岡山理科大学紀要　第57号 A pp. 41-46（2021）

TLS-DQNに基づくAAVの実装と評価

齋藤　伸樹・小田　哲也*・平田　蒼人・クラ　エリス*

岡山理科大学大学院工学研究科修士課程情報工学専攻

*岡山理科大学工学部情報工学科

(2021年11月1日受付、2021年12月9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制御が行われている．一方，AAVは測量や点検，物資輸送，監視など様々な用途での応用が期待されている [1, 2, 3]．こ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解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，深層強化学習に基づくAAVの実装を示すとともに，高速に解を得る手法としてタブーリスト戦略に基づくDQN (TLS-DQN)を提案する．また，TLS-DQNに起因する移動時の変動を低減させるための移動経路補正手法の提案を行う. 加えて，性能評価のため，実環境を想定した屋内単一経路環境におけるTLS-DQNによるAAVの自律制御シミュレーションと，シミュレーション結果に対する移動経路補正手法を適用した結果を示す.

2. 提案システム

2.1 Autonomous Aerial Vehicle

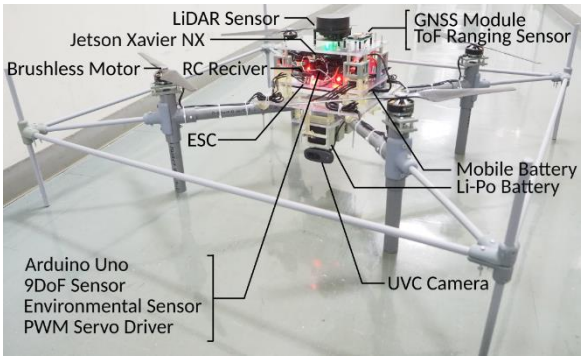

図1　AAVの機体

AAVを様々な用途で応用するために，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なAAVを開発する．開発したAAVの機体を図1に示す．開発した機体は4枚の回転翼を備えることにより空中で停止行動ができ，定点での活動が可能なクワッドロータ型である．クアッドロータのフレームは，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いる．また，バッテリやモータ，センサなどをフレームに接続するための部品は，Computer Aided Design (CAD)ソフトであるFusion360を用いて設計し，光学式3Dプリンタにより出力したものを用いている．開発したAAVの構成部品を表1に示し，AAVの制御のイメージを図2に示す．AAVの大きさは高さ40 [cm]，幅90 [cm]，奥行き90 [cm]となっている．開発したAAVは128コアGPUを搭載するシングルボードコンピュータJetson Xavier NXを搭載し，GNSSモジュールを用いた位置情報の取得，UVCカメラによる動画像の取得，Light Detection and Ranging (LiDAR)より取得したデータを基にリアルタイム性が高いLiDAR Simultaneous Localization and Mapping (LiDAR SLAM)を用いた自己位置推定，以上によって自己と周囲の環境をストリーミングで取得することが可能である．Jetsonは，後述

するTLS-DQNを用いて導出した結果に基づいて，上下左右前後停止といった動作命令をArduinoへ送信する．Arduinoは，動作命令と各センサを用いて取得したデータから，フィードバック制御の一手法であるProportional Integral Differential (PID)制御に基づいてモータの回転数を決定し，Pulse Width Modulation (PWM)ドライバとElectric Speed Controller (ESC)を介し，各モータを制御する．

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	12000mAh
モバイルバッテリ	23000mAh
測距センサ	VL53L0X
配電盤	MES-PDB-KIT

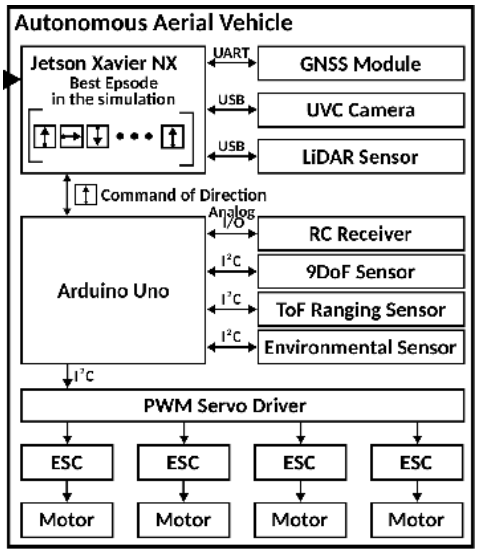

図2　AAV制御のイメージ

2.2　AAVのためのDeep Q-Network

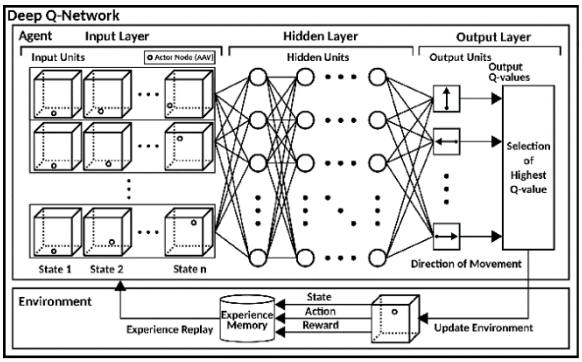

図3　Deep Q-Network

図3にDQNの構成を示す．DQNは強化学習の一手法である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り，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の行動制御のため，DQNは移動を行った際，領域内に任意で設置される目的地へ接近すると良い出力値，目的地から離れるもしくは遮蔽物へ衝突すれば悪い出力値であることを報酬として学習する．報酬関数に関しては，後述するタブーリスト戦略にて示す．DQNはRustプログラミング言語を用いて実装しており，DNN部にはCNNより計算負荷の小さいDeep Belief Network (DBN)を用いている．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．

2.3　タブーリスト戦略

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

Eq. 1　報酬関数

Alg. 1　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).

if $(x_{before} \leq x_{current}) \wedge (x_{current} \leq x)$ **then**
  $tabu\,list \Leftarrow ((x_{min} \leq x_{before}) \wedge (y_{min} \leq y_{max}) \wedge (z_{min} \leq z_{max}))$
else if $(x_{before} \geq x_{current}) \wedge (x_{current} \geq x)$ **then**
  $tabu\,list \Leftarrow ((x_{before} \leq x_{max}) \wedge (y_{min} \leq y_{max}) \wedge (z_{min} \leq z_{max}))$
else if $(y_{before} \leq y_{current}) \wedge (y_{current} \leq y)$ **then**
  $tabu\,list \Leftarrow ((x_{min} \leq x_{max}) \wedge (y_{min} \leq y_{before}) \wedge (z_{min} \leq z_{max}))$
else if $(y_{before} \geq y_{current}) \wedge (y_{current} \geq y)$ **then**
  $tabu\,list \Leftarrow ((x_{min} \leq x_{max}) \wedge (y_{before} \leq y_{max}) \wedge (z_{min} \leq z_{max}))$
else if $(z_{before} \leq z_{current}) \wedge (z_{current} \leq z)$ **then**
  $tabu\,list \Leftarrow ((x_{min} \leq x_{max}) \wedge (y_{min} \leq y_{max}) \wedge (z_{min} \leq z_{before}))$
else if $(z_{before} \geq z_{current}) \wedge (z_{current} \geq z)$ **then**
  $tabu\,list \Leftarrow ((x_{min} \leq x_{max}) \wedge (y_{min} \leq y_{max}) \wedge (z_{before} \leq z_{max}))$

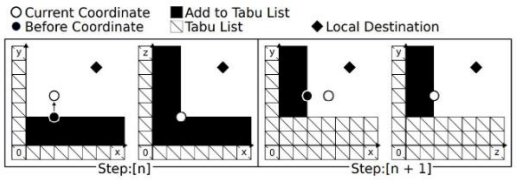

図4　タブールール追加イメージ

TLS-DQNのためのタブーリスト戦略(TLS)はF. Glover [11]によって提案されたTabu Search (TS)に基づいており，様々な最適化問題に対して，局所最適解に陥らないように，以前に訪れた探索領域への移動を禁止することで，効率的な探索を実現する．TLSにおいて，x，y，zはそれぞれX軸，Y軸，Z軸を示す．また，$current$は，DQNにおけるAAVの現在の座標を示し，$before$は移動方向を決定し移動する前の座標を示す．また，$Global\ Destination$は，問題領域内における目的地を示しており，$Local\ Destination$は，初期位置から$Global\ Destination$までの目標通過点を示す．また，対象となる領域を目標通過点に基づいて分割し，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．現在の座標が目的地である場合，報酬値を3とする．また，現在の座標が移動前の座標に対して，目的地へ接近した場合，報酬値は3とする．それ以外の場合，報酬の値は-1とする．TLSにおけるタブーリストは，DQNにおけるAAVの行動を選択する際と，その行動に対する報酬を決定する際に用いる．タブーリストは，ランダムに移動方向が決定される際に参照し，決定された移動方向の領域がタブーリストに含まれている場合，AAVは移動方向の再選択を行う．また，報酬値が決定された際に報酬値が-1である場合，Alg. 1に基づいて，目的地から離れた領域を禁止領域としてタブーリストへ追加する．タブーリストは，追加された移動禁止領域を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. 1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，移動前よりも目的地に接近しているため，Alg. 1における$((y_{before} < y_{current}) \wedge (y_{current} \leq y_{local\ destinations}))$を満たしている．したがって，$[(x_{min} \leq x_{max})$，$(y_{min} \leq y_{before})$，$(z_{min} \leq z_{max})]$を満たす，黒塗りされている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，移動前よりも目的地に接近しており，Alg. 1における$((x_{before} < x_{current}) \wedge (x_{current} \leq x_{local\ destinations}))$を満たしている．したがって，$[(x_{min} \leq x_{before})$，$(y_{min} \leq y_{max})$，$(z_{min} \leq z_{max})]$を満たす黒塗りされている領域がタブーリストに追加される．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．

2.4　移動経路補正手法

Alg. 2　移動経路補正手法

Algorithm 2 Movement Adjustment Decision.

Input: *Movement Coordinates* ← The movement of coordinates (*X*, *Y*, Z) by TLS-DQN
Output: *Adjustment Point Coordinates List*.
Number of divided list ← Any number.
Number of coordinates ← $\frac{Number\ of\ Iterations\ in\ TLS-DQN}{Number\ of\ divided\ list}$.
$i \leftarrow 0, j \leftarrow 0$
for $k = 0$ **to** Number of coordinates in *Movement Coordinates* **do**
  Divided List$[j]$ ← *Movement Coordinates*$[k]$.
  $j \leftarrow j+1$.
  if $j \geq$ *Number of coordinates* **then**
    (x_{min}, x_{max}) ← Min. and Max. values for X-axis in the *Divided List*.
    (y_{min}, y_{max}) ← Min. and Max. values for Y-axis in the *Divided List*.
    (z_{min}, z_{max}) ← Min. and Max. values for Z-axis in the *Divided List*.
    $(x_{center}, y_{center}, z_{center}) \leftarrow (\frac{x_{min}+x_{max}}{2}, \frac{y_{min}+y_{max}}{2}, \frac{z_{min}+z_{max}}{2})$
    Adjustment Point Coordinates List$[i]$ ← $(x_{center}, y_{center}, z_{center})$.
    $i \leftarrow i+1, j \leftarrow 0$

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. 2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(X, Y, Z)からなる移動座標リストを入力し，移動経路補正のための補正点のリストとなる$Adjustment\ Point\ Coordinates\ List$を出力する．また，$Number\ of\ Divided\ List$は，座標移動リストを分割する数を示しており，$Number\ of\ Coordinate$は，$Divided\ List$に含まれる座標の数を示している．また，$x_{center}$，$y_{center}$，$z_{center}$は，$Divided\ List$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による屋内単一経路環境を対象としたAAVの自律制御のためのシミュレーション結果と，移動経路補正手法の性能評価について述べる．

3.1　TLS-DQNのシミュレーション結果

図5(a)　初期位置から目的地方向を撮影した写真

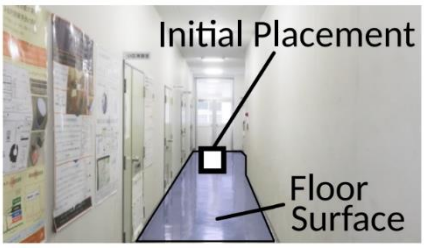

図5(b)　目的地から初期位置方向を撮影した写真

図5　屋内単一経路環境の写真

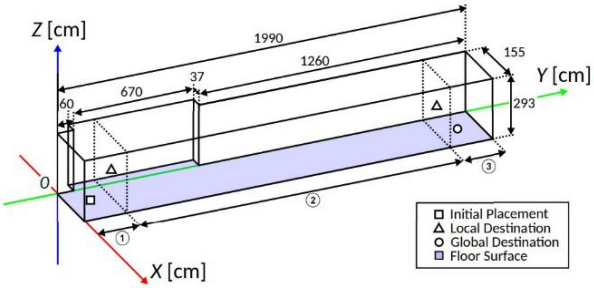

図6　屋内単一経路環境

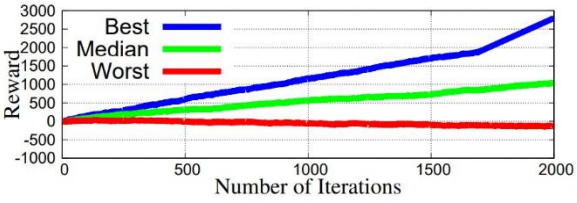

図7　報酬値の累積

シミュレーションでは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，青色で着色している部分は床表面を示している．シミュレーションでは，初期位置となるInitial Placementから目的地となるGlobal Destinationまでの間で，離陸，飛行，着陸を行うことを目標とする．初期位置を[75, 75, 0]，エリア①，②における通過目標点となるLocal Destinationをそれぞれ[75, 150, 150]，[75, 1850, 150]，エリア③では目的地を[75, 1925, 0]とした．DQNによるシミュレーションに用いたパラメータを表2に示す．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．

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	5000
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(ε)	0.999 - (t / エピソード数) (t = 0, 1, 2, …, エピソード数)
学習率(α)	0.04
割引率(γ)	0.9
経験メモリ	400×100
バッチサイズ	32

3.2　移動経路補正手法の可視化結果

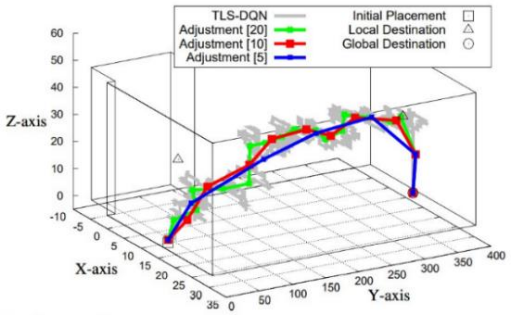

図8(a)　3次元空間での移動経路

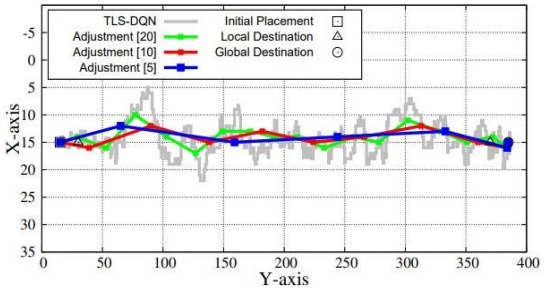

図8(b)　XY平面での移動経路

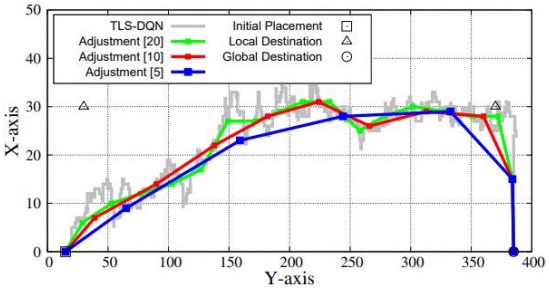

図8(c)　YZ平面での移動経路

図8　移動経路の可視化

表3　XY平面，YZ平面における移動距離

対象平面	XY平面	YZ平面
最短経路	370．00	406．19
TLS-DQN	850．00	890．00
分割リスト数 = 20	97．61	395．06
分割リスト数 = 10	390．94	390．39
分割リスト数 = 5	370．65	387．91

TLS-DQNによるシュミレーション結果における累計獲得報酬値の最も高いエピソードに対して移動経路補正手法の分割リスト数を20，10，5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．図8(a)は3次元空間での移動経路の可視化結果を示しており，移動経路補正手法を適用することによって，移動経路の変動が低減されていることが見て取れる．図8(b)，図8(c)に，TLS-DQNと移動調整法の適用結果のXY平面とYZ平面での移動経路の可視化結果を示す．表3は，最小移動距離，TLS-DQNによる移動経路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移動距離を示す．移動距離は，各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，移動調整法はXY平面とYZ平面の両方において，移動距離と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，深層強化学習に基づくAAVの実装と，TLS-DQNの提案，TLS-DQNに起因する移動時の変動を減少させるための移動経路補正手法を提案した．また，屋内単一経路環境を考慮したTLS-DQNによるAAV制御のシミュレーション，移動経路補正手法を適用した．性能評価結果から，提案した移動経路補正方法を用いることで，動きの変動を抑えることができ，移動変動を抑制することによって，移動距離を削減できることを確認した．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，深層強化学習に基づくAAVのためのTLS-DQNを改善していきたいと考えている．

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 TLS-DQN Based AAV

Nobuki Saito, Tetsuya Oda*, Aoto Hirata and Elis Kulla*

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 November 1, 2021; accepted December 9, 2021)

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 designed and implement a DQN-based Autonomous Aerial Vehicle (AAV) testbed and propose a Tabu List Strategy based DQN (TLS-DQN). Also we propose a movement adjustment method for decreasing the movement fluctuations caused by TLS-DQN during autonomous movement control. The performance evaluation results show that the proposed method can be reached destination and decrease the movement fluctuation.

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