

## 自律行動制御手法に基づくAAVの実装及び評価

齋藤 伸樹・小田 哲也<sup>\*</sup>・平田 蒼人・片山 謙吾<sup>\*</sup>

岡山理科大学大学院工学研究科修士課程情報工学専攻

<sup>\*</sup>岡山理科大学工学部情報工学科

(2022年10月31日受付、2022年12月5日受理)

### 1. はじめに

Autonomous Aerial Vehicle (AAV)は屋外の広大な空間での運用が主であるため、Global Navigation Satellite System (GNSS)を用いた自己位置情報に基づく自律行動制御が行われている。一方、AAVは測量や点検、物資輸送、監視等の様々な用途で応用が期待されている[1, 2, 3]。これらの応用を実現するには、トンネルや下水道のような電波環境の劣悪な場所、屋内や建築物同士の隙間、橋梁の下部空間等のGNSSによる位置情報の取得が困難、もしくは不可能な環境下での運用等の問題を考慮する必要がある。そこで、AAVの自律飛行行動制御アルゴリズムとして、環境に応じて行動を変化させることが可能な深層強化学習の一手法であるDeep Q-Network (DQN)を適用することで、前述した問題を解決もしくは影響の低減が可能である。深層強化学習は、強化学習における価値関数や政策関数をDeep Neural Network (DNN)によって近似する手法であり、DQNは、強化学習[4, 5]におけるQ値の関数近似としてConvolution Neural Network (CNN)を用いた手法である。DQNは、Neural fitting Q-iteration [6, 7]とExperience replay [8]を組み合わせることにより、行動パターンごとに行動価値関数の隠れ層を共有しており、CNNのような非線形関数でも学習を安定して行うことが可能である[9, 10]。しかし、AAVにおける空中移動のように、3次元空間での動作を決定することが求められる問題は探索空間も広大であるため、通常のDQNは報酬を得るまでに非常に多くの時間を要する。そこで本稿では、自律行動制御手法に基づくAAVの実装を示すとともに、AAVのための自律行動制御手法を提案する。自律行動制御手法は、AAVの自律的な行動を実現するため、LiDARを用いた目的地の決定手法、高速に移動経路を得る手法としてタブーリスト戦略に基づくDQN (TLS-DQN)、TLS-DQNに起因する移動時における座標の変動の低減を目的とした移動経路補正手法から構成される。加えて、実環境を想定して、曲がり角を含む屋内単一経路環境における自律行動制御手法の性能評価を行う。

### 2. 提案システム

#### 2.1 Autonomous Aerial Vehicle

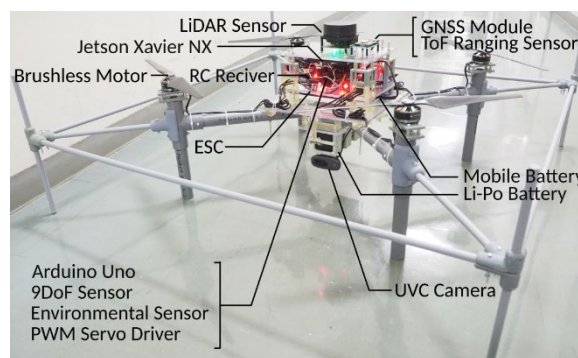


図1 AAVの機体

図1に開発したAAVの機体を示す。本機体は4枚の回転翼を備えるクワッドロータ型であり、空中での定点監視活動が可能である。本機体のフレームは、主にポリ塩化ビニル (PVC)パイプとアクリル板で構成されており、用途に応じてセンサとアクチュエーションの機構を拡張することが可能である。また、バッテリーやモータ、センサ等の機体構成部品をフレームに固定するため、固定補助用の部品を3D Computer Aided Design (CAD)ソフトを用いて設計し、3Dプリンタを用いて作成している。表1に本機体の構成部品を示す。本機体サイズは、高さ40 [cm]、幅90 [cm]、奥行き90 [cm]となっている。また、位置情報の取得のためのGNSSモジュール、動画の取得のためのUVCカメラ、周囲の遮蔽物取得のためのLight Detection and Ranging (LiDAR)、LiDARにより取得したデータに基づき自己位置の推定が可能なSimultaneous Localization and Mapping (SLAM)、以上により自己と周囲の環境取得が可能である。また、図2に機体の制御のイメージを示す。本機体は後述する自律行動制御手法の機体上での実行を考慮し、知的アルゴリズムの高速な実行が可能なシングルボードコンピュータであるJetson Xavier NXを搭載している。Jetsonは、自律行動制御手

法を用いて導出した移動経路に基づき、上下左右といった動作命令をArduinoへ送信する。Arduinoは動作命令及び各センサにより取得したデータを用い、フィードバック制御の一手法であるProportional Integral Differential (PID)制御に基づき、モータの回転数を制御するためのデューティ比を決定する。また、決定したデューティ比に基づき、Pulse Width Modulation (PWM)ドライバとElectric Speed Controller (ESC)を介し、各ブラシレスモータを制御する。

表1 AAVの構成部品

部品	モデル
Jetson	Xavier NX
Arduino	UNO
環境センサ(気温, 気圧, 湿度, ガス)	BME680
PWMドライバ	PCS9685
LiDAR	A1M8
GNSSモジュール	Ultimate 66
UVCカメラ	Logicool C270
プロペラ	15 × 5.8
モータ	MN3508 700kv
ESC	F45A 32bitV2
リポバッテリー	12000mAh
モバイルバッテリー	23000mAh
測距センサ	VL53L0X
配電盤	MES-PDB-KIT

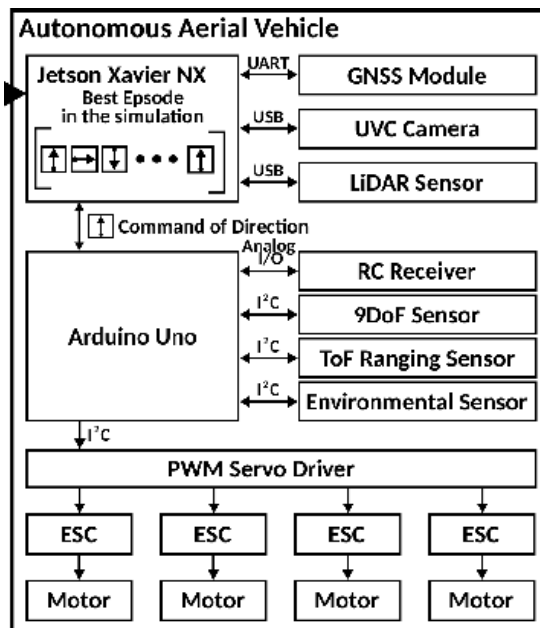


図2 AAV制御のイメージ

## 2.2 LiDARを用いた目的地の決定手法

Alg. 1 LiDARに基づく目的地の決定手法

### Algorithm 1 LiDAR Based Mobile Area Decision Method.

**Input:** Point Cloud List  $\leftarrow$  The coordinates list of obstacles (distance, angle) obtained by LiDAR  
 $(x_{LiDAR}, y_{LiDAR}) \leftarrow$  The coordinates of LiDAR Placement.  
**Output:** Destination (X, Y).  
1: **for**  $i = 0$  to 360 **do**  
2:  $x_{Point\ Cloud\ List}[i] \leftarrow Point\ Cloud\ List[i][0] \times \cos(Point\ Cloud\ List[i][1])$ .  
3:  $y_{Point\ Cloud\ List}[i] \leftarrow Point\ Cloud\ List[i][0] \times \sin(Point\ Cloud\ List[i][1])$ .  
4: **if**  $Point\ Cloud\ List[i][0] > Any\ Distance$  **then**  
5:  $Distant\ Point\ Cloud\ List[i] \leftarrow (x_{Point\ Cloud\ List}[i], y_{Point\ Cloud\ List}[i])$ .  
6:  $(x_{min}, x_{max}) \leftarrow$  Min. and Max. value for X-axis in the Distant Point Cloud.  
7:  $(y_{min}, y_{max}) \leftarrow$  Min. and Max. value for Y-axis in the Distant Point Cloud.  
8:  $(x_{center}, y_{center}) \leftarrow (\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2})$ .  
9:  $flag \leftarrow 0$ .  
10: **for**  $x = x_{LiDAR}$  to  $x_{center}$  **do**  
11:  $y \leftarrow (\frac{y_{center} - y_{LiDAR}}{x_{center} - x_{LiDAR}}) \times (x - x_{LiDAR}) + y_{LiDAR}$ .  
12: **for**  $i = 0$  to 360 **do**  
13: **if**  $\sqrt{(x - x_{Point\ Cloud\ List}[i])^2 + (y - y_{Point\ Cloud\ List}[i])^2} > Any\ Distance$  **then**  
14:  $Destination \leftarrow (x, y)$ .  
15: **else**  
16:  $flag \leftarrow 1$ .  
17: **break**  
18: **if**  $flag = 0$  **then**  
19:  $Destination$  is local destination.  
20: **else**  
21:  $Destination$  is global destination.

自律行動制御における目的地の決定手法をAlg. 1に示す。目的地の決定には、LiDARで取得した遮蔽物を示す点群とLiDARの位置座標を用いる。また、対象環境においてAAVが移動可能な範囲のZ軸の値を用いる。目的地は、最終的な目的地もしくは目的地までの目標通過点のどちらかを設定し、AAVが目標通過点へ到達時、目的地もしくは目標通過地点を連続的に決定する。

## 2.3 AAVのためのDeep Q-Network

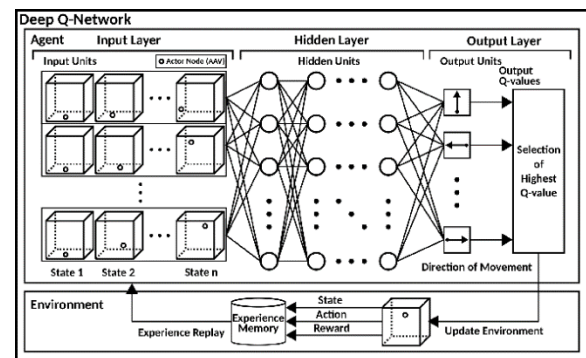


図3 Deep Q-Network

図3にDQNの構成を示す。DQNは深層強化学習の一手法であり、Q学習におけるQ値の推定をDNN等の深層学習を用いて近似する手法である。DQNは、マルコフ性を有する問題であれば、適用することが可能である。AAVの自律行動制御のため、DQNは移動を行った際に、目的地及び目標通過点に接近した場合は良い出力値、目的地から離れるもしくは遮蔽物へ衝突した場合は悪い出力値であること学習する。また、DQNのDNN部にはCNNより計算負荷の小さいDeep Belief Network (DBN)を用いており、AAVの行動選択肢として、上下左右前後、停止の7パターンを考慮している。

## 2.4 タブーリスト戦略

$$r = \begin{cases} 3 & \text{if } (x_{\text{current}} = x_{\text{global destinations}}) \wedge \\ & (y_{\text{current}} = y_{\text{global destinations}}) \wedge \\ & (z_{\text{current}} = z_{\text{global destinations}}) \vee \\ & ((x_{\text{before}} < x_{\text{current}}) \wedge (x_{\text{current}} \leq x_{\text{local destinations}})) \vee \\ & ((x_{\text{before}} > x_{\text{current}}) \wedge (x_{\text{current}} \geq x_{\text{local destinations}})) \vee \\ & ((y_{\text{before}} < y_{\text{current}}) \wedge (y_{\text{current}} \leq y_{\text{local destinations}})) \vee \\ & ((y_{\text{before}} > y_{\text{current}}) \wedge (y_{\text{current}} \geq y_{\text{local destinations}})) \vee \\ & ((z_{\text{before}} < z_{\text{current}}) \wedge (z_{\text{current}} \leq z_{\text{local destinations}})) \vee \\ & ((z_{\text{before}} > z_{\text{current}}) \wedge (z_{\text{current}} \geq z_{\text{local destinations}})) \\ -1 & (\text{else}). \end{cases}$$

Eq. 1 報酬関数

Alg. 2 TLS-DQNのためのタブーリスト

### Algorithm 1 Tabu List for TLS-DQN.

**Require:** The coordinate with the highest evaluated value in the section is  $(x, y, z)$ .

```

1: if  $(x_{\text{before}} \leq x_{\text{current}}) \wedge (x_{\text{current}} \leq x)$  then
2:    $\text{tabu list} \leftarrow ((x_{\min} \leq x_{\text{before}}) \wedge (y_{\min} \leq y_{\max}) \wedge (z_{\min} \leq z_{\max}))$ 
3: else if  $(x_{\text{before}} \geq x_{\text{current}}) \wedge (x_{\text{current}} \geq x)$  then
4:    $\text{tabu list} \leftarrow ((x_{\text{before}} \leq x_{\max}) \wedge (y_{\min} \leq y_{\max}) \wedge (z_{\min} \leq z_{\max}))$ 
5: else if  $(y_{\text{before}} \leq y_{\text{current}}) \wedge (y_{\text{current}} \leq y)$  then
6:    $\text{tabu list} \leftarrow ((x_{\min} \leq x_{\max}) \wedge (y_{\min} \leq y_{\text{before}}) \wedge (z_{\min} \leq z_{\max}))$ 
7: else if  $(y_{\text{before}} \geq y_{\text{current}}) \wedge (y_{\text{current}} \geq y)$  then
8:    $\text{tabu list} \leftarrow ((x_{\min} \leq x_{\max}) \wedge (y_{\text{before}} \leq y_{\max}) \wedge (z_{\min} \leq z_{\max}))$ 
9: else if  $(z_{\text{before}} \leq z_{\text{current}}) \wedge (z_{\text{current}} \leq z)$  then
10:   $\text{tabu list} \leftarrow ((x_{\min} \leq x_{\max}) \wedge (y_{\min} \leq y_{\max}) \wedge (z_{\min} \leq z_{\text{before}}))$ 
11: else if  $(z_{\text{before}} \geq z_{\text{current}}) \wedge (z_{\text{current}} \geq z)$  then
12:   $\text{tabu list} \leftarrow ((x_{\min} \leq x_{\max}) \wedge (y_{\min} \leq y_{\max}) \wedge (z_{\text{before}} \leq z_{\max}))$ 
    
```

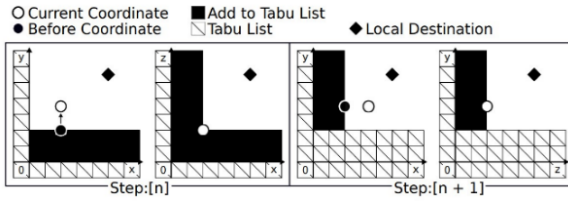


図4 タブーリスト追加イメージ

TLS-DQNのためのタブーリスト戦略(TLS)はF. Glover [11]により提案されたTabu Search (TS)に基づいており、最適化問題において、局所最適解への収束を抑制するため、エピソード中に既に探索した領域への移動を禁止する。TLSにおいて、 $x, y, z$ はそれぞれ対象環境のX軸、Y軸、Z軸を示す。 $\text{current}$ ,  $\text{before}$ は、それぞれDQNにおけるAAVの現在座標及びAAVの移動前座標を示している。また、 $\text{Local Destination}$ ,  $\text{Global Destination}$ は、それぞれ問題領域内における目標通過点と目的地を示している。また、対象領域を分割し、各領域に1つの目標通過点もしくは目的地を設定する。DQNにおけるAAVの行動に対する報酬値を、Eq. 1に基づき導出する。報酬値は、現在座標が目的地である場合と現在座標が移動前座標に対して目的地へ接近した場合は、3とする。それ以外の場合、報酬値は-1とする。タブーリストは、DQNにおけるAAVのランダムな行動選択時及び行動に対する報酬値の決定時に用いる。ランダムな移動方向の決定時にタブーリストを参照し、移動方向の領域がタブーリストに含まれている

場合、移動方向を再選択する。また、報酬値が決定された際、Alg. 2に基づいて、目的地から離れた領域を移動禁止領域としてタブーリストへ追加する。TLSはタブーリストをDQNにおけるAAVの行動選択の反復回数が終了するまで保持し、エピソードごとに初期化する。図4は、Alg. 2に従って移動禁止領域をタブーリストに追加する例を示している。図4において $n$ は自然数であり、DQNにおけるAAVの行動選択の反復回数である。図4のStep :  $[n]$ では、AAVがY軸方向に移動し、移動前座標よりも現在座標が目的地に接近している。そのため、Alg. 2に基づき、 $[(x_{\min} \leq x_{\max}), (y_{\min} \leq y_{\text{before}}), (z_{\min} \leq z_{\max})]$ を満たす黒塗りの領域をタブーリストに追加する。また、Step: $[n+1]$ では、AAVがX軸方向に移動し、移動前座標よりも現在座標が目的地に接近している。そのため、Alg. 2に基づき、 $[(x_{\min} \leq x_{\text{before}}), (y_{\min} \leq y_{\max}), (z_{\min} \leq z_{\max})]$ を満たす黒塗りの領域をタブーリストに追加する。TLSをDQNに適用することにより、ランダムな移動方向の決定による探索と比較し、再訪した座標での報酬の取得を制限することが可能であるため、解探索空間内をより広範囲に探索を行うことが期待される。

## 2.5 移動経路補正手法

Alg. 3 移動経路補正手法

### Algorithm 2 Movement Adjustment Decision.

**Input:** Movement Coordinates  $\leftarrow$  The movement of coordinates  $(X, Y, Z)$  by TLS-DQN

**Output:** Adjustment Point Coordinates List.

```

Number of divided list  $\leftarrow$  Any number.
2: Number of coordinates  $\leftarrow \frac{\text{Number of iterations in TLS-DQN}}{\text{Number of divided list}}$ 
 $i \leftarrow 0, j \leftarrow 0$ 
4: for  $k = 0$  to Number of coordinates in Movement Coordinates do
   Divided List  $[j] \leftarrow$  Movement Coordinates  $[k]$ .
    $j \leftarrow j + 1$ .
6: if  $j \geq \text{Number of coordinates}$  then
8:    $(x_{\min}, x_{\max}) \leftarrow$  Min. and Max. values for X-axis in the Divided List.
    $(y_{\min}, y_{\max}) \leftarrow$  Min. and Max. values for Y-axis in the Divided List.
10:   $(z_{\min}, z_{\max}) \leftarrow$  Min. and Max. values for Z-axis in the Divided List.
    $(x_{\text{center}}, y_{\text{center}}, z_{\text{center}}) \leftarrow (\frac{x_{\min} + x_{\max}}{2}, \frac{y_{\min} + y_{\max}}{2}, \frac{z_{\min} + z_{\max}}{2})$ 
12:  Adjustment Point Coordinates List  $[i] \leftarrow (x_{\text{center}}, y_{\text{center}}, z_{\text{center}})$ .
    $i \leftarrow i + 1, j \leftarrow 0$ 
    
```

移動経路補正手法は、TLS-DQNに起因する移動時における座標の変動を抑制するために用いる。Alg. 3は、TLS-DQNで得られた累計獲得報酬値が最も高かったエピソードにおける移動経路の座標リストを用いて、補正点の座標リストを出力する。また、補正点をB-スプライン曲線における制御点として用いて導出した曲線に基づき、移動経路の補正を行う。Alg. 3において、 $\text{Number of Divided List}$ ,  $\text{Number of Coordinate}$ は、それぞれ移動経路の座標リストを分割する数とリストに含まれる座標の数を示している。また、 $x_{\text{center}}$ ,  $y_{\text{center}}$ ,  $z_{\text{center}}$ は、分割された移動経路の座標リストであるDivided Listに含まれる座標のX軸、Y軸、Z軸における最大値と最小値から得られる座標を示す。



### 3. 性能評価

本節では、TLS-DQNによるAAVの自律行動制御のための実験及びシミュレーション結果について述べる。実環境を考慮するため、屋内単一経路環境である岡山理科大学のC4号館1階廊下からC5館1階渡り廊下を対象環境とした。図5は、対象環境にて実際に撮影した写真である。図6は、対象環境の実測値に基づく問題領域を示しており、赤色と青色で着色している部分は、それぞれ曲がり角と床表面を示している。

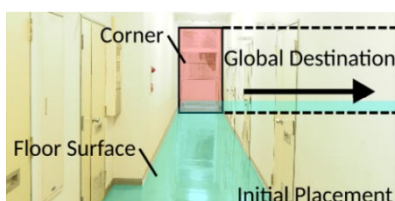


図5(a) 初期位置から曲がり角方向を撮影した写真



図5(b) 曲がり角から初期位置方向を撮影した写真

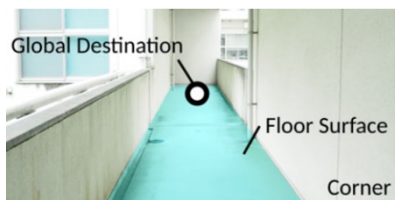


図5(c) 曲がり角から目的地方向を撮影した写真

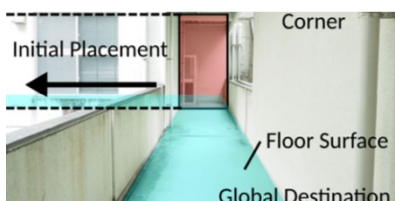


図5(d) 目的地から曲がり角方向を撮影した写真

図5 屋内単一経路環境の写真

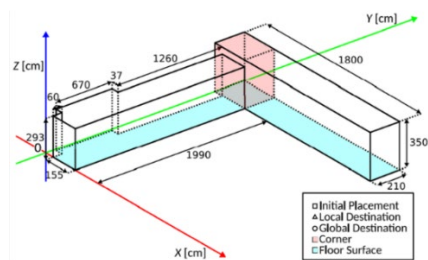


図6 曲がり角を含む屋内単一経路環境

#### 3.1 LiDARを用いた目的地の決定手法の結果

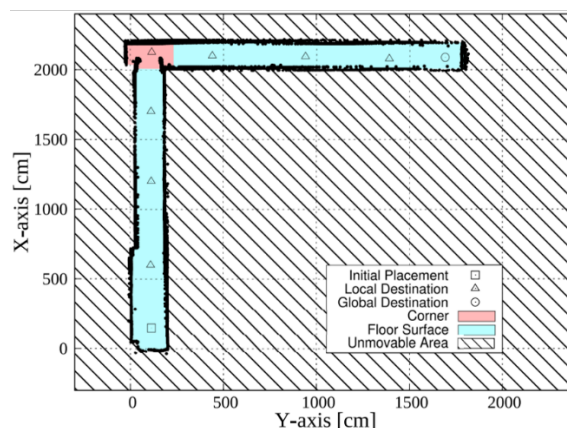


図7 LiDARを用いた目的地の決定手法の結果

図7は、LiDARで取得した周囲の遮蔽物とLiDARを用いた目的地の決定手法により決定された目標通過点及び目的地を示している。実験結果より、初期位置から目標通過点を連続的に決定し、経路の終点で目的地が決定されている。

#### 3.2 TLS-DQNのシミュレーション結果

表2 シミュレーションに用いたパラメータ

機能	パラメータ
エピソード数	30000
反復回数	2000
隠れ層	3
隠れユニット	15
初期の重み	Normal Initialization
活性化関数	ReLU
学習使用率( $\epsilon$ )	$0.999 - (t / \text{エピソード数})$ ( $t = 0, 1, 2, \dots, \text{エピソード数}$ )
学習率( $\alpha$ )	0.04
割引率( $\gamma$ )	0.9
経験メモリ	$400 \times 100$
バッチサイズ	32

シミュレーションでは、初期位置から目的地までの間で、離陸、飛行、着陸を行うことを目標とする。表2にDQNによるシミュレーションに用いたパラメータを示す。図7は、TLS-DQNにおけるWorst, Median, Bestの各エピソードについて、各反復での行動に対する報酬値の累積を示す。図7からBestとMedianのエピソードでは、報酬値が上昇傾向にあることが見て取れる。

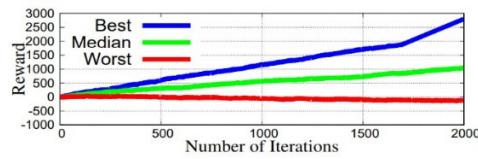


図7 報酬値の累積

### 3.3 移動経路補正手法の可視化結果

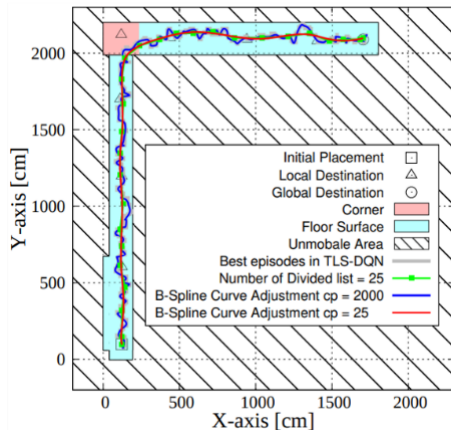


図8(a) XY平面での移動経路

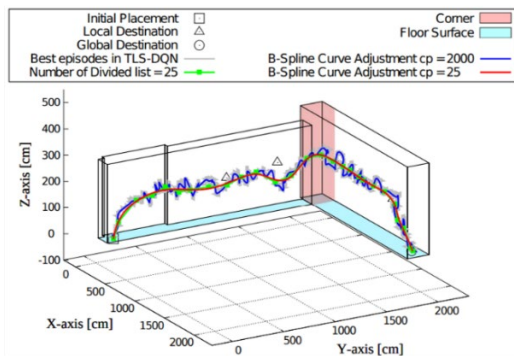


図8(b) 3次元空間での移動経路

図8 移動経路の可視化

表3 各手法により導出した経路の移動距離

手法	XYZ	XY平面	YZ平面
TLS-DQN	1449.000	936.000	985.000
B-スプライン曲線	363.189	351.918	245.515
移動経路補正手法(B-スプライン曲線無)	580.221	496.838	447.300
移動経路補正手法(B-スプライン曲線有)	361.740	351.526	238.615

TLS-DQNによるシミュレーション結果において累計獲得報酬値の最も高いエピソードに対し、移動経路補正手法を分割リスト数が25の場合で適用し、比較を行う。図8にTLS-DQNによる移動経路と移動経路補正手法による移動経路の可視化結果を示す。移動経路補

正手法を適用することにより、移動経路の変動が低減されていることが見て取れる。表3は、TLS-DQNと移動経路補正手法を適用した結果の移動経路におけるそれぞれの移動距離を示す。移動距離は、座標間のユークリッド距離の総和により導出した。性能評価の結果、移動経路補正手法は移動距離を削減し、移動時の変動を低減できることがわかった。

## 4. まとめ

本稿では、AAVの実装と自律行動制御手法を提案した。また、性能評価のため、曲がり角を含む屋内単一経路環境を対象としたTLS-DQNによるAAVの自律行動制御のシミュレーションを行った。性能評価の結果から、提案手法は、対象環境において目的地を決定するとともに、初期位置から目的地までの移動経路を導出することができた。そのため、提案手法は、屋内単一経路環境において、有用なアプローチであると考えられる。今後は、様々なシナリオを考慮して、自律行動制御手法及びAAVの改善を行いたいと考えている。

## 参考文献

- 1) O. Artemenko, et al., "Energy-aware Trajectory Planning for the Localization of Mobile Devices using an Unmanned Aerial Vehicle", Proc. of The 25-th International Conference on Computer Communication and Networks (ICCCN-2016), pp. 1-9, (2016).
- 2) M. Popovic, et al., "An Informative Path Planning Framework for UAV-Based Terrain Monitoring", Autonomous Robots, Vol. 44, pp. 889-911, (2020).
- 3) H. Nguyen, et al., "LAVAPilot: Lightweight UAV Trajectory Planner with Situational Awareness for Embedded Autonomy to Track and Locate Radio-tags", arXiv:2007.15860, pp. 1-8, (2020).
- 4) V. Mnih, et al., "Human-Level Control Through Deep Reinforcement Learning", Nature, Vol. 518, pp. 529-533, (2015).
- 5) V. Mnih, et al., "Playing Atari with Deep Reinforcement Learning", arXiv:1312.5602, pp. 1-9, (2013).
- 6) T. Lei and L. Ming, "A Robot Exploration Strategy Based on Q-learning Network", IEEE International Conference on Real-time Computing and Robotics (IEEE RCAR-2016), pp. 57-62, (2016).
- 7) M. Riedmiller, "Neural Fitted Q Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method", Proc. of The 16-th European Conference on Machine Learning (ECML-2005), pp. 317-328, (2005).
- 8) L. J. Lin, "Reinforcement Learning for Robots Using Neural Networks", Proc. of Technical Report, DTIC Document, (1993).
- 9) S. Lange, and M. Riedmiller, "Deep Auto-Encoder Neural Networks in Reinforcement Learning", Proc. of The International Joint Conference on Neural Networks (IJCNN-2010), pp. 1-8, (2010).
- 10) L. P. Kaelbling, et al., "Planning and Acting in Partially Observable Stochastic Domains", Artificial Intelligence, Vol. 101, No. 1-2, pp. 99-134, (1998).
- 11) F. Glover, "Tabu Search - Part I", ORSA Journal on Computing, Vol. 1, No. 3, pp. 190-206, (1989).

# Implementation and Evaluation of Mobility Control Method based AAV

Nobuki Saito, Tetsuya Oda\*, Aoto Hirata and Kengo Katayama\*

*Graduate School of Engineering,*

*\*Department of Information Science and Computer Engineering,*

*Okayama University of Science,*

*1-1 Ridai-cho, Kita-ku, Okayama 700-0005, Japan*

(Received October 31, 2022; accepted December 5, 2022)

The Deep Q-Network (DQN) is one of the deep reinforcement learning algorithms which uses deep neural network structure to estimate the Q-value in Q-learning. In this paper, we design and implement a DQN-based Autonomous Aerial Vehicle (AAV) testbed and propose a DQN-based AAV mobility control method. Also, we show the performance evaluation results for a scenario of an indoor single-path environment with a corner space. The performance evaluation results show that the proposed method can be reached the destination and decrease the movement fluctuation.

**Keywords:** Deep Q-Network: Autonomous aerial vehicle: Deep reinforcement learning: Unmanned aerial vehicle