強化学習を用いた適応的サブサンプションアーキテクチャとその障害物回避への応用

○長尾　確，森　優介（名古屋大学情報学研究科），坂田 悠馬（名古屋大学工学部）

Adaptive Subsumption Architecture Based on Reinforcement Learning and its Application to Robust Obstacle Avoidance

○Katashi NAGAO, Yusuke MORI (Graduate School of Informatics, Nagoya University), and Yuma SAKATA (School of Engineering, Nagoya University)

Abstract: In automatic driving of electric wheelchairs, map generation, self-position estimation, route generation, and route following using 3D LiDAR are realized, but robust obstacle avoidance remains almost unsolved. Therefore, in this research, a flexible obstacle avoidance mechanism is realized based on an adaptive subsumption architecture, that is, a subsumption architecture in which the hierarchical relationship of tasks changes depending on the situation. Tasks are as follows: (1) Avoid an obstacle to the right. (2) Avoid an obstacle on the left. (3) Move towards a point on the route. (3) Move towards the target point, etc. Situation dependence is acquired by reinforcement learning. This is a mechanism for determining which task is to be prioritized in a certain situation based on a reward calculated from a distance from an obstacle or an automatic travel route or a movement distance. We will also report on some mechanisms of an automatic wheelchair that implements this architecture and autonomous driving techniques for Tsukuba Challenge 2019.

1．はじめに

電動車いすの自動走行において，3D LiDAR を用いた地図生成・自己位置推定・経路生成・経路追従が実現されているが，柔軟で頑健な障害物回避に関してはほぼ未解決のままである．そこで、本研究では，適応的サブサンプションアーキテクチャ，つまり，タスク（障害物を右に回避する，障害物を左に回避する，経路上の点に向かって進む，目標点に向かって進む，など）の階層関係が状況に依存して変化するサブサンプションアーキテクチャに基づいて頑健な障害物回避の仕組みを実現する．状況依存性は，強化学習によって獲得する．これは，ある状況においてどのタスクを優先するかについて，障害物や自動走行経路との距離や移動距離から計算される報酬に基づいて決定する仕組みである．つくばチャレンジ 2019 のために，このアーキテクチャおよび 3D LiDAR に基づく自動走行手法を実装した車いすの各種メカニズムについても説明する．

2．自動走行車いす

2.1 構成

図 1 に自動走行可能な電動車いすの構成を示す．

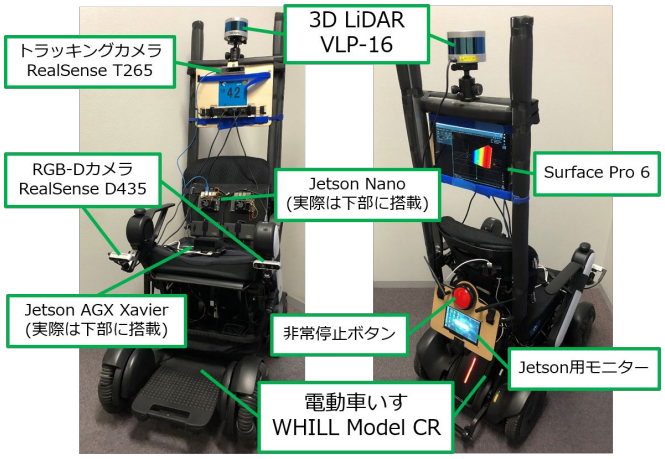

図 1　自動走行車いす

基盤となる電動車いすには WHILL 社の研究開発モデル WHILL Model CR を用いた．これは，外部機器から入力信号を RS232C を介して送信することで，本体を制御することができ，また，本体の情報（速度，加減速値，エンコーダー情報，加速度センサ値，コントローラ入力情報，バッテリー情報など）を取得することができる．

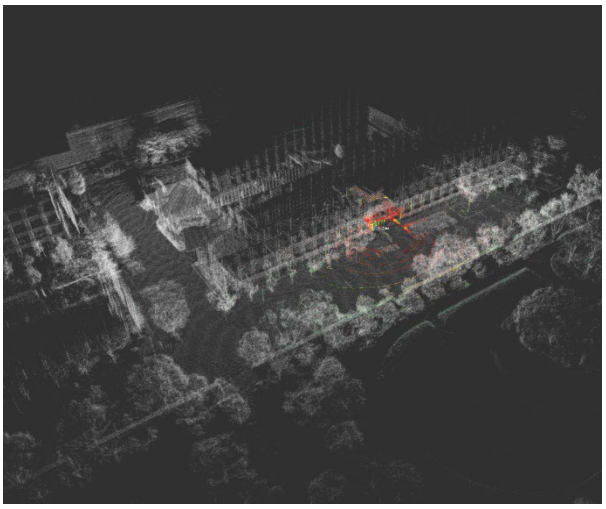

図 2　生成された 3 次元地図の例

電動車いすの上部には，全方位レーザーLiDAR イメージングユニットである Velodyne 社の VLP-16 を固定してる. 車いす下部には, 組み込み PC である NVIDIA 社の Jetson AGX Xavier を搭載している．LiDAR から得られる周囲のデータを Jetson で処理することで自己位置を推定する．Jetson で行う処理には Autoware [1]と呼ばれる，名古屋大学を中心に開発され，自動運転の研究開発用途に公開されている Linux と ROS (Robot Operating System)をベースとした自動走行システム用オープンソースソフトウェアを利用している．また，Jetson から WHILL に対して制御信号を送信することで自動走行を実現する．この自動走行に関する詳細は 2.2 項で説明する.

また，電動車いすには前方の障害物を検出するために RGB-D カメラ RealSense D435 を 2 台搭載している．1 台は前方を向き，もう 1 台は前方から若干下を向いている．2 台とも 1 台の PC（Microsoft Surface Pro）に接続して稼働している．そのうちの前方を向いている 1 台のカメラの深度画像を用いて，前方の障害物の発見および発見した障害物との距離を計算する．もう 1 台の斜め下向きのカメラでは，主に画像処理を行い，路面上の白線を認識する．障害物および白線の認識には機械学習を用いる．この機械学習については 2.3 項で説明する.

2 台の RGB-D カメラによって，LiDAR の死角となる車いすの近距離部分を補い，前方の障害物を検知した場合はそれとの距離に応じて減速あるいは停止する．これによって，自動走行時の安全性を考慮している.

2.2 自動走行システム

指定した目的地に対して自動で移動する仕組みは，移動ロボットの基本的な機能として数多く研究されており，次の 4 つのステップの実現を主な課題としている[2].

1. 環境地図生成
2. 自己位置推定
3. 経路生成
4. 経路追従のための駆動系の制御

前述の Autoware には，これらの 4 ステップのそれぞれを実現するための機能が用意されている.

環境地図生成に関しては，NDT (Normal Distributions Transform)スキャンマッチングにより自己位置推定を行い LiDAR の 3D スキャンデータを追加していくことで 3 次元地図を作成する．図 2 に生成された 3 次元地図の例を示す. 3 次元地図は自動走行を行う前に，手動で走行したデータを用いて作成する.

自己位置推定に関しては，1 で作成した 3 次元地図と LiDAR のスキャンデータから NDT スキャンマッチングにより自己位置推定を行う.

経路生成に関しては，Autoware では waypoint と呼ばれる位置・方向・速度の情報を持った点の一定間隔の離散的な集合で経路を表現しており，この waypoint を生成する機能が用意されている．waypoint の生成手法は大きく分けて 3 つあるが，内 2 つは車線や信号といった情報を持つベクターマップを用いて自動生成する方法で，自動車に特化した経路生成である．そのため，今回は 3 つ目の方法である実際に走行した経路から waypoint を生成する方法を用いる．その結果，図 3 のような経路が生成される.

経路追従のための駆動系の制御に関して，Autoware

図 3　waypoint に基づく経路の例

には経路に対して追従し，目標となる速度と角速度を出力する機能が存在する．これを用いて，目標速度と目標角速度から，WHILL の制御信号に変換することで経路に沿った自動走行を実現する．

屋内で，実際に自動走行の試験を行った．図 3 に試験走行の際の経路を示す．経路としては，部屋の中から走行を開始し，廊下に出て，建物中央のオープンスペースに向かい，その内部を通った後に元の部屋に戻るというものである．生成した経路に追従して自動走行ができることを確認した．また，進路上に人が現れた場合は，RGB-D カメラにより障害物を検出して，衝突する前に自動で停止できることを確認した．

2.3 機械学習に基づく障害物認識

進路上の障害物や路面に引かれた白線を認識することは，自動走行において重要であり，地図に基づく自己位置推定や経路追従とは独立に解決しなければならない問題である．そこで，自動走行のモジュールと並列に認識処理を行い，白線の場合は減速・停止のコマンドを自動走行モジュールに送信する．また，障害物の場合は，後述する障害物回避モジュールに制御権を委譲する．

障害物に関しては，深度から最も接近した物体との距離を計算して閾値処理をするモジュールと，画像から物体認識をして障害物の種類に応じた処理をするモジュールが並列に稼働している．物体認識には，既存の学習済みディープラーニングモデルを利用して，収集した画像データを使って転移学習したモデルを用いている．

物体認識は障害物が自律移動するもの（人，動物，移動ロボットなど）とそれ以外のものを識別する．二つのモジュールの出力は統合され，自動走行モジュールへのコマンドを出力する．

白線の認識に関しても，障害物と同様に，白線の写っている画像とそうでない画像を用いて既存のニューラルネットワーク(Alexnet)から転移学習したものを用いている．この場合の白線には，直線と T 字型の 2 種類があり，直線に関して 328 枚，T 字に関して 708 枚の画像を用意した．さらに白線の写っていない画像を 890 枚用意して学習を行った．

転移学習では Alexnet の最終層の出力を 1000 から 3 に変更し 30 エポックで学習させた．オプティマイザは SGD を使用し，損失関数はクロスエントロピーを使用した．全画像中 300 枚をバリデーションデータとして使用したところ，約 8 割の精度が得られた．

白線および障害物の認識モジュールは，自己位置推定・経路追従モジュールと並列に稼働しており，白線に関しては停止処理を行い，障害物に関しては，次節で説明する障害物回避モジュールに制御を切り替える．

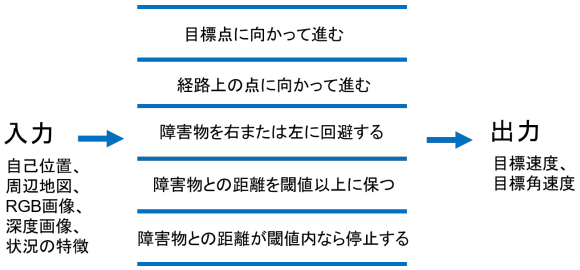

図 4　サブサンプションアーキテクチャの例

3．適応的サブサンプションアーキテクチャ

サブサンプションアーキテクチャは，MIT のロドニー・ブルックス（Brooks, R.）が提案した知能ロボットのためのアーキテクチャである[3]．図 4 に示すようにロボットの行動を複数の層に分けて記述し，必要な場合には上位層が下位層の機能を抑制（subsume）するためサブサンプションアーキテクチャと呼ばれる．

それ以前のアーキテクチャでは，観測-認識-計画-動作というように入力から出力までを直列につないだものであったが，これでは外界の突発的な変化に速やかに反応できない．脊髄反射のような反射行動と，計画策定による熟考行動を同じループで行うのは間違っているとし，それぞれの機能を並列に実装し，動作させるのがこのアーキテクチャの考え方である．

この仕組みの問題点は，各層として表現された複数のタスクの処理メカニズムの優先度を階層関係によって固定してしまう点である．そこで，この階層関係を動的に変更可能にする．ただし，すべてのセンサー情報を基づいて，タスクの関係をルール的に決定するのでは，サブサンプションアーキテクチャのボトムアップ処理による即応性を損ねてしまう可能性がある．それを防ぐために，走行経路をその特徴（道幅や周辺の環境など）に基づいていくつかのパターンに分割し，パターンごとにタスク間の階層関係をあらかじめ決定しておく．走行経路の各パターン（これを状況と呼ぶ）を位置情報から判定し，そのときの最適な階層関係を，その部分の走行の直前に確定する．確定後は，通常のサブサンプションアーキテクチャと同様に機能する．この最適な階層関係を戦略と呼び，強化学習によって決定する．

サブサンプションアーキテクチャの実装には ROS を用いる．タスク（あるいは行動）は ROS のノードとして実装され，固有の ID を持つ．階層関係は，どのノードがどのノードを抑制するかを記述した行列（サブサ

ンプション行列と呼ぶ）として表現される．この行列はトピックとして常に送信され，タスク間の関係がリアルタイムに調整される．

この場合のタスク（行動）は以下の通りである．

- 障害物を右に回避する
- 障害物を左に回避する
- 障害物との距離が閾値内なら停止する
- 障害物との距離を閾値以上に保つ
- 経路上の点に向かって進む
- 目標点に向かって進む

初期値としては「障害物との距離が閾値内なら停止する」という行動を最下層とするが，それを抑制する行動は存在しない（すべてのサブサンプション行列値が 0 となっている）．しかし．そのままだと障害物が進路上にあると常に停止してしまう．そこで，状況ごとに最適なサブサンプション行列を設定して，柔軟に対処する．次に述べる強化学習は，状況を入力としてサブサンプション行列を出力する学習モデルの構築を目的とする．

3.1 強化学習

近年，機械学習技術が発展し，コンピュータビジョンや自然言語処理などの分野においてさまざまな研究成果が挙げられている．その中で、強化学習を用いたロボットの制御に関する研究が活発に行われている．強化学習では，目標とするロボットの最適な行動列（計画）を正解として与える代わりに，ロボットの局所的な各行動に対して報酬を与える．ロボットはどのように行動するとどれくらいの報酬が得られそうかを学習していき，最大の報酬が得られそうな行動を選択することで，結果的に最適な行動をとる戦略が学習できる．

強化学習に関する研究は，主にゲーム環境やシミュレーション環境で稼働するものであり，実世界の複雑な環境に対する解析が少なく，OpenAI gym [4]で公開された訓練環境で訓練してテストし，既存手法と比較することが多く，本研究の課題とのギャップが大きい．そのため，できるだけ実世界の環境を用いてロボットを学習させることが望ましい．

強化学習モデルの訓練のためには，大量なサンプリングデータが必要であり，ロボットが行動する環境において繰り返して試行錯誤することが必要である．しかしながら，それを実現するために，ロボットを実環境において動作させ，さまざまな障害物を認識・回避した行動データを収集することは非常に困難であり，多大なコストがかかる．この問題を解決する，つまり学習訓練を効率的に行うために，仮想環境を用いて強化学習の学習モデルを訓練するのが，一般的である．そこで，本研究は 3D LiDAR で計測した 3 次元点群データを仮想環境に取り込み，さまざまな形状の障害物をさまざまな位置に配置して強化学習のシミュレーション環境とし，仮想的なセンサーを備えた仮想的なロボットによって訓練用データを生成して学習を行った．

仮想環境ではセンサーをシミュレートし，観測信号を出力するだけでなく，報酬信号，終了信号を学習モデルに与える必要がある．本研究で設計した報酬は移動距離による報酬 Rm，移動時間による報酬 Rt，障害物との距離による報酬 Ro，目標点との距離による報酬 Rd，そして，経路上の点との距離による報酬 Rr から構成される．

- 移動距離による報酬 Rm

Rm の値はロボットのある行動によって，どの程度移動できたかに依存する．仮想環境は，各ステップのロボットの移動を記録し，ステップ開始から終了までの移動距離を積算する．そして，前回のステップより増加した移動距離に応じたプラスの報酬を与える．

- 移動時間による報酬 Rt

Rt の値はロボットのある行動が，どの程度時間がかかったかに依存する．仮想環境は，各ステップのロボットの移動にかかる時間を記録し，ステップ開始から終了までの時間を計算する．そして，前回のステップより増加した移動時間に応じたマイナスの報酬を与える．

- 障害物との距離による報酬 Ro

ロボットがある閾値より少ない距離で障害物に接近した場合，マイナスの報酬が与えられる．

- 目標点との距離による報酬 Rd

ロボットが前回のステップ終了時より目標点との距離が短くなったときにはプラスの報酬，長くなったときにはマイナスの報酬を与える．Rd の絶対値は目標点との距離の差分に比例する．

- 経路上の点との距離による報酬 Rr

ロボットが前回のステップ終了時より経路上の点（ただし前述の waypoint であればどの点でもよいので前回のステップと同じ点とは限らない）との距離が短くなったときにはプラスの報酬，長くなったときにはマイナスの報酬を与える．Rr の絶対値は経路上の点との距離の差分に比例する．

学習モデルがいくつかのステップを経て目標点（状況ごとに経路上の中継点を目標点に設定する）に到達し，終了信号が与えられると，終了して内部状態を初期化する．初期化から終了までの流れを 1 エピソードとして定義する．本研究では 1 エピソードを終了するパターンは以下の三つである．

- 回数による終了

実世界のロボットが稼働できる時間は限られているので，回数の上限を考慮する必要がある．本研究での訓練実験では回数の上限を 100 に設定し，1 エピソードで 100 ステップを超えると終了する．ちなみに 1 ステップの時間は 10 秒とする．

- 目標点到達

状況ごとに設定された目標点にロボットが到達するとエピソードが終了する．実際の環境においてはロボットが必ずしも完璧に目標点に到達できるわけではなく，ロボットの自己位置推定の精度によって正確に目標点に到達できない場合があるので，目標点とロボットの重心の距離が閾値以下になると到達とする．

- 障害物との衝突による終了

シミュレータでは，ロボットのセンサーにはガウス分布に基づく誤差を含ませている．そのため障害物との距離を正しく計測できず衝突してしまう場合がある．この場合，回避失敗として終了する．

構築した仮想環境を用いて学習モデルを訓練した．本研究で使用する学習モデルは，OpenAI 社が提案した PPO (Proximal Policy Optimization)手法[5]をベースにしたモデルである．PPO 手法は入力した状態に対して行動を出力し，最大の報酬を得られるために行動を選択するニューラルネットワーク Actor と，Actor のパフォーマンスを評価するニューラルネットワーク Critic を組み合わせて利用する．

４．つくばチャレンジに向けて

つくばチャレンジ 2019 では，走行エリアがいくつかのパターン（状況）に分割できる．具体的には，建物（つくば市役所庁舎）周辺，屋内（つくば市役所内），歩道，横断歩道，公園内である．このため，あらかじめ収集した 3D LiDAR のデータ，RGB 画像，深度画像を用いて，状況ごとの特徴を抽出した．具体的には道幅，経路の平均曲率，白線に遭遇する頻度，障害物に遭遇する頻度などである．

また 3D LiDAR で収集した点群データを利用してシミュレータを作成し，前述した強化学習の実験を行っている．

現在では障害物回避の成功率があまり高くなく，9 月 14 日に行われた確認走行（つくば市役所庁舎周辺の短距離の自動走行の達成度を確認する）では，1 回目の走行で他のロボットと接触し（実際には接触する直前に人間が非常停止ボタンを押して停止した）達成できなかった（2 回目には成功した）．

接触したロボットは小型だったため，検出漏れが発生し，減速・停止処理が間に合わなかった．この問題は，シミュレーションを繰り返して学習することで解決できると思われる．

また，今回のつくばチャレンジの課題には，直前に指定される公園内のチェックポイント（ただしその候補となる点は限定されている）を通過する，さらに，公園内に封鎖エリアを設け（その位置については事前に通知されない）それを迂回する，というものがあるため，動的な経路変更は不可欠である．そのため，公園内の全チェックポイント候補に関して，近傍の 2 つの点を結ぶ経路から成るグラフを構成し，そのエッジにその部分の経路の走行に関するコストを割り当てて，指定されたチェックポイントをすべて通過する（さらに封鎖エリアを迂回する）経路を最小コスト経路探索問題として解く仕組みも実装している．

５．おわりに

自動車のような専用の道路を高速で移動する移動体に比べて，車いすや移動ロボットのような低速で小型の移動体は，一般的な人間とほぼ同様の環境で活動する（歩道を走行するなど）必要がある．その場合，目的地に自律的に移動することだけでなく，人や障害物との衝突を避け，安全に移動するための課題がより複雑になる．本研究で提案する適応的サブサンプションアーキテクチャの仕組みはその問題を解決する有効な解決策の一つである．

つくばチャレンジは実世界で活動するロボットに関する最新の技術を実験する非常に有益な機会であり，すでに多くの実績がある．シミュレーションでは成功しても実際にはうまく機能しない場合は容易に想像できる．そのため，この機会を有効に活用したい．

本走行までにまだ時間がある（本稿の執筆時において）ため，提案手法の精度を向上させるためのさらなる改良を行っていく予定である．

謝辞

本研究の機会を与えてくださったつくばチャレンジ実行委員会のみなさまに感謝いたします．

参考文献

[1] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. “An Open Approach to Autonomous Vehicles,” IEEE Micro, Vol. 35, No. 6, pp. 60-69, (2015).

[2] S. Thrun, W. Burgard, and D. Fox, “Probabilistic Robotics,” The MIT Press, (2005).

[3] R. Brooks, “A robust layered control system for a mobile robot,” IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 14–23. doi:10.1109/JRA.1986.1087032 (1986).

[4] OpenAI. Accessed 2019-09-25, https://gym.openai.com/envs/ (2019).

[5] J. Schulman et al., “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, (2017).