# 動的環境における学習ベースおよびルールベースの切り替え手法を用いた移動ロボットナビゲーション 第二報 実機実装と実環境での動作実験

○兵頭 侑樹（九州大学） 松本 耕平（九州大学） 富田 湧（九州大学） 倉爪 亮（九州大学）

We have been studying about a method for mobile robot navigation in dynamic environments, including pedestrians, that switches between learning-based and rule-based methods. By introducing the switching mechanism, the proposed method can reach the destination safer than the learning-based method and faster than the rule-based method. In this paper, we developed an omni-directional robot and conducted real world experiments to verify the applicability of the proposed method.

## 1. 緒言

人々が生活する日常環境にロボットを安全に導入するには，歩行者が行き交う混雑した環境でもスムーズに移動できる移動ロボットナビゲーション手法の実現が不可欠である．近年では，深層強化学習を用いた移動ロボットナビゲーションについて盛んに研究が行われている．深層強化学習手法は学習済みの環境に対して，ルールベース手法より良いパフォーマンスを発揮する．しかし，想定していない環境に直面した場合の対処は難しい．また実環境で移動ロボットを動作させる場合，シミュレーション環境と実環境のギャップが存在し，学習ベースが想定していない環境での行動は歩行者との衝突のリスクがあり，未だ課題が残る．我々は第一報 [1] として，状況に応じて学習ベースの行動とルールベースの行動を切り替える手法を提案し，想定していない環境に対しても柔軟に対応できることをシミュレーション環境上で確認した．具体的には，限られたデータセットをオフラインで強化学習させた学習ベース手法と衝突回避アルゴリズムからなるルールベース手法を，グラフ正規化フローを用いた尤度推定を基に切り替える手法を提案した．本稿では，移動ロボットの実機実装と実環境での実験を行い，提案した切り替えによるナビゲーション手法の実環境への適用可能性を検証する．

## 2. 問題設定

本研究では，X-Y 平面においてロボットが周囲の歩行者を回避ながら目的地に到達する問題を考える．環境の状態，行動，報酬は以下のように設定する．

- 状態：ロボットの状態は $\boldsymbol{s}^r = [|\boldsymbol{p}^g - \boldsymbol{p}^r|, \boldsymbol{v}^r, \theta]$ と表し，それぞれロボットから目的地までの距離，ロボットの速度，進行方向とする．対して，歩行者の状態は $\boldsymbol{s}^n = [{}^{rc}\boldsymbol{p}^n, {}^{rc}\boldsymbol{v}^n]$ であり，歩行者の位置と速度を表すが，それぞれの歩行者の位置，速度はロボット中心の座標系に変換される．観測はベクトル $(p_x^r, p_y^r, v_x^r, v_y^r, g_x, g_y, p_x^n, p_y^n)$ であり，$(p_x^r, p_y^r)$ はロボットの位置，$(v_x^r, v_y^r)$ はロボットの速度，$(g_x, g_y)$ は目的地の位置，また $i$ 番目の歩行者に対して，$(p_x^i, p_y^i)$ は歩行者の位置，$(v_x^i, v_y^i)$ は歩行者の速度を表している．
- 行動：本研究では，ホロノミックな全方位移動ロボットを想定し，2 次元空間におけるロボットの x 軸方向の入力速度 $v_x$ と y 軸方向の入力速度 $v_y$ からなる 2 次元ベクトル $(v_x, v_y)$ を用いる．
- 報酬：報酬には式 (1) を用いる．$d_t$ は時刻 $t$ におけるロボットと周囲の歩行者間の最小距離を表し，$p_t^r$ は時刻 $t$ におけるロボットの位置，$p_g$ はロボットの目標位置を表す．

$$R_t = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \boldsymbol{p}_t^r = \boldsymbol{p}_t^g \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

## 3. 学習ベースとルールベースの切り替えを用いたナビゲーション

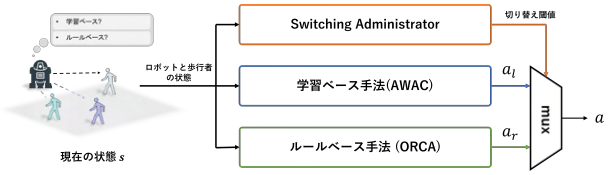


図 1 切り替え手法の概念図

学習ベースとルールベースの切り替えを用いたナビゲーション手法の概念図を図 1 に示す．以降この手法は切り替え手法とする．ロボットと歩行者の情報を学習ベース手法およびルールベース手法のモデルに入力すると，入力速度がそれぞれのモデルから出力される．これらの速度は Switching Administrator からの出力よって切り替えられる．ルールベース手法には，エージェント間の衝突回避手法として広く用いられる ORCA [2] を利用する．学習ベース手法にはオフライン強化学習に基づいた AWAC [3] によって学習された方策を利用する．AWAC は重み付き回帰学習により学習ベースで行動生成を行うナビゲーション手法であり，他の強化学習の手法に比べてデータセットに近い行動を学習しやすいために用いている．各手法の切り替え機構の役割を担う Switching Administrator の基本構造として，グラフ正規化フローを利用する．

### 3.1 Switching Administrator

#### 3.1.1 アーキテクチャ

Switching Administrator の構造を図 2 に示す．Switching Administrator は正規化フローに基づいた

構造を持ち，歩行者の状態を MLP によりエンコードした後，正規化フローに入力され，尤度が計算される．正規化フローの構造としてはグラフ正規化フロー [4] を用いる．グラフ正規化フローは RealNVP [5] をベースとしており，affine coupling layer [5] 内の非線形関数を，グラフニューラルネットワークで置き換えることで実現される．このグラフニューラルネットワークには Graph Attention Network (GATv2) [6] を利用する．また，バイナリマスクを利用したパーティションを採用しており，Switching Administrator 内のフローブロックの変換は以下のように表すことができる．

$$\bar{\boldsymbol{h}}^n = \boldsymbol{b} \odot \boldsymbol{h}^n + (1 - \boldsymbol{b}) \odot \left(\boldsymbol{h}^n \odot \exp(f_g^s(\boldsymbol{b} \odot \boldsymbol{h}^n)) + f_g^t(\boldsymbol{b} \odot \boldsymbol{h}^n)\right) \quad (2)$$

$$\hat{\boldsymbol{h}}^n = \tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n + (1 - \tilde{\boldsymbol{b}}) \odot \left(\bar{\boldsymbol{h}}^n \odot \exp(f_g^s(\tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n)) + f_g^t(\tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n)\right). \quad (3)$$

ここで，$\boldsymbol{h}^n \in \mathbb{R}^d$ は MLP によりエンコードされた歩行者の特徴量である．また，$\boldsymbol{b}$ はバイナリマスクであり，$\tilde{\boldsymbol{b}} = 1 - \boldsymbol{b}$ である．$f_g^s$ と $f_g^t$ はそれぞれ GATv2 による関数である．$f_g^s$ と $f_g^t$ の計算は以下の式で表される．

$$f_g(\boldsymbol{h}^i) = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}^j. \quad (4)$$

ここで，$\boldsymbol{W} \in \mathbb{R}^{d \times d}$ は重みであり，$\mathcal{N}_i$ は歩行者の $i$ 番目の隣接の集合である．アテンション係数 $\alpha_{ij}$ は以下の式で計算される．

$$e\left(\boldsymbol{h}^i, \boldsymbol{h}^j\right) = \boldsymbol{a}_w^\top \mathrm{LeakyReLU}\left(\boldsymbol{W}_l \boldsymbol{h}^i + \boldsymbol{W}_r \boldsymbol{h}^j\right) \quad (5)$$

$$\alpha_{ij} = \mathrm{softmax}_j\left(e\left(\boldsymbol{h}^i, \boldsymbol{h}^j\right)\right). \quad (6)$$

ここで，$\boldsymbol{a}_w \in \mathbb{R}^d$, $\boldsymbol{W}_l \in \mathbb{R}^{d \times d}$, and $\boldsymbol{W}_r \in \mathbb{R}^{d \times d}$ はいずれも学習する時の重みである．

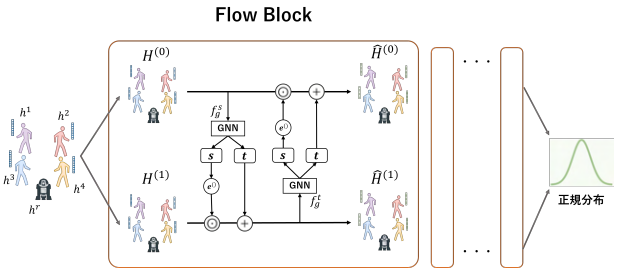


図 2　Switching Administrator の構造

### 3.1.2　切り替え方法

図 3 は切り替えを含めた実行のフローチャートを表している．学習ベースとルールベースの切り替えは Switching Administrator によって計算される尤度が，閾値 $\theta_{th}$ を下回るときに行われる．1 ステップごとに出力される行動は以下のように表される．

$$\boldsymbol{a}_t = \begin{cases} \boldsymbol{a}_t^r & \text{for } \tau(\boldsymbol{s}_t^h) > \theta_{th} \\ \boldsymbol{a}_t^l & \text{for } \tau(\boldsymbol{s}_t^h) \leq \theta_{th} \end{cases}. \quad (7)$$

ここで，$\boldsymbol{a}_t^r$ と $\boldsymbol{a}_t^l$ はそれぞれ時刻 t においてルールベースと学習ベースから生成される行動である．スイッチングスコア $\tau(\boldsymbol{s}^h)$ は入力状態の尤度であり，以下のように計算される．

$$\tau(\boldsymbol{s}^h) = -\log p_Z\left(f_{\mathrm{sa}}(\boldsymbol{s}^h)\right). \quad (8)$$

この閾値は，学習後の Switching Administrator により算出した，学習データの尤度の平均値を利用する．

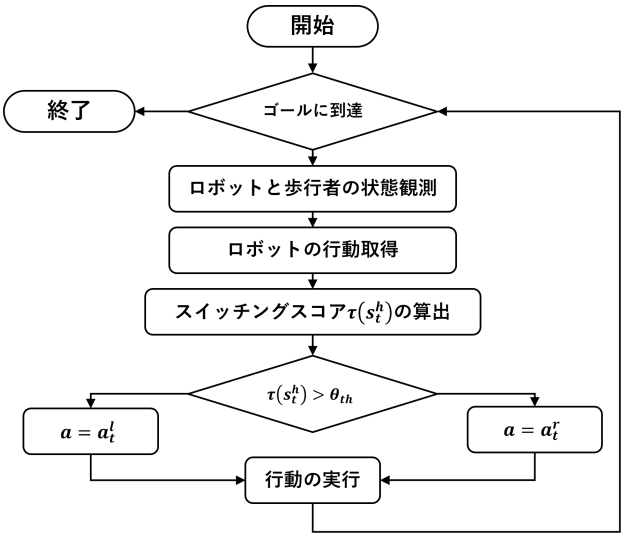


図 3　実行のフローチャート

## 4.　実機実装

### 4.1　実機のハードウェア構成

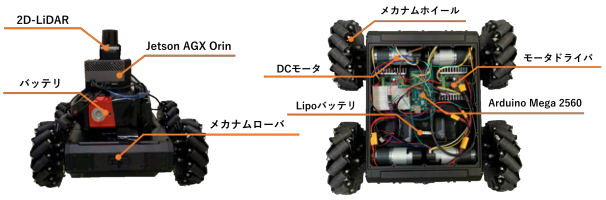


図 4　開発した実機のハードウェア構成

本研究で開発した実機のハードウェア構成を図 4 に示す．移動台車部分は Lynxmotion 社製のメカナムローバを用いており，全方向に移動可能な 4 輪のメカナムホイールとアルミニウム合金のフレームからなる．車輪としてメカナムホイールを用いているのは，本研究におけるシミュレーション環境で全方位に移動可能なロボットを想定しているためである．また，センサーとして 2D-LiDAR (UST-20LX) を用いている．本実験で，歩行者とロボットが衝突したと判定する際にロボットから近い距離にいる歩行者を検知する必要があるため，測距距離の最小値が小さい UST-20LX を用いている．制御用コンピュータとして Jetson AGX Orin を用いている．制御基板として Arduino Mega 2560，モータドライバとして Savertooth2x12，モータは直交エンコーダ付きの DC モータを用いている．

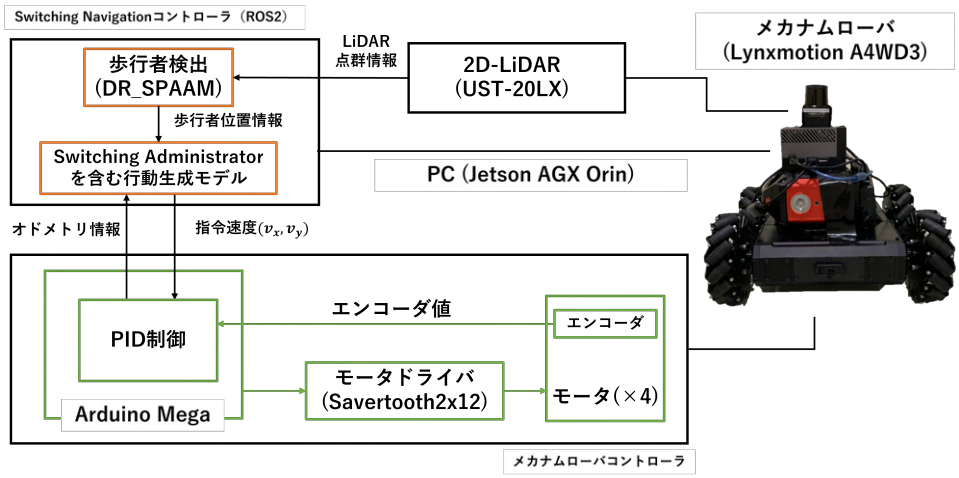


図 5　開発した実機のソフトウェア構成

### 4.2　実機のソフトウェア構成

実機のソフトウェア構成としては ROS2 Humble を用いた実装を行っており，歩行者検出と学習ベースとルールベースの切り替え手法による行動生成からなる Switching Navigation コントローラと，指令速度を基にモータをコントロールするメカナムローバコントローラからなる．まず LiDAR で読み取った点群情報を制御用コンピュータに送信する．制御用コンピュータでは，この LiDAR データを基にした歩行者位置推定と [1] の切り替え手法による実機の速度計算を行う．本研究の歩行者検出手法として，DR-SPAAM [7] を用いている．DR-SPAAM は歩行者検知のための深層学習モデルであり，2 次元のレーザスキャンデータを入力として，歩行者の位置を出力とする．検知した歩行者の位置情報を基に切り替え手法のモデルからロボット側に指令速度が送信される．実機の速度データはシリアル通信により Arduino Mega 2560 に送信される．Arduino Mega 2560 では，実機の速度データの受信，実機の速度データのモータの回転速度への変換，エンコーダの値を基にしたモータの PID 制御を行っている．また，Arduino Mega 2560 からはエンコーダの値を基にしたオドメトリ情報がフィードバックとして制御用コンピュータに送信される．

## 5.　実機実験

### 5.1　実験環境

実環境として 7.5m 四方の部屋を用意した．シミュレーション環境における circle-crossing シナリオを模して，5 人の歩行者が部屋の中央を交差するようにしてロボットが目的地まで移動する．切り替え手法，学習ベースのみ，ルールベースのみの 3 つの手法それぞれについて 25 回の実験を行った．

### 5.2　結果

図 6 は実機実験の様子を示している．この図においてロボットは [1] の切り替え手法を用いた動作を行っており，歩行者を回避しながらのナビゲーションを実環境において達成した．ここで表 1 に [1] のシミュレーションの結果である circle-crossing シナリオの数値評価を示す．この結果から，square-crossing シナリオ上で学習した学習ベース手法は circle-crossing シナリオでテストすると，環境の違いによる影響から成功率が下がっていることが分かる．切り替え手法は circle-crossing シナリオにおいて，学習ベース手法よりも高い成功率で，ルールベースよりも速く目的地に到達できていることから，切り替えによって，学習ベース手法が想定していない環境に対応できることが分かる．

表 1　シミュレーションによる数値評価 [1]

| 手法 | 成功率 [%] | 衝突率 [%] | 平均到達時間 [s] |
|---|---|---|---|
| ルールベース手法 | 100.0 | 0.0 | 10.02 |
| 学習ベース手法 | 63.8 | 36.2 | 8.70 |
| 切り替え手法 | 97.8 | 2.2 | 8.84 |

表 2 に実環境実験における，ルールベース手法，学習ベース手法，切り替え手法の 3 つの手法の数値評価を示す．各手法それぞれ 25 回の実験を行い，それぞれの成功回数はルールベース手法は 21 回，学習ベース手法は 4 回，切り替え手法は 21 回であった．この結果からシミュレーション環境と比較すると，実環境において学習ベースのみの手法は成功した回数が少なく，シミュレーション環境と実環境の差異の影響が表れたと考えられる．切り替え手法は学習ベースのみの手法に対して成功率が改善しており，切り替えを行うことで学習ベースのみでは対応できない場面に対処できていると考えられる．このことからシミュレーション環境での実験と同様に，実環境でも切り替え手法は有効であると考えられる．また実環境実験での平均到達時間は切り替え手法が 15.1s，学習ベース手法が 11.8s，ルールベース手法が 15.5s であった．切り替え手法はルールベース手法に比べてわずかに平均到達時間が速く，また学習ベース手法は他手法に比べて到達時間が速いことを確認した．今回の実機実験の切り替え手法において，ルールベースへの切り替えが多く発生しており，エピソード全体において，切り替えの割合は 87.8% であった，これはロボットが実環境のデータを学習しておらず，シミュレーションと実環境の差異が大きいことから現在の状態の尤度が閾値を下回ってしまい，ルールベース手法に切り替えたこ

とが原因と考えられる．また歩行者検出の際に誤って机や壁の角などを歩行者と誤認識した影響も原因の一つであると考えられる．

表 2　実環境実験における数値比較

| 手法 | 成功率 [%] | 衝突率 [%] | 平均到達時間 [s] |
|---|---|---|---|
| ルールベース手法 | 84.0 | 16.0 | 15.5 |
| 学習ベース手法 | 16.0 | 84.0 | 11.8 |
| 切り替え手法 | 84.0 | 16.0 | 15.1 |

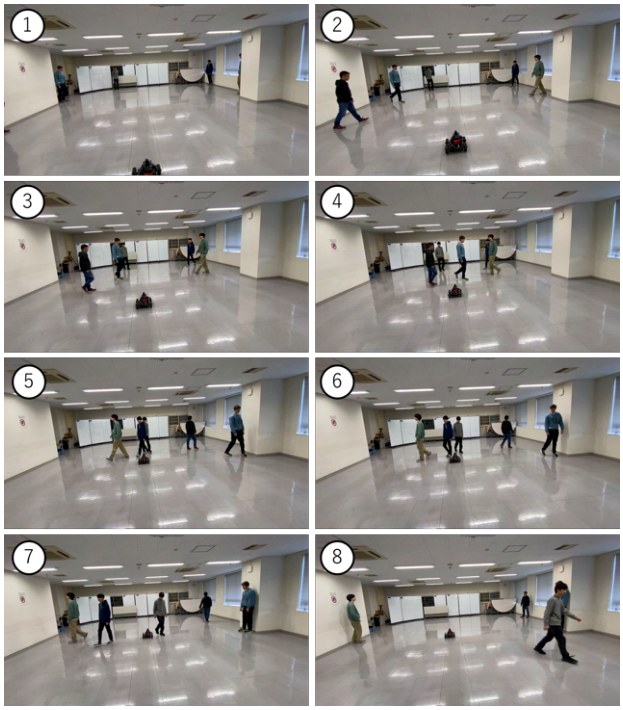


図 6　実環境での実機実験の様子

## 6.　まとめと今後の展望

本研究では，歩行者が行き交う動的な環境での移動ロボットナビゲーションにおいて，第一報で提案した学習ベースの行動とルールベースの行動を切り替える手法に対する実環境実験の詳細を示した．実機実験の結果から，切り替えを行うことによって学習ベースよりも安全に目的地に到達できることを確認した．しかしながらルールベースへの切り替え割合が多く，学習ベース手法のメリットである効率性を十分に活かすことができなかった．今後はこの問題点を解決するために，実環境でのデータ収集と追加の学習を行うことで，効率性も考慮したナビゲーションを実現したい．また，本研究の Switching Administrator は学習したデータセットに対する尤度を評価しているため，厳密には直面している状況を方策が対応できるかどうかを直接評価できるものではない．今後はこのように方策の性能を考慮して切り替えを行うような手法の開発に取り組んでいく．

## 謝辞

本研究の一部は，JSPS 科研費 JP20H00230 の助成を受けたものである．

## 参　考　文　献

[1] 兵頭侑樹, 松本耕平, 倉爪亮. 「動的環境における学習ベースおよびルールベースの切り替え手法を用いた移動ロボットナビゲーション」. 計測自動制御学会システムインテグレーション部門講演会 (SI), pp. 275–278, 2023.

[2] Jur Van Den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-Body Collision Avoidance. In *Proceedings of the International Symposium of Robotic Research*, pp. 3–19, 2011.

[3] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[4] Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph Normalizing Flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13556–13566, 2019.

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR)*, April 2017.

[6] Shaked Brody, Uri Alon, and Eran Yahav. How Attentive are Graph Attention Networks? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[7] Dan Jia, Alexander Hermans, and Bastian Leibe. Dr-spaam: A spatial-attention and auto-regressive model for person detection in 2d range data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10270–10277, 2020.