

# 強化学習による仮想環境と実環境における自動走行車いすの 障害物回避

坂田 悠馬<sup>†</sup> 長尾 確<sup>‡</sup>

名古屋大学 大学院情報学研究科<sup>†‡</sup>

## 1. はじめに

人よりも密接な移動手段の一つとして、パーソナルモビリティに対する需要が近年高まっている。また、パーソナルモビリティに自動運転技術を組み合わせることで、より安全で快適な移動が実現可能となる。しかし、動的に変化する環境を走行する際には、事前に取得した情報には含まれない障害物があり、危害が加わらないように回避しなければならない。そのため一般の車よりも精密な制御が必要となる。

そこで、本研究ではパーソナルモビリティのシミュレーション環境を作成し、強化学習により障害物回避行動を獲得する。また、構築した学習モデルを実機に適用し、動作を評価する。本研究ではパーソナルモビリティとして、当研究室で研究開発している自動走行車いす[1]を対象とする。

## 2. 自動走行車いすの構成

図1に本研究で使用する自動走行車いすを示す。自動走行車いすのベースとして WHILL 社の電動車いす研究開発モデルである WHILL Model CR を用いている。電動車いすの上部には、全方位3次元 LiDAR である Velodyne 社の VLP-16 を設置している。下部前方には2次元 LiDAR である SLAMTEC 社の RPLIDAR S1 を配置している。また、座席下部には NVIDIA 社の Jetson AGX Xavier を2台搭載している。1台で自動走行の処理を行い、もう1台で障害物認識を行う。

自動走行のプログラムとして、Autoware[2]と呼ばれる ROS (Robot Operating System) をベースとしたオープンソースソフトウェアを利用している。自動走行を行う際は、VLP-16 から送られる3次元点群情報と WHILL から送られるオドメトリ情報を用いて自己位置推定を行う。

障害物認識には RPLIDAR S1 から送られる2次元スキャンデータを用いて行う。障害物回避では強化学習で獲得したモデルを用いて行動できるようにする。強化学習については4節で述べる。

自動走行や障害物回避などにおける速度情報

Obstacle Avoidance of Autonomous Wheelchairs in Virtual and Real Environments Based on Reinforcement Learning

<sup>†</sup>SAKATA, Yuma (sakata@nagao.nuie.nagoya-u.ac.jp)

<sup>‡</sup>NAGAO, Katashi (nagao@i.nagoya-u.ac.jp)

<sup>†‡</sup>Graduate School of Informatics, Nagoya University

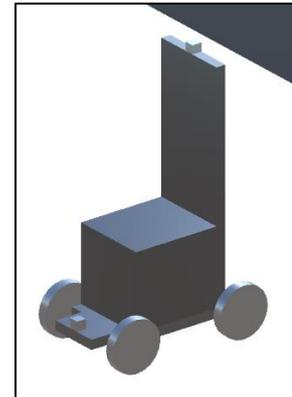


図1 自動走行車いす(左図)

図2 車いすの3Dモデル(右図)

は、優先度付き速度情報として車いすに送信する。優先度を考慮することで、緊急時のみ動作に介入するなど、車いすの動作プログラムの切り替えを容易に行うことができる。

## 3. シミュレーション環境の作成

実世界を模した環境で自動走行車いすを走行させ、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を作成する。

シミュレーション環境の基盤には3Dゲームエンジンである Unity を使用する。Unity は物理エンジンを搭載しており、またシミュレーション環境の作成を容易に行うことができる。これにより、実世界の複雑な環境も再現することが可能である。

シミュレーションと実機を対応しやすくするため、Unity が公式で配布しているパッケージである Unity Robotics Hub[3]を使用し、Unity で ROS のトピックが使用できるようにする。

シミュレーション環境で自動走行車いすを走らせるために、車いすの3Dモデルを作成する。大きさや重量に関しては実機のものと同様にしている。図2に車いすの3Dモデルを示す。

車いすモデルには障害物認識における環境情報を取得するために実機と同様の位置に2DのLiDARが搭載されている。このLiDARを用いて周囲の環境情報を取得する。

車いすモデルは ROS のトピック通信で速度情報を送信すると Unity 上でその速度に合わせて走

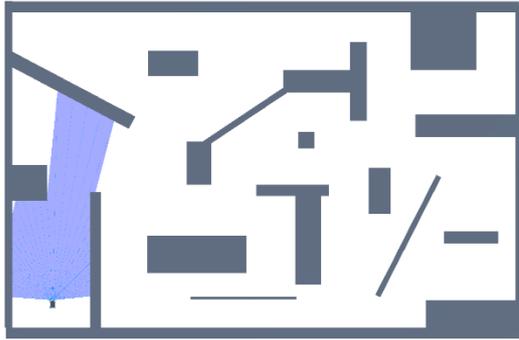


図3 障害物回避のシミュレーション環境

行する。

#### 4. シミュレーション環境での強化学習

作成したシミュレーション環境で障害物回避のための強化学習を行う。

##### 4.1. 強化学習環境の作成

ROS で強化学習を行うためのライブラリとして `openai_ros`[4]を使用する。ROS は非同期通信により情報交換を行うが、行動選択や学習は同期的に行われる。`openai_ros`は非同期なROSの情報を統合して強化学習を実装しやすいプログラム群となっている。Unityのシミュレーション環境とROS通信を行い、観測や行動に関する情報をやりとりする。

Unityではシミュレーション環境に関する処理を行う。車いすモデルには直方体の当たり判定を用意し、障害物が当たり判定内に入ったら衝突したことを`openai_ros`に送信する。強化学習の各エピソードの開始時にはUnityが車いすの初期位置や目標地点をランダムに設定する。

##### 4.2. 強化学習を用いた障害物回避の獲得

強化学習のアルゴリズムとしてDDPG (Deep Deterministic Policy Gradient)[5]を採用する。DDPGは強化学習の中でもActor-Criticと呼ばれる手法の一つで、行動の選択と行動に対する価値を別々のニューラルネットワークで学習する。Q値などの価値を行動ごとに出力する強化学習では、行動パターンが出力ベクトルの次元数と一致し離散的になるが、DDPGは行動を直接出力できるため、行動空間を連続的にすることが可能である。図3に強化学習で使用した環境を示す。

まず、シンプルな障害物のある環境で、障害物を避けるだけの強化学習を行う。入力となる観測値は、2D LiDARから得られるスキャンデータを40次元のベクトルにしたものである。前進や回転しないといった情報にステップごとの報酬を与え、障害物に接触したタイミングでリセットする。

3000エピソードほど学習することで、障害物を回避し前進するモデルを作成できた。



図4 名大モデルのシミュレーション環境

次に、自動走行における目的地を想定して、Unity環境上にゴールを設定し、ゴールに向かうようなタスクを行う。観測値には2D LiDARに加えてゴールに対する相対的な角度を与え、ゴールに到着したときにはプラスの報酬を与える。

#### 5. 実機での障害物回避実験

学習済みモデルを用いて実機に適用し、実機でも障害物回避ができることを確認する。

その前段階として、実世界を模したシミュレーション環境で強化学習が正しく動くことを確認する。環境として名古屋大学の3Dモデルを使用した。図4にその3Dモデル上で車いすモデルを動かした様子を示す。

その後、強化学習プログラムを実機に移行し、障害物回避を行う。RPLIDARの値を取得して、一定値よりも近ければ障害物が近くにあると認識し、自動走行動作から障害物回避動作への切り替えを行う。障害物が周囲から除外されたことを確認できた場合に自動走行動作へと戻る。

#### 6. まとめと今後の課題

本研究では、自動走行車いすを作成し、さらに車いすの動作を確認できるシミュレーション環境を作成した。シミュレーション環境上で障害物回避の強化学習を行い、実機に適用した。

今後の課題として、移動障害物やより複雑な地形などの難易度の高い環境下で障害物回避ができるようになる必要がある。また、今回はルールベースを用いて自動走行と障害物回避の切り替えを行ったが、この切り替えについても学習ベースで行うことができれば、複雑な環境下でも柔軟に対応可能だと考えられる。

#### 参考文献

- [1] Y. Mori and K. Nagao, "Automatic Generation of Multidestination Routes for Autonomous Wheelchairs," *Journal of Robotics and Mechatronics*, Vol. 32, No. 6, pp. 1121-1136, 2020.
- [2] Autoware, <https://www.autoware.org/>
- [3] Unity Robotics Hub, <https://github.com/Unity-Technologies/Unity-Robotics-Hub>
- [4] ROS.org, `openai_ros`, [http://wiki.ros.org/openai\\_ros](http://wiki.ros.org/openai_ros)
- [5] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, et al., "Continuous control with deep reinforcement learning," *ICLR*, 2016