Autonomous UAV Trajectory for Localizing Ground Objects: A Reinforcement Learning Approach

Dariush Ebrahimi, Sanaa Sharafeddine, Pin-Han Ho, Chadi Assi

Abstract—Disaster management, search and rescue missions, and health monitoring are examples of critical applications that require object localization with high precision and sometimes in a timely manner. In the absence of the global positioning system (GPS), the radio received signal strength index (RSSI) can be used for localization purposes due to its simplicity and cost-effectiveness. However, due to the low accuracy of RSSI, unmanned aerial vehicles (UAVs) or drones may be used as an efficient solution for improved localization accuracy due to their agility and higher probability of line-of-sight (LoS). Hence, in this context, we propose a novel framework based on reinforcement learning (RL) to enable a UAV (agent) to autonomously find its trajectory that results in improving the localization accuracy of multiple objects in shortest time and path length, fewer signal-strength measurements (waypoints), and/or lower UAV energy consumption. In particular, we first control the agent through initial scan trajectory on the whole region to 1) know the number of nodes and estimate their initial locations, and 2) train the agent online during operation. Then, the agent forms its trajectory by using RL to choose the next waypoints in order to minimize the average location errors of all objects. Our framework includes detailed UAV to ground channel characteristics with an empirical path loss and log-normal shadowing model, and also with an elaborate energy consumption model. We investigate and compare the localization precision of our approach with existing methods from the literature by varying the UAV's trajectory length, energy, number of waypoints, and time. Furthermore, we study the impact of the UAV's velocity, altitude, hovering time, communication range, number of maximum RSSI measurements, and number of objects. The results show the superiority of our method over the state-of-art and demonstrates its fast reduction of the localization error.

Index Terms—Localization, Reinforcement Learning, Q-Learning, Unmanned Aerial Vehicles (UAVs), Drones, Trajectory Planning, Received Signal Strength (RSS)

1 Introduction

Most of the distributed wireless Internet-of-Thing (IoT) systems, in order to make their collected data insightful and meaningful, require to know the location of their component devices. **Since not all communicating devices are equipped with a global positioning system (GPS) due to its expensive cost and vulnerability to jamming, in addition to its bad performance in poor weather conditions [1], and also due to unavailability of a base station in a disaster situation to collect objects' locations using GPS, hence, alternative localization techniques have been extensively studied in the literature [2]. Among those, the radio received signal strength (RSS) is more attractive due to its simplicity and cheap functionality (does not require extra antennas or time synchronization) [3].** However, its localization accuracy is significantly affected by the randomness of the received signal and shadowing, particularly in urban areas. As an enhancement, an unmanned aerial vehicle (UAV) or drone may be used to localize ground objects [4]. The UAV has the ability to measure the RSS of multiple objects from different angles (or waypoints) with higher probability of line-of-sight (LoS), and thus better localization accuracy [5]. Examples of such application can vary from delivering packages to different addresses to finding expensive devices in an area.

In addition to the accurate positioning, timely localization is also indispensable for many operations such as in search and rescue missions. For example, finding locations of trapped people after a disaster or a patient who needs rescue in a serious life threat [6]. Therefore, finding the right flight path (trajectory) and aerial anchors (waypoints) is crucial for both timely and accuracy of the objects' localization. On the other hand, a UAV has limited energy which restricts its operational lifetime. Therefore, different criteria such as UAV's velocity, hovering time, and path length impact the energy consumption of the UAV, and hence affect the localization accuracy due to fewer collected RSSI measurements. Another challenge is that the UAV, before its mission, does not know the number and locations of the objects, therefore, none of the existing pre-path planing algorithms from the literature are efficient for the fast localization operation. To this end, the necessity in creating an autonomous UAV so as to observe the environment while localizing becomes crucial [7].

In this paper, a framework using reinforcement learning (RL) is proposed to optimize the operation of the UAV in urban areas. Based on the capacity factors, whether it is the UAV energy, operational time, number of waypoints, or allowed path length, a Markov decision process (MDP) model

D. Ebrahimi and P. Ho are with the department of Electrical and Computer Engineering at University of Waterloo, Ontario, Canada, e-mail: darebra@yahoo.com, p4ho@uwaterloo.ca. S. Sharafeddine is with the Department of Computer Science and Mathematics, Lebanese American University, Lebanon, e-mail: sanaa.sharafeddine@lau.edu.lb. C. Assi is with the Faculty of Engineering and Computer Science, Concordia University, Montreal, Canada, e-mail: assi@ciise.concordia.ca.

is formulated. Then, the proposed RL algorithm (known as Q-learning algorithm) grant the UAV the required artificial intelligence to autonomously find the trajectory and consecutive waypoints so as to optimize the localization precision with considered capacity factor. **The novelty of our work concentrates on the fact that a smart UAV autonomously observes the environment and finds the trajectory that will result in the fastest multi-object localization with minimum errors, by only relying on RSS information, and taking into account the variation of shadowing with UAV elevation angle in urban areas.** The RL of the UAV operation is summarized as follows.

1) Initiate initial scan over the region to know the number of objects and estimate their initial positions, in addition, to training the RL agent online in a real scenario. Note that unlike the other works in the literature, since we assumed the number of objects is unknown, this phase is crucial.

2) Divide the region into equal cells (where the center of each cell is considered as a waypoint), observe the environment from the current UAV location (current state or cell), and estimate the probability of reward (i.e., the average localization error deduction) that may be gained by choosing any of the available actions (neighbor cells).

3) Exploit the optimal estimated policy by choosing the best action that maximizes the localization accuracy.

Note that, in the first step, through fast initial scan, the UAV will find the number of objects and their positions, however with low accuracy. In the next step, based on the time given to the UAV, it will improve the location accuracy of the objects. Therefore, in the rescue mission, the UAV through initial scan will find all the objects within shortest time possible, and then, will try to localize their positions more accurately. It should be noted that the proposed algorithm does not localize objects one by one. Instead, it does localization for multiple objects simultaneously based on UAV's communication range. When the algorithm through the initial scan finds the inaccurate position of all objects, the rescue mission can be started. Consequently, as we give more time to the algorithm, the location precision gets better, and the rescuers, in case they have not found trapped people yet, get more accurate information in their rescue mission.

In our proposed framework, we use detailed UAV to ground channel characteristics with an empirical path loss and log-normal shadowing model, in addition to an elaborate energy consumption model. We investigate the impact of different factors that affect the performance of UAV operation in localizing multiple objects. These factors range from UAV's altitude, velocity, hovering time, communication range, number of waypoints to the number of objects to be localized. Furthermore, we compare the performance of our proposed RL approach with methods from the literature that rely on pre-path algorithm. The results of our performance evaluation show that RL significantly helps in achieving better localization accuracy faster with available UAV energy, time, path length, or number of waypoints. Moreover, the results show that increasing the UAV's velocity, hovering time, communication range, and number of waypoints can remarkably decrease the localization error at the cost of longer path, higher energy consumption, or operational time. However, increasing the UAV's altitude does not always improve the localization performance. Although, higher altitude increases the probability of LoS and hence better localization accuracy, but, at the same time, it decreases the coverage area of the UAV and consequently results in fewer number of objects to be localized.

The remainder of this paper is organized as follows. Section 2 summarizes related work from the literature. Section 3 presents the system model and introduces the employed channel and energy models. The complete proposed RL framework is explained in Section 4, followed by UAV localization procedure in Section 5. The performance evaluation and analysis are presented in Section 6. Finally, Section 7 concludes the paper and proposes future directions.

2 Related Work

There is a quite number of works in the literature that investigated the localization problem. Among those, [3,8–12] studied object(s) localization using terrestrial anchors based on RSS measurements. In [3], the authors analyzed the main factors that affect the accuracy of the RSS measurements and suggested some techniques to alleviate the negative impacts of these factors. [8] proposed a distributed-based localization technique to achieve high accuracy without dense deployment. In [9], new schemes (cooperative and noncooperative) based on convex optimization are proposed to improve the localization accuracy. The authors of [10] evaluated the accuracy obtained through changing the height and distance of the anchors to terrestrial objects. While, [11] and [12] showed the importance of anchors' position and the requirement for their replacement in the objects localization accuracy.

Furthermore, several research studies addressed the localization problem using mobile anchors [2,13–19]. A survey of mobile node assisted localization problem is presented in [2]. [13] proposed a location verification using a random anchor movement. In [14], the authors studied three different pre-determined trajectories for a mobile anchor to traverse the whole area, and showed that any deterministic trajectory offers significant benefits compared to a random movement. In [15], a novel trajectory is proposed, where in this method, all deployed nodes are localized with high precision and short required time. In [16], another trajectory, named LMAT, is proposed. The authors in [17] presented a novel localization algorithm, where in their method, one mobile anchor incorporates least square method to estimate the location of terrestrial nodes. In [18], multiple location-aware mobile anchors localize the unknown nodes. For this purpose, the authors proposed two algorithms; one to control the trajectory of the mobile anchor, and another to extract the direction and distance of unknown nodes. In [19], the authors proposed a distributed technique using multiple mobile anchors which periodically broadcast beacon messages for localizing static sensors.

Localizing terrestrial objects using UAV or drone anchor(s) is studied thoroughly in the literature. [20] studied the advantages of using drone anchor. The authors of [21] proposed multiple path planing algorithms based on traveling salesman problem (TPS) for a UAV to localize all objects' positions. They used multilateration [22] to measure the position. Similarly, [23] presented a technique using triangulation that guarantees the localization precision. However, in

both approaches, only the instrumental error is considered. [24] improved the localization approach by equipping a UAV with directional antennas. [25] extended the approach even further by using omnidirectional antenna. Nonetheless, none of these works consider the characteristic model of UAV to ground channel (i.e., ground error due to UAV's altitude).

Different from the above studies, [26] proposed a solution on basis of an empirical path loss and log-normal shadowing model. In [27], the authors expressed the measurement error through conducting real experiments. The authors in [28] proposed a generic framework for the air-to-ground channel model that incorporates both height-dependent path loss exponent and small-scale fading. Moreover, they derived the optimal UAV height that minimizes the outage probability of an arbitrary air-to-ground link. In the same context, [29] introduced a scenario which results in an optimum UAV altitude for minimum localization error. The same authors of [29], in their new work [30], included a highly detailed UAV energy consumption model [31]. This enabled them to explore different tradeoffs between optimizing UAV trajectory and minimizing localization error. However, they did not consider the importance of timely localization. In addition, in their work, the number of terrestrial objects is known in advance, which is not the case in our work. Furthermore, our work is different from them in such, the autonomous UAV, by observing the environment, can better localize multiple objects simultaneously. Whereas, in [30], the UAV moves in circular trajectory to localize one object, with the objective of optimizing the energy consumption of the UAV subject to the number of waypoints and trajectory radius, while the distance between any two waypoints is fixed. In [4], the authors, not directly considering the path loss and shadowing characteristics, proposed a hybrid path planning algorithm to maximize the localization accuracy and minimize the energy cost represented by the length of the trajectory taken by a drone.

On the other hand, [6,32,33] discussed the importance of timely localization. The authors in [32] presented a study on smart phone localization of missing persons in search and rescue operations. However, they considered mobile anchors. Whereas, [33] used UAV system for search, rescue and surveillance based on RSS information. Moreover, the authors in [6], in order to improve the accuracy of radio-localization technology, introduced GuideLoc, a highly efficient aerial wireless localization system. GuideLoc allows a UAV, by getting RSS and angle-of-arrival (AOA) information, flies over a target device and provides positioning coordinates. The cost of installation of multiple antennas is one of the disadvantages of GuideLoc. Also, a UAV has to locate one target at a time which delays the process of localizing multiple objects.

To the best of our knowledge, no work has considered using a smart UAV to autonomously observe the environment and find the trajectory that results in faster multiple-object localization with minimum errors, by only relying on RSS information, and taking into account the variation of shadowing with UAV elevation angle in urban areas. There are several researches in the literature that focused on automating a UAV to navigate [34], or track object(s) [35]. However, few research works, like [36] and [37], have investigated in automating UAV to localize objects. In [36], the authors divided the geographical area into multiple zones, and based on continuously capturing the WiFi probe requests at different locations, using random-forest based machine learning technique, the UAV finds the zone where the terrestrial device is located. In [37] illegal radio station localization using a Q-learning technique is developed to process RSS values collected by a directional antenna, and determine the UAV's trajectory. Nevertheless, none of the autonomous UAV work presented in the literature considered the dependency of path loss and shadowing characteristics on the UAV altitude.

3 System Model

In this paper, we consider a UAV flying over an urban area at a fixed altitude h, acting as an aerial anchor to localize multiple terrestrial objects $N = \{n_1, n_2, n_3, ..., n_j, ...\}$. These objects are equipped with a wireless communication device which periodically broadcast a probe request. The UAV, in its trajectory, hovers for few seconds (τ) over certain points (referred to as waypoints $W = \{w_1, w_2, w_3, ..., w_i, ...\}$) to collect RSSI measurements from different objects in its communication range. The UAV obtains its distance to the object from the well-known path loss model equation [5]. Subsequently, the location of each object is estimated by the RSS measurements collected at different waypoints using the multilateration technique. As illustrated in Fig. 1, at each waypoint, a UAV may have a line-of-sight (LoS) or non-line-of-sight (NLoS) link with an object. In the figure, the direct distance between the UAV at waypoint w_i and object n_j is denoted by d_{ij}, and the ground distance is represented by r_{ij}. Moreover, the elevation angle is denoted by θ_{ij}. The search area is divided into equal cells. Each cell represents a waypoint (at the center of the cell). These cells are used for the UAV to traverse to maximize the localization precision. However, in our method, in order to train the agent (auto controller) of the UAV, and also to know the number of objects for better autonomous localization, an initial scan is needed. Hence, first we find a minimum number of initial scan waypoints (or scan-waypoints in short) and its shortest trajectory. Then, we let the UAV, using the RL method, autonomously find the optimal trajectory through the specified cell-waypoints (also referred as RL cells). In the following subsections, we define the initial scan, RL cells, and thoroughly explain the channel and energy consumption models used in our localization procedure.

3.1 Initial scan

In order to know the number of objects in the search area, a UAV has to scan and cover the whole area using a minimum number of waypoints so as to optimize the cost (i.e., cost of energy consumption, number of waypoints, path length, or time to scan the area). To do that, we first divide the area into minimum number of equal cells, where each cell is covered by the communication range of one UAV when placed in the middle of the cell. As depicted in Fig. 2, if we let L_x and L_y respectively be the length and width of the area. Then,

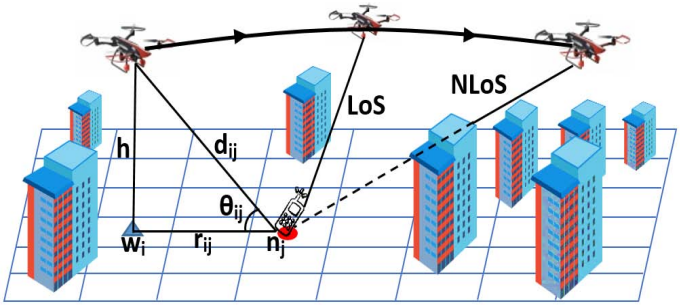


Fig. 1. **Illustration example of collecting RSSI measurements using one UAV in localizing a terrestrial object. The arrows show the moving direction of the UAV.**

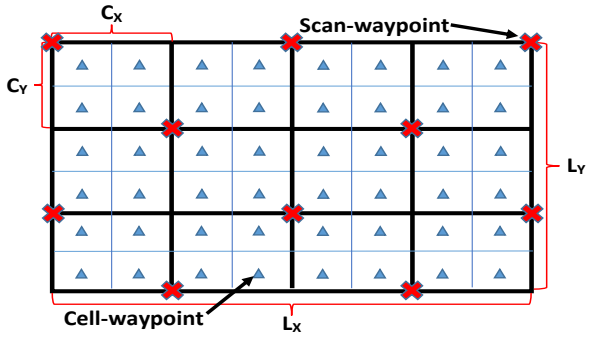


Fig. 2. Illustration example of dividing the area into RL-cells and finding initial scan waypoints.

the length C_x and width C_y of the guaranteed covered cell is obtained from the following:

$$C_x = \frac{L_x}{\lceil \frac{L_x}{R} \rceil} \tag{1}$$

$$C_y = \frac{L_y}{\lceil \frac{L_y}{R} \rceil} \tag{2}$$

where $R = \sqrt{D^2 - h^2}$ is the ground coverage range of the UAV, and D is the actual communication range of the UAV. The location of the scan-waypoints is obtained through Algorithm 1, and illustrated in Fig. 2 by red X-signs. As shown in Fig. 3 by blue circles, all the search area has been covered by the UAV's communication range using scan-waypoints. Moreover, the trajectory of the UAV over the scan-waypoints is shown in the figure by a black line, where it sequentially follows the nearest scan-waypoint.

3.2 Reinforcement learning cells

To get the RL cells, it is as simple as dividing each coverage cell (i.e., length C_x and width C_y) into equal multiple cells. The minimum number of possible RL cells in a coverage cell is four (i.e., dividing the edges into two equal parts, $\omega = 2$), as illustrated in Fig. 2. Alternatively, the number of edge partitions ω can be increased as needed. We analyze the effect of the number of RL cells on localization precision in the numerical result section (Section 6). Algorithm 2 demonstrates the steps to find the RL cell waypoints, and Fig. 4 shows an example of RL trajectory.

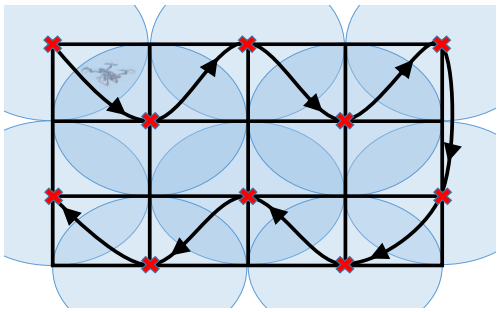


Fig. 3. Initial scan-waypoints, their coverage range, and trajectory.

Algorithm 1: Finding initial scan waypoints

Data: L_x, L_y, C_x, and C_y
Result: Set of initial scan waypoints $Snodes$

```
Snodes = ∅
x = 0, y = 0
while x ≤ L_x do
    temp = y
    while y ≤ L_y do
        Snodes.append((x, y))
        y = y + 2C_y
    if temp = 0 then
        y = C_y
    else
        y = 0
    x = x + C_x
```

3.3 Channel model

The air to ground channel model, by incorporating the dependencies of shadowing and path loss exponent with the elevation angle ($\theta = \tan^{-1}(h/r)$), is given by [38]:

$$PL = 20\,log\,(d) + 20\,log\,(\frac{4\pi f}{c}) + \Psi(\theta) \tag{3}$$

where f and c are respectively the system frequency and speed of light, and $\Psi(\theta)$ is a log-normal distributed random variable with mean μ and variance $\sigma^2(\theta)$ [5], i.e.,

$$\Psi(\theta) \sim \mathcal{N}(\mu, \sigma^2(\theta)) \tag{4}$$

given that $\mu = 0$, and $\sigma^2(\theta)$ can be written as:

$$\sigma^2(\theta) = \mathbb{P}^2_{LoS}(\theta)\,\sigma^2_{LoS}(\theta) + [1 - \mathbb{P}_{LoS}(\theta)]^2\,\sigma^2_{NLoS}(\theta) \tag{5}$$

where $\sigma_{LoS}(\theta)$ and $\sigma_{NLoS}(\theta)$ correspond respectively to the shadowing effect of LoS and NLoS links between the UAV and object, and they are expressed as:

$$\sigma_{LoS}(\theta) = a_{LoS}\exp(-b_{LoS}, \theta) \tag{6}$$

$$\sigma_{NLoS}(\theta) = a_{NLoS}\exp(-b_{NLoS}, \theta) \tag{7}$$

and $\mathbb{P}_{LoS}(\theta)$ is the probability of having LoS link, which is given by:

$$\mathbb{P}_{LoS}(\theta) = \frac{1}{1 + a_0\exp(-b_0, \theta)} \tag{8}$$

Algorithm 2: Finding RL cell waypoints

Data: L_x, L_y, C_x, C_y, and ω
Result: Set of initial scan nodes $Snodes$

```
Cnodes = ∅
x = C_x/2ω
y = C_y/2ω
while x ≤ L_x do
    while y ≤ L_y do
        Cnodes.append((x, y))
        y = y + C_y/ω
    y = C_y/2ω
    x = x + C_x/ω
```

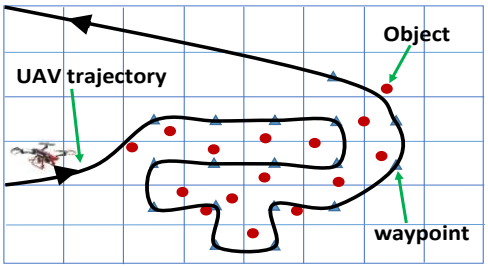


Fig. 4. Illustration example of RL trajectory.

where a_0, b_0, a_{LoS}, b_{LoS}, a_{NLoS}, and b_{NLoS} are environment dependent parameters. The reader is referred to [38] for more insights regarding the channel model.

3.4 Power consumption model

In this subsection, we present a suitable simple power consumption model for a UAV following the work presented in [30] and [39]. From the fact that the energy consumption of data communication is negligible compared to the energy required to keep the UAV aloft and fly, we compound the model into three main power consumption sources:

3.4.1 Blade profile power

This power is required to turn the rotors' blade, and it is given by:

$$\mathbf{P}_{blade} = K(1 + 3\frac{v^2}{v_b^2}) \tag{9}$$

where v is the UAV velocity, v_b is the blade's rotor speed, and K represents a constant which depends on the dimensions of the blade.

3.4.2 Parasite power

The power is used to overcome the drag force resulted from moving through the air.

$$\mathbf{P}_{parasite} = \frac{1}{2}\rho v^3 F \tag{10}$$

ρ is the air density, and F represents a constant that depends on the UAV drag coefficient and reference area.

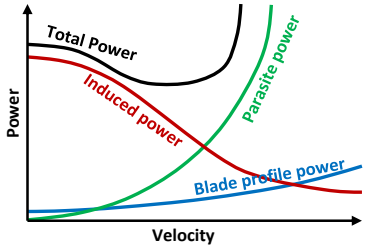


Fig. 5. Three main sources of power consumption vs. UAV velocity [30, 39].

Note that this power is proportional to the UAV velocity v; it is zero when hovering and gradually increases by the speed of the UAV.

3.4.3 Induced power

This power is required to lift the UAV and overcome the drag caused by the gravity. Whenever a UAV is moving, the airflow coming at it redirects the UAV and helps to lift it. Hence, the induced power has inverse proportion to the airspeed. When hovering, all the airflow needed to lift the UAV has to be created by the blade rotors, which results in more power consumption. The induced power can be written as follows:

$$\mathbf{P}_{induced} = mgv_i \tag{11}$$

where m and g respectively denote the mass of the UAV and the standard gravity, whereas, v_i represents the mean propellers' induced velocity in the forward flight, and it is given by:

$$v_i = \sqrt{\frac{-v^2 + \sqrt{v^4 + (\frac{mg}{\rho A})^2}}{2}} \tag{12}$$

with A being the area of the UAV.

Now, when the UAV is flying from one waypoint to another, the total power consumption is obtained from the following:

$$\mathbf{P}_{total} = \mathbf{P}_{blade} + \mathbf{P}_{parasite} + \mathbf{P}_{induced} \tag{13}$$

However, in case of hovering, when the UAV needs to collect RSSI measurements (i.e., when v = 0), the total power consumption is limited to hovering power and is calculated accordingly:

$$\mathbf{P}_{total} = \mathbf{P}_{hover} = K + \sqrt{\frac{(mg)^3}{2\rho A}} \tag{14}$$

In Fig. 5, we demonstrate the trend of the three power consumption factors along with the total power versus the UAV velocity. From the figure, we can conclude that at optimal speed, the UAV consumes less power compared to hovering time (when $v = 0$). Therefore, in order to maximize the localization precision with the knowledge of limited UAV energy, it is not always advisable to maximize the number of waypoints.

4 The Reinforcement Learning Approach

As mentioned, using the multilateration technique, in order to find the position of objects with less localization errors, a UAV has to follow more waypoints. However, with limited number of waypoints, UAV energy, flying time, or path length, a certain UAV trajectory results in optimal localization precision. Hence, in this section, we let the UAV, by observing the environment and using RL, learn and autonomously find the best trajectory that leads to minimum localization errors. In the following, we first briefly review RL, a machine learning technique which is suitable for controlling an autonomous machine such as UAV. Then, we introduce our approach using RL for efficient UAV localization.

4.1 Reinforcement learning background

RL is a branch of machine learning paradigm, which deals with multi-state decision process of a software agent (UAV in our case) while interacting with an environment. In general, RL assumes the system consists of multiple states S (waypoints in this case), where at each state $s_t \in S$, the agent has a finite number of actions A (i.e., neighboring waypoints) to choose from. After choosing an action $a_t \in A$, the agent receives a reward $r(s_t, a_t)$, and moves to the next state s_{t+1}. The goal of RL is to learn from the transition tuple $\langle s_t, a_t, r(s_t, a_t), s_{t+1}\rangle$, and find an optimal policy π^* that will maximize the cumulative sum of all future rewards. Note that the policy $\pi = \{a_1, a_2, ..., a_T\}$ defines which action a_t should be applied at state s_t. If we let $r(s_t, \pi(a_t))$ denote the reward obtained by choosing policy π, the cumulative discount sum of all future rewards using policy π is given by:

$$R_\pi = \sum_{t=1}^{T} \gamma^{t-1} r(s_t, \pi(a_t)) \tag{15}$$

where $\gamma \in [0, 1)$ is a discount factor, which measures the weight given to the future rewards (i.e., when $\gamma = 0$, the agent considers only the current received rewards, whereas, when the factor approaches one, the agent strives for future higher reward). Now, let Λ denote the set of all admissible policies. Then, the optimal policy is given by:

$$\pi^* = \underset{\pi \in \Lambda}{\operatorname{argmax}} R_\pi \tag{16}$$

Note that RL is modeled as a Markov Decision Process (MDP), where the tuple $\langle s_t, a_t, r(s_t, a_t), s_{t+1}\rangle$ is conditionally independent of all previous states and actions. Therefore, the agent does not need to memorize or save all the state-action tuples, just the last one, and subsequently updates it at each cycle or iteration. In this work, we use Q-learning [40], one of the widely used RL algorithms, which allows the agent to optimally act in an environment represented by an MDP. Q-learning iteratively improves the state-action value function (also known as Q-function or Q-value), and by estimating the future reward if action a_t is taken, presents the probability of going from state s_t to s_{t+1} using policy π. The optimal Q-value function is given by:

$$Q^*(s_t, a_t) = E[R(s_t, a_t) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})] \tag{17}$$

Once we have the optimal Q-function at state s_t, it is easy to obtain the optimal policy simply by choosing the best Q-value from the current available action as follows:

$$\pi^*(s_t) = \underset{a_t}{\operatorname{argmax}} Q^*(s_t, a_t) \tag{18}$$

It should be noted that, the Q-value function is usually stored in a table. Now, starting from an arbitrary Q-value, each time the agent wants to take an action, it approximates the optimal Q-function based on the observations of the environment, updates the Q-value according to equation (19) and stores it into the table. The parameter $\alpha \in [0, 1]$ denotes the learning rate. In other words, it determines to what extent the old Q-values are overridden (i.e., when $\alpha = 0$, Q-value is not updated and thus nothing is learnt, whereas, when $\alpha = 1$, it means the agent learns quickly).

$$\begin{aligned} Q(s_t, a_t) \leftarrow (1 - \alpha) Q(s_t, a_t) \\ + \alpha[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})] \end{aligned} \tag{19}$$

From the fact that Q-learning is an iterative algorithm, under certain conditions [40], the Q-value function will converge to optimal policy $Q^*(s_t, a_t)$, if the number of iterations approaches infinity. For more background information on RL, the reader is referred to [41].

4.2 Proposed solution approach

In this subsection, we introduce our RL approach for UAV multi-object localization. As explained earlier, the current RL state s_t is the waypoint (or cell) that the UAV is hovering at time t to measure RSSI from all objects in its communication range. Subsequently, all available neighbor waypoints (cells) are actions to choose from to move to a next waypoint in state s_{t+1}. While visiting a waypoint, the UAV by taking the RSSI measurements, observes the environment and calculates the reward $r(s_t, a_t)$ obtained from choosing action a_t. Concurrently, for each available action a_t, the probability of going from state s_t to state s_{t+1}, i.e., $Q(s_t, a_t)$, is estimated through equation (19). The way to obtain the reward and Q-value for our RL-UAV localization is explained thoroughly in Section 5.

The main problem with RL on autonomous UAV localization is in the early stage of the learning process. It is obvious that the agent, at the early stage, knows very few or nothing about the environment, and thus, somehow chooses an arbitrary action. As the agent starts learning by iteratively taking more actions and receiving rewards from the environment, it can improve its approximation value $Q(s_t, a_t)$ and better decide on its next step. Hence, similar to the work in [42] for mobile robots, to boost the learning curve of the RL system, as illustrated in Fig. 6, we split the learning policy into two phases: 1) initial controlled scan trajectory, and 2) standard RL implementation.

In Phase 1, the UAV is controlled by a pre-path trajectory algorithm. The algorithm is designed to let the UAV visit scan waypoints (as shown in Fig. 3) in order to train the agent online and get information from the environment for learning. During Phase 1, the agent by observing the environment (i.e., watching the states, actions, and rewards), bootstraps information into its Q-value function approximation $Q(s_t, a_t)$. Subsequently, after this learning phase, the

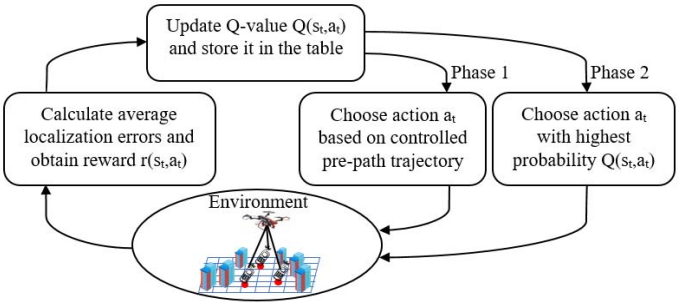


Fig. 6. Learning policy using two phases: 1) initial controlled scan trajectory, and 2) standard RL implementation.

agent will be ready to control the UAV. In Phase 2, the UAV, using the information of Q-value function approximation, autonomously traverses the area and visits waypoints to increase the average localization accuracy. In this learning phase, as the standard RL implementation, the learning policy is in the control of the UAV.

5 The UAV Localization Error

In this section, we explain how the UAV estimates the position of multiple objects through received RSSI, and regularly using multilateration minimizes the average location errors. In other words, this section describes how to obtain the reward $r(s_t, a_t)$ and estimated future Q-value function $Q(s_{t+1}, a_{t+1})$ for RL implementation. Here, we illustrate the localization procedure for one object and similarly is done for other objects. Eventually, the average localization errors from all objects will be our measured quantity for the RL reward and Q-value at each state.

Fig. 7 shows the localization error reduction of an object using the multilateration technique. The object is shown by a red point, waypoints by blue triangles, and object estimated location area by shaded blue color. In the first step (depicted in Fig. 7(a)), by getting RSSI measurement at one waypoint, following the air to ground channel model (3) in section 3.3, the position of the object is estimated in the shaded blue area between the inner (I_1) and outer (O_1) circles. The radius of these circles is dependent on the shadowing and path loss exponent. Next, when the UAV moves to the next waypoint and takes another RSSI measurement (Fig. 7(b)), the localization area shrinks. Whenever the number of measurements becomes three (Fig. 7(c)), the position of the object can be estimated using trilateration, and subsequently, the calculation of the localization error. As the number of waypoints and RSSI measurements increase, the localization error likely decreases (as illustrated in Fig. 7(d).

Fig. 8 shows how we obtain the error for one object using three waypoints. The intersection point between three lines that connect inner and outer circles presents the estimated location of the object. Consequently, the localization error can be obtained by finding the farthest border point to the estimated object point as shown by the black line in the figure. Let us assume that the Cartesian coordinate for the estimated location of the object is $(\hat{x}, \hat{y})$. Let (x_{w_i}, y_{w_i}) be the known ground position of the UAV at waypoint i, and $\bar{r_i} = \frac{O_i + I_i}{2}$ be the distance from waypoint i to the middle of the two circles, then the estimated position $(\hat{x}, \hat{y})$ using M number of waypoints can be obtained from the following optimization model:

$$(\hat{x}, \hat{y}) = \underset{\hat{x},\hat{y}}{\operatorname{argmin}} \{ \sum_{i=1}^{M} (\sqrt{(x_{w_i} - \hat{x})^2 + (y_{w_i} - \hat{y})^2} - \bar{r_i})^2 \} \tag{20}$$

The border points of the estimated area of the object are created each by the intersection of two communication circles. Fig. 9 illustrates how a border point is found. From the figure, r_1 and r_2 are respectively the communication radius of waypoints w_1 and w_2, and k is the distance between the two waypoints. P_1 and P_2 are the required intersection points between two circles, and P_0 is the intersection point of the perpendicular line connecting P_1 and P_2 with line k. Respectively, q_1 and q_2 denote the distances from w_1 to P_0, and from P_0 to w_2, respectively. Now, if we let (x_{w_1}, y_{w_1}), (x_{w_2}, y_{w_2}), (x_{P_0}, y_{P_0}), (x_{P_1}, y_{P_1}), and (x_{P_2}, y_{P_2}) denote respectively the Cartesian coordinates for points w_1, w_2, P_0, P_1, and P_2, then the border points are calculated through the following equations:

$$x_{P_1,P_2} = x_{P_0} \pm \frac{(y_{w_2} - y_{w_1})h}{k} \tag{21}$$

$$y_{P_1,P_2} = y_{P_0} \mp \frac{(x_{w_2} - x_{w_1})h}{k} \tag{22}$$

where $(x_{P_0}, y_{P_0}) = (x_{w_1} + \frac{(x_{w_2} - x_{w_1})q_1}{k}, y_{w_1} + \frac{(y_{w_2} - y_{w_1})q_1}{k})$, $q_1 = \frac{r_1^2 - r_2^2 + k^2}{2k}$ and $h = \sqrt{r_1^2 - q_1^2}$.

After a new RSSI measurement, the accuracy of the estimated object localization area is updated through the following steps and illustrated in Fig. 10:

1) Remove border points, if any, that position outside the outer circle (O_{new}) and inside the inner circle (I_{new}) as shown in the figure as red points.
2) Add new intersection points (shown in the figure as blue points) if they do not reside inside and outside of any inner and outer circles of old measurements.
3) Find distances from all obtained area points to the estimated object point, and the one with farthest distance is the object's localization error.

After obtaining the localization error, as explained earlier, for all terrestrial objects which are within the UAV's communication radius in current state s_t, we average over all these error values. We retrieve then the stored localization error values from previous state s_{t-1} and compute their average. The difference between these two average errors is considered as the current reward $r(s_t; a_t)$. Then, we store the obtained error values from current state into the table, which will be used for subsequent reward computation. Similarly, we estimate the future average localization errors for all available neighbor waypoints or actions, and we update the approximated Q-value function for all actions and store them into the table. Subsequently, for the next iteration, we choose the action that results in higher reward by looking at the stored Q-value functions. To be noted that the future estimated average localization errors is not obtained through the RSSI measurements, however, it is calculated based on estimation of RSSI without visiting the new waypoint or taking the action.

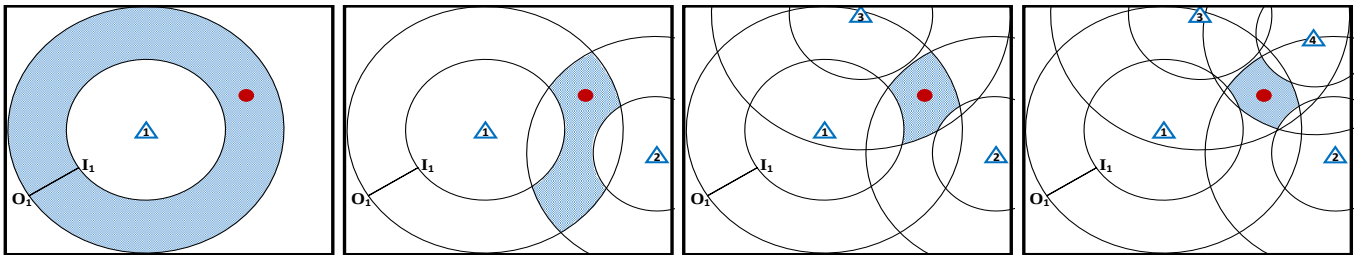


Fig. 7. Illustration of localization error reduction for one object using four UAV measurements.

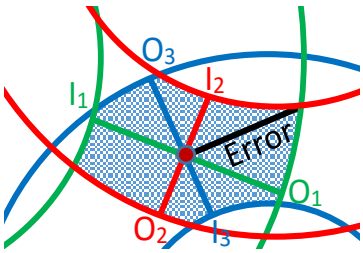


Fig. 8. Illustration of position estimation and error calculation for one object.

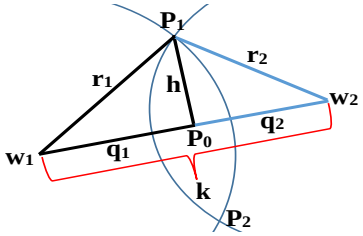


Fig. 9. Illustration of obtaining the intersection point of two communication circles.

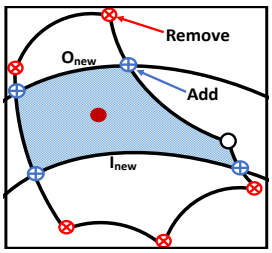


Fig. 10. Example of how the border nodes are updated after a new UAV measurement.

Algorithm 3: LMAT traverse steps from the down right most cell.

1 From the current location, if there is no neighbor cell to traverse, terminate.
2 Else if there is one untraversed neighbor cell, choose it as the next traverse node (cell).
3 Else if the down corner cell is untraversed, choose it.
4 Else if the down cell is untraversed, choose it.
5 Else if upper cell is untraversed, choose it.
6 Else traverse any available upper corner cell.

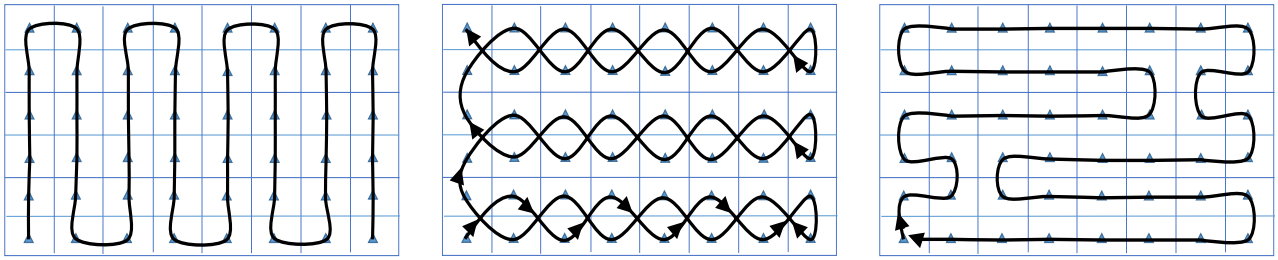
6 Performance Evaluation

In this section, we evaluate the performance of our RL approach in localizing terrestrial objects numerically. We generate at random the locations of the IoT devices which we want the UAV to localize. Based on UAV's altitude and the probability of LoS, as explained in Section 3.3 for variance $\sigma^2(\theta)$, we compute the range (between inner-circle I_i and outer-circle O_i) where the object is located from the ground position of UAV's waypoint i. Consequently, the area obtained from the intersection of multiple inner and outer circles (or ranges) is considered as the location area of the object. Hence, the localization error or accuracy can be measured by calculating the distance from the farthest border node in this area to the center of the area. Subsequently, by adding more UAV waypoints, the location error is minimized.

We compare the performance of our RL approach with a method that chooses a random direction for a UAV to localize objects (Random Path), and three other state-of-the-art pre-path trajectory methods: 1) SCAN path [25] (see Fig. 11(a)): the UAV follows a path formed by vertical straight lines interconnected by horizontal lines. 2) LMAT path [16] (see Fig. 11(b)): the UAV follows a path formed by equilateral triangles such that all the waypoints are visited once. This path here is updated to fit our region and cell division. Algorithm 3 illustrates the UAV traverse steps to create the LMAT path for our environment. 3) MAZE path (see Fig. 11(c)): the UAV follows a path which eventually creates a shape of maze. This path is deduced from the path planing algorithm named LocalizerBee [21]. Algorithm 4 presents the steps to build up the MAZE path for UAV trajectory.

In this section, we first study the performance of all mentioned methods above for localizing 20 and 30 terrestrial objects by varying UAV's energy, trajectory length, number of waypoints, and UAV flying time. We then study the performance of our RL method by varying the UAV altitude and communication range. We further evaluate the localization accuracy by modifying the UAV velocity and hovering time. Finally, we observe the localization error by changing the number of terrestrial nodes and cells.

For the numerical study, we assume N terrestrial nodes which are randomly distributed in a region of $900 \times 700m^2$, where the region is divided into M equal cells. We also assume a UAV is flying at a fixed altitude h, and the hovering time τ is equal at each waypoint. Further, we assume the communication range of all nodes is equal and the UAV can measure the RSSI from nodes within radius D. The parame-



(a) (b) (c)

Fig. 11. Three different pre-path trajectory methods from the literature: (a) SCAN, (b) LMAT, and (c) MAZE.

Algorithm 4: MAZE traverse steps from the down right most cell.

```
Traverse to the upper cell if the number of X-axis cells is even.
Let the number of allowed moves be equal to one (i.e., NoMove = 1).
while we did not reach the most upper cell do
    Based on the NoMove, traverse to the right cells.
    Traverse one cell up.
    Traverse to the left most cell.
    NoMove++;
    if NoMove = number of Y-axis -2 then
        NoMove = 1;
    Traverse one cell up.
Traverse to the right most cell.
Traverse one cell down.
NoMove = Number of Y-axis cells - 2;
while we did not reach the most down cell do
    Do the same procedure similar to traversing upwards but in opposite direction.
```

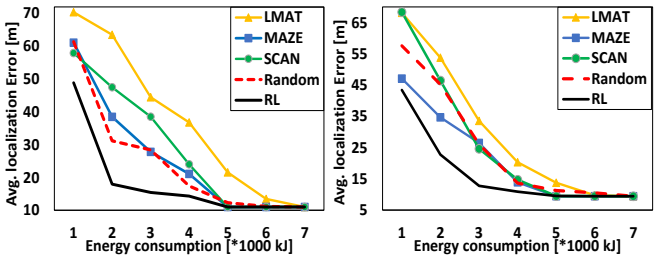
ters used in these numerical results and their corresponding values (taken and recommended by [5,28,30,38] for urban environments) are listed in Table 1, unless otherwise stated. We use Python as a programming language to simulate the operation of the proposed methods, and the numerical results are averaged over ten runs.

6.1 Comparing the performance of different methods with limited UAV energy, path length, number of waypoints, or flying time

We start by examining the results obtained by solving the RL approach and compare it with the results obtained from the random path, SCAN, LMAT, and MAZE. For comparison we acquire the localization error of 20 and 30 terrestrial objects, and the region is divided into $M = 120$ equal cells (or waypoints). Fig. 12 shows the average localization error by varying the energy consumption of the UAV from $1000kJ$ to $7000kJ$. **As depicted in the figure, if the UAV's energy is sufficient to traverse all the waypoints (for example 7000kJ), all methods eventually perform equally** (the ultimate average localization error for 20 (respectively 30) nodes is around $11m$ (res. $9.4m$) for a total of 120 waypoints). Note that this shows the fairness for all

TABLE 1
Description of the parameters used and their corresponding values.

Parameter	Description	Value
$L_x \times L_y$	Area dimensions [m*m]	900×700
h	UAV's altitude [m]	100
D	UAV's com. range [m]	200
τ	Hovering time [sec]	5
v	UAV's velocity [Km/h]	40
v_b	Rotor speed	100
K	Blade dimension constant	570
ρ	Air density	1.225
F	Drag and reference area coefficient	0.4
m	UAV mass [Kg]	5
A	UAV surface area [m^2]	0.25
a_0	Environment parameter for P_{LoS}	45
b_0	Environment parameter for P_{LoS}	10
a_{LoS}	shadowing constant for LoS	10
b_{LoS}	shadowing constant for LoS	2
a_{NLoS}	shadowing constant for NLoS	30
b_{NLoS}	shadowing constant for NLoS	1.7



(a) (b)

Fig. 12. Average localization error in meter versus UAV energy consumption. Number of localizing objects is (a) 20 nodes, and (b) 30 nodes.

methods. However, when the UAV's energy is limited, the performance of different methods varies. For instance, for localizing 20 nodes, when the energy is limited by $2000kJ$ (respectively $4000kJ$), the RL approach, Random, SCAN, LMAT, and MAZE perform $18m$ (res. $14.3m$), $31.2m$ (res. $17.4m$), $47.5m$ (res. $24m$), $63.5m$ (res. $36.8m$), and $38.5m$ (res. $21m$) respectively. The RL approach, as expected, always outperforms the other methods for both 20 (Fig. 12(a)) and 30 (Fig. 12(b)) terrestrial nodes. It should be noted here that such gains are attributed to the intelligent movement and trajectory of the UAV. As for the Random method, because of randomness, we can not predict its behavior.

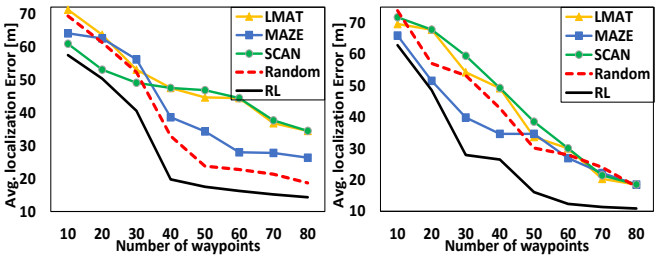


Fig. 13. Average localization error in meter versus number of UAV waypoints. Number of localizing objects is (a) 20 nodes, and (b) 30 nodes.

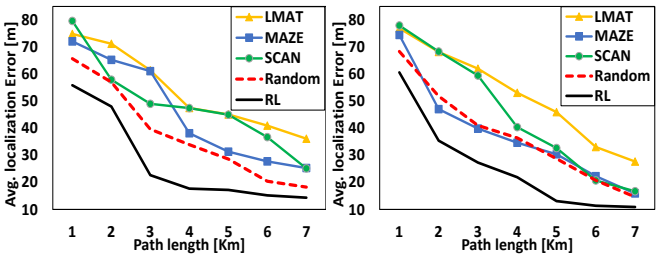


Fig. 14. Average localization error in meter versus UAV trajectory distance. Number of localizing objects is (a) 20 nodes, and (b) 30 nodes.

Whereas, for the other methods, their performance depends on the random distribution of terrestrial nodes. However, in all of the methods, the localization accuracy improves by consuming more energy and hence traversing more waypoints. This improvement is also shown in Fig. 13. In Fig. 13(a) (res. Fig. 13(b)), the average localization error for the RL approach reduces from $57.4m$ (res. $62.9m$) to $14.3m$ (res. $10.9m$).

Fig. 14 depicts the localization accuracy by varying the path length of the UAV trajectory from one to seven kilometers. As plotted in the figure, the RL approach, for localizing 20 (respectively 30) nodes, shown in Fig. 14(a) (res. Fig. 14(b)) respectively performs in the worst case 14.9% (res. 11.3%), 17.1% (res. 22.2%), 25.3% (res. 21.2%), and 22.5% (res. 18.6%), and in the best case the RL approach performs 47.8% (res. 54.6%), 62.7% (res. 60%), 63% (res. 71.6%), and 62.9% (res. 57%) better than Random, SCAN, LMAT, and MAZE. The figure also shows that the average localization error for 20 nodes, by increasing the path length of UAV trajectory, reduces faster than localizing for 30 nodes. However, the latter shows better accuracy than the former one. Fig. 15 illustrates the average localization error by varying the UAV flying time from one to 20 minutes. Similar to the above figures, as the UAV invests more time in localizing terrestrial nodes, the average localization error decreases. For instance, the average localization error for 30 nodes after five minutes fly using the RL approach is $33.6m$, whereas, after 15 minutes fly, the average localization error reaches $11.4m$.

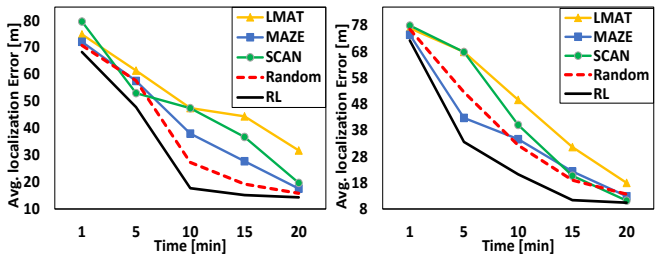


Fig. 15. Average localization error in meter versus UAV flying time. Number of localizing objects is (a) 20 nodes, and (b) 30 nodes.

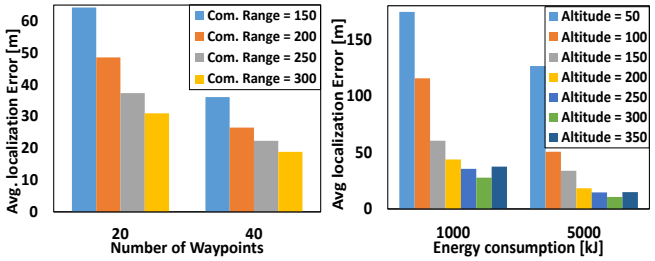
6.2 The effect of UAV altitude and communication range on the performance of RL approach

In this subsection, we study the performance of our RL approach in terms of average localization error, and we set the number of terrestrial nodes to 30. Fig. 16(a) shows the effect of UAV communication range on the localization accuracy by varying the range from $150m$ to $300m$ with 50 meters interval. For comparison, we limit the number of waypoints to 20 and 40. As shown in the figure, by increasing the communication range of a UAV, the average localization error decreases exponentially. It should be noted that when the communication range increases, a UAV can measure the RSSI from more terrestrial nodes and hence the average localization error decreases. The figure also shows that the localization accuracy is enhanced by increasing the number of waypoints. For instance, when the communication range of a UAV is 200 meters, the average localization errors are $48.6m$ and $26.5m$ after visiting 20 and 40 waypoints respectively, and when the communication range is 300 meters, the localization errors are $31m$ and $18.9m$ respectively.

In Fig. 16(b), we illustrate the localization accuracy by varying the UAV's altitude from $50m$ to $350m$ with interval of 50 meters. Here, we set the communication range of the UAV to 400 meters, and for comparison purposes, we limit the UAV's energy consumption to $1000kJ$ and $5000kJ$. As the figure shows, increasing the UAV's altitude does not always improve the localization performance. It should be noted that although higher altitude increases the probability of LoS and thus better localization accuracy, but, at the same time, it decreases the coverage area of the UAV and consequently results in fewer number of objects to be localized. In this example, the optimal altitude is 300 meters; as seen from the figure, the average localization error after consuming $5000kJ$ (res. $1000kJ$) is $10.6m$ (res. $27.8m$).

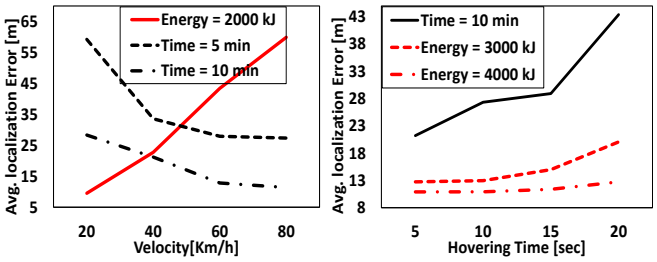
6.3 Localization accuracy versus UAV velocity and hovering time

In this subsection, we evaluate the performance of RL approach by varying the UAV velocity and hovering time. Here, the number of localized objects is set to 30 nodes. We start by evaluating the performance by changing the UAV velocity (20, 40, 60 , and 80 Km/h) in Fig. 17(a). The figure plots the average localization error and UAV velocity under three different stopping criteria: 1) five minutes flying time, 2) ten minutes flying time, and 3) stopping after consuming 2000kJ UAV energy. It is clear that with limited energy



(a) (b)

Fig. 16. Performance of the Reinforcement Learning (RL) method. Here the number of localizing objects is 30 nodes. The UAV communication range for figure (b) is 400 meters.

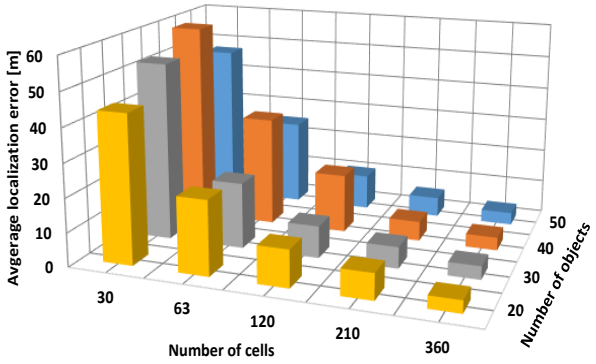


(a) (b)

Fig. 17. Performance of the Reinforcement Learning (RL) method. Here the number of localizing objects is 30 nodes.

the localization error increases along with increasing the UAV velocity. Since, the UAV, in order to move from one waypoint to another, requires more energy to accelerate and maintain higher speed. Therefore, as shown in the figure, the error increases linearly (e.g., $9.5m$ error with velocity $20km/h$, $22.7m$ error with velocity $40km/h$, and $43.4m$ error with velocity $60km/h$). From the figure, we can also observe that, with five or ten minutes flying time, the localization error decreases with the increasing of UAV velocity. This is expected since, with higher speed and fixed flying time, the UAV can visit more waypoints and take more RSSI measurements, and thus, more chances to enhance the localization accuracy. In addition, the longer the flying time, the better the accuracy is. For instance, with five (res. ten) minutes flying time, when we change the UAV velocity from $20km/h$ to $80km/h$, the localization accuracy enhances 53.8% (res. 59.8%).

Fig. 17(b) shows the performance of our approach by varying the hovering time (5 to 20 with 5 seconds interval) of the UAV to take RSSI measurements. The figure plots the average localization error for $3000kJ$ and $4000kJ$ UAV energy, and ten minutes flying time. In fact, whenever we increase the hovering time, the average localization error increases. Recall that from Section 3.4 and equation (14), the UAV consumes power even when it is hovering. Hence, by increasing the hovering time, more energy is depleted and consequently, with limited available energy (whether $3000kJ$ or $4000kJ$ as illustrated in the figure), fewer waypoints can be visited and therefore the accuracy will not be enhanced intensively. However, as explained earlier, the system can achieve better accuracy when it has energy to traverse more waypoints. Furthermore, by looking at the



	30	63	120	210	360
20	43.849	21.961	10.992	7.659	3.663
30	52.719	19.391	9.428	6.294	3.908
40	59.034	32.434	17.382	5.769	3.832
50	47.336	25.301	10.401	5.774	3.652

Fig. 18. Performance of the RL method by varying the number of localization objects and the total number of waypoint cells.

figure, it is clear that, with a total of ten minutes flying time, the performance of the system degrades by increasing the hovering time.

6.4 Localization error versus number of terrestrial nodes and RL cells

Finally, we study the performance of our approach by considering changing the number of terrestrial objects to be localized (20 to 50 nodes with 10 nodes interval), and number of RL cells (30, 63, 120, 210, and 360 cells) for RSSI measurements. The results are shown in Fig. 18. Note that the number of cells depends on region's dimensions (L_x and L_y). The plots in the figure show that, for any number of terrestrial objects, the localization error decreases exponentially with increasing the number of cells. The reason goes for taking more RSSI measurements, and consequently shrinking the localization area for most of objects. For instance, for 50 objects, the average localization error decreases 46.5% from 30 to 63 cells, 58.9% from 63 to 120 cells, 44.5% from 120 to 210 cells, and 36.8% from 210 to 360 cells. However, as shown in the figure, the average localization accuracy does not depend on the number of terrestrial objects distributed randomly in the region. So, localizing larger number of objects does not mean the average localization accuracy is better. For example, as plotted in the figure, for 120 cells, the average localization error for localizing 40 objects is 45.8% worse than localizing 30 objects, and 36.8% worse than 20 objects. Whereas, for 210 cells, the average localization error for localizing 40 objects is 8.3% better than localizing 30 objects, and 24.7% better than 20 objects.

7 Conclusions

In this paper we proposed a novel framework using RL to let a UAV autonomously traverse a trajectory that results in finding the position of multiple ground objects with minimum average localization error under fixed amount of UAV energy consumption, trajectory length, number of

waypoints, or flying time. The framework for localization considers detailed UAV to ground channel characteristics along with an empirical path loss and log-normal shadowing model, in addition to an elaborate energy consumption model. Our RL approach consists of two phases: In phase one, the UAV is controlled through an initial scan trajectory to know the number of terrestrial objects and to train the UAV's agent online, in a real scenario. In the second phase, the UAV, based on what it learned in phase one, controls its movement. Through numerical evaluation we showed the superiority of our approach in terms of average localization error compared to existing methods in the literature. Furthermore, we studied the impact of UAV's velocity, altitude, hovering time, communication range, number of maximum RSSI measurements, and number of objects on the localization accuracy. For future work, we intend to study the situation where a UAV can change its altitude based on probability of LoS and on communication range to better localize multiple objects. In addition, we would like to see the impact of using multiple collaborative UAVs in localizing ground objects.

REFERENCES

[1] W. Alshrafi, U. Engel, and T. Bertuch, "Compact controlled reception pattern antenna for interference mitigation tasks of global navigation satellite system receivers," *IET Microwaves, Antennas & Propagation*, vol. 9, no. 6, pp. 593–601, 2014.

[2] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 2220–2243, 2016.

[3] A. Zanella, "Best practice in rss measurements and ranging," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2662–2686, 2016.

[4] A. Rubina, O. Artemenko, O. Andryeyev, and A. Mitschele-Thiel, "A novel hybrid path planning algorithm for localization in wireless networks," in *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. ACM, 2017, pp. 13–16.

[5] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Global Communications Conference (GLOBECOM), 2014 IEEE*. IEEE, 2014, pp. 2898–2904.

[6] A. Wang, X. Ji, D. Wu, X. Bai, N. Ding, J. Pang, S. Chen, X. Chen, and D. Fang, "Guideloc: Uav-assisted multitarget localization system for disaster rescue," *Mobile Information Systems*, vol. 2017, 2017.

[7] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[8] C. Liu, D. Fang, Z. Yang, H. Jiang, X. Chen, W. Wang, T. Xing, and L. Cai, "Rss distribution-based passive localization and its application in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2883–2895, 2016.

[9] S. Tomic, M. Beko, and R. Dinis, "Rss-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2037–2050, 2015.

[10] T. Stoyanova, F. Kerasiotis, C. Antonopoulos, and G. Papadopoulos, "Rss-based localization for wireless sensor networks in practice," in *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2014 9th International Symposium on*. IEEE, 2014, pp. 134–139.

[11] R. Zhang, W. Xia, Z. Jia, L. Shen, and J. Guo, "The optimal placement method of anchor nodes toward rss-based localization systems," in *Wireless Communications and Signal Processing (WCSP), 2014 Sixth International Conference on*. IEEE, 2014, pp. 1–6.

[12] W. Suwansantisuk and H. Lu, "Localization in the unknown environments and the principle of anchor placement," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2488–2494.

[13] K. Rasmussen, M. Srivastava *et al.*, "Secure location verification with hidden and mobile base stations," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 470–483, 2008.

[14] D. Koutsonikolas, S. M. Das, and Y. C. Hu, "Path planning of mobile landmarks for localization in wireless sensor networks," *Computer Communications*, vol. 30, no. 13, pp. 2577–2592, 2007.

[15] J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz, "Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 9, pp. 3052–3064, 2014.

[16] J. Jiang, G. Han, H. Xu, L. Shu, and M. Guizani, "Lmat: Localization with a mobile anchor node based on trilateration in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[17] R. Sumathi and R. Srinivasan, "Rss-based location estimation in mobility assisted wireless sensor networks," in *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, vol. 2. IEEE, 2011, pp. 848–852.

[18] X. Zhang, Z. Duan, L. Tao, and D. K. Sung, "Localization algorithms based on a mobile anchor in wireless sensor networks," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*. IEEE, 2014, pp. 1–6.

[19] S. K. Rout, A. Mehta, A. R. Swain, A. K. Rath, and M. R. Lenka, "Algorithm aspects of dynamic coordination of beacons in localization of wireless sensor networks," in *Computer Graphics, Vision and Information Security (CGVIS), 2015 IEEE International Conference on*. IEEE, 2015, pp. 157–162.

[20] Z. Gong, C. Li, F. Jiang, R. Su, R. Venkatesan, C. Meng, S. Han, Y. Zhang, S. Liu, and K. Hao, "Design, analysis, and field testing of an innovative drone-assisted zero-configuration localization framework for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 322–10 335, 2017.

[21] P. Perazzo, F. B. Sorbelli, M. Conti, G. Dini, and C. M. Pinotti, "Drone path planning for secure positioning and secure position verification," *IEEE Transactions on Mobile Computing*, no. 1, pp. 1–1, 2017.

[22] S. Capkun and J.-P. Hubaux, "Secure positioning in wireless networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, no. LCA-ARTICLE-2007-002, pp. 221–232, 2006.

[23] C. M. Pinotti, F. Betti Sorbelli, P. Perazzo, and G. Dini, "Localization with guaranteed bound on the position error using a drone," in *Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access*. ACM, 2016, pp. 147–154.

[24] F. B. Sorbelli, S. K. Das, C. M. Pinotti, and S. Silvestri, "Precise localization in sparse sensor networks using a drone with directional antennas," in *Proceedings of the 19th International Conference on Distributed Computing and Networking*. ACM, 2018, p. 34.

[25] ——, "Range based algorithms for precise localization of terrestrial objects using a drone," *Pervasive and Mobile Computing*, 2018.

[26] J. Liang and Q. Liang, "Rf emitter location using a network of small unmanned aerial vehicles (suavs)," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[27] F. B. Sorbelli, S. K. Das, C. M. Pinotti, and S. Silvestri, "On the accuracy of localizing terrestrial objects using drones," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[28] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Ultra reliable uav communication using altitude and cooperation diversity," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 330–344, 2018.

[29] H. Sallouha, M. M. Azari, A. Chiumento, and S. Pollin, "Aerial anchors positioning for reliable rss-based outdoor localization in urban environments," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 376–379, 2018.

[30] H. Sallouha, M. M. Azari, and S. Pollin, "Energy-constrained uav trajectory design for ground node localization," *arXiv preprint arXiv:1806.02055*, 2018.

[31] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Trans. Wireless Commun*, vol. 16, no. 6, pp. 3747–3760, 2017.

[32] J. Sundqvist, J. Ekskog, B. J. Dil, F. Gustafsson, J. Tordenlid, and M. Petterstedt, "Feasibility study on smartphone localization using mobile anchors in search and rescue operations," in *Information*

Fusion (FUSION), 2016 19th International Conference on. IEEE, 2016, pp. 1448–1453.

[33] Z. Liu, Y. Chen, B. Liu, C. Cao, and X. Fu, "Hawk: An unmanned mini-helicopter-based aerial wireless kit for localization," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 287–298, 2014.

[34] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, 2019.

[35] F. Koohifar, I. Guvenc, and M. Sichitiu, "Autonomous tracking of intermittent rf source using a uav swarm," *IEEE Access*, vol. 6, pp. 15 884 – 15 897, 2018.

[36] V. Acuna, A. Kumbhar, E. Vattapparamban, F. Rajabli, and I. Guvenc, "Localization of wifi devices using probe requests captured at unmanned aerial vehicles," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE, 2017, pp. 1–6.

[37] S. Wu, "Illegal radio station localization with uav-based q-learning," *China Communications*, vol. 15, no. 12, pp. 122–131, 2018.

[38] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[39] L. Sankar, "Steady, level forward flight," accessed on Feb. 2019. [Online]. Available: www.wpri.info/wpcontent/uploads/2013/08/Part2.ppt

[40] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[41] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[42] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 4. IEEE, 2002, pp. 3404–3410.