Resource Efficient Actuation in UAV-aided Sensor-Actuator Networks

Resource-Efficient Actuation in UAV-Aided Sensor-Actuator Networks

Amit Goel[1], *Student Member, IEEE*, Swades De[1], *Senior Member, IEEE*,
Carla-Fabiana Chiasserini[2], *Fellow, IEEE*, Claudio E. Casetti[3], *Senior Member, IEEE*

***Abstract*—Integrating uncrewed aerial vehicles (UAVs) into wireless sensor-actuator networks (WSANs) offers flexibility and improved system performance. However, imprecise localization and limited battery capacity constrain the UAV operation. This paper explores the use of battery swap station (BSS)-assisted UAV to facilitate timely actuation in WSANs while addressing the above limitations. The UAV collects data via backscatter communication from the sensor nodes and delivers to the energy-constrained actuator nodes along with the required energy. Incorporating UAV location uncertainty and Nakagami-m wireless channel fading, closed-form expressions are derived for the ergodic capacity in backscatter communication and the expected energy harvesting rate. To minimize the maximum delay in actuation, an optimization problem is formulated. To reduce complexity, the problem is transformed into an equivalent node visit sequence optimization and solved using sequential deep reinforcement learning (SDRL). We verify the accuracy of our analysis through Monte Carlo simulations. Our results show that the proposed SDRL-based strategy consistently offers reduced actuation delay with a significantly small computation overhead.**

***Index Terms*—UAV, wireless sensor-actuator network, backscatter communication, wireless energy transfer, DRL**

I. Introduction

Recent advances in wireless communication have enabled distributed sensing and actuation through wireless sensor and actuator networks (WSANs) [1]. However, these networks, which consist of wirelessly connected sensor and actuator nodes, are constrained by communication range, energy, and infrastructure costs. Conventional nodes with limited battery life need frequent replacements, which is infeasible in hazardous deployments. To overcome these issues, emerging technologies such as backscatter communication (BSC) and wireless energy transfer (WET) have gained attention [2], [3].

In WSANs, a key objective is to deliver sensor data to actuators for timely action while ensuring sufficient energy availability. The Age of Information (AoI) metric quantifies the freshness of sensor updates [4]. However, minimizing AoI alone does not ensure timely data-driven actuation. While [5] adds data uncertainty to AoI, it does not fully address end-to-end responsiveness. In [6], actuation delay is modeled assuming a direct sensor-actuator communication link, which limits its utility in remote deployments. To address this, [7] proposes using a static controller to relay information. However, a static controller may not be sufficient for communication and actuation operations, as these power-intensive tasks require the controller to be close to the sensor and actuator nodes. To this end, UAVs can be employed in BSC- and WET-based WSANs. While UAV-assisted communication and WET have been individually studied, their joint application with monostatic backscatter in sensor data relaying under practical constraints such as limited UAV energy, imprecise localization, and low WET efficiency has received limited attention. Although approaches like establishing BSSs [8], [9] can extend UAV service time, optimizing UAV-aided WSAN operation under such constraints requires in-depth analysis.

To address the above challenges, this work contributes as follows: *(i)* Using stochastic geometry tools, we derive closed-form expressions for BSC ergodic capacity $\bar{r}_{\text{BSC}}$ and the expected energy harvesting rate $\bar{P}_{\text{EH}}$, incorporating Nakagami-m channel fading and UAV location uncertainty. *(ii)* Based on $\bar{r}_{\text{BSC}}$ and $\bar{P}_{\text{EH}}$ and WSAN requirements, we propose a novel UAV-assisted framework to minimize the actuation delay. *(iii)* To achieve the objective, we reformulate the problem into an equivalent form that optimizes the node visiting sequence. *(iv)* Given the high complexity due to the constrained combinatorial nature of the problem, a sequential deep reinforcement learning (SDRL) approach is employed for solving it. *(v)* Our numerical results demonstrate significant performance benefits with the proposed method compared to the benchmark schemes. While this study focuses on a single UAV, future multi-UAV extensions [10] offer promising scalability. We remark that the proposed system is practical for scenarios like avalanche-prone areas, enabling timely, sensor-driven warning actuation for road safety.

II. System Model

Consider a UAV-aided WSAN, having N sensor-actuator pairs, a UAV, and a BSS, deployed in a circular region of radius R_{max} (Fig. 1). The nodes are assumed to be located far apart; thus, the UAV can serve only one node at a time. The UAV visits each sensor before its corresponding actuator once per round while accessing the BSS as required. It follows predefined inter-node paths at a speed V_{UAV} and altitude h_{UAV}, enabling low control complexity and energy-aware mobility. UAV power consumptions for movement P_{mov} and hovering P_{hov} are modeled as in [11]. The sensor and actuator nodes

[1]Dept. of Electrical Engineering and Bharti School of Telecommunication Technology and Management, IIT Delhi, India (bsz208009@iitd.ac.in; swadesd@ee.iitd.ac.in). [2]Dept. of Electronics and Telecommunications, Politecnico di Torino, Italy (chiasserini@polito.it). [3]Dept. of Control and Computer Engineering, Politecnico di Torino, Italy (claudio.casetti@polito.it).

This work was supported in part by the Dept. of Science and Technology India-Italy Research Mobility under Grant INT/Italy/P-34/2022; in part by the Tehri Hydroelectric Development Corporation under Grant THDC/RKSH/R&D/F-2076/1036; and in part by the Science and Engineering Board under Grant CRG/2023/005421.

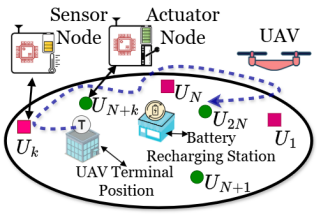

Figure 1 UAV and BSS aided wireless sensor actuator network.

have communication and energy harvesting modules; sensors include sensing units, whereas actuators contain mechanical actuation modules. The combined set of nodes is denoted by $\mathcal{U}=\{1,\ldots,2N\}$, where index $k\in\{1,\ldots,N\}$ refers to the k-th sensor and $k+N\in\{N+1,\ldots,2N\}$ to its corresponding actuator. Nodes are assumed static with known positions, reflecting real-world scenarios where coordinates are fixed during deployment or obtained via localization methods, without loss of generality. The coordinates of n-th node is denoted as $w_{\mathrm{n}}=\{x_{\mathrm{n}},y_{\mathrm{n}},z_{\mathrm{n}}\}$. For convenience, indices $'0'$ and $'2N+1'$ represent BSS and UAV terminal station with respective coordinates $\{x_0,y_0,z_0\}$ and, $\{x_{2\mathrm{N}+1},y_{2\mathrm{N}+1},z_{2\mathrm{N}+1}\}$. The 2D Euclidean distance between any two indices is denoted as $d_{\mathrm{i,j}}$.

The UAV collects the sensor data using monostatic BSC by hovering over it and transmitting an unmodulated signal u_1 toward the sensor node. The received signal at the sensor node is $y_{\mathrm{S}}=h_1\sqrt{P_{\mathrm{tx}}}u_1+n_1$. The sensor node reflects back the received signal after modulating it with the information signal u_2. The signal received by the UAV is [12]: $y_{\mathrm{U}}=h_2h_1\sqrt{P_{\mathrm{tx}}}u_1u_2+n_1+n_2$, where P_{tx} is the transmit power of the UAV, and n_1 and n_2 denote additive white Gaussian noise (AWGN). There can be some imperfection in hovering position due to UAV localization error, which are discussed in Section III. The LoS-dominant UAV channel is modeled by the Nakagami-m distribution, owing to its versatile fading representation and analytical tractability. The channel gain is $h_{\mathrm{i}}=\sqrt{G_1G_2(\lambda/4\pi)^2d^{-\alpha_P}}\tilde{h}_{\mathrm{i}}\ \forall i\in\{1,2\}$. Here, $\tilde{h}_{\mathrm{i}}$ denotes the small-scale fading coefficients, α_{P} is the path loss exponent, λ is the transmission frequency, and G_1,G_2, respectively, denote the UAV and node antenna gain. The communication data rates from the sensor node to UAV $r_{\mathrm{S-U}}$ and the UAV to actuator node $r_{\mathrm{U-A}}$ are expressed as $r_{\mathrm{S-U}}=\log_2\!\left(1+P_{\mathrm{tx}}|h_1|^2|h_2|^2/\sigma_1^2+\sigma_2^2\right)$, and $r_{\mathrm{U-A}}=\log_2\!\left(1+P_{\mathrm{tx}}|h_1|^2/\sigma_1^2\right)$ The harvested power P_{EH},with $P_{\mathrm{rx}}=P_{\mathrm{tx}}|h_1|^2$,is modeled as [13]

$$P_{\mathrm{EH}}(P_{\mathrm{rx}})=(aP_{\mathrm{rx}}+b)\cdot\mathbf{1}_{[\varrho_1,\varrho_2)}(P_{\mathrm{rx}})+P_{\mathrm{EH}}^{\max}\cdot\mathbf{1}_{[\varrho_2,\infty)}(P_{\mathrm{rx}}) \quad (1)$$

where $\mathbf{1}_{[m,n)}(x)=1$ if $m\le x<n$, else 0, $P_{\mathrm{EH}}^{\max}$ is maximum harvest power, ϱ_1 and ϱ_2 are RF energy harvesting sensitivity and saturation threshold, a and b are shaping parameters.

III. Statistical Analysis

In this section, $\bar{r}_{\mathrm{BSC}}$ and $\bar{P}_{\mathrm{EH}}$ are derived based on the above channel model and UAV location uncertainty.

Distance distribution: While serving the n-th node, the UAV is meant to hover above w_{n}. Due to location error, its actual position is considered uniformly distributed in a ball $B(w_{\mathrm{n}},\Delta_{\mathrm{R}})$. Thus, the UAV-to-node distance distribution is

$$f_{\mathrm{d}}(d)=2d/\sqrt{d^2+h_{\mathrm{UAV}}^2};\quad h_{\mathrm{UAV}}\le d\le\sqrt{h_{\mathrm{UAV}}^2+\Delta_{\mathrm{R}}^2}\,. \quad (2)$$

BSC ergodic capacity: It is defined as the achievable average data rate, considering Nakagami-m wireless fading channel and the spatial distribution of the UAV-to-node distance as modeled in (2) and is expressed as $E_d\left[E_{h_1h_2}\left[\log_2(1+\gamma)\right]\right]$.

Theorem 1. *The closed-form expression for BSC ergodic capacity is expressed in* (4)*, where* $C_3=(\Gamma m_1\Gamma m_2)^{-1}, C_4=c_{41}/c_{42}$, $c_{41}=\alpha_1\alpha_2P_{tx}(G_1G_2)^2(\lambda/4\pi)^4$, *and* $c_{42}=m_1m_2$.

Proof. See Appendix A. ∎

Expected energy harvesting rate: It is defined as the average amount of power that can be harvested by the node via WET process.

Theorem 2. *The closed-form expression for expected energy harvesting rate considering Gamma-distributed wireless fading link gain and spatial distribution as modeled in* (2) *is expressed in* (5)*, where* $K=P_{tx}E[d^{\alpha_p}](\lambda/4\pi)^2G_1$.

Proof. See Appendix B. ∎

IV. Optimization Framework and Constraint

We now formulate the actuation delay minimization based on the node requirements and statistical measures from (4) and (5). Then we present the associated feasibility constraints.

The actuation delay minimization is formulated as a sequence prediction problem. After the UAV leaves the terminal, it travels to each sensor node to gather data and proceeds to the associated actuator node to transfer information and energy. This process continues until data are collected from all sensor nodes and the respective actuators are served, and the UAV recharges itself as required. UAV visiting sequence is denoted as $S=\{s_k\}_{k=1}^{L}$ where $s_k\in\{0,1,\ldots,2N+1\}$ denotes the indices having one-to-one mapping with the nodes, BSS, and terminal station, L is the sequence length, which is at least $2(N+1)$, but could be longer since the UAV may need to visit the BSS multiple times. S needs to be optimized for reduced actuation delay. In finding the optimal S, the feasibility needs to be ensured, as the constraints can affect the viability of S.

1) Feasible sequence: We define a variable $\mathbf{I}(k)$ that indicates the index of the location visited by the UAV at k-th position in the sequence and is defined as $\mathbf{I}(k)=m$, s.t.$s_{\mathrm{m}}=k; m\in\{1,\ldots L\}, k\in\{1,\ldots 2N\}$. While serving, each node should be visited only once, with an actuator visited only after its corresponding sensor is served. The constraints are represented as

$$\text{if } s_{\mathrm{i}}=k\Rightarrow s_{\mathrm{j}}\ne k;\forall j\ne i,\forall i,j\in\{1,\cdots,L\},k\in\{1,\cdots,2N\} \quad (6)$$

$$\mathbf{I}(k)<\mathbf{I}(N+k)\quad\forall k\in\{1,\cdots,N\}\,. \quad (7)$$

Moreover, the total time the UAV spends on each node is required to be greater than or equal to the required service time for that node. The variable τ_j denotes the expected time required to serve the j-th node and is defined as

$$\bar{r}_{\mathrm{BSC}} = \frac{C_3 h_{\mathrm{UAV}}^{-2} \Delta_{\mathrm{R}}^3}{2} H_{1,0;4,3;1,2}^{0,1:1,4:1,1} \left(\begin{matrix} (0,2,1) \\ - \end{matrix} \middle| \begin{matrix} (1-m_1,1)\,(1-m_2,1)\,(1,1)\,(1,1) \\ (1,1)\,(1/2,2)\,(0,1) \end{matrix} \middle| \begin{matrix} (-1/2,1) \\ (-1,1)\,(-3/2,1) \end{matrix} \middle| C_4 h_{\mathrm{UAV}}^{-4}, \frac{\Delta_{\mathrm{R}}^2}{h_{\mathrm{UAV}}^2} \right) \tag{4}$$

$$\bar{P}_{\mathrm{EH}} = \frac{a\alpha_1 K}{m_1 \Gamma m_1} \left(\gamma\left(m_1{+}1, \tfrac{m_1 \varrho_2}{K\alpha_1}\right) - \gamma\left(m_1{+}1, \tfrac{m_1 \varrho_1}{K\alpha_1}\right) \right) + \frac{b}{\Gamma m_1} \left(\gamma\left(m_1, \tfrac{m_1 \varrho_2}{K\alpha_1}\right) - \gamma\left(m_1, \tfrac{m_1 \varrho_1}{K\alpha_1}\right) \right) + \frac{P_{\mathrm{EH,max}}}{\Gamma m_1} \left(1 - \frac{1}{\Gamma m_1} \gamma\left(m_1, \tfrac{m_1 \varrho_2}{K\alpha_1}\right) \right) \tag{5}$$

$$\tau_{\mathrm{j}} = \begin{cases} \frac{D_{\mathrm{j}}^{\mathrm{req}}}{\bar{r}_{\mathrm{S\text{-}U,j}}} (1 + \frac{P_{\mathrm{c}}}{\bar{P}_{\mathrm{EH}}}) & \forall j \in \{1, \cdots, N\} \\ \frac{E_{\mathrm{j}}^{\mathrm{req}}}{\bar{P}_{\mathrm{EH}}} + \frac{D_{\mathrm{j}}^{\mathrm{req}}}{\bar{r}_{\mathrm{U\text{-}A,j}}} \approx \frac{E_{\mathrm{j}}^{\mathrm{req}}}{\bar{P}_{\mathrm{EH}}} & \forall j \in \{N+1, \cdots, 2N\} \\ \tau_{\mathrm{BSS}} & j = 0 \end{cases} \tag{8}$$

where $D_{\mathrm{j}}^{\mathrm{req}}$ denotes the amount of data required to be transmitted, $E_{\mathrm{j}}^{\mathrm{req}}$ denotes the required energy for actuation, P_{c} denotes the sensor node circuit power consumption while performing BSC [3], and τ_{BSS} denotes the time required for battery swapping. Thus, the total energy required to be spent by the UAV on the j-th node is expressed as $\mathcal{E}_{\mathrm{j}}^{\mathrm{ser}} \approx (P_{\mathrm{Hov}} + P_{\mathrm{tx}})\tau_{\mathrm{j}} \quad \forall j \in \{1, \ldots, N\}$. The UAV transmits with $P_{\mathrm{tx}}^{\mathrm{a}}$ for actuator nodes and $P_{\mathrm{tx}}^{\mathrm{s}}$ for sensor nodes. Once a node has been served, $\mathcal{E}_{\mathrm{j}}^{\mathrm{ser}}$ is reset to null. Furthermore, throughout the process, it is necessary to monitor the on-board battery and recharge it when required. The variable $B_{\mathrm{rem,j}}$ denotes the remaining on-board battery capacity of UAV after it has completed traversing the sequence up to the j-th element, $\forall j{\in}\{0, 1, \ldots, 2N{+}1\}$. Furthermore, to represent the relationship between two nodes that are visited in sequence, a binary variable $\mathcal{X}_{\mathrm{ij}}(n)$ is defined as $\mathcal{X}_{\mathrm{ij}}(n){=}1$ if $s_{\mathrm{n}}{=}\mathrm{i}$ and $s_{n+1}{=}\mathrm{j}$, where, $i, j{\in}\{0, \ldots 2N{+}1\}$ and $n{\in}\{0, \ldots L{-}1\}$. Given that $\mathcal{B}_{\mathrm{rem,n}}$ is known and $\mathcal{X}_{\mathrm{ij}}(n){=}1$ where $n{\in}\{0, \ldots L{-}1\}$, then the battery status is updated as

$$\mathcal{B}_{\mathrm{rem,n+1}} = \begin{cases} \mathcal{B}_{\mathrm{rem,n}} - \mathcal{E}_{\mathrm{ij}}^{\mathrm{mov}} - \mathcal{E}_{\mathrm{j}}^{\mathrm{ser}} & j \in \{1, \cdots, 2N\} \\ \mathcal{B}_{\mathrm{full}} & j = 0 \\ \mathcal{B}_{\mathrm{rem,n}} - \mathcal{E}_{\mathrm{ij}}^{\mathrm{mov}} & j = 2N+1\,. \end{cases} \tag{9}$$

It is notable that the onboard battery status restricts the UAV movement. The constraints are represented as

$$\mathcal{X}_{\mathrm{ij}}(n) \geq 0 \text{ if } \begin{cases} \mathcal{B}_{\mathrm{rem,n}} \geq \mathcal{E}_{\mathrm{ij}}^{\mathrm{mov}} + \mathcal{E}_{\mathrm{j}}^{\mathrm{ser}} + \mathcal{E}_{\mathrm{j0}}^{\mathrm{mov}} & j \in \{1, \cdots, 2N\} \\ \mathcal{B}_{\mathrm{rem,n}} \geq \mathcal{E}_{\mathrm{ij}}^{\mathrm{mov}} & j \in \{0, 2N+1\} \end{cases} \tag{10}$$

$$\mathcal{X}_{\mathrm{i0}}(n) = 1 \text{ if } \mathcal{B}_{\mathrm{rem,n}} - \mathcal{E}_{\mathrm{i,0}} < \min\left\{\mathcal{E}_{\mathrm{ij}}^{mov}\right\} \forall n \in \{1, \cdots, L-1\}, \quad i \in \{1, \cdots, 2N\}, j \in \{1, \cdots, 2N+1\} \setminus \{i\} \tag{11}$$

where (10) indicates that after serving the i-th node, the UAV can visit the j-th node only if the remaining battery capacity is sufficient for both serving the j-th node and returning to the BSS from that node. Moreover, (11) states that if the remaining battery capacity is not enough to serve any unserved nodes, the UAV should proceed to the BSS. The subtraction in (11) ensures that the UAV always has sufficient battery to reach the BSS. Furthermore, (12) highlights that the UAV in any feasible sequence starts and ends at the UAV terminal position.

$$\textstyle\sum_{\mathrm{i}} \mathcal{X}_{\mathrm{2N+1,i}}(0) = 1, \sum_{\mathrm{i}} \mathcal{X}_{\mathrm{i,2N+1}}(L-1) = 1 \forall \mathrm{i} \in \{0, \cdots, 2\mathrm{N}\}\,. \tag{12}$$

2) Problem formulation: The UAV performance is measured by the maximum delay of actuation (MDA), defined as the maximum time difference between the start of operation and when the actuator takes action. The overall optimization problem considering this metric and the feasibility constraints is expressed as:

$$\begin{aligned} (\mathcal{P}1): \quad & \min_{\mathbf{S}} \quad \max_{\mathbf{k}} |t_{\mathrm{k+N}} - t_0| \\ & \text{s.t.}: \quad (6), (7), (10) - (12) \end{aligned} \tag{13}$$

where, t_0 is the time at which UAV starts the operations and t_{k+N} denotes the time at which actuator associated with k-th sensor takes action which is expressed as $t_{\mathrm{j}} = \sum_{i=1}^{\mathbf{I}(j)} (d_{\mathrm{s_i,s_{i+1}}}/V_{\mathrm{UAV}}) + \tau_{\mathrm{i}}, \forall j \in \{N{+}1, \ldots, 2N\}$. The problem $\mathcal{P}1$ is an NP-hard combinatorial optimization problem. Considering the UAV battery capacity sufficiently large and homogeneity among all the nodes, the transformed problem becomes an equivalent NP-hard problem as in [11]. Consequently, the problem $\mathcal{P}1$ is inherently NP-hard, rendering it difficult to address through conventional solution methods. Therefore, we propose a sequential deep reinforcement learning approach building on [14], [15] to effectively address this challenge. The details of the solution are outlined in the following section.

V. Proposed SDRL-based Solution

The proposed solution utilizes a Markov Decision Process framework, where the UAV (agent) interacts with the dynamic environment by selecting actions that generate rewards while transitioning among states. Here, the UAV is responsible for making decisions and executing actions. A state S_l includes information about the last visited node and relevant parameters such as $\{w_{\mathrm{n}}, \tau_{\mathrm{n}}, \mathcal{E}_{\mathrm{n}}^{\mathrm{ser}}\}_{n=0}^{n=2N+1}$, and $B_{\mathrm{rem,l}}$. The action set consists of the available choices at a given state S_l, which includes unvisited node indices, BSS, and the UAV terminal station, all subject to the constraints defined in $\mathcal{P}1$. The action a_l represents the index of the location the UAV visits. The reward is calculated at the end of the episode and is defined as the negative of the maximum actuation delay, as described in $\mathcal{P}1$.

1) Sequential neural network architecture: To map the current state to a probability distribution of possible actions, a sequential neural network-based policy is used at the UAV. Primarily, sequential neural network architecture is composed of an encoder and decoder module.

Encoder: The encoder input consists of static components $C_{\mathrm{s,n}} = \{w_{\mathrm{n}} \cup \tau_{\mathrm{n}}\}$ (location and expected serving time) and dynamic components $C_{\mathrm{d,n}} = \{B_{\mathrm{UAV}} \cup \mathcal{E}_{\mathrm{n}}^{\mathrm{ser}}\} \forall n = \{0 \cdots 2N+1\}$ (UAV battery and energy requirements). These are processed through an embedding layer comprising a convolutional encoder, which maps the low-dimensional data to a high-dimensional space and produces embedded outputs $C_{\mathrm{emb},s_{\mathrm{n}}}$ and $C_{\mathrm{emb},d_{\mathrm{n}}}$, with total inputs and outputs denoted as C_{in} and C_{emb}.

Decoder: For a policy ϕ, the probability that the UAV follows a sequence S conditioned on C_{emb} is defined as $P_\phi(S|C_{\mathrm{emb}}) = \prod_{l=1}^{L} P(s_{l+1}|S_l, C_{\mathrm{emb}})$, where S_l denotes the sequence up to l steps. At each step $l \in \{1, \cdots, L\}$, the decoder generates the conditional probability distribution

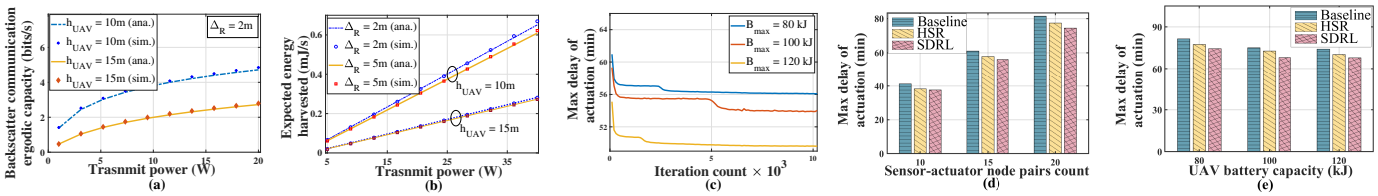

Figure 2 (a) Ergodic capacity vs. UAV transmit power. (b) Energy harvest vs. UAV transmit power for various location error. (c) Convergence of proposed SDRL algorithm; N=15. (d) Actuation delay, with battery size 80 kJ. (e) Actuation delay for 20 sensor-actuator node pairs.

Table I Compute overhead for different sensor-actuator pair counts

Method	Computation Time (ms)		
	N = 10	N = 15	N = 20
Baseline	0.1675	0.19	0.26
HSR	1504	2100	3080
SDRL	86.76	104.96	134.65

$P(s_{l+1}|S_l, C_{\text{emb}})$, that determines the agent action a_1. The optimal policy ϕ^* gives the optimal sequence S^* with probability 1. We aim to minimize the optimality gap between ϕ and ϕ^*.

In the l-th decoding step, the static information of the last visited index is encoded as $C_{\text{emb},s_{l-1}}$, and Gated Recurrent Unit (GRU) processes this encoded data along with the hidden state h_{l-1} maintained by GRU, yielding $h_l, Y_l = \text{GRU}(h_{l-1}, C_{\text{emb},s_{l-1}})$. The GRU output Y_l is then passed through the attention block which incorporates the importance of the locations at the current decoding step. This is achieved through the following operation: a_l=softmax($v_{a1}^T \tanh(W_{a1}\cdot C_{\text{emb}})+v_{a2}^T \tanh(W_{a2}\cdot Y_l)$). Using a_l, a context vector c_l, is generated as $c_l = \sum_{i=0}^{2N+1}(a_l^i C_{\text{emb},i})$. The encoded output C_{emb} is then combined with c_l to produce κ_l as $\kappa_l = v_{c1}^T \tanh(W_{c1}\cdot C_{\text{emb}})+v_{c2}^T \tanh(W_{c2}\cdot c_l)$. Before generating the action probability distribution, the masking vector $\mathcal{M}_l$ is applied to κ_l as $\kappa_l = \kappa_l + \mathcal{M}_l$, where $\mathcal{M}_{l,i} \in \{0, -\infty\} \forall i \in \{1,\ldots,2N+2\}$, enforces that the feasibility conditions in $\mathcal{P}1$, are satisfied. In addition, the actuator nodes are masked until all the sensor nodes are visited. Finally, the conditional probability distribution which determines the agent action a_l, is calculated as $P(s_{l+1}|S_l, C_{\text{emb}}) = \text{softmax}(\kappa_l)$. The process repeats until the termination condition is satisfied.

2) Training: The UAV policy network is trained using the REINFORCE algorithm to maximize expected cumulative reward, as outlined in Algorithm 1. The algorithm utilizes both an actor and a critic neural network to optimize the policy. The actor adjusts the policy parameters ϱ_a, increasing the likelihood of selecting actions that lead to positive outcomes. During each episode, the agent collects experience from the environment and computes cumulative reward R^b at the end, which serves as a performance estimate. Following the policy gradient approach, the policy parameters are then updated in the direction that increases the likelihood of actions that lead to higher rewards. The critic estimates the value function V^b and associated parameters ϱ_c are trained using stochastic gradient descent on the mean squared error (MSE) between predicted value and actual reward, helping to reduce variance in policy updates. To enhance learning efficiency, parallel learning and batch processing are used, addressing correlation between successive states and improving training stability.

Algorithm 1 Pseudo-code for SDRL training stage

Input: batch size $\mathcal{B}$, training dataset $\mathcal{D}$.
Output: Policy parameters ϱ_a , ϱ_c
for itr = $1, 2, \cdots,$ i_{tot} **do**
  reset gradient $d\varrho_a \leftarrow 0, d\varrho_c \leftarrow 0$
  Obtain training batch $\{\mathcal{D}_{(\text{itr}\times\mathcal{B})+1}, \cdots, \mathcal{D}_{(\text{itr}+1)\times\mathcal{B}}\}$
  for $b = 1, 2, \ldots, \mathcal{B}$ **do**
    $l \leftarrow 0$
    do
      compute $P(s_{l+1}^b | S_l^b, C_{emb}^b)$
      choose action accordingly
      $l \leftarrow l + 1$
    while terminal state is reached
    compute reward R^b
  end for
  calculate gradient
  $d\varrho_a \leftarrow \frac{1}{\mathcal{B}} \sum_{b=1}^{B} \left(R^b - V^b\right) \nabla_{\varrho_a} \log\left(P^b(S|C_{emb})\right)$
  $d\varrho_c \leftarrow \frac{1}{\mathcal{B}} \sum_{b=1}^{B} \nabla_{\varrho_c} \left(R^b - V^b\right)^2$
  Update ϱ_a and ϱ_c
end for

VI. Numerical Results

The SDRL simulations are performed in Python with the PyTorch library on a system having an Intel Xeon W-2145 3.70 GHz processor, 64 GB RAM, and Nvidia Quadro P5000 GPU. Simulation parameters include: $m_1, m_2 = 2$; $\alpha_1, \alpha_2 = 1$; $D_{\text{req}} = 24$–40 Bytes; $E_{\text{req}} = 31$–47 mJ; $G_1 = 10$ dBi; $G_2 = 0$ dBi; $\tau_{\text{BSS}} = 180$ s; $h_{\text{UAV}} = 10$ m; $\lambda = 915$ MHz; $R_{\max} = 150$ m; $\Delta R = 2$ m; $\alpha_P = 2$; $\varrho_1 = 0.000064$; $\varrho_2 = 0.02$; $P_c = 10.6$ μW; $P_{\text{EH,max}} = 0.00492$ μW; $\sigma_1^2, \sigma_2^2 = 10^{-9}$; constants $a = 0.24714$; $b = -1.5817 \times 10^{-5}$; $P_{\text{tx}}^{(a)} = 40$ W; $P_{\text{tx}}^{(s)} = 10$ W; $V_{\text{UAV}} = 9.8$ m/s; training size = 1.28×10^6; validation size = 10^3; convolution layers=1; hidden size $h_s = 128$; learning rate=5×10^{-4}; dropout = 0.1; batch size = 256. Fig. 2(a) highlights the BSC ergodic capacity. The plot illustrates a positive correlation between ergodic capacity and transmit power, while elevated hovering altitudes result in a noticeable reduction in capacity due to increased path loss. The low data rate is primarily due to the multiplicative fading and dual pathloss in BSC link and compounded by UAV position imprecision. Fig. 2(b) illustrates the performance of the UAV energy transfer capability in terms of expected energy harvesting rate. The plot captures the effect of UAV altitude variation and localization inaccuracy. At lower elevations, the impact of localization uncertainty on performance is more pronounced. The results reflect the characteristics of monostatic BSC systems, where ultra-low-power operation is prioritized over high data rates. The observed performance aligns with low-rate, energy-constrained application scenarios, where small payloads and efficient energy usage

are key. The analytical expressions align well with Monte Carlo simulations, confirming both their accuracy and practical relevance within the operational context considered.

Fig. 2(c) shows the learning curves of the proposed algorithm for different battery capacities. The curves demonstrate eventual convergence to the long-term return. The maximum delay is observed with the lowest battery capacity. Figures 2(d) and 2(e) compare the proposed algorithm, the baseline solution, and the hybrid swap-and-reverse (HSR) algorithm. The baseline solution satisfies the feasibility constraints but does not necessarily yield an optimal result. In contrast, the HSR algorithm employs a 2-Opt-based approach [16], starting from a feasible solution and iteratively refining it by randomly selecting two indices to either swap nodes or reverse the subsequence between them. The update is retained if it improves the reward and maintains feasibility, progressively refining the solution. Fig. 2(d) demonstrates that the maximum actuation delay in the proposed approach is consistently lower than the benchmark schemes. Also, Fig. 2(e) shows a clear reduction in maximum actuation delay. For example, in a network with 20 sensor-actuator node pairs and 80 kJ UAV battery capacity, the proposed strategy reduces actuation delay by up to 182 s compared to benchmark methods across various scenarios.

The computational complexity of the proposed SDRL is $O(LNh_s^2C_1)$, while that of HSR is $O(I_{max}N^2C_2)$, where C_1,C_2 are constants, h_s is the hidden size, and I_{max} is the number of iterations. The computation times are compared in Table I, showing up to 22 times reduced overhead in the proposed approach compared to HSR, highlighting its efficiency. The proposed algorithm is robust to changes in node count or configuration, requiring no retraining and thereby ensuring effective adaptation to network variations.

VII. Conclusion

This paper introduced UAV and BSS-assisted WSAN for minimizing actuation delay with imperfect UAV localization and wireless fading channel constraints. We derived closed-form expressions of BSC ergodic capacity for data collection and expected energy harvesting rate. Based on node requirements and the statistical properties of communication and energy harvesting, the node visit sequence of UAV was optimized using sequential DRL. Our results demonstrate that the proposed DRL-aided strategy offers consistently reduced actuation delay with multifold reduction in computation overhead compared to the benchmark approaches.

Appendix

A. *Proof of* (4): The pdf of the product of two independent not identical (i.n.i.d.) gamma random variables h_1 and h_2 is

$$f_{h_1h_2}(z) = \frac{2}{\Gamma m_1 \Gamma m_2}\left(\frac{m_1m_2}{\alpha_1\alpha_2}\right)^{\frac{m_1+m_2}{2}} z^{\frac{m_1+m_2-2}{2}} K_{m_1-m_2}\left(2\sqrt{\frac{m_1m_2z}{\alpha_1\alpha_2}}\right). \quad (.1)$$

The ergodic capacity is defined as

$$\frac{1}{\log 2}\int_{d_{\min}}^{d_{\max}}\int_0^{\infty} \log(1+\gamma)\, f_{h_1h_2}(x) f_{\mathrm{d}}(d)\,\mathrm{d}x\,\mathrm{d}d \quad (.2)$$

where $\gamma{=}\eta d^{-2\alpha}x$, $d_{\min}{=}h_{\mathrm{UAV}}$, $d_{\max}{=}\sqrt{\Delta_{\mathrm{R}}^2+h_{\mathrm{UAV}}^2}$, using (.1) and [17, 07.34.03.0456.01]. Consider the following integral:

$$I_1 = C_1\int_0^{\infty} G_{2,2}^{1,2}\left(\begin{matrix}11\\10\end{matrix}\middle| C_2x\right) x^{\frac{m_1+m_2-2}{2}} K_{m_1-m_2}\left(2\sqrt{x}\right)\mathrm{d}x. \quad (.3)$$

Using [18, Eq. (7.821.3)], and substituting in (.2), we have

$$\int_{d_{\min}}^{d_{\max}} C_3 G_{4,2}^{1,4}\left(\begin{matrix}1-m_1, 1-m_2, 1, 1\\ 1, 0\end{matrix}\middle| C_4d^{-2\alpha}\right)\frac{d}{\sqrt{d^2-h_{\mathrm{UAV}}^2}}\mathrm{d}d\,. \quad (.4)$$

Thereafter, using [18, Eq. (9.34.7)], and representing the output in terms of bivariate Fox-H function, we get (4).

B. *Proof of* (5) : The expected energy harvesting rate is

$$\bar{P}_{\mathrm{EH}} = \int_{\varrho_1}^{\varrho_2}(aP+b)\, f_{\bar{P}_{\mathrm{rx}}}(P)dP + P_{\mathrm{EH,\,max}}\int_{\varrho_2}^{\infty} f_{\bar{P}_{\mathrm{rx}}}(P)dP \quad (.1)$$

where $\bar{P}_{\mathrm{rx}} = E_{\mathrm{d}}[P_{\mathrm{rx}}] = G_1\,(\lambda/4\pi)^2\,E[d^{-\alpha_P}]h_1P_{tx}$. For $\alpha_P = 2$, after simple mathematical calculations, we get

$$E[d^{-\alpha_{\mathrm{P}}}] = \int_{d_{\min}}^{d_{\max}} d^{-\alpha_{\mathrm{P}}} f_{\mathrm{d}}(d)\mathrm{d}d = \frac{2}{h_{\mathrm{UAV}}\Delta_{\mathrm{R}}}\tan^{-1}\left(\frac{\Delta_{\mathrm{R}}}{h_{\mathrm{UAV}}}\right). \quad (.2)$$

Finally, substituting (.2) into (.1) and using [18, Eq. (8.350.1)], the closed-form expression (5) is obtained.

References

[1] N. Primeau *et al.*, "A review of computational intelligence techniques in wireless sensor and actuator networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2822–2854, 2018.

[2] D. Mishra *et al.*, "Smart RF energy harvesting communications: challenges and opportunities," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 70–78, 2015.

[3] T. Jiang *et al.*, "Backscatter communication meets practical battery-free internet of things: A survey and outlook," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 2021–2051, 2023.

[4] H. Fu *et al.*, "Analysis and optimization of age of information for area sensing," *IEEE Commun. Lett.*, vol. 28, no. 1, pp. 103–107, 2024.

[5] K. Fizza *et al.*, "Age of data aware internet of things applications," in *IEEE CCNC*, 2022, pp. 399–404.

[6] A. Nikkhah *et al.*, "Age of actuation in a wireless power transfer system," in *IEEE INFOCOM Wksp.*, 2023, pp. 1–6.

[7] B. Chang *et al.*, "Age of information for actuation update in real-time wireless control systems," in *IEEE INFOCOM Wksp.*, 2020, pp. 26–30.

[8] M. Hoang *et al.*, "Design of autonomous battery swapping for UAVs," in *IEEE/ASME Int. Conf. AIM*, 2024, pp. 353–358.

[9] T. Cokyasar *et al.*, "Designing a drone delivery network with automated battery swapping machines," *Comput. Oper. Res.*, vol. 129, p. 105177, 2021.

[10] Y. Li *et al.*, "Energy-efficient UAV-driven multi-access edge computing: A distributed many-agent perspective," *IEEE Trans. Commun.*, pp. 1–1, 2025.

[11] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[12] Y. H. Al-Badarneh *et al.*, "Performance analysis of monostatic multi-tag backscatter systems with general order tag selection," *IEEE Wireless Commun. Lett.*, vol. 9, no. 8, pp. 1201–1205, 2020.

[13] D. Mishra, S. De, and D. Krishnaswamy, "Dilemma at RF energy harvesting relay: Downlink energy relaying or uplink information transfer?" *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4939–4955, 2017.

[14] N. Mazyavkina *et al.*, "Reinforcement learning for combinatorial optimization: A survey," *Comput. Oper. Res.*, vol. 134, p. 105400, 2021.

[15] M. Nazari *et al.*, "Reinforcement learning for solving the vehicle routing problem," in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[16] Y. Ren and V. Friderikos, "Path planning optimization based interference awareness for mobile robots in mmwave multi cell networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 13 639–13 650, 2024.

[17] I. Wolfram, "Wolfram, research, mathematica edition: Version 10.0. champaign," *Wolfram Research, Inc.*, 2010.
[18] I. S. Gradshteyn and I. M. Ryzhik, "Table of Integrals, Series and Products," *New York, NY, USA: Academic*, 2000.