15th European Conference on Antennas and Propagation (EuCAP),
22-26 March 2021, Dusseldorf, Germany
DOI:10.23919/EuCAP51087.2021.9411153

Introduction to UAV swarm utilization for communication on the move terminals tracking evaluation with reinforcement learning technique

Saki Omi*, Hyo-Sang Shin†, Antonios Tsourdos‡, Joakim Espeland§, Andrian Buchi¶
*The School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, U.K., s.omi@cranfield.ac.uk
†The School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, U.K., h.shin@cranfield.ac.uk
‡The School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, U.K., a.tsourdos@cranfield.ac.uk
§QuadSAT, Odense, Denmark, je@quadsat.com
¶QuadSAT, Odense, Denmark, ab@quadsat.com

***Abstract*—As the growth of communication and satellite industry, the demand of satellite antenna evaluation is increasing. Particularly Communication On The Move (COTM) terminal antenna, including electronically steerable antennas (ESA) and for the communication between new constellations on LEO and MEO, requires tracking accuracy test for the communication on moving vehicles. The measurement capability of conventional methodologies have been limited due to their location fixed facilities and non-adjustable sensor's positions during the measurement. To overcome this drawbacks, we will present how multi-agent system of UAVs could be utilized for COTM tracking accuracy evaluation. This measurement needs instant actions for UAVs to keep them navigating in order to achieve accurate and stable measurement. Reinforcement learning (RL) techniques are investigated for this purpose in this paper. The performance improvement is demonstrated with the system using RL technique to adjust UAVs with sensors during the measurement.**

***Index Terms*—Communication On The Move, de-pointing, antenna measurements, UAV, multi-agent reinforcement learning**

I. Introduction

Along with the satellite industry growth, the needs to evaluate the satellite antenna and its system has become eminent in order to guarantee that the satellite communication system on the ground does not hinder the other communication by emitting unwanted signal and solid network can be created between satellites and terrestrial terminal. COTM is a type of system whose terminal antennas are mounted on the moving vehicle such as a ship, a train, a vehicle or an airplane and establishes the satellite communication. The key requirement for COTM application is to keep tracking the intended satellite during the operation by steering the beam direction physically or electronically. To provide COTM antenna complete testing without satellite involvement, Fraunhofer IIS in collaboration with the Teschniche Univeraität has established Facility for Over-the-air Research and Testing (FORTE) [1] which has been authorized by a test entity of Global VSAT Forum [2]. However, the measurement sensors of FORTE are fixed on the mast and their positions cannot be adjusted during the evaluation. This could limit the available test scenarios and types of antenna under the test (AUT). Also, building such kind of bulky facilities would be expensive and test process may be logistically time consuming. Currently, there would not be enough number facilities to evaluate all newly developed COTM antennas under Satellite Operator's Minimum Antenna Performance (SOMAP) criteria [3] due to the steep increase of the demand.

On the other hand, UAV systems are nowadays widely used for many purposes such as surveillance, delivery and patrol. Airborne RF measurement is one of the highly focused areas within telecommunication industry and academia [4], [5]. However most of the existing applications are dedicated to radiation pattern measurements and pre-defined flight paths are applied. As long as we know, there is no examples of using autonomous multi-agent system of UAVs for evaluating the tracking accuracy of COTM system.

In this paper, it is investigated how autonomous UAVs system could be utilized for the tracking accuracy evaluation, so called de-pointing measurement. The final success of the development would bring three main benefits. Firstly, the airborne measurement system does not require heavy facility and can be delivered for on-site evaluation. Secondly, this solution would offer the capability to evaluate any COTM antenna including pattern variable antennas like ESA by continuously adjusting the sensor position during the measurement. Thirdly, the solution will add the capability to evaluate the antennas communicating with new satellite constellations operated in LEO and MEO by emulating the trajectory of those orbit in addition to the de-pointing formation. These advantages will eventually make the COTM de-pointing evaluation more accessible and extend the possible measurement scenarios.

De-pointing measurement requires simultaneous signal measurement from different locations since it is calculated from the correlation between measured signal strength and pre-collected radiation pattern. One of the foreseeable challenges of de-pointing measurement is positioning the system to optimal locations in real-time in the changing environment to keep the de-pointing measurement as accurate as possible. In this sense, the system needs to make a cooperative motion

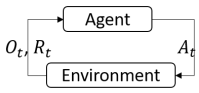

Fig. 1. Illustration of reinforcement learning

to work as a team in order to maximize the measurement accuracy. It is natural approach to calculate the existing de-pointing and then decide the next action from the estimated de-pointing. However, if de-pointing is calculated from the matching between reference radiation pattern as in [1], the computation time may not be sufficiently fast since there is a large amount of possible combinations on the table. Hence, the system which instantly provides the next action just by giving the row data is beneficial.

RL is one of the machine learning techniques which interacts with a dynamic environment and decides the action to take from its observation in order to maximize the accumulated "reward" during the episode. Alpha-go, which plays the complicated board game Go, is one of the big achievement of the recent development [6]. In [7], it was applied for a vehicle control of Mars landing. Recently utilizing this technique for multi-agent system, multi-agent reinforcement learning (MARL), has been extensively studied. MARL can efficiently work on the cooperative task such as multi-robot navigation [8], traffic control [9], team video games [10] and cooperative escort [11]. Using RL for COTM de-pointing could also be a valid approach for the reason mentioned above. This paper is to show initial investigation of using MARL for de-pointing measurement which nobody has studied yet apparently.

The background of RL will be described in section 2. The methodology, experiment set-up and its results are shown in section 2, 3 and 4 respectively. Finally, conclusion is provided in the section 5.

II. Background of RL

A. Concept of RL and MARL

RL is a type of machine learning techniques which consists of environment and agents. The agent learns to make decision about which action A_t to take at each time step based on their state S_t [12]. The agent gets observation O_t from the environment representing the information of new state S_{t+1} of the agent as a result of the action and also receives feedback signal, called reward R_t, which evaluates the selected action (Fig. 1). The purpose of the entire learning is to maximize the accumulated reward at the end of the episode. Originally, RL was structured based on Markov decision processes (MDPs). In a most general environment, the interaction from the environment is depending on entire history of states and actions. On the other hand, on MDP, the probability distribution of the response from the environment only depends on the current state as described in (1).

$$\begin{aligned} &P_r\left\{R_{t+1}=r, S_{t+1}=s' | S_0, A_0, R_1, \ldots, S_t, A_t\right\} \\ &= P_r\left\{R_{t+1}=r, S_{t+1}=s' | S_t, A_t\right\} \end{aligned} \tag{1}$$

RL techniques have been well studied so far and applying those techniques for multi-agent system is a recent open discussion. The objective of state-of-the-art MARL algorithm can be categorized into two [11]. One is to maximize the global reward for the success as a team as it can be found in COMA[10]. Another one is focusing on maximizing the local rewards like in MADDPG [13]. Then, MARL algorithm can have either centralized, decentralized or centralized training and decentralized execution structure. The centralized structure has only one single agent with a large state and action space for all objects to be controlled. The size of the parameters exponentially increases as the number of the objects grows. In the decentralized structure, each object has their own agent, thus the number of the parameters to be learned stays affordable. However, this approach generates selfish actions and does not suitable for credit assignment. Also, this structure tends to violate the Markov assumption since the other agents would be considered as a part of environment though their behaviours can vary because they are also learning. Therefore the current trend is to have centralized leaning and decentralized execution so that the Markov assumption can be kept during the training.

B. Policy Gradient Algorithms

In RL, an agent includes these component in general; "policy (π)", "value function" and "model" where the policy decides the action of the agent's behaviour, the value function represents evaluation of the state and action and the model predict the next state from the current state and action. Q-value describes the expected reward as $Q^{\pi}(s,a) = \mathbf{E}[r|s_t = s, a_t = a]$ (r : total discounted reward) and this value is used to evaluate the policy. To optimize the policy in continuous state-action space, the policy is parameterized with θ and the policy gradient decent is attractively calculated during the training. The objective function is to maximize the return formulated as $J(\theta) = \mathbf{E}_{\pi_\theta}[r]$. Then the policy gradient is derived as in (2) [12].

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{s\sim\rho^{\pi}, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q^{\pi_\theta}(s,a)] \tag{2}$$

,where ρ^{π} is a state distribution. There are several approaches to estimate the Q-value. One of the common ways is actor-critic method. In this method, the Q-value is also parameterized and the critic is used for the estimation of the Q-value function by taking the gradient decent. The actor is trained to optimize the policy parameter θ by taking the estimated gradient from the critic [14].

In [15], actor-critic is extended to deterministic policy gradient (DPG) algorithm where the policies are deterministic as $\mu_\theta : \mathcal{S} \mapsto \mathcal{A}$ presenting more efficient learning than stochastic policy algorithm. Deep neural network is normally applied for the approximation of the policy and the critic and this algorithm is called deep deterministic policy gradient (DDPG) [16]. Then, the gradient (2) can be rewritten as (3)

$$\nabla_\theta J_(\rho_\theta) = \mathbf{E}_{s\sim\rho^{\mu}}[\nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu}(s,a)|_{a=\mu_\theta(s)}] \tag{3}$$

It has been observed that DDPG tends to overestimate Q-value and end up with slow convergence. To overcome this drawback, twin delayed deep deterministic policy gradient (TD3) was proposed in [17] motivated by double Q-learning and double DQN. One of the key features of TD3 is "clipped double Q-learning", where it has two deterministic actors and two corresponding critics. The Q-functions are updated with the minimum target value among these two networks. Also its "delayed update of target and policy networks" feature reduces the variance of the value estimation by keeping the update frequency of policy slower than Q-value function update. In addition, "target policy smoothing" was introduced which adds clipped Gaussian noise to the selected action to avoid overfitting to the narrow peaks in the value estimation due to a concern with deterministic policies.
Recently there are many cases which TD3 have been applied for multi-agent system as multi-agent TD3 (MATD3) [18], [11]. [18] has a structure of decentralized actor-critic which is similar to [13] but instead of DDPG, this has TD3 network. [11] distinguished the local reward and the global reward and decomposed the structure with a centralized critic which is shared with all agents and local critics for each agent.

C. Recurrent Neural Network

One of the expected challenges of the de-pointing measurement is that the observation from the UAVs does not directly represent the state of the agents. De-pointing angle needs to be calculated based on the measured signal strength and the measured position. Then, the agent state w.r.t. RF sphere around AUT can be calculated. This situation is categorized as Partially Observable Markov Decision Process (POMDP) which requires to explicitly model the environment when the agents no longer have access to the true system state and receive observations instead. In POMDP, Q-function is $Q(o, a|\theta) \neq Q(s, a|\theta)$. Under this condition, agents need to construct their own state representation. Recently recurrent neural network (RNN) such as Long Short-Term Memory and Gated Recursive Unit (GRU) [19] has been extended to be applied for MARL to address the challenge of the POMDP [20]. Those RNNs can estimate the hidden state by giving the past sequence of the estimation and new observation [7], [10], [21].

III. Method

A. Fully centralized structure

For this de-pointing measurement application, the considered number of the UAVs are not so many. Therefore, it is valuable to investigate how feasible it is to apply a fully centralized reinforcement learning structure (Fig. 2). TD3 is assigned as the RL application to produce the trajectory of the UAVs. Specifically, the output value from the actor is regarded as the vector to move from the current position. Like COMA presented in [10], the goal of the system is to maximize the global reward as a group, i.e. to keep measuring de-pointing angle of AUT as accurate as possible. Hence, the structure for this application is designed to have a shared

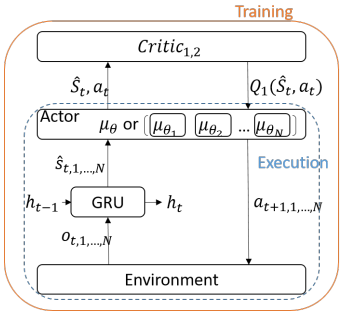

Fig. 2. RL structure (fully centralized / decentralized actor)

critic rather than having one for each agent, unlike MADDPG where each agent has their own critic because their interest is in maximize the local rewards [13]. For fully centralized structure, all state information from each agent is the input of the actor and the single actor generates the output for all UAVs.

B. Centralized Critic, Decentralized Actor

The structure with centralized critic and decentralized actor is also tested as shown in (Fig. 2). The actors' inputs can be either the estimated states of all of the agents; $a_i = \mu_{\theta_i}(\hat{s}_1 \ldots \hat{s}_N)$ or only the estimated state of the agent; $a_i = \mu_{\theta_i}(\hat{s}_i)$. Then, the policy gradient can be written as (4).

$$\nabla_{\theta_i} J_{(\rho_{\theta_i})} = \mathbf{E}_{s\sim\rho^\mu}[\nabla_\theta \mu_\theta(\hat{\mathbf{s}}(\hat{s}_i)) \nabla_{a_i} Q^\mu(\mathbf{s}, \mathbf{a})|_{a_i=\mu_{\theta_i}(\hat{\mathbf{s}}(\hat{s}_i))}] \tag{4}$$

C. RL setting

The reward is calculated based on the accuracy of the de-pointing estimation (R_{error}) and also a factor in order to avoid collision of the UAVs (R_{ca}) (7).

$$\begin{cases} R = R_{error} + R_{ca}, & \text{if } steps \geq 50 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

, where

$$R_{error} = -|\Delta\boldsymbol{\theta}| \tag{6}$$

$$R_{ca} = -0.1 \times M. \tag{7}$$

M is the number of the UAVs pairs of which distance is closer than $0.1°$ in any direction between them. One episode consists of 1000 steps. All UAVs are randomly located in the test area initially and rewards are counted 50 steps after starting de-pointing measurements to give agents to locate UAVs to their optimal positions. The observation $O_{t,1,\ldots,N}$ is defined as a data-set which consists of the positions of N number of UAVs and measured signal strength (and evaluation angle for ESA). The observation is stored in the block which includes the observation data set for the previous 5 steps and the block is passed to GRU as a shaded part in grey in Fig. 3. Then, this data frame are processed through GRU and current de-pointing estimation is generated as its output ($\Delta\theta_5$ in Fig. 3). The state $\hat{S}_t$ is the relative angle between UAVs' positions and

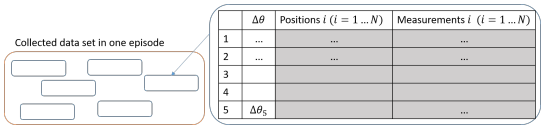

Fig. 3. GRU training data block

the estimated main beam direction of the AUT from GRU. Based on this estimated state, the position to go in the next time step is calculated from the actor. As an additional input of observation, the estimated angular velocity of the AUT (i.e. transition of the estimated de-pointing from the previous time step) is also implemented to examine if the behaviour information has effect on the accuracy inspired by [22].

For the training phase of TD3 network, the episode based on the decision of the actor and dynamic environment is proceeded and the transition data consists of $\left[\hat{S}_t, \hat{S}_{t+1}, \mathbf{a}_t, R_t\right]$ is collected and stored in the replay buffer. Also, GRU is trained every episode. The training data is accumulated during one episode for GRU as a block of data consists of true de-pointing angle, the positions of N number of UAVs and measured signal strength (and evaluation angle for ESA) as shown in the Fig. 3. The parameters of TD3 follow [17].

IV. Environment

A. Scenario

During the training, de-pointing angles are calculated on each steps from the correlation between pre-collected radiation pattern and measured signal strength with noise characterised by signal to noise ratio. Given UAVs' position data, the best matched angles are assigned as measured de-pointing.

B. AUT and UAV model

Theoretical radiation patterns of a parabolic antenna and an Uniform Linear Array (ULA) are used for the experiments. ULA's radiation pattern varies depending on its steering angle and evaluation angle for ULA is set to $5°$. The de-pointing around the evaluation angle is measured during the test. It is assumed that the radiation pattern has a granularity of $0.05°$ and multiple radiation patterns are available for each steering angle every $0.01°$ for ULA. The random angular acceleration is added to AUT in each step and it is tested if the movement of AUT is detected accurately from the developed system. In this experiment, the perfect control of the UAVs are assumed to reach the proposed points in the next time step and the Boresight of probe antenna is always directed to AUT.

V. Result

A. 1D test result

Firstly the test dimension is limited to 1 dimension and the system navigating 2 UAVs is trained for de-pointing measurement with parabolic antenna and ULA pattern. Fig. 4 shows the leaning transition for both antenna cases. Also system excluding GRU is tested. It can be found that the system cannot learn without GRU implementation because

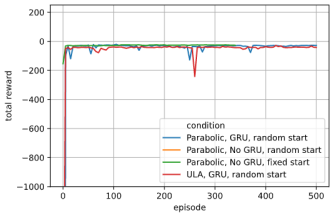

Fig. 4. Fully centralized RL, Learning transition in 1D

it cannot estimate their current state $\mathbf{s}$ accurately especially when their initial positions are strongly affected by SNR at low EIRP and it keeps collecting training data based on the wrong estimated states. If the initial positions of the UAVs are fixed in reasonable place, it can learn without GRU as shown in Fig. 4. RMSE results from the each trained system can be found in TABLE I. Regardless the varying amount of de-pointing and type of AUT, the stable performance of the measurements can be obtained when UAVs (i.e. dynamic sensors) are applied compared to the system with static sensors since the trained system keeps adjusting the position of UAVs based on the observation.

B. 2D test result

The learning transition for fully centralized and decentralized actor structure can be found in Fig. 6. $\Delta\boldsymbol{\theta}$ describes if the estimated de-pointing angular velocity is included in the actor's input. There is no clear difference in the final accuracy (TABLE I) between 3 UAVs with and without $\Delta\boldsymbol{\theta}$ cases although learning transition is more stable with $\Delta\boldsymbol{\theta}$. This would be because data update frequency is so high against UAVs' mobility that the system does not find this data effective. The achievable reward and RMSE are effectively improved when 3 UAVs are applied compared to 2 UAVs in fully centralized structure. In this case, the learning transition gets unstable at the beginning. It is expected that the more parameters there is, the harder it is to converge as it can be also seen from the 4 UAVs fully centralized case.

Also, the example formations from decentralized actor system are shown in Fig. 5. The system with 3 UAVs decentralized system achieved slightly worse level of reward than the fully centralized RL structure. Both 4 UAVs decentralized networks with the inputs, $\hat{\mathbf{s}}$ and $\hat{s}_i$ cases, converge. They show more stable RMSE, though neither of them do not improve the average accuracy compared to the 3 UAVs fully centralized structure. It is also noticeable there is no obvious difference in RMSE between $\hat{s}_i$ and $\hat{\mathbf{s}}$ cases. The accuracy from the trained system highly depends on the final state of the training and this may need to be less tighten. Further investigations are required to configure the network structure and the test set-up.

VI. Conclusion

In this paper, several examples to apply MARL for de-pointing measurement with UAVs were presented. Utilizing dynamic system which can adjust the sensors' locations during the measurement shows advantages in accuracy and stable

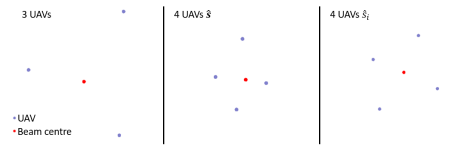

Fig. 5. Generated formation in 2D from decentralized actor

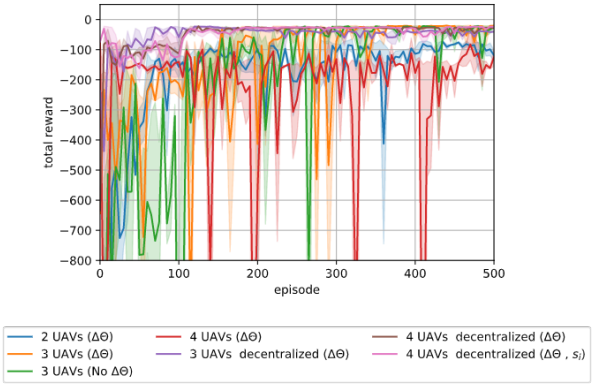

Fig. 6. Learning transition in 2D

performance compared to static system. This paper addresses a challenge of real-time navigation for these mobile sensors due to the massive amount of possible actions and reference data to compare to calculate de-pointing angle numerically. MARL could overcome this issue by generating next agent's action spontaneously without referencing the data by training the decision making system beforehand. This system would make COTM antenna evaluation possible for any types of antennas including ESA and test scenarios for LEO and MEO satellite communication. Future work will be to extend MARL to more realistic environments with constrains and variety of antenna patterns for COTM evaluation. Also, the RL structure will be further investigated for more efficient and stable learning which can quickly adapt its system for never experienced AUT so that the time for evaluation process can be shortened.

TABLE I
DE-POINTING MEASUREMENT ACCURACY RESULT FROM TRAINED SYSTEM IN 500 TIMES TESTS

Condition	Average RMSE	Min / Max RMSE
1D UAVs, Parabolic	0.039	0.036 / 0.124
1D Static Parabolic	0.054	0.0459 / 0.059
1D UAVs ULA	0.060	0.029 / 0.164
1D Static ULA	0.108	0.048 / 0.426
2D 2UAVs ($\Delta\boldsymbol{\theta}$)	0.373	0.204 / 0.684
2D 3UAVs ($\Delta\boldsymbol{\theta}$)	0.057	0.047 / 0.377
2D 3UAVs (no $\Delta\boldsymbol{\theta}$)	0.069	0.049 / 0.176
2D 4UAVs ($\Delta\boldsymbol{\theta}$)	0.588	0.468 / 0.737
2D 3UAVs Dec* ($\Delta\boldsymbol{\theta}$)	0.105	0.092 / 0.213
2D 4UAVs Dec* ($\Delta\boldsymbol{\theta}$)	0.071	0.065 / 0.149
2D 4UAVs Dec* ($\Delta\boldsymbol{\theta}$) $\hat{s}_i$)	0.060	0.049 / 0.115

Dec*: Decentralized actor, $\Delta\boldsymbol{\theta}$: input parameter for actor

VII. ACKNOWLEDGMENT

This work is partially funded by QuadSAT. We would like to thank them for their support.

REFERENCES

[1] M. Alazab, M. Rieche, G. Del Galdo, W. Felber, F. Raschke, G. Siegert, and M. Landmann, "On-earth performance evaluation of SatCom on-the-move (SOTM) terminals," *Proceedings - IEEE Military Communications Conference MILCOM*, pp. 634–640, 2013.

[2] G. V. Forum, "Global VSAT Forum PERFORMANCE AND TEST GUIDELINES FOR TYPE APPROVAL OF ' COMMS ON THE MOVE ' MOBILE SATELLITE GVF-105 COMMUNICATIONS," pp. 1–30.

[3] A. P. Requirements, "Satellite Operator's Minimum Antenna Performance Requirements," no. November, 2019.

[4] T. Fritzel, H. J. Steiner, and R. Straus, "Advances in the Development of an Industrial UAV for Large-Scale Near-Field Antenna Measurements," *13th European Conference on Antennas and Propagation, EuCAP 2019*, no. EuCAP, pp. 1–3, 2019.

[5] J. Schreiber, "Antenna pattern reconstitution using unmanned aerial vehicles (UAVs)," *2016 IEEE Conference on Antenna Measurements and Applications, CAMA 2016*, pp. 6–8, 2017.

[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, and e. a. Van Den Driessche, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[7] B. Gaudet and R. Linares, "Adaptive guidance with reinforcement meta learning," *Advances in the Astronautical Sciences*, vol. 168, pp. 4091–4109, 2019.

[8] S. H. Semnani, H. Liu, M. Everett, A. De Ruiter, and J. P. How, "Multi-Agent Motion Planning for Dense and Dynamic Environments via Deep Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3221–3226, 2020.

[9] S. Rahili, B. Riviere, and S.-J. Chung, "Distributed Adaptive Reinforcement Learning: A Method for Optimal Routing," pp. 1–13, 2020.

[10] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 2974–2982, 2018.

[11] H. U. Sheikh and L. Bölöni, "Multi-Agent Reinforcement Learning for Problems with Combined Individual and Team Reward," 2020.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, United States, 2018.

[13] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 6380–6391, 2017.

[14] S. S. Y. M. Richard S. Sutton, David McAllester, "Policy gradient methods for reinforcement learning with function approximation," 7 1999.

[15] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *31st International Conference on Machine Learning, ICML 2014*, vol. 1, pp. 605–619, 2014.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[17] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *35th International Conference on Machine Learning, ICML 2018*, vol. 4, pp. 2587–2601, 2018.

[18] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, "Reducing Overestimation Bias in Multi-Agent Domains Using Double Centralized Critics," 2019.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, pp. 2067–2075, 2015.

[20] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, *A survey and critique of multiagent deep reinforcement learning*, vol. 33. Springer US, 2019.

[21] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, pp. 2145–2153, 2016.

[22] H. S. Shin, A. J. Garcia, and S. Alvarez, "Information-driven Persistent Sensing of a Non-cooperative Mobile Target Using UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 92, no. 3-4, pp. 629–643, 2018.

2021-04-27

Introduction to UAV swarm utilization for communication on the move terminals tracking evaluation with reinforcement learning technique

Omi, Saki

IEEE

Omi S, Shin H-S, Tsourdos A, et al., (2021) Introduction to UAV swarm utilization for communication on the move terminals tracking evaluation with reinforcement learning technique. In: 15th European Conference on Antennas and Propagation (EuCAP), 22-26 March 2021, Dusseldorf, Germany

https://doi.org/10.23919/EuCAP51087.2021.9411153