FlyBeam: Echo State Learning for Joint Flight and Beamforming Control in Wireless UAV Networks

Sabarish Krishna Moorthy[1], Zhangyu Guan[1], Scott Pudlewski[2], and Elizabeth Serena Bentley[3]
[1]Dept. of Electrical Engineering, State University of New York (SUNY) at Buffalo, Buffalo, NY 14260, USA
[2]Georgia Tech Research Institute (GTRI), Atlanta, GA 30332, USA
[3]Air Force Research Laboratory (AFRL), Rome, NY 13440, USA
Email: {sk382, guan}@buffalo.edu, scott.pudlewski@gtri.gatech.edu, elizabeth.bentley.3@us.af.mil

***Abstract*—This paper aims at designing high-data-rate swarm UAV networks with distributed beamforming capabilities. The primary challenge is that the beamforming gain in swarm UAV networks is highly affected by the UAVs' flight altitude, their movements and the resulting intermittent link blockages, as well as the availability of channel state information (CSI) at individual UAVs. To address this challenge, we propose *FlyBeam*, a learning-based framework for joint flight and beamforming control in swarm UAV networks. We first present a mathematical formulation of the control problem with the objective of maximizing the throughput of swarm UAV networks by jointly controlling the flight and distributed beamforming of UAVs. Then, a distributed solution algorithm is designed based on a combination of Echo State Network learning and online reinforcement learning. The former is adopted to approximate the utility function for individual UAVs based on online measurements, by jointly considering the unknown blockage dynamics and other factors that affect the beamforming gain. The latter is used to guide the exploitation and exploration in *FlyBeam*. The effectiveness of *FlyBeam* is evaluated through an extensive simulation campaign. Results indicate that significant (up to 450%) beamforming gain can be achieved by *FlyBeam*. We also investigate the effects of blockages and UAV flight altitude on the beamforming gain. It is found that, which is somewhat surprising, higher *(rather than lower)* beamforming gain can be achieved by *FlyBeam* with denser blockages in swarm UAV networks.**

***Index Terms*—Swarm UAV Networks, Distributed Beamforming, Echo State Network, Reinforcement Learning.**

I. Introduction

Unmanned aerial vehicles (UAVs) have been envisioned as an enabling technology for a wide set of new applications, because of their features of fast deployment, high mobility and small size [1]–[3]. Examples of these applications include small cells with flying base stations, UAV-aided guidance, swarm networking for battlefield sensing and data collection, emergency wireless networking in the aftermath of disasters, among others. While UAVs can certainly enable a wide set of new applications, their wide deployment will impose a significant burden on the capacity of the underlying wireless networks. In this work, we focus on designing high-data-rate wireless UAV networks by exploring spatial diversity through collaborative beamforming among the UAVs.

Since it is not easy to mount many antennas on individual UAVs because of their small size, in this work we consider a swarm of UAVs collaborating with each other to perform distributed beamforming. One of the primary challenges is in the formation of UAV clusters for collaborative beamforming. In swarm UAV networks, beamforming can be typically accomplished in two phases. In the first phase, the UAVs estimate the channel state information (CSI) of the wireless links from them to the users they serve and then share the obtained CSI among the UAVs; in the second phase, the UAVs collaborate with each other to perform distributed beamforming for data transmission. While higher beamforming gain can be achieved by forming a larger UAV cluster with more antennas, it takes longer for the CSI sharing in the first phase and hence reduces the time available for data transmission in the second phase. Therefore, a tradeoff needs to be achieved between beamforming gain and channel utilization. Moreover, the beamforming gain is closely coupled with the statistical behaviors of the wireless channels, which are affected by the flight altitude of the UAVs, the dynamic movements of UAVs and the resulting intermittent existence of blockages. Roughly speaking, the wireless communication is dominated by Non-line-of-sight (NLOS) transmissions when the UAVs fly lower, while line-of-sight (LOS) channels are dominant when they fly higher. Additionally, the interference level in the network can be effectively lowered with more and larger blockages, and this also affects the beamforming gain.

To account for these coupled factors that jointly affect the beamforming gain in swarm UAV networks, in this paper we propose *FlyBeam*, a learning-based framework for joint flight and beamforming control in swarm UAV networks with unknown blockage dynamics. The main contributions of the paper are as follows.

- We first present a mathematical formulation of the *FlyBeam* control problem, where the objective is to maximize the aggregate capacity of swarm UAV networks with a set of single-antenna UAVs collaborating with each other to perform distributed beamforming, by jointly

ACKNOWLEDGMENT OF SUPPORT AND DISCLAIMER: (a) Contractor acknowledges Government's support in the publication of this paper. This material is based upon work funded by AFRL, under AFRL Contract No. FA8750-20-1-0501. (b) Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

Distribution A. Approved for public release: Distribution unlimited AFRL-2020-0533 on December 17, 2020.

considering the movements of the UAVs, the blockage- and altitude-dependent wireless channels as well as CSI sharing among the UAVs.

- We then design a distributed solution algorithm to solve the *FlyBeam* control problem based on a combination of Echo State Network (ESN) learning and reinforcement learning (RL). ESN provides an approximation of the utility functions of the UAVs with unknown blockage dynamics based on online measured data, and RL is adopted to achieve a good tradeoff between exploitation and exploration in *FlyBeam*.
- We evaluate the effectiveness of *FlyBeam* by conducting an extensive simulation campaign. Results indicate that that up to 450% capacity gain can be achieved by enabling collaborative beamforming among UAVs. It is also shown that the beamforming gain can be increased (*rather than decreased*) in swarm UAV networks with denser blockages. The effects of periodic training of ESN on the capacity achievable by *FlyBeam* are also studied.

To the best of our knowledge, *FlyBeam* is the first reinforcement learning framework for swarm UAV networks with ESN-based utility function approximation jointly considering UAV movement, blockage- and altitude-dependent wireless channels as well as CSI sharing in distributed beamforming.

II. Related Work

Wireless UAV networking has drawn significant research attention over the past years [4]–[9]. For example, in [4] Azari et al. study power control for wireless communications among single-antenna UAVs in cellular networks. In [5] the authors maximize the throughput in UAV-enabled orthogonal frequency-division multiple access (OFDMA) systems with delay-constrained data traffic. The authors of [6] jointly optimize the trajectory and communication in multi-UAV wireless networks to achieve better fairness among users. In [7], we propose a new framework for automated control of swarm UAV networks based on the recent results on principled software-defined wireless networking. Readers are referred to [8], [9] and references therein for a good survey of the main results in this area. *Different from these works, which focus on non-collaborative single-antenna UAV communications, in this paper we explore spatial diversity in swarm UAV networks by allowing the UAVs to perform distributed beamforming.*

Distributed beamforming has also been extensively studied in wireless networks [10]–[19]. For example, in [10], Mohanti et al. propose an experimental framework that uses software-defined radios to assign beamforming weights that ensure a high level of directivity. In [11], the authors study beamforming vector generation and updating based on recursive channel estimation and updating. The beamforming algorithms for UAV swarm is studied in [12] where the swarm is modeled as a morphing volumetric random array with an assumption that each element is enabled with orientation awareness. However, different from our work, [12] did not consider the dynamic movements and the effects of blockages on the beamforming gain. The authors of [13] present a cooperative communication for cache-enabled UAVs to jointly decide the UAV placement and transmit beamforming based on outdated CSI information. Similar to [12], [13] did not consider the effect of blockages either. In [14] Yuan et al. consider single UAV-user pair and develop a deep learning-based predictive beamforming scheme that can recover from beam misalignment caused by UAV jittering. Position-based beamforming is studied in [15] to enhance the capacity of UAV communications in LTE networks in the presence of direction-of-arrival estimation errors. In [16], the authors discuss the feasibility and enabling techniques for distributed beamforming in swarm UAV networks. In our previous work [17], we design distributed algorithms for joint power, association and flight control in swarm UAV networks with each UAV endowed with a large number of antennas. Please refer to [18], [19] and references therein for an extensive survey of the latest results in this area. *None of these of work has studied distributed beamforming in swarm UAV networks by explicitly considering all the factors that affect the beamforming gain discussed in Section I, including the flight of UAVs, the resulting dynamic blockages, and CSI sharing among UAVs.*

III. System Model and Problem Formulation

We consider swarm UAV networks with a set $\mathcal{M}$ of single-antenna UAVs collaborating with each other to serve a set $\mathcal{U}$ of ground users. The UAVs are allowed to form a virtual MIMO UAV cluster to enhance the quality of the aerial-ground wireless links through distributed beamforming. Focusing on the downlink communications in this setting, our objective is to investigate the effects of those factors that affect the beamforming gain, including blockages in the network, flight altitude of the UAVs as well as the altitude-dependent fading channels, among others. The results obtained in this paper can also be extended to uplink scenarios. Next we describe the blockage, channel and beamforming models sequentially.

Blockage Model. Denote $\mathcal{S}$ and $|\mathcal{S}|$ as the set and the number of blockages in the network, respectively. For each blockage $s \in \mathcal{S}$, let C_s, L_s, W_s, H_s and θ_s represent the center, length, width, height and orientation of the blockage, respectively. The orientation θ_s is considered to be uniformly distributed in $[0,\ 2\pi]$, and the other blockage parameters are randomly generated with the average corresponding to the typical size of blockages in real networks, e.g., the buildings. Let $\mathbf{P}_s^{\text{blk}}(C_s, L_s, W_s, H_s, \theta_s)$ represent the set of points contained in blockage $s \in \mathcal{S}$.

Denote the location vector of user $u \in \mathcal{U}$ as $\mathbf{cod}_u = (x_u,\ y_u,\ z_u)$, with $x_u,\ y_u,\ z_u$ representing the x-, y- and z-axis coordinates, respectively. Similarly, denote $\mathbf{cod}_m = (x_m,\ y_m,\ z_m)$ as the location vector of UAV $m \in \mathcal{M}$. Let $\mathbf{P}_{mu}^{\text{link}}$ denote the set of points on the line connecting UAV $m \in \mathcal{M}$ and user $u \in \mathcal{U}$. Further denote $\mathbf{P}_{mus}^{\text{xt}} = \mathbf{P}_{mu}^{\text{link}} \cap \mathbf{P}_s^{\text{blk}}(C_s, L_s, W_s, H_s, \theta_s)$ as the resulting intersection set of points. Then, given the set $\mathcal{S}$ of blockages, the total number of blockages on the link between user u and UAV m, denoted as S_{mu}, can be expressed as

$$S_{mu} = \sum_{s \in \mathcal{S}} \mathrm{I}(\mathbf{cod}_m, \mathbf{cod}_u, s), \ \forall m \in \mathcal{M}, \ \forall u \in \mathcal{U}, \tag{1}$$

where $\mathrm{I}(\mathbf{cod}_m, \mathbf{cod}_u, s) = \begin{cases} 1, & \text{if } \mathrm{P}^{\mathrm{xt}}_{mus} \neq \phi \\ 0, & \text{otherwise} \end{cases}$ is the indicator function taking the value of 1 if blockage s is blocking the link and 0 otherwise.

Channel Model. Denote $H_{mu}(f)$ as the path loss for the link between UAV $m \in \mathcal{M}$ and ground user $u \in \mathcal{U}$ operating in frequency band $f \in \mathcal{F}$. Then we model $H_{mu}(f)$ as in [20] as follows:

$$H_{mu}(f) = \eta_0^{S_{mu}} \left(\frac{4\pi f}{C} \right)^2 (d_{mu})^{\alpha_{mu}(f)}, \tag{2}$$

where C is the speed of light, $\alpha_{mu}(f)$ is the path-loss exponent for the link between UAV $\in \mathcal{M}$ and ground user $u \in \mathcal{U}$ in frequency band f, S_{mu} defined in (1) represents the number of blockages in the link, $\eta_0 \in [0, \ 1]$ is the per-blockage absorption coefficient [21], and finally $d_{mu} = d_{mu}(\mathbf{cod}_m, \mathbf{cod}_u)$ denotes the distance between UAV $m \in \mathcal{M}$ and user $u \in \mathcal{U}$.

The transmission time is divided into a set of consecutive time slots. We consider block fading channels in each time slot, i.e., the channel coefficient is considered to be fixed in each time slot and change to another random value following certain distribution in the next. Rician fading model is adopted to characterize the fading behavior of the wireless channels, with the Rician factor K depending on whether the link is blocked or not. For NLOS links, i.e. $S_{mu} \neq 0$ in (2), factor K is set to 0; for LOS transmissions, i.e., $S_{m,u} = 0$, factor K is given as $K = 13 - 0.03 \times d_{mu}$ [22]. Denote the resulting channel fading coefficient as h_{mu} for the wireless link between UAV $m \in \mathcal{M}$ and ground user $u \in \mathcal{U}$.

We further consider pilot-based channel estimation, which can be accomplished in either sequential or simultaneous manner. In the former case, the ground users broadcast their pilot signals in each time slot sequentially, and hence the problem of pilot contamination can be avoided but at the cost of more estimation time. In contrast, with simultaneous channel estimation all the users broadcast their pilot signals at the same time, and hence it takes less time at the cost of lower estimation accuracy. In this work, we take sequential channel estimation as an example, while the results can be easily extended to the simultaneous case. Denote $\mathbf{y}_m = [y_{m1}, \cdots, y_{mN_{\mathrm{plt}}}]$ as the pilot signal received by UAV $m \in \mathcal{M}$, with $y_{m\nu}, \ \nu = 1, \cdots, N_{\mathrm{plt}}$ being the vth symbol of the received pilot signal. Then $\mathbf{y}_m$ can be given as $\boldsymbol{y}_m = \sqrt{H_{mu}} h_{mu} \boldsymbol{p}_u + n_m^0$, where $\boldsymbol{p}_u = [p_{u1}, \cdots, p_{uN_{\mathrm{plt}}}]$ is the pilot sequence of user $u \in \mathcal{U}$, and $n_m^0 = (n_{m\nu}^0)_{\nu=1}^{N_{\mathrm{plt}}}$ is the vector of Additive White Gaussian Noise (AWGN) at UAV $m \in \mathcal{M}$. Let $\widetilde{h}_{mu}$ represent the estimated channel gain for the link between UAV $m \in \mathcal{M}$ and user $u \in \mathcal{U}$. Then, if a least-square estimator [23] is considered, we have

$$\widetilde{h}_{mu} = \boldsymbol{y}_m \times \boldsymbol{p}_u^{\dagger} (\boldsymbol{p}_u \times \boldsymbol{p}_u^{\dagger})^{-1}, \tag{3}$$

where $(\cdot)^{\dagger}$ denotes the conjugate transpose and $(\cdot)^{-1}$ represents the matrix inverse operation.

Let $t_{\mathrm{est},u}$ denote the time overhead for the channel estimation for user $u \in \mathcal{U}$ and t_{est} as the total time overhead for all the users in $\mathcal{U}$. Then we have $t_{\mathrm{est}} = \sum_{u \in \mathcal{U}} t_{\mathrm{est},u}$. where $t_{\mathrm{est},u} = N_{\mathrm{plt}}/r$ with N_{plt} being the number of bits in each pilot sequence and r the data rate for pilot transmission.

Beamforming Model. Each UAV $m \in \mathcal{M}$ sends its CSI obtained above to a pre-selected leading UAV of the swarm, denoted as m' with $m' \in \mathcal{M}/m$, which will then calculate the beamforming weights for the whole swarm. Denote t_{csi} as the resulting time overhead, then we have $t_{\mathrm{csi}} = \sum_{m \in \mathcal{M}/m'} N_{\mathrm{csi}}/C_{mm'}$, where $C_{mm'}$ represents the capacity of the link between UAVs m and m', and N_{csi} is the amount of CSI data in bits to be shared by UAV $m \in \mathcal{M}$, i.e., $\widetilde{\boldsymbol{h}}_m = (\widetilde{h}_{mu})_{u \in \mathcal{U}}$ with $\widetilde{h}_{mu}$ obtained in (3). Here, we consider again sequential transmission for CSI sharing. Based on the collected CSI, the beamforming weights can be calculated at the leading UAV. Denote the resulting beamforming weight vector as $\boldsymbol{w} = (\boldsymbol{w}_m)_{m \in \mathcal{M}}$, where $\boldsymbol{w}_m = (w_{mu})_{u \in \mathcal{U}}$ with w_{mu} being the beamforming weight of UAV m for ground user u. For example, based on *Zero Forcing (ZF)* beamforming [24], $\boldsymbol{w}_m$ can be given as $\boldsymbol{w}_m = \boldsymbol{h}_m^{\dagger} \ (\boldsymbol{h}_m \boldsymbol{h}_m^{\dagger})^{-1}$, where $(\cdot)^{\dagger}$ denotes the conjugate transpose and $(\cdot)^{-1}$ is the matrix inverse operation. The obtained beamforming weight is then sent back to the corresponding UAVs for the actual use in the following data transmission. Denote the resulting time overhead as t_{beam}. Then, the overall time overhead denoted as t_{ovhd} can be written as

$$t_{\mathrm{ovhd}} = t_{\mathrm{est}} + t_{\mathrm{csi}} + t_{\mathrm{beam}}, \tag{4}$$

where t_{est}, t_{csi} and t_{beam} are the above defined time overhead for channel estimation, CSI sharing and beamforming weight feedback, respectively.

FlyBeam Control Problem. For a given frequency f, bandwidth B and coordinates $\mathbf{cod}_u, u \in \mathcal{U}$, as well as the set of blockages $\mathcal{S}$, the control objective of *FlyBeam* is to maximize the aggregate network capacity by jointly controlling the flight and beamforming of the UAVs, as formulated as follows.

$$\begin{aligned} & max \, \gamma B \sum_{u \in \mathcal{U}} \log_2 \left(1 + \frac{\sum\limits_{m \in \mathcal{M}} P_{mu} \widehat{H}_{mu} |\widehat{h}_{mu} \widehat{w}_{mu}|^2}{\sum\limits_{u' \in \mathcal{U}/u} \sum\limits_{m \in \mathcal{M}} P_{mu'} \widehat{H}_{mu'} |\widehat{h}_{mu'} \widehat{w}_{mu'}|^2 + N_u^0} \right), \\ & s.t. : \quad \sum_{u \in \mathcal{U}} P_{mu} \leq P_{\max}, \ \forall m \in \mathcal{M}, \end{aligned} \tag{5}$$

where $\widehat{H}_{mu} = H_{mu}(\mathbf{cod}_m)$, $\widehat{h}_{mu} = h_{mu}(\mathbf{cod}_m)$ and $\widehat{w}_{mu} = w_{mu}(\mathbf{cod}_m, \pi)$ denote the path loss, channel fading and beamforming weights for user u and UAV m, respectively, with π denoting the beamforming strategy; P_{mu} and $P_{mu'}$ are the transmission power of UAV $m \in \mathcal{M}$ allocated to users u and u', respectively; $P_{\max}$ is the maximum transmission power of each UAV; N_u^0 is the power of noise at ground user $u \in \mathcal{U}$; and finally $\gamma = \frac{t_{\mathrm{slt}} - t_{\mathrm{ovhd}}}{t_{\mathrm{slt}}}$ represents the channel utilization coefficient with t_{slt} being the duration of each time slot and t_{ovhd} defined in (4).

IV. FlyBeam Algorithm Design

The primary challenge in solving problem (5) is that the wireless channel hence the beamforming weight w_{mu} and the channel utilization coefficient γ are closely coupled with the UAV location variables $\mathbf{cod}_m$ and hence the resulting dynamic blockages, for which the complete information is unknown to the UAVs. To address this challenge, in this work we solve the problem by designing distributed control algorithms based on a combination of ESN learning [25] and reinforcement learning techniques. Particularly, the ESN is used to approximately model the mapping from the input signals to the output signals of a system, by training its input weights $\mathbf{W_{in}}$, the reservoir weights $\mathbf{W}$ and output weights $\mathbf{W_{out}}$ using a sigmoidal transfer function (e.g., hyperbolic tangent). Compared to traditional Neural Networks, which are computationally expensive, it is incredibly simple to train ESNs, while they are still able to model the complex time-varying behaviors of dynamical systems. In this work, we use ESN to approximate the utility function in (5), by, as described in Sec. III, jointly considering the UAV flight, the channel estimation and beamforming strategies, as well as the effects of blockages on beamforming. Based on the ESN approximated utility function, online reinforcement learning is then adopted to guide the exploitation and exploration in favor of higher aggregate capacity. The ESN module consists of four components: *Agent*, *Input*, *Action* and *Reward Function*. In *FlyBeam*, the ESN is implemented in individual UAVs, i.e., each UAV is an *Agent*.

ESN Input Design. Let $\mathcal{T}$ denote the total network running time. In each time slot $t \in \mathcal{T}$, each UAV $m \in \mathcal{M}$ feeds an *Input* (denoted as ρ_m^t) and a candidate *Action* (denoted as ξ_m^t) to its ESN module, which will then output the expected *Reward Function* value of the UAV in the next time slot. For UAV m the input to *FlyBeam*'s ESN module in time slot t, defined as $\rho_m^t \triangleq \{\mathbf{cod}_{-m}^t\}$, comprises of the locations of all the other UAVs $\mathbf{cod}_{-m}^t = (\mathbf{cod}_{m'}^t)_{m' \in \mathcal{M}/m}$ with $\mathbf{cod}_{m'}^t$ being the coordinate vector of UAV m' in time slot t. The dimension of ρ_m^t increases quadratically with the scale of the network which can slow down the training of *FlyBeam*'s ESN module and hence degrade the utility approximation accuracy in large-scale networks. To address this challenge, the network area is divided into a number $N_x \times N_y \times N_z$ of three-dimensional rectangles, each with $\frac{L_x}{N_x}$, $\frac{L_y}{N_y}$ and $\frac{L_z}{N_z}$ for width, length and height, respectively. Denote $\mathcal{N}$ as the set of the resulting rectangles. Each rectangle $n \in \mathcal{N}$ is represented using a vector $\mathbf{r}_n = (\widetilde{\mathbf{cod}}_n, \mathit{\Gamma}_n)$, where $\widetilde{\mathbf{cod}}_n$ is the coordinate vector of the center point of rectangle $n \in \mathcal{N}$, and $\mathit{\Gamma}_n = 0, 1, \cdots, N_x \times N_y \times N_z - 1$ is the index of the rectangle. Based on this policy the input of *FlyBeam*'s ESN module can be rewritten as $\rho_m^t = (\mathit{\Gamma}_{m'}^t)_{m' \in \mathcal{M}/m}$, with $\mathit{\Gamma}_{m'}^t$ is the the index of UAV m''s rectangle in time slot t.

ESN Action and Reward. Given input ρ_m^t for *FlyBeam*'s ESN module for UAV m in time slot t, UAV m makes its action decisions and observes an output of the action. To this end, UAV m chooses to move to a new rectangle in $\mathcal{N}$ except those occupied by other UAVs. The set of actions for UAV m, denoted as ξ_m^t for time slot t, can be written as $\xi_m^t = \{\mathit{\Gamma}_v | v \in \mathcal{N}/\{n^t(m'),\ m' \in \mathcal{M}/m\}\}$, where $n^t(m')$ represents the rectangle index of UAV m' in time slot t. The corresponding reward is defined as the aggregate network capacity achievable through distributed beamforming as defined by (5). Denote the resulting output capacity as C^t for time slot t.

ESN Training and RL Control. We train the designed ESN module based on online collected capacity data. In the training phase, the objective of the *FlyBeam* ESN module is to learn a model with output C^t that minimizes the root-mean-square error (RMSE) between C^t (i.e., predicted sum capacity) and C_{tgt}^t (i.e., target sum capacity) defined as

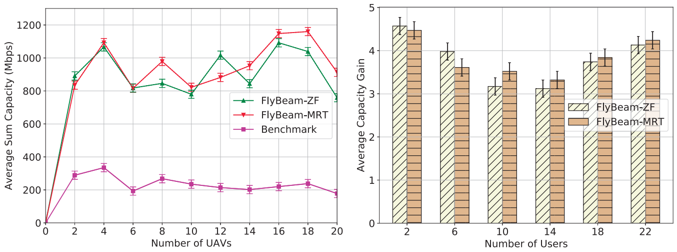
$$E(C^t, C_{\mathrm{tgt}}^t) = \frac{1}{N_{\mathrm{out}}} \sum_{i=1}^{N_{\mathrm{out}}} \sqrt{\frac{1}{|\mathcal{T}|} \sum_{t=1}^{|\mathcal{T}|} (C^t - C_{\mathrm{tgt}}^t)^2}, \quad (6)$$

where N_{out} denotes the output units of the ESN, $|\cdot|$ represents the cardinality of a set and $\mathcal{T}$ denotes the number of time slots in the training phase. To this end, we measure the sum capacity based on (5) and the measured sum capacity is used as the C_{tgt}^t in (6).

Based on the trained ESN-module, each UAV $m \in \mathcal{M}$ can determine its own optimal next-step location m as $\mathit{\Gamma}_m^{t*}$. However, this may lead to a local optimum for our swarm control problem defined in (5), which is not desirable. In this work we use reinforcement learning (RL) to guide the *exploration* and *exploitation* in the flight control of the UAVs. Reinforcement learning [26] has been widely used to solve complex problems that cannot be solved by conventional techniques. In this work, we consider RL algorithm with ϵ-greedy exploration strategy [26]. As the network runs, the online collected data will be used to update the training dataset and then retrain the ESN module. The details of the algorithm for RL control and ESN updating are omitted due to lack of space.

V. Performance Evaluation

We consider a swarm UAV network with area of $200 \times 200 \times 200\ \mathrm{m}^3$. The frequency and bandwidth are set to 2.1 GHz and 40 MHz, respectively. The transmission power of each UAV is set to 1 W. The LOS and NLOS path-loss exponents are considered to be 2 and 4, respectively. The duration of each time slot is set to 100 ms. Two beamforming schemes are considered: *Zero Forcing (ZF)* and *Maximum Ratio Transmission (MRT)* [24]. The corresponding *FlyBeam* schemes are referred to as *FlyBeam-ZF* and *FlyBeam-MRT*, respectively. We set $\mathcal{T}$ and $\mathcal{T}_{\mathrm{updt}}$ as 3000 and 1000 for initial training and training update phases, respectively. The simulations have been conducted over *SimBAG, an event-driven Simulator for Broadband Aerial-Ground wireless networks that has been developed in our recent research [2].* Based on SimBAG, a benchmark scheme has been implemented for performance comparison. In the benchmark scheme, no collaborative beamforming is considered, and the UAVs access the channel in a frequency division multiple access (FDMA)



(a) (b)

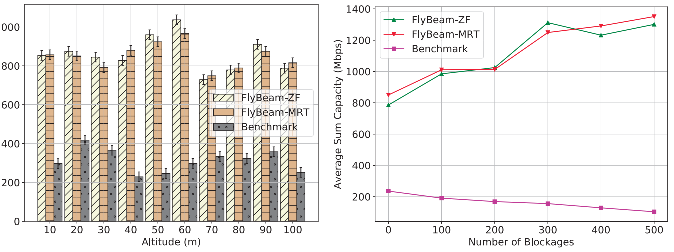
Fig. 1: (a) Average sum capacity with different number of UAVs; (b) Capacity gain with different number of users.

manner and further serve their users based on time division multiple access (TDMA). Each UAV maximizes the aggregate capacity of the users it serves based on Sequential Least Squares Programming (SLSQP) [27], to dynamically allocate their transmission power among different sub-channels to avoid mutual interference.

Performance Analysis. In the first experiment, we evaluate the capacity performance of *FlyBeam*. The number of UAVs is varied from 2 to 30 in step of 2 and the number of ground users is set to 10. The results are reported in Fig. 1. It can be seen from Fig. 1(a) that significant capacity gain can be achieved by *FlyBeam*. For example, average capacities of $915.20\,\mathrm{Mbps}$ and $960.12\,\mathrm{Mbps}$ can be achieved with *FlyBeam-ZF* and *FlyBeam-MRT*, respectively, which are 3.56 and 3.73 times higher than that achievable by the benchmark scheme ($257\,\mathrm{Mbps}$). The effectiveness of distributed beamforming in swarm UAV networks is further verified in Fig. 1(b) in terms of average capacity gain with 10 UAVs and the number of users varying from 2 to 22 in step of 4. We can see that both *FlyBeam-ZF* and *FlyBeam-MRT* are able to achieve a capacity gain between 312% and 457% with an average of 380%.

Next, we investigate the effects of the UAVs' flight altitude on the beamforming gain, considering 10 ground users and 10 UAVs. The flight altitude of the UAVs is varied from $10\,\mathrm{m}$ to $100\,\mathrm{m}$ in steps of $10\ \mathrm{m}$. The results are plotted in Fig. 2(a). Similar to that in Fig. 1, significant capacity gains can be achieved by *FlyBeam* in all the tested cases. It can also be found that the sum capacity achievable by *FlyBeam* is only slightly affected by the UAV flight altitude. For example, an average sum capacity of $854\ \mathrm{Mbps}$ can be achieved by *FlyBeam-ZF* with a flight altitude of $10\ \mathrm{m}$, which is $816\ \mathrm{Mbps}$ for an altitude of $100\ \mathrm{m}$. Similar results can also be achieved by the *FlyBeam-MRT*. This is not without reasons. As discussed in Section I, while flying lower can reduce the propagation distance to the ground users, the transmissions may experience higher attenuation because of a higher probability of being blocked.

The effects of blockages on the beamforming gain is further studied in Fig. 2(b). The number of blockages is varied from 0 to 500 in steps of 100. The minimum and maximum dimensions of the blockages are set to $2\ \mathrm{m}$ and $8\ \mathrm{m}$ for length and width, respectively. The height of the blockages is fixed to $15\ \mathrm{m}$, and the UAV flight altitude is set to $30\ \mathrm{m}$.



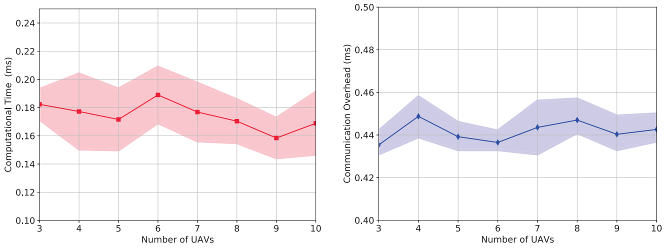
(a) (b)

Fig. 2: Average sum capacity with (a) varying flight altitude (b) varying number of blockages.

It can be found that *FlyBeam-ZF* and *FlyBeam-MRT* can yield an average capacity of $1106.9\ \mathrm{Mbps}$ and $1127.5\ \mathrm{Mbps}$, respectively, which are 6.74 and 6.87 times larger than the benchmark scheme. The sum capacity achievable with the benchmark scheme decreases monotonically with the number of blockages because of larger signal attenuation with more blockages. A surprising observation is that a *higher rather than lower* sum capacity can be achieved by *FlyBeam* with more blockages in the network. For example, an average sum capacity of $786\ \mathrm{Mbps}$ can be achieved by the *FlyBeam-ZF* with no blockages, which are $1025\ \mathrm{Mbps}$ and $1301\ \mathrm{Mbps}$ with 200 and 500 blockages, respectively. This is primarily because denser blockages introduce more diversity in the wireless channels, and hence higher spatial diversity gain can be achieved through beamforming.

Complexity and Convergence. We study the computational complexity and communication overhead of *FlyBeam* control in Fig. 3. The experiments are conducted on a workstation with Intel(R) Core(TM) $i7-10510$U CPU @ 1.80 GHz 2.30 GHz, memory of 16.0 GB, and 64-bit Windows Operating System. We fix the number of users as 10 and vary the number of UAVs from 3 to 10 at a step of 1. In Fig. 3(a) we report the average computational time taken by the UAVs' *FlyBeam* ESN module, with the confidence interval indicated by the shaded area. It can be seen that the computational time ranges from $0.14\ \mathrm{ms}$ to $0.21\ \mathrm{ms}$ and is barely affected by the network scale. Figure 3(b) plots the corresponding communication overhead, i.e., the time taken for CSI and beamforming weights sharing. Results indicate that the communication overhead of individual UAVs varies only slightly with the number of UAVs, from $0.43\ \mathrm{ms}$ to $0.46\ \mathrm{ms}$ with an average of $0.443\ \mathrm{ms}$. This verifies the good scalability of *FlyBeam*.

We further study the effects of periodic updating of the ESN training on the network's sum capacity, taking *FlyBeam-*



(a) (b)

Fig. 3: (a) Computational complexity and (b) Communication overhead of *FlyBeam*.

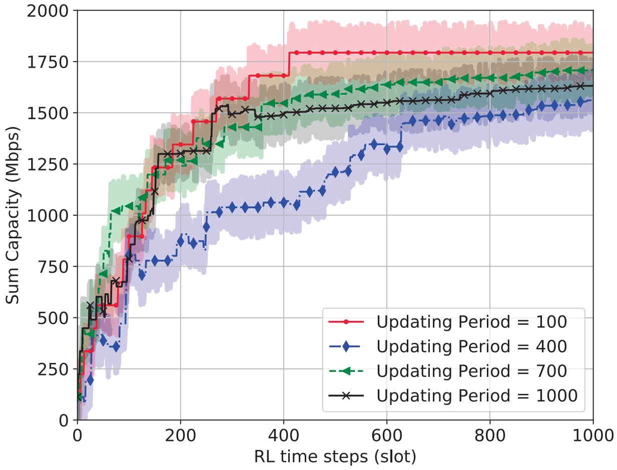


Fig. 4: Convergence of *FlyBeam* with different updating periods.

MRT as an example. The results are plotted in Fig. 4, where each curve is obtained by averaging over 20 simulations. The updating period is varied from 100 to 1000 time slots at steps of 300 time slots. It can be seen that the convergence speed is significantly affected by the updating period. In most of the test cases, *FlyBeam* converges faster with shorter updating periods. For example, *FlyBeam* converges in around 400 time slots with an updating period of 100 time slots, which is twice as fast as with updating periods of 700 and 1000 time slots. However, interestingly, we found that there is no monotonic mapping between the convergence speed and the updating period. This can be seen when the updating period is set to 400 time slots. In future work, we will investigate the optimal updating frequency for online reinforcement learning by considering the involved interaction between ESN-based utility function approximation and RL-based state exploration.

VI. Conclusions

In this paper we designed a high-data-rate swarm UAV with distributed capabilities by taking into account different factors that affect the beamforming gain. We proposed *FlyBeam* by jointly controlling the flight and beamforming in swarm UAV networks, based on a combination of ESN learning and reinforcement learning. Simulation results indicate that i) up to 450% capacity gain can be achieved by enabling distributed beamforming in swarm UAV networks, and ii) with distributed beamforming, *higher (rather than lower)* network capacity can be achieved with denser blockages. In future work we will further verify these findings based on testbed experiments over the aerial network experimentation facilities that have been developed at University at Buffalo [28].

References

[1] V. Sharma and R. Kumar, “A Cooperative Network Framework for Multi-UAV Guided Ground Ad hoc Networks,” *Journal of Intelligent Robotic Systems*, vol. 77, pp. 629–652, Jan. 2015.

[2] S. K. Moorthy and Z. Guan, “FlyTera: Echo State Learning for Joint Access and Flight Control in THz-enabled Drone Networks,” in *Proc. of IEEE SECON*, Como, Italy, June 2020.

[3] K. G. Panda, S. Das, D. Sen, and W. Arif, “Design and Deployment of UAV-Aided Post-Disaster Emergency Network,” *IEEE Access*, vol. 7, pp. 102 985–102 999, July 2019.

[4] M. Mandi Azari et al., “Cellular UAV-to-UAV Communications,” in *Proc. of PIMRC*, Istanbul, Turkey, Sept. 2019.

[5] Q. Wu and R. Zhang, “Common Throughput Maximization in UAV-Enabled OFDMA Systems With Delay Consideration,” *IEEE Transactions on Commun.*, vol. 66, no. 12, pp. 6614–6627, Nov. 2018.

[6] Q. Wu and Y. Zeng, “Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks,” *IEEE Transactions on Wireless Commun.*, vol. 17, no. 3, pp. 3417–3442, Nov. 2018.

[7] L. Bertizzolo et al., “SwarmControl: An Automated Distributed Control Framework for Self-Optimizing Drone Networks,” in *Proc. of IEEE INFOCOM (virtual)*, July 2020.

[8] M. Mozaffari et al., “A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems,” *IEEE Commun. Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, Nov. 2019.

[9] A. Fotouhi et al., “Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges,” *IEEE Commun. Surveys Tutorials*, vol. 21, no. 4, pp. 2109–2121, Nov. 2019.

[10] S. Mohanti et al., “AirBeam: Experimental Demonstration of Distributed Beamforming by a Swarm of UAVs,” in *Proc. of IEEE MASS*, Monterey, CA, USA, Nov. 2019.

[11] Y. Lu, J. Fang, Z. Guo, and J. A. Zhang, “Distributed Transmit Beamforming for UAV to Base Communications,” *China Commun.*, vol. 16, no. 1, pp. 15–25, 2019.

[12] S. Yeh, J. Chamberland, and G. H. Huff, “An Investigation of Geolocation-Aware Beamforming Algorithms for Swarming UAVs,” in *Proc. of IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, 2017, pp. 641–642.

[13] P. Dinh, T. M. Nguyen, C. Assi, and W. Ajib, “ Joint Beamforming and Location Optimization for Cooperative Content-Aware UAVs,” in *Proc. of IEEE WCNC*, Marrakesh, Morocco, Apr. 2019.

[14] W. Yuan, C. Liu, F. Liu, S. Li, and D. W. K. Ng, “Learning-based Predictive Beamforming for UAV Communications with Jittering,” *IEEE Wireless Commun. Letters*, 2020.

[15] W. Miao et al., “Position-Based Beamforming Design for UAV Communications in LTE Networks,” in *Proc. of IEEE ICC*, Shanghai, China, May 2019.

[16] I. Kocaman, “Distributed beamforming in a swarm UAV network,” *Calhoun: The NPS Institutional Archive DSpace Repository*, 2008. [Online]. Available: https://calhoun.nps.edu/handle/10945/4215

[17] Z. Guan, N. Cen, T. Melodia, and S. Pudlewski, “Joint Power, Association and Flight Control for Massive-MIMO Self-Organizing Flying Drones,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1491–1505, August 2020.

[18] D. C. Araújo et al., “Massive MIMO: Survey and Future Research Topics,” *IET Commun.*, vol. 10, no. 15, pp. 1938–1946, Oct. 2016.

[19] P. Zhang and J. Chen and X. Yang and N. Ma and Z. Zhang, “Recent Research on Massive MIMO Propagation Channels: A Survey,” *IEEE Commun. Magazine*, vol. 56, no. 12, pp. 22–29, Dec. 2018.

[20] K. Ntontin and C. Verikoukis, “Toward the Performance Enhancement of Microwave Cellular Networks Through THz Links,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5635–5646, July 2017.

[21] T. Bai, R. Vaze, and R. W. Heath, “Analysis of Blockage Effects on Urban Cellular Networks,” *IEEE Transactions on Wireless Commun.*, vol. 13, no. 9, pp. 5070–5083, Sept. 2014.

[22] “Spatial channel model for multiple input multiple output (MIMO) simulations (3GPP TR 25.996 version 11.0.0 release 11).” [Online]. Available: https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html

[23] J. Dang, C. Li, and Y. Meng, “Research on Least Square Channel Estimation Algorithm Based on Phase Compensation,” in *Proc. of IEEE ICCC*, Chengdu, China, Oct. 2016.

[24] T. Parfait, Y. Kuang, and K. Jerry, “Performance analysis and comparison of ZF and MRT based downlink massive MIMO systems,” in *Proc. of ICUFN*, Shanghai, China, July 2014.

[25] M. Lukoševičius, “A Practical Guide to Applying Echo State Networks,” *Lecture Notes*, Jan. 2012.

[26] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction.” Cambridge, MA, USA: MIT Press, 1998.

[27] “Sequential Least Squares Programming (SLSQP).” [Online]. Available: http://www.pyopt.org/reference/optimizers.slsqp.html

[28] A. Anand et al., “UBSpot: A Universal Broadband Flying Hotspot Experimental Testbed Toward Programmable Aerial-Ground Wireless Networks,” in *Proc. of IEEE Radio Wireless Symposium (RWS): Internet of Things (IoT) and the mmWave Frontier*, San Antonio, Texas, USA, January 2020.