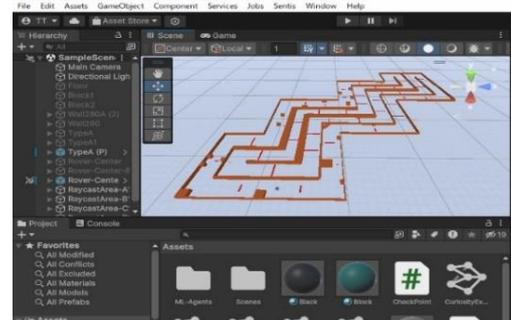


図 3 コースモデル (Type C)

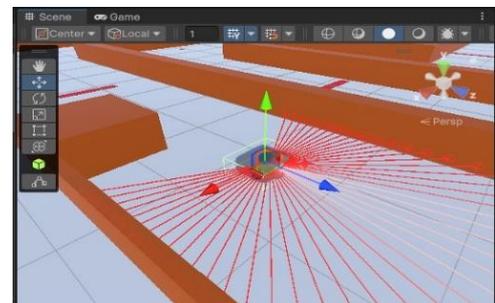


図 4 走行路上の搬送ロボット

2.4 深層強化学習プログラムの作成

シミュレーション環境上で動作する深層強化学習プログラムのフローチャートを図 5 に示す。

シミュレーション環境上で 1Step は 0.02 秒となる。「状態取得」では、Lidar センサから障害物や壁までの距離と角度情報を取得する。取得するデータは -120° ~ $+120^{\circ}$ の範囲で 5° 間隔 (49 点) とした。「行動決定」は 10Step 毎に行い、行動の種類は「前進」、「後進」、「右回転」、「左回転」の 4 種類とした。また、障害物や壁を避け、走行路に沿って最短で走行できるように、報酬の設定値は図 6 のようにした。障害物や壁に接触した場合にはマイナスの報酬を与え、走行路に沿って設置した報酬ラインを通過した場合にはプラスの報酬を与える。「ポリシー更新」は、20480Step の経験を蓄積した後に行う。また、強化学習アルゴリズムには PPO を使用し、200 万 Step まで学習を行う。学習に使用した PC のスペックを図 7 に示す。

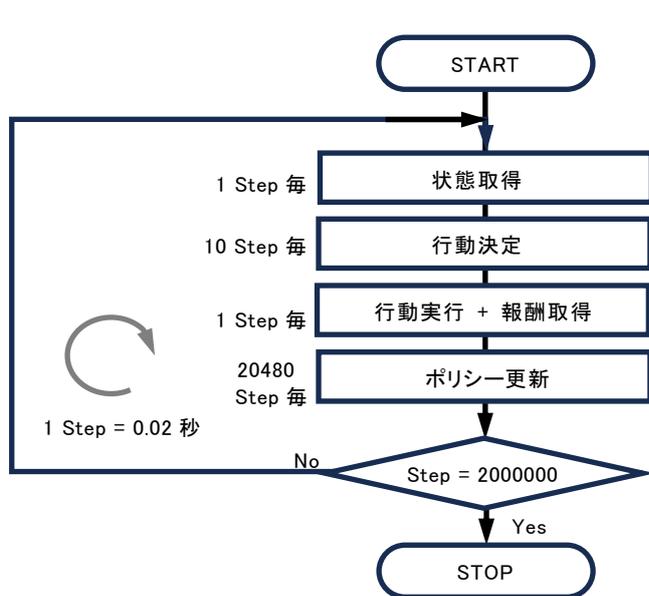


図 5 プログラムフローチャート

報酬設定	
壁や障害物に接触	-0.1
1 Step	-0.001
報酬ラインを通過	+0.1

図 6 報酬設定

PC スペック	
CPU	Intel Core i9 12900K
Memory	64 GB
GPU	GeForce RTX 3090

図 7 PC スペック

2.5 深層強化学習の実施と結果

強化学習では、学習が収束せず安定しないことも多い。ここでは、3つのコース (Type A,B,C) を同時に並行して学習を行った。これにより、カリキュラム学習と並列学習の効果が得られる。図 8 に深層強化学習の推移を示す。横軸は Step 数、縦軸は平均報酬である。灰色のグラフデータが実際の学習推移であるが傾向が分かりづらいため移動平均を取ったもの (黒色のグラフデータ) を重ねて示した。110 万 Step を超えたあたりで学習は収束しており、安定した学習結果が得られた。ここで得られた学習結果 (学習モデル) を実環境で行う自律走行制御実験で使用する。

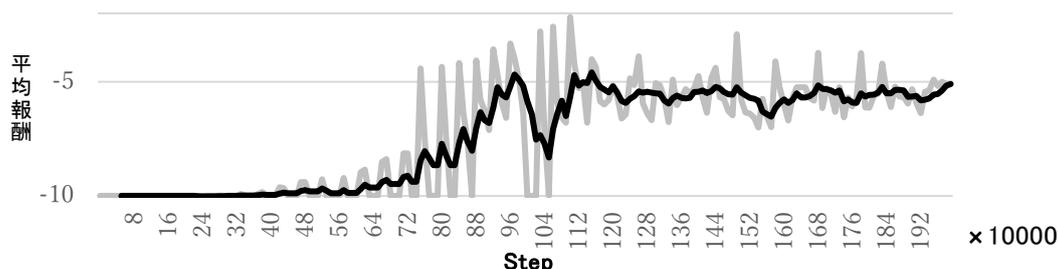


図 8 深層強化学習の推移

3. 深層強化学習を用いた自律走行制御実験

3.1 搬送ロボットの制御方法

搬送ロボットにはヴィストン(株)社製のメガローバーに LRF オプション(YDLiDAR TG30)、及び ROSPC (オプション) を搭載したものをを使用した。メガローバー本体 (以下、本体)、YDLiDAR TG30 (以下 TG30) は共に ROS (Robot Operation System) のトピック通信で制御可能である。本体は、トピック名「/rover_twist」

(geometry_msgs/Twist 型) で走行指示 (前進、後進、右回転、左回転) をサブスクライブする。TG30 は、トピック

名「/scan」で障害物までの距離、方向などの LiDAR 情報をパブリッシュする。本体には ROSPC が搭載されているため搬送ロボット単独で自律走行可能であるが、走行時の状態確認とパラメータ調整を容易にするため、ROS をセットアップした Note PC を用意し図 9 のような形で「LiDAR 情報」や「走行指示」をトピック通信でやり取りする形を取った。深層強化学習で得られた学習モデルを使用して Note PC に AI システムを構築し、学習モデルに「LiDAR 情報」を入力すると学習モデルからは「走行指示」が出力される。出力された「走行指示」をトピック名「/rover_twist」でパブリッシュすることで搬送ロボットを走行制御できる。

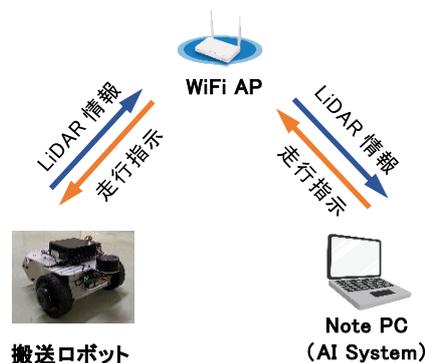


図 9 搬送ロボットの制御

3.2 自律走行実験と結果

図 10 に自律走行実験の様子を示す。シミュレータ環境と実環境の違いから、センサデータ、モーター駆動等に関してパラメータ調整を行ったが、おおむね走行路に沿って障害物を避けながら走行できることを確認した。



図 10 自律走行制御実験の様子

4. まとめ

搬送ロボットが自律走行できるようになるまでに様々な試行錯誤と調整を繰り返したが、研究を通して得られた多くのノウハウを今後の工業教育に活かしていく予定である。また、十分なカメラ性能が得られなかったため、LiDAR センサの情報のみで自律走行制御を行った。カメラ画像を用いた自律走行制御については、今後の課題である。

謝辞

本研究は、公益財団法人天野工業技術研究所工業教育研究助成金の支援を受けたものです。ここに感謝申し上げます。

参考文献

- 1)Unity ではじめる ROS・人工知能ロボットプログラミング実践入門 布留川英一 著 (株)ポーンデジタル
- 2)作りながら学ぶ! 深層強化学習 PyTorch による実践プログラミング 小川雄太郎 著 (株)マイナビ出版