# Reinforcement Learning for Multiple HAPS/UAV Coordination: Impact of Exploration-Exploitation Dilemma on Convergence

Ogbonnaya Anicho, Philip B Charlesworth, Gurvinder S Baicher, and Atulya Nagar
Department of Mathematics and Computer Science
Liverpool Hope University
Email: anichoo@hope.ac.uk

Abstract—This work analyses the application of Reinforcement Learning (RL) for coordinating multiple Unmanned High Altitude Platform Stations (HAPS) or Unmanned Aerial Vehicles (UAVs) for providing communications area coverage to a community of fixed and mobile users. Multiple agent coordination techniques are essential for developing autonomous capabilities for multi-UAV/HAPS control and management. This paper examines the impact of exploration-exploitation dilemma on the application of RL for coordinating multiple UAVs/HAPS. In the work, it is observed that RL convergence is a challenge, as the RL algorithm struggles to find optimal positioning for maximum user coverage. This paper attempts to establish the source of the convergence issue with the RL technique for this specific application scenario. The work goes on to suggest methods to minimise this impact, and some insights for applying RL techniques for multi-agent coordination for communications area coverage.

Index Terms—Reinforcement Learning, Multi-Agent Coordination, HAPS.

## I. Introduction

High Altitude Platform Station (HAPS) is defined by the International Telecommunications Union (ITU) as “a station located on an object at an altitude of 20 to 50 Km and at a specified, nominal, fixed point relative to the earth” [1]. This stratospheric altitude (20-50 Km) is characterised by mild wind profile suitable for hosting platforms with minimal station keeping requirements [2]. HAPS is suitable for providing persistent communications coverage to mobile and fixed users using its unique technical strengths which combines those of terrestrial and satellite communication systems [2]–[4]. The capacity to offer large footprints with signal latency similar to terrestrial systems further places it as a dominant aerial infrastructure. As an aerial platform, it can be easily recovered and redeployed to meet various operational scenarios, an additional capability that neither satellite nor terrestrial systems can offer effectively [5].

Unmanned HAPS aircraft do not have a human pilot physically on board to control the aircraft, however, such aircraft can be controlled remotely [6]. The concept of coordination is at the core of this research because the state of the art in operating unmanned HAPS systems require at least two (2) or more ground-based crew members overseeing various aspects of mission planning, flight control, sensor operation and data assessment; also known as many-to-one ratio [7], [8]. The current capability suggests therefore, that deploying multiple HAPS platform may be technically and economically challenging, as operating complexity and cost will scale with increase in the number of HAPS platforms. The challenge of flipping the many-to-one ratio to one-to-many ratio is at the core of the multiple HAPS coordination problem. To solve the operating ratio problem will involve designing HAPS platforms with some level of autonomy. Autonomy will eliminate the need for direct human intervention on many operational levels and elevate HAPS platforms/systems to higher layers in the decision making logic hierarchy. Another challenge lies in defining, designing and integrating autonomy solutions and concepts relevant to each use-case or problem. Deploying multiple HAPS is required to extend area coverage capacity using a network of HAPS. Area coverage in this context refers to a form of blanket coverage as defined by Gage and Howard et al [9], [10], but achieved in this case by the dynamic arrangement of HAPS to provide communications services over an area of interest.

However this work focuses on analysing the application of Reinforcement learning (RL) in the multiple HAPS coordination problem for communications area coverage. The paper specifically examines the impact of the exploration-exploitation dilemma on the convergence of the RL algorithm as applied to multiple HAPS coordination. RL also known as adaptive (or approximate) dynamic programming (ADP) is now a popular technique in solving complex sequential decision-making problems [11]. It is a paradigm of learning whereby the agent (HAPS in this case) learns through exploring or interacting with its environment. These interactions involve the agent taking actions that trigger transitions from one state to another with associated rewards or punishments. In the literature it is stated that RL algorithms should converge to optimal solutions with probability of one [11], however, this has not been the case with the RL implementation in this work. The motivation of this work is to establish the impact of the exploration-exploitation dilemma on the convergence of the RL algorithm in the context of this work.

In this paper, section I gives an overview of HAPS and the multiple HAPS coordination problem and the RL concept. Section II reviews the application of RL in various contexts in unmanned aerial systems; while section III, describes the modelling and simulation background of the work. In section V, simulation results and analysis are presented. Finally, section VI draws conclusions on the work and considers future work.

## II. Reinforcement Learning technique in Unmanned Aerial Systems

RL has been applied in varying contexts to address problems in the unmanned aerial systems for instance, Pham et al [12], proposed a distributed Multi-Agent Reinforcement Learning (MARL) algorithm to tackle the problem of UAV team cooperation to address the issue of fully covering an unknown area, however, the work highlighted that the mission could be achieved without a mathematical model but was limited in application specifics. The work by [13] attempted to address the issue of learning convergence by applying adaptive state focus Q-learning. In order to solve the convergence problem, the learner dynamically expands the state space by adding more state information (state information of other agents). The challenge with this approach is the assumption that the state information of other agents will be available or better which may not be the case. This approach breaks down in any event where the learner is unable to access the required information and introduces a weak link in the solution and amplifies the risk of slow or no convergence. Similarly RL was applied to asymptotically converge UAV agents in optimal configurations [14]; address UAV flocking problem (using Q-learning) [15]; and achieve a UAV and UGV (Unmanned Ground Vehicle) coordination task [16]. The cited papers did not fully analyse the exploration-exploitation dilemma or how it impacted the convergence of the algorithm. There is a consensus that exploration-exploitation dilemma is a standard challenge in any RL implementation [11], [17], however, it is important to examine the impact of this phenomenon in different contexts. This paper does not intend to provide an exhaustive list of all reinforcement learning based UAV applications but to sample out implementations that reinforces the context of the work.

### A. Q-learning Approach

In this work Q-learning approach was adopted to implement reinforcement learning; the central idea in the Q-learning algorithm is to store the state-action pair value Q(s, a) called Q-values of each iteration as the agents interact with the environment (Q stands for “Quality”). At the beginning of the simulation the Q-values are initialised to zero and stored in a table or an array. The agent visits some state s, and takes action a, and then transits another state. The immediate reward gained from this action is stored and the Q-value updated using the following mathematical relationship [11], [17];

$$Q(\mathrm{s},\mathrm{a}) \approx (1-\alpha)Q(\mathrm{s},\mathrm{a}) + \alpha\Big[\mathrm{r} + \gamma\, max_{t+1} Q(\mathrm{s}_{t+1},\mathrm{a}_{t+1})\Big] \tag{1}$$

Where r denotes the reward at time t, $0 < \alpha < 1$ is a given learning rate and $\gamma$ is discount factor. The expression is used to update the Q-table until the values converge to a near-optimal solution (see algorithm 1). In the simulation carried out, the HAPS are defined as agents and user mobility modelled as part of the environment and ’states’ are pre-selected and fixed coordinates (beacons). The agent can execute two action set: Relocate from or Remain within a ’state’ as user density changes due to user mobility. Reward (or penalty) signals are fed back to the HAPS to reinforce actions that influence goals (e.g. maximise user coverage) positively or otherwise.

Algorithm 1 Q-Learning Pseudo-code

```
1: Input: States and Actions Set (M and N)
2: Create Q-table : M by N matrix
3: Initialise Q-table to zero.
4: Visit state (s), Take action (a), Receive reward (r)
5: Estimate Q-value of next state and Update Q-table.
6: if 1-$\epsilon$ = True then ▷ Probability of choosing Greedily
7:     Choose action from Maximum Q-value (Exploitation)
8:     Go To step 3
9: else if 1-$\epsilon$ = False then ▷ Probability of not choosing Greedily
10:    Choose random action (Exploration)
11:    Go To step 3
12: end if
```

## III. Modelling and Simulation background

The simulation was run with four (4) HAPS covering an area with about 500 users distributed over the area of interest. The initial distribution of the HAPS and the ground users is shown in figure 1. The ground users move randomly with a mobility model that is not predictable or known to the HAPS in advance.

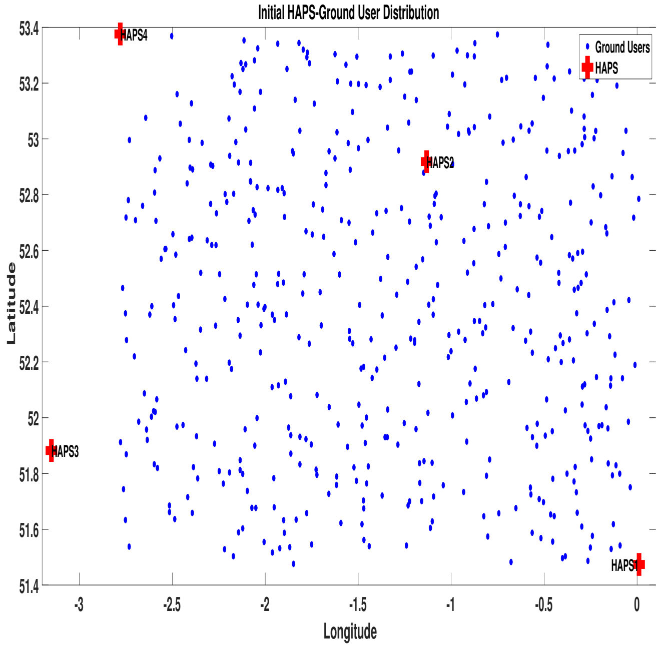

Fig. 1. Initial HAPS versus Ground Users Distribution

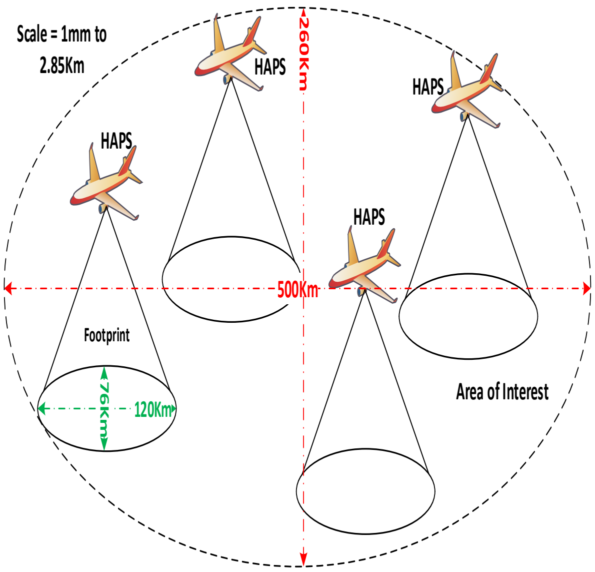

Fig. 2. Area of Interest versus HAPS

At 20km altitude, and about 22 degrees elevation (angle from the user's local horizon to the HAPS), and 135 degrees HPBW, each of the simulated HAPS has a footprint covering an area of about 7160km$^2$, with semi-minor axis of 38km and semi-major axis of 60km. The entire area of interest covers about 102,101km$^2$ with semi-minor axis of 130km and semi-major axis of 250km, see figure 2 (drawn to scale). Due to the size of the area of interest and the HAPS footprint size, only 4 HAPS were deployed to allow room for testing out the coordination algorithm. Making the area of interest more crowded by deploying more HAPS would have defeated the aim of the experiment. In practical applications the number of HAPS that can be deployed will always be a constraint due to operational and economic reasons. The users are randomly spread across the area of interest and the goal is to maximise communications area coverage for the users through autonomous coordination of the 4 HAPS.

## IV. Analysing RL Hyper-Parameters

In the RL algorithm design and application, hyper-parameters are essential to achieving optimal outcomes. Hyper-parameters are those parameters that are fixed before the algorithm is applied to the simulation. There are 3 hyper-parameters critical to RL algorithms i.e. epsilon-greedy ($\epsilon$ ), learning rate ($\alpha$) and discount factor ($\gamma$). For the purpose of this paper only the epsilon-greedy ($\epsilon$ ) hyper-parameter was analysed as it directly controls the exploration-exploitation phenomenon. The learning rate ($\alpha$) and discount factor ($\gamma$) hyper-parameters were kept constant in order to isolate the impact of varying the epsilon-greedy ($\epsilon$ ) values.

### A. Epsilon-greedy ($\epsilon$) Parameter

This parameter controls how the HAPS balances the exploration-exploitation dilemma. In this method the HAPS takes a greedy action with the probability of 1-$\epsilon$ and random action with probability $\epsilon$. However, RL environments can be stationary or non-stationary stochastic depending on how the properties of the environment vary with time [11], [17]. Each environment may require a different approach to implementing $\epsilon$ to achieve optimal results. The HAPS environment can be described as non-stationary stochastic and therefore requires careful approach to tuning or decaying of $\epsilon$. The methodology applied involved varying the values ($\epsilon$) and thereby altering the probability with which the HAPS chooses a random action (exploitation) over a known action with highest pay-off (exploration). Which implies that the HAPS will be expected to deliberately skip taking a greedy action (exploitation) and instead try out a new action without any guarantees of maximum pay-off (exploration). In the next section the result of the experiment carried out is analysed to further demonstrate the exploration-exploitation dynamics and its relationship to RL algorithm convergence in the context of this work.

## V. Results & Analysis

A practical approach for applying $\epsilon$ to the research problem was to experimentally test how each value of $\epsilon$ performed and to draw inferences based on outcomes. A simulation was carried out where the values of $\epsilon$ was tested by randomly choosing 4 different values of $\epsilon$ that covered the possible spectrum of values. Each extreme of the spectrum represents maximal or minimal exploration/exploitation tendencies i.e. epsilon ($\epsilon$ ) values range from 0 to 1; 0 signifies maximum exploitation while 1 means maximum exploration. For a non-stationary stochastic environment like the one under consideration, an RL algorithm design that can exploit and explore in the right balance continuously is needed. In the following simulation, the 4 HAPS are deployed and coordinated using the RL algorithm. The performance of the HAPS were measured based on the number of users covered locally by each HAPS (local coverage) and collectively by all HAPS (global coverage). The values of $\epsilon$ are changed for each run of the experiment and the performance captured. The convergence of the algorithm for the individual HAPS and the entire HAPS system is further analysed below after running the simulation for 21600 time steps.

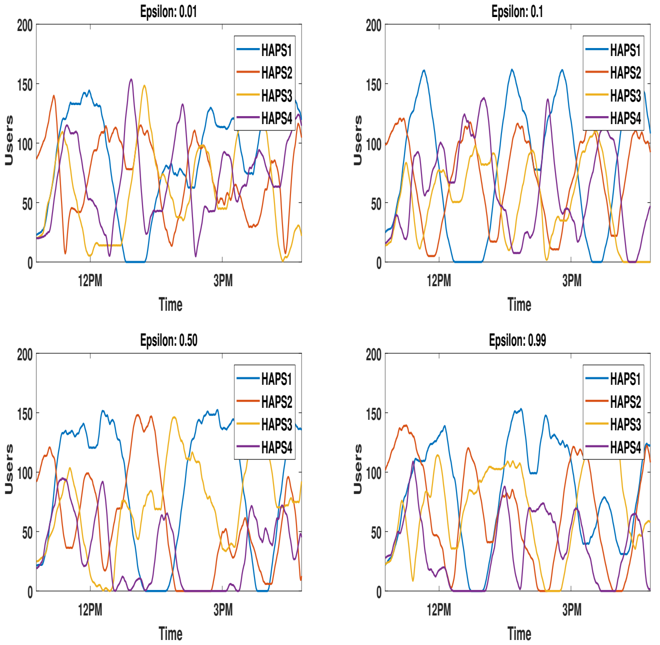

Fig. 3. Local Coverage Performance with different Epsilon Values

Figures 3 and 4 shows the local and global coverage results for 4 different values of epsilon (0.01, 0.1, 0.5 and 0.99). As explained, the 0.01 epsilon value will represent a highly exploitative HAPS with a policy based on 99% exploitation and 1% exploration. Conversely 0.99 epsilon value signifies very high exploration policy with 99% exploration and 1% exploitation. The results show that regardless of the value of $\epsilon$ the RL algorithm still had issues with convergence.

At the local HAPS performance level (see figure 3), each HAPS showed convergence challenges for all values of $\epsilon$. This is readily explained as the impact of the HAPS trying to balance the decision of either remaining in one location or relocating to an entirely new location to cover users. The HAPS will make less random relocation decisions if the value of $\epsilon$ is low and higher random relocation decisions at higher thresholds of $\epsilon$. In figure 4, the global performance of all the HAPS combined is displayed. The convergence noise is reduced at this global level of the results. However, to further understand which value of $\epsilon$ performed better at the global level, a statistical test was performed to provide an empirically verifiable analysis. Analysis of Variance (ANOVA) tests were carried out to establish any statistical significance in the results shown in figures 3 and 4.

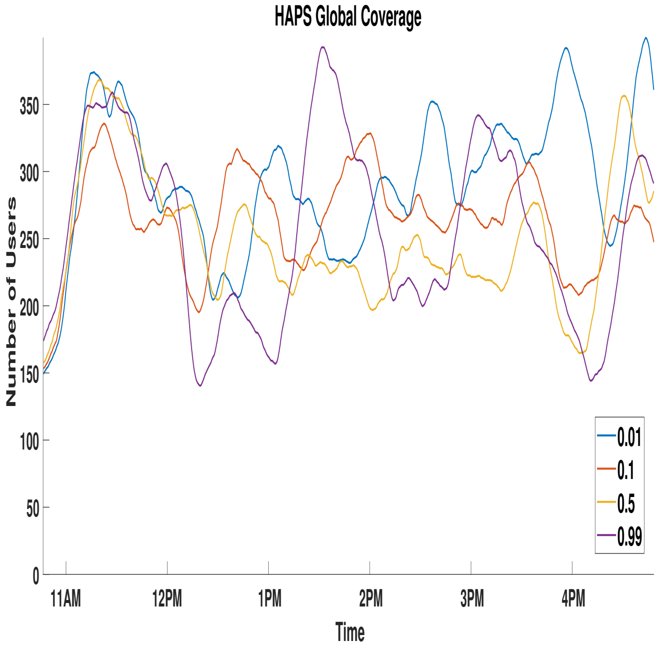

Fig. 4. Global Coverage Performance with different Epsilon Values

TABLE I
ANOVA Data - Coverage Variance

| Source | SS | df | MS | F | Prob>F |
|---|---|---|---|---|---|
| Columns | 23660857.51 | 3 | 7886952.50 | 2852.52 | 0.00 |
| Error | 243300708.60 | 87996 | 2764.91 | - | - |
| Total | 266961566.10 | 87999 | - | - | - |

The ANOVA data from table I showed a p-value (Prob>F) of 0.00; signifying that the results for the global coverage in figure 4 have different group means. Which implies that the null hypothesis that the group means are the same can be rejected; while the alternate hypothesis of different group means can be accepted. The box plot of the global coverage result (see figure 5) graphically supported this position as the 4 groups showed different statistical profiles (note different median, maximum and minimum values). Furthermore, the multiple comparison plot (figure

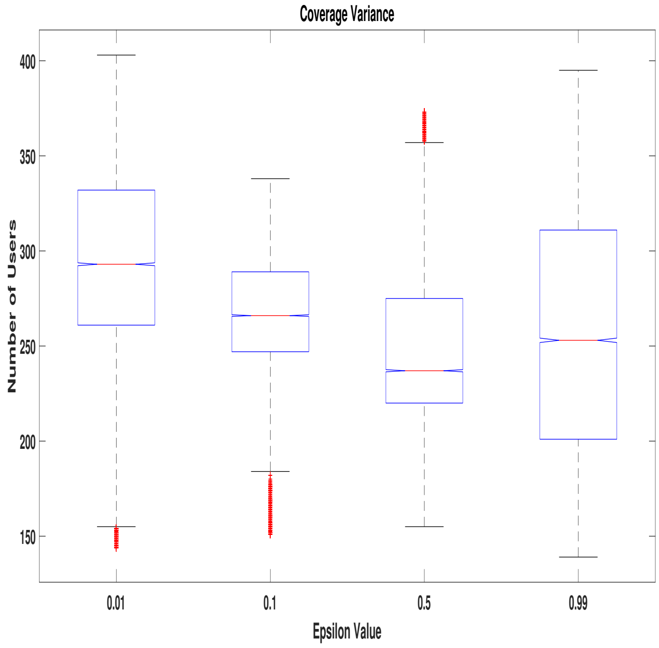

Fig. 5. Global Coverage Performance Box Plot for all Epsilon Values

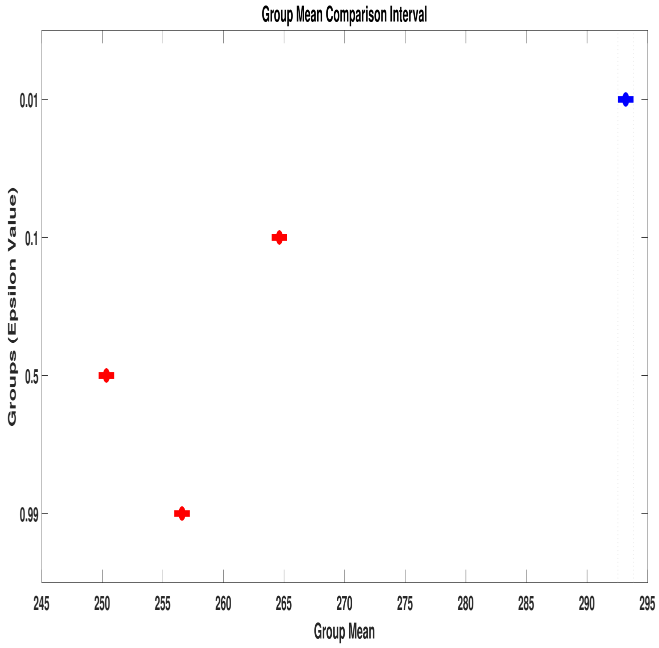

Fig. 6. Group Mean Comparison Interval for all Epsilon Values

6) showed that 0.01 epsilon value had the highest group mean signifying better performance. Another insight from the comparison plot is that none of the groups showed any overlap (another justification that all groups are statistically different from each other). From the analysis it can be seen that performance improved or declined as the exploration-exploitation balance was adjusted by varying $\epsilon$, thereby impacting the convergence behaviour of the algorithm. The lower $\epsilon$ value of 0.01 recorded the best average coverage results which implies that the HAPS exploited more and explored very less. This outcome further highlights the impact the value of $\epsilon$ can have on the convergence of the algorithm and by extension the performance of the HAPS in providing coverage. However, there may be some risk to this policy as the density of users may evolve with time and a longer run of this experiment may provide a different outcome.

## VI. Conclusions and Future Work

This paper investigates the impact of the exploration-exploitation dilemma on the convergence of the RL algorithm in the context of this work where multiple HAPS are coordinated for communications area coverage. The exploration-exploitation dilemma is a well known issue in the application of RL algorithms where the agents (in this case the HAPS) have to decide the probability with which exploration is done over exploitation. The dilemma lies in the inevitable risk each decision poses as continuous exploitation may deprive the HAPS of finding new and better solutions while continuous exploration may mean never settling or converging to a solution. The impact of the exploration-exploitation dilemma on the convergence of the algorithm therefore affects the performance of the RL algorithm. This work analysed the impact of this phenomenon in the task of 4 HAPS providing communications coverage to about 500 users. The results show that the exploration-exploitation dilemma impacts RL algorithm convergence and therefore affects coverage performance. However, the lower values of $\epsilon$ showed better performance statistically and may be suitable for dynamic environments.

Future work will consider running the simulation for much longer time steps and analysing convergence and coverage performance. In the current implementation the $\epsilon$ values were static, future work will consider applying dynamic $\epsilon$ values to further investigate how convergence may be impacted.

## References

[1] I. T. U. (ITU), “Terms and definitions,” Radio Regulations Articles, 2016.

[2] F. A. d’Oliveira, F. C. L. de Melo, and T. C. Devezas, “High-Altitude Platforms - Present Situation and Technology Trends,” Journal of Aerospace Technology and Management, vol. 8, pp. 249 – 262, 09 2016.

[3] D. Grace and M. Mohorcic, Broadband Communications via High Altitude Platforms. Wiley, 2011.

[4] ITU, “Identifying the Potential of New Communications Technologies for Sustainable Development,” Broadband Commission For Sustainable Development: Working Group on Technologies in Space and the Upper-Atmosphere, Tech. Rep., 2017.

[5] O. Anicho, P. B. Charlesworth, G. S. Baicher, and A. Nagar, "Integrating Routing Schemes and Platform Autonomy Algorithms for UAV Ad-hoc & Infrastructure Based Networks," in 28th International Telecommunication Networks and Applications Conference (ITNAC), 28th International Telecommunication Networks and Applications Conference (ITNAC). IEEE, Nov. 2018.

[6] R. Stengel, Flight Dynamics. Princeton University Press, 2004.

[7] V. Hehtke, J. Kiam, and A. Schulte, "An Autonomous Mission Management System to Assist Decision Making for a HALE Operator," Deutscher Luft-und RaumfahrtKongress, 2017.

[8] T. B. Chen, "Management of Multiple Heterogenous Unmanned Aerial Vehicles Through Capacity Transparency," Ph.D. dissertation, Queensland University of Technology, 2016.

[9] D. W. Gage, "Command and Control of Many-Robot Systems," Unmanned Systems, 1992.

[10] A. Howard, M. Mataric, and G. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem." International Symposium on Distributed Autonomous Robotic Systems, Jun. 2002.

[11] Abhijit Gosavi, A Tutorial for Reinforcement Learning. Springer, 2017.

[12] H. Xuan Pham, H. La, D. Feil-Seifer, and L. Nguyen, "Cooperative and Distributed Reinforcement Learning of Drones for Field Coverage," 03 2018.

[13] L. Busoniu, B. Schtter, and R. Babuska, "Multiagent Reinforcement Learning with Adaptive State Focus," in BNAIC 2005, K. Verbeeck, K. Tuyls, A. Nowe, B. Manderick, and B. Kuijpers, Eds. Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence, Oct. 2005, pp. 35–42.

[14] A. Adepegba, S. Miah, and D. Spinello, "Multi-Agent Area Coverage Control using Reinforcement Learning," in Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Society Conference, 2016.

[15] S.-M. Hung and S. Givigi, "A Q-Learning Approach to Flocking with UAVs in a Stochastic Environment," IEEE Transactions on Cybernetics, vol. 47, no. 1, pp. 186–197, Jan. 2017.

[16] H. Nguyen, L. Bui, M. Garratt, and H. Abbass, "Apprenticeship Bootstrapping: Inverse Reinforcement Learning in a Multi-Skill UAV-UGV Coordination Task," in Proceedings of the 17th International Conference on Autonomous and Multiagent Systems, M. Dastani, G. Sukthankar, E. Andre, and S. Koenig, Eds., Jul. 2018, pp. 2204–2206.

[17] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction. MIT Press, 2017.