Emitter Localization using Q-Learning Optimized Sensor Positioning

Parker Brown
Computer Science
Southern Methodist University
Dallas, USA
pbrown@smu.edu

Suma Chackola
Electrical and Computer Engineering
Southern Methodist University
Dallas, USA
schackola@mail.smu.edu

Christopher Peters
Electrical and Computer Engineering
Southern Methodist University
Dallas, USA
peterscl@mail.smu.edu

Oliver Raney
Computer Science
Southern Methodist University
Dallas, USA
oraney@smu.edu

***Abstract*— This study presents a novel approach to enhancing emitter localization accuracy using a cooperative unmanned aerial system (UAS) equipped with a sensor array. By integrating Q-learning with dynamic adjustments to the array geometry, the system effectively optimizes sensor positions in real-time to converge more rapidly on the true location of a sensed emitter. Employing time difference of arrival (TDOA) measurements, the Q-learning algorithm is tailored to process iterative geometry updates, thereby minimizing localization errors in complex environments, such as urban areas where signal scattering is prevalent. Simulation results confirm the efficacy of our approach, demonstrating significant improvements in localization accuracy with fewer measurements compared to traditional methods. The potential applications of this technology are broad, with significant implications for search and rescue operations and other critical tasks in inaccessible or hazardous environments.**

Keywords*— *Q-learning optimization, emitter localization, time difference of arrival, LOCA

I. MOTIVATION

In recent years, unmanned autonomous vehicles and systems (UAVs, UAS) have become more abundant and their missions have drastically increased in scope. These systems can be small and versatile, allowing the ability to reach locations that manned systems are unable to access, such as hostile locations, underground tunnels, and confined spaces within a building. Emitter-locating UAS have been studied [1] and developed for applications such as search and rescue missions, where it is required to obtain the geolocation estimate of an emergency transponder beacon whose location was previously unknown. Because of scenarios like this, it is highly desirable to obtain an estimate of a sensed emitter's location that is accurate and requires as few measurements and computations as possible. This is particularly true when the emitter of interest is transmitting signals of very short time duration.

It is well-known [2] that an array's collective Field of View (FOV) with respect to a transmitter location greatly affects emitter location estimates, particularly for time-based emitter location algorithms that are sensitive to the relative geometry of the sensor array and its orientation to the emitter. Therefore, the agility and adaptability of a UAS network configured as a cooperative wireless sensor array is desirable, since it can be enabled to dynamically reconfigure its array geometry such that iterative measurements cause the emitter location estimate to converge to the true emitter location more quickly and with fewer measurements. Further, in an environment such as a dense urban area, buildings and other vehicles become scatterers to RF signals and lead to measurement errors, and a UAS has versatility over stationary sensors because they have the ability to maneuver to avoid physical structures that block signals and lead to measurement errors.

We are motivated to utilize a cooperative unmanned aerial system that operates as a sensor array to locate an emitter with an unknown location. For this system, we assume that the sensors employ omnidirectional antennas, are wirelessly connected with a wireless ad hoc network (WANET), they have the ability to self-localize to a high level of accuracy using an on-board LIDAR system, they have a highly accurate on-board timing source and reference that is synchronized prior to the mission, and we assume the system can accurately correlate signals that are received. With these capabilities, the use of a time difference of arrival (TDOA) measurement of the emitter signal at each sensor can be implemented to localize the emitter. A similar system was studied in [1], where the hardware limitations were analyzed to find a lower accuracy bound.

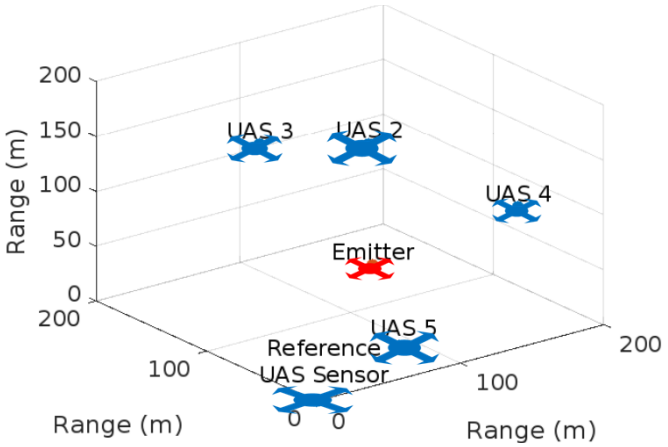

Fig. 1. Scenario utilizing a collection of unmanned aerial systems to locate an emitter.

To leverage the maneuverability of this system, we analyze in this work how the change in array geometry can lead to improved accuracy in emitter localization. The use of reinforcement learning in this scenario can evaluate iterative geometry updates and localization error and reward states that lead to improved accuracy, thus determining an optimal geometry for the system. Reinforcement approaches have been applied to robotic path planning [3] and specifically Q-learning has been used for UAS in emitter location scenarios in single agent [4] and multi-agent [5] approaches. In many methods, the system relies on increasing received signal strength (RSS) from the emitter at each sensor to quantify the reward in geometry state updates. We hypothesize that the utilization of a TDOA localization algorithm in a multi-agent Q-learning provides a reward metric that enables the cooperative use of sensor array and leads to a geometry that provides a highly accurate emitter localization. To this end, we will show the comparison in reinforcement learning convergence using both RSS and TDOA localization.

II. Related Work

Previous approaches have considered Reinforcement Learning, specifically Q-Learning, to search for the highest received signal strength (RSS) and reward states that move the UAS closer to the highest RSS.

A. Baseline

In [4] the approach taken was to use directional antennas used in conjunction with directional-aware Q-learning for localization. This was compared with two baseline tests. The first baseline test used an omnidirectional antenna and Q-learning was used to measure the RSS values.

The RSS is found by the Friis free space equation:

$$RSS_i(dBm) = -20 \times \log_{10}\left(\frac{4\pi R_i f}{c}\right) + P_e \tag{1}$$

where R_i is the distance from the emitter to the sensor in m, f is the emitter operational frequency in Hz, which is 5 GHz, c is the signal propagation speed, which is $3\text{x}10^8$ m/s in a vacuum, and P_e is the emitter output power in dBm.

The second baseline used a directional antenna on a UAV, with no reference to Q-learning, where the device moved in the direction that received the highest RSS value. The modified Q-learning approach with a directional antenna outperformed Q-learning with an omnidirectional antenna and the maximum RSS method with a directional antenna.

B. Multi-Agent Q-Learning

By utilizing a multi-agent Q-learning (reinforcement learning) approach as demonstrated in [5], we can study the effect of rewarding the UAS within shared state and action spaces. In a MQL (Multi-Agent Q-learning) algorithm, "each UAV takes account of other UAVs' flight decisions with the objective of promoting cooperative actions in order to achieve the highest reward". MQL algorithms behave similarly to SQL (Single-Agent Q-learning) algorithms in that each agent iteratively interacts with the environment to determine the optimal policy for achieving the maximal long-term reward. Using "exchanged rewards" broadcasted by each agent during specific actions, the MQL value function is updated to reflect the overall success of the system.

C. Location on a Conic Axis (LOCA) Algorithm

Because we intend to utilize a localization approach in addition to RSS, we consider an algorithm developed by Ralph Schmidt [6] called "Location on a Conic Axis" (LOCA). This algorithm differentiates itself from the geometry of traditional hyperbolic ranging approaches in that it considers the sensors to be on the perimeter of a conic and the emitter to be at the focus of the conic, which is contained on the conic axis. In a two dimensional geometry, three sensors are needed to find the focus. If more sensors are used, the algorithm can be used with any combination of a subset of three sensors and leads to many emitter location estimates (for example, using 5 sensors leads to $\binom{5}{3} = 10$ results). With this approach, since the algorithm finds the emitter on the straight-line conic axis and with conics there may be two foci, to resolve the focus ambiguity each resulting conic axis can be analyzed together as a system of intersecting linear equations so that their intersection is the focus. The advantage of this method provides a computationally efficient approach to TDOA processing.

The LOCA algorithm uses the difference in estimated range to the target for a pair of sensors,

$$\Delta R_{\text{ij}} = \hat{R}_{\text{j}} - \hat{R}_{\text{i}} \tag{2}$$

and implements the range difference from a triad of sensors (i, j, k):

$$\Sigma = \Delta R_{\text{ij}} + \Delta R_{\text{jk}} + \Delta R_{\text{ki}}. \tag{3}$$

Using a_i to represent the absolute range to the origin of the coordinate system:

$$a_i = \sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2}, \tag{4}$$

the algorithm implements these range differences in a set of linear equations to find the conic axis:

$$\begin{gathered}[x_1\Delta R_{23} + x_2\Delta R_{21} + x_3(\Delta R_{12} - \Sigma)]x_e + \\ [y_1\Delta R_{23} + y_2\Delta R_{21} + y_3(\Delta R_{12} - \Sigma)]y_e + \\ [z_1\Delta R_{23} + z_2\Delta R_{21} + z_3(\Delta R_{12} - \Sigma)]z_e = \\ \frac{1}{2}[\Delta R_{12}\Delta R_{23}\Delta R_{31} + {a_1}^2\Delta R_{23} + {a_2}^2\Delta R_{31} + {a_3}^2(\Delta R_{12} - \Sigma)]\end{gathered} \tag{5}$$

For three dimensions, (5) is in the form of the plane equation (6), and the emitter location $[\text{x}_e, y_e, z_e]$ can be found with linear algebra techniques:

$$\begin{bmatrix} A_{123} & B_{123} & C_{123} \\ A_{124} & B_{124} & C_{124} \\ \vdots & \vdots & \vdots \\ A_{ijk} & B_{ijk} & C_{ijk} \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \approx \begin{bmatrix} D_{123} \\ D_{124} \\ \vdots \\ D_{ijk} \\ \vdots \end{bmatrix}. \tag{6}$$

In [1], a similar system was studied with the LOCA algorithm that utilized on-board sensor positioning and timing references. There, the UAS sensor positioning accuracy was bounded to 2 cm, and the use of commercial chip scale atomic clocks bounded the timing reference to as low as 1 ns. However,

while feasible with this hardware, practically 1 ns is very difficult to achieve, therefore we analyze a 10 ns timing accuracy.

D. Error Analysis

In [7], the system in [1] was further studied to quantify the impacts of environmental errors to emitter localization. In that study, the use of urban propagation models such as ITU-R 1411-12 [8] were utilized to formulate stochastic models of a dense urban environment. It was found that for this system, with eight or more sensors, an urban environment has similar error as that of a system with 10 ns local timing inaccuracy. We utilize the results of these studies to generalize the error contributions to our system.

For RSS, an additional environmental loss term, $\varepsilon_P = \mathcal{N}(3,5)$, based on a generalization of [8] is included in (1) to incorporate multipath and fading effects. Therefore the estimated RSS is generalized to

$$\hat{P}_{r,i} = RSS_i + \text{abs}(\varepsilon_P). \tag{7}$$

Each LOCA estimate incorporates error contributions from studies in [1] and [7], where we apply an individual UAS position error into (3) based on the 2 cm accuracy of the UAS on-board self-positioning systems, $\varepsilon_U = \mathcal{N}(0,0.02)$, as:

$$\widehat{U}_i = \left[U_{i,x} + \varepsilon_U, U_{i,y} + \varepsilon_U\right] \tag{8}$$

where $U_{i,x}$, $U_{i,y}$ represent the coordinates of the UAS.

The timing error $\varepsilon_{t,i}$ at each i UAS is incorporated similarly into (2) as

$$\hat{R}_i = c\left(t_i + \varepsilon_{t,i}\right) \tag{9}$$

where t_i is the time-of-arrival of the emitter signal at the UAS. $\varepsilon_{t,i}$ is distance-dependent with variance σ_t, incorporates the sensor local clock inaccuracy of 10 ns, and incorporates signal delay effects due to a dense multipath environment, $\varepsilon_{m,i}$. The entire effect of is found by:

$$\varepsilon_{t,i} = \mathcal{N}\left(0, \sigma_{t,i} + 10\ ns\right) + \varepsilon_{m,i} \tag{10}$$

$\varepsilon_{m,i}$ is found using constants $(C_\alpha, \gamma_\alpha, C_\sigma, \gamma_\sigma)$ from Table 12 of [8] representing the delay spreading effects in dense multipath environments, and can be considered normally distributed, $\varepsilon_{m,i} = \mathcal{N}\left(C_\alpha R_i^{\gamma_\alpha}, C_\sigma R_i^{\gamma_\sigma}\right)$. Because these values are height-dependent and stochastically-developed, we only consider a 50% error contribution for the multipath effects. $\sigma_{t,i}$ is dependent on range on additional sensor characteristics, such as the sensor minimum signal to noise ratio (SNR_0), the sensor bandwidth (B), sensor integration time (T_s), and the minimum operational range for the sensor (r_0), as found in [1]:

$$\sigma_{t,i} = \frac{3R_i^2}{4\pi^2 f T_s B^2 r_0^2 SNR_0}. \tag{11}$$

III. Methods

The following methods aim to refine the localization and tracking capabilities of the UAS collection by iteratively improving its accuracy and efficiency in determining the position of a target emitter. Each approach uses Q-learning to optimize the localization performance

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s,a)\right] \tag{12}$$

where s is the current state, a is the action taken, r is the reward received, s' is the new state after the action a is taken, α is the learning rate (how much new information overrides old information), γ is the discount factor (importance of future rewards). Each element of the UAS is considered a Q-learning agent and maintains its own Q-table, thus we are using multi-agent Q-learning. The actions are defined as directional movements, and the states are defined by the localization outcome for the specific action taken. The approach also considers an epsilon-greedy strategy, which defines a probability of choosing a random action and promotes exploration of the state space. The hyperparameters for this approach were tuned to the following: $\alpha = 0.1$, to maintain history as the space is explored, $\gamma = 0.9$, indicating that the future rewards are important but less important than immediate rewards, and $\epsilon = 0.1$, so that the system maintains an ability to explore the environment. This hyperparameter strategy allows the system to adapt its approach in the presence of noise, such that an excess error contribution in a single step would not bias the learning process.

We compare each method according to the rewards given to UAS. The success of each approach is measured by the decreasing radius of the circular error probable (CEP) [9] and the reduction in the Euclidean distance between the center of the CEP, or centroid, and the true target position. We determine the centroid by accumulating all emitter location estimates found using LOCA and determining the median (x_m, y_m) of those estimates. The absolute deviation from each estimate to the centroid is determined in each coordinate $(\text{mad}_x, \text{mad}_y)$ and the CEP radius is found as

$$CEP(m) = \sqrt{\text{mad}_x^2 + \text{mad}_y^2} \tag{13}$$

The median error is found by the Euclidean distance from the centroid to the true target location:

$$error_{med} = \sqrt{(x_m - x_e)^2 - (y_m - y_e)^2}. \tag{14}$$

The result of (14) is only captured as a figure of merit for the performance of the algorithm, and it is not used in the actual Q-Learning process so that we can evaluate the performance of a system with no knowledge of the true location of the target.

The system iterates until it reaches a predefined number of steps, a time limit, or the emitter is within a specified CEP, cep_0, indicating successful localization. We set $cep_0 = 2$ meters and keep this metric the same across all methods to maintain a fair comparison of success.

In each method, the system can make nine directional movements – up, down, left, right, diagonally, and zero movement – to hone in on the signal source. As CEP evolves in the Q-learning process, we use its value as an additional dynamic hyperparameter to determine a step size of how far the UAS should travel for the current step:

$$ss = \frac{CEP - cep_0}{cep_0} \tag{15}$$

The structure of the Q-learning algorithm is as follows:

---

1. Initialize $Q_n(s_n, a_n) \coloneqq 0$, set $flag\ done = false$, $target = (x_e, y_e)$, $next\_move$ is 9-directional with initial step size $ss = 0$, random initial drone location U_i
2. Determine initial CEP radius (m), using (5), (13)
3. Determine state, s, as a function of U_i
4. While $done = false$:
   a. Determine next action, a', by random selection: If exploration: $a' = random()$, else $a' = \max_a Q(s, a) \bmod num\_directions$
   b. Take action a':
      i. Increment $step$ by 1
      ii. Update ss using (15)
      iii. Update $U_i = U_i + next\_move\ (ss * a')$
      iv. Update distance from target, and apply error using (8)
      v. Update time of arrival and apply error using (9)-(11)
      vi. Calculate next state, s', using LOCA (5):
         I. Calculate centroid, (x_m, y_m), and deviation from centroid (mad_x, mad_y)
         II. Update CEP using (13)
         III. Determine median error using (14)
      vii. Update power received and error, RSS_i, using (1), (7)
      viii. Update reward, r, using applicable method. If the condition is met, then $r = r + 1$ else $r = r - 1$
      ix. Check for $done$: if $step \geq max$ or $CEP \leq cep_0$ then $done = true$
   c. Update $Q(s, a)$ with s, a, r, s':
      i. Update s' as a function of (7)
      ii. Update $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max'_a Q(s', a') - Q(s, a)]$ using (12)

A. Iteration 1: RSS Maximization

In the first method, the system maximizes the reward by increasing RSS for each sensor, similar to the approach in [4]. The goal in this method is to form a geometry where the UAS geometry converges on the emitter. For this reward structure, we determine step 4.b.viii as follows: if the individual UAS received signal strength $RSS_i > RSS_i^*$ then $r = r + 1$ else $r = r - 1$, where RSS_i^* is the RSS_i from the previous step.

B. Iteration 2: Distance Minimization

The second method shifts focus to state updates informed by the LOCA algorithm, where the UAS is rewarded for moving towards the estimated position of the emitter. The goal in this approach, similar to the first approach, is to form a sensor geometry that converges on the emitter. The estimated position is found by computing the centroid of the collection of estimates of all combinations of sensors using the LOCA algorithm. For this reward structure, we determine step 4.b.viii by calculating the Euclidean distance of each U_i to the centroid (x_m, y_m):

$$dist_i = \sqrt{\left(U_{x_i} - x_m\right)^2 - \left(U_{y_i} - y_m\right)^2} \tag{16}$$

Then, if the individual UAS distance $dist_i < dist_i^*$ then $r = r + 1$ else $r = r - 1$, where $dist_i^*$ is the $dist_i$ from the previous step.

C. Iteration 3: LOCA Variance Minimization

In the third method, the strategy considers minimizing CEP in each state update by changing the geometry of the individual UAS without considering a convergence on the target. Each step does not require the UAS to necessarily move closer to the emitter, instead the only requirement is that CEP is reduced, so that the geometry is optimized for cep_0. In this way, in each state update, the LOCA estimates (5) are determined for all sensor combinations and are associated to each contributing sensor. The deviation, dev_l, from the centroid is determined for each estimate l and stored in a lxN matrix for sensor U_i of N sensors:

$$dev_{l,i} = \sqrt{\left(x_{l,i} - x_m\right)^2 - \left(y_{l,i} - y_m\right)^2}. \tag{17}$$

Because every sensor is not used for each LOCA estimate, the non-contributing sensors receive a deviation of 0. dev_l is summed for each sensor, $Dev_i = \sum_l dev_{l,i}$, and the sensor associated with the largest total deviation from the centroid is then penalized in step 4.b.viii of the algorithm, such that for $\max Dev_i$ then $r = r - 1$ else $r = r + 1$. Last, because this method is not biased to a specific geometry or distance to the emitter, we impose an additional penalty on any movements outside a specified maximum distance from the centroid to ensure the receiver sensitivity of each UAS is not violated. Applying this boundary with this method allows the maximum utilization of all sensors in the UAS while allowing the UAS to explore a geometry and not necessarily converge on the emitter, provided that the CEP is minimized with geometry updates.

D. Iteration 4: RSS Maximization + LOCA variance minimization

In this last method, we combine elements from the first and third methods, where the geometry is optimized for cep_0 but is informed by maximizing RSS_i on each step. The reward for this approach is weighted equally for increasing RSS_i and decreasing CEP.

IV. Results and Conclusion

The Q-Learning methods were performed for a simulated environment and each saw 10,000 Monte Carlo runs where the initial positions of the UAS were randomized in a 4000m x 4000m square space. Each UAS utilizes an omnidirectional antenna, and contains an self-positioning system with accuracy of 2 cm and a reference clock with accuracy 10 ns. The emitter is stationary and broadcasting a 5 GHz signal in a simulated urban environment, where the scatterers followed a Rayleigh distribution according to [8]. In all methods, we bound the learning to ≤ 1000 steps per Monte Carlo or $cep_0 = 2$ m. In methods 3 & 4, we set a maximum distance of 1500 m and penalize the UAS for an attempt to move outside that distance.

In Fig. 2 through Fig. 5, we visualize a single run of the Monte Carlo simulations, showing trajectories of the UAS system towards and around the target during the Q-Learning process and the number of steps for convergence in the single run. We also show the progression of the centroid with the various UAS geometries to demonstrate the varying nature of the estimates with the respective geometries. In all cases, the final centroid converged with the true target location.

In Table I. , we summarize the pertinent metrics averaged from the 10,000 Monte Carlo runs of each Q-learning method as follows: average number of steps to achieve cep_0 of 2 m,

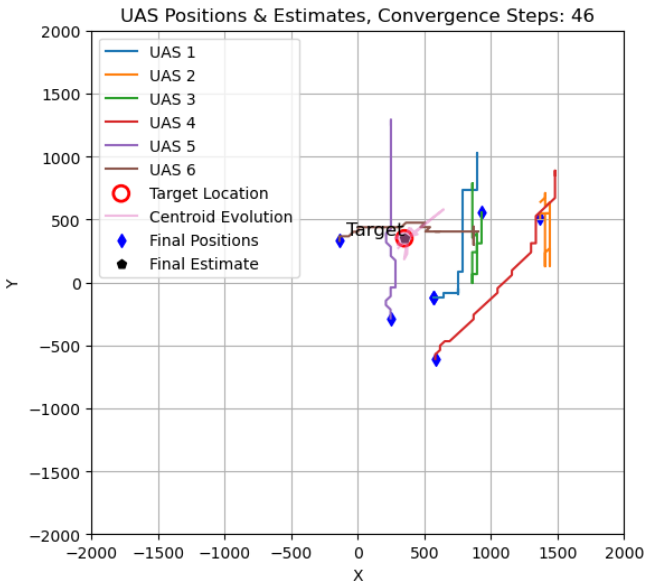

Fig. 2. Example of UAS trajectories and centroid estimate evolution during Q-Learning for Iteration 1: RSS Maximization.

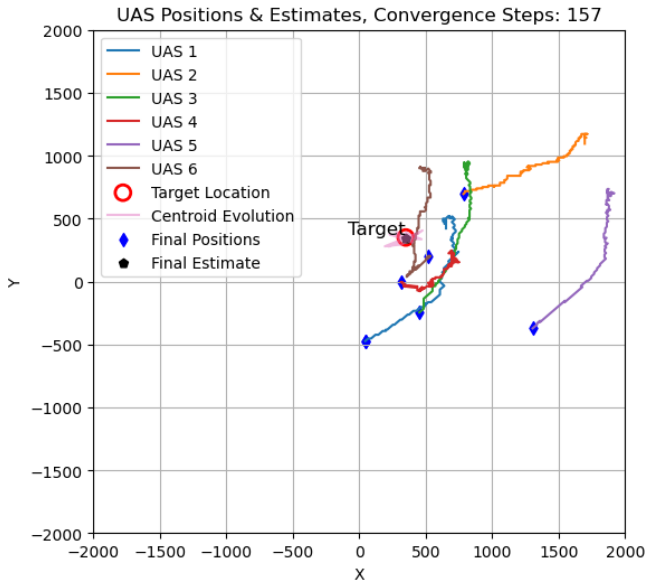

Fig. 3. Example of UAS trajectories and centroid estimate evolution during Q-Learning for Iteration 2: Distance Minimization.

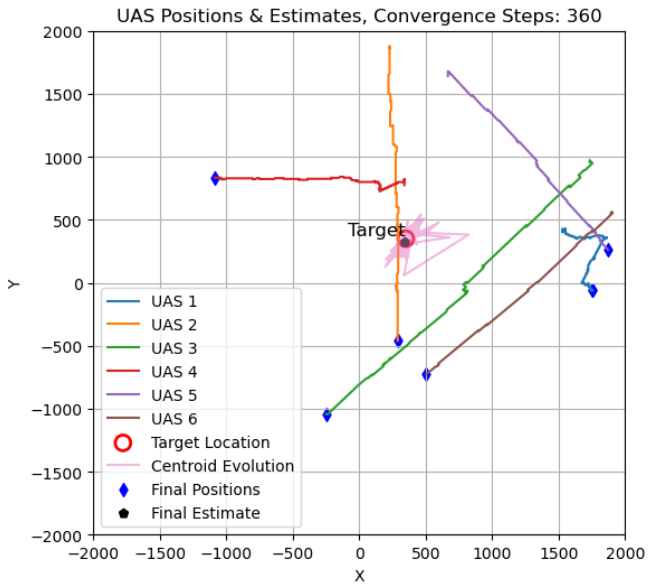

Fig. 4. Example of UAS trajectories and centroid estimate evolution during Q-Learning for Iteration 3: LOCA variance minimization.

TABLE I. Summary of Pertinent Metrics

Method	Average Number of Steps	Average Median Error (m)	Average Final CEP Radius (m)	Average Total Distance Moved (m)	Average Total Reward
RSS Maximization	118	12.7	3.2	10008	123
Distance Minimization	165	17.7	6.5	12100	-2
LOCA Variance Minimization	220	19.9	6.7	8490	183
RSS Maximization + LOCA Variance MInimization	***129***	***13.7***	***2.4***	***7741***	***113***

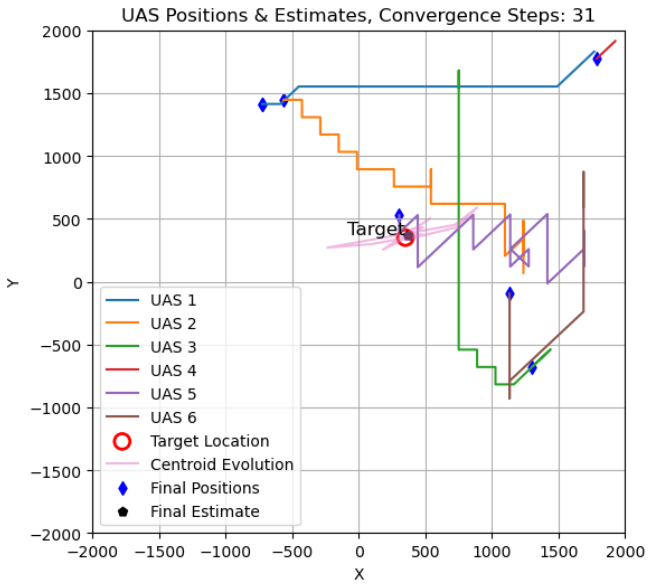

Fig. 5. Example of UAS trajectories and centroid estimate evolution during Q-Learning for Iteration 4: balanced RSS Maximization & LOCA variance minimization.

average error found by (14), average CEP (13), average total distance moved by the system (the sum of the distance moved by all individual UAS), and average total reward for each run. The consideration of the total distance moved by the system incorporates an acknowledgement of a mission constraint, such that the UAS system moves the shortest path to converge on the emitter. We conclude from these results that the approach in Iteration 4, where the reward is gained in equal parts from the minimum variance and the maximum RSS, provides the fastest convergence and a consistently minimized CEP. We visualize the distribution of these parameters from the 10,000 Monte Carlo runs fitted to normal distributions in Fig. 6.

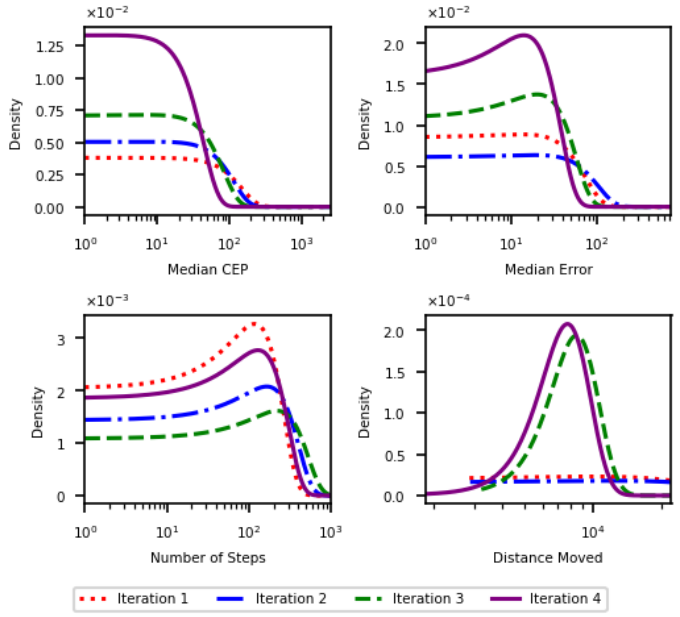

Fig. 6. Density of pertinent metrics approximated to normal distributions.

While the RSS Maximization method alone shows a lower number of convergence steps, Iteration 4 provides the highest density of optimized metrics for median CEP, median error, and distance moved, providing substantial evidence that the combination of maximizing the RSS (Iteration 1) while optimizing a geometry that minimizes the variance in the localization (Iteration 3) is a superior method to geometric convergence (Iterations 1 & 2) and to geometric optimization (Iteration 3).

In Fig. 7, we visualize the average final confidence region from the Monte Carlo runs as a circle centered at the median of the LOCA estimates (centroid) with a radius equal to the CEP. We show how Iteration 4, whose CEP has the smallest radius, also benefits from a lower error from the centroid to the true target position. In Fig. 8, we visualize the collection of all final UAS positions from the 10,000 Monte Carlo runs for Iteration 3, showing that the optimized position for the UAS collection is roughly circular around the target.

Our results demonstrate that this cooperative UAS system for emitter localization achieves the most efficient and optimum performance when the sensors utilize both the emitter RSS and the signal time-of-arrival, such that the combination of these two available information sources achieves a desired CEP with minimal error.

In a future study, we will consider expansion of our simulation parameters to consider different sensor operating frequency and bandwidth, varying sensor quantities, and expand our geometry to three dimensions. We will also consider refining our algorithm to include continuous directional decisions and more sophisticated Q-learning techniques to include pathfinding with obstacle avoidance.

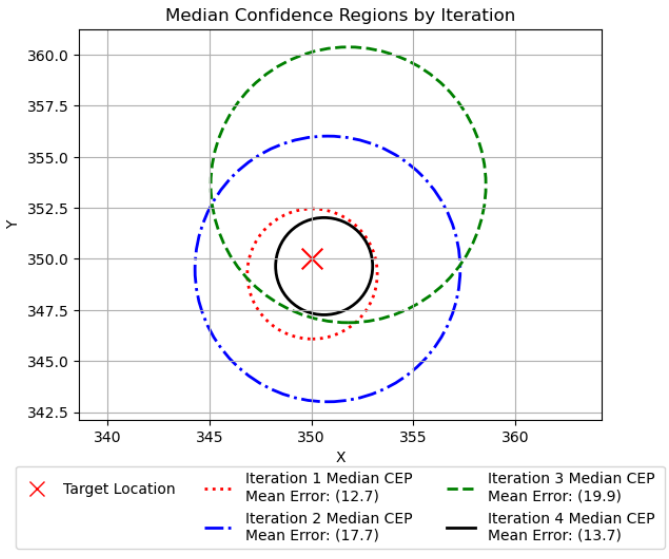

Fig. 7. Median confidence regions defined by circular error probable and mean Euclidean error for each Q-Learning iteration compared to true target location.

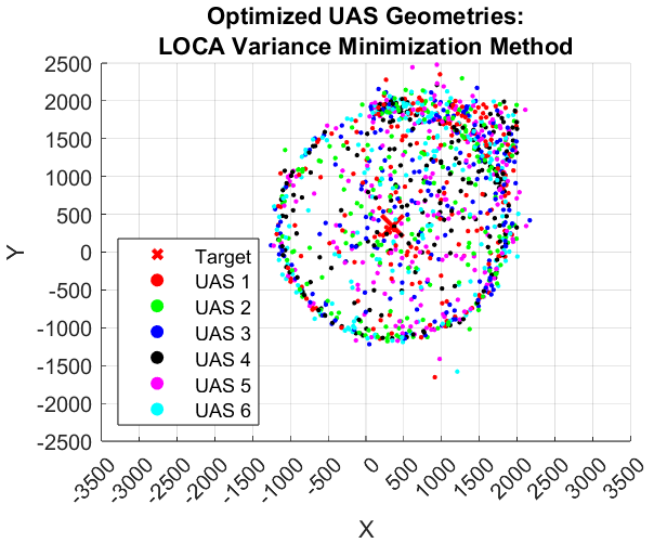

Fig. 8. Optimized UAS geometries compiled for all Monte Carlo runs for the LOCA variance minimization method.

V. Acknowledgment

We would like to express our gratitude to Dr. Eric Larson for providing invaluable guidance and feedback throughout this project. We also appreciate the support and resources offered by the Lyle School of Engineering at Southern Methodist University, which were essential for our research.

References

[1] C. Peters and M. A. Thornton, "Cooperative UAS Geolocation of Emitters with Multi-Sensor-Bounded Timing and Localization Error," 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 2023, pp. 1-13, doi: 10.1109/AERO55745.2023.10116023.

[2] C. Yan, L. Fu, J. Zhang and J. Wang, "A Comprehensive Survey on UAV Communication Channel Modeling," in IEEE Access, vol. 7, pp. 107769-107792, 2019, doi: 10.1109/ACCESS.2019.2933173.

[3] J. Jyoti and R. S. Batth, "Unmanned Aerial vehicles (UAV) Path Planning Approaches," 2021 International Conference on Computing Sciences (ICCS), Phagwara, India, 2021, pp. 76-82, doi: 10.1109/ICCS54944.2021.00023.

[4] S. Wu, "Illegal radio station localization with UAV-based Q-learning," in China Communications, vol. 15, no. 12, pp. 122-131, Dec. 2018, doi: 10.12676/j.cc.2018.12.010.

[5] Y. J. Chen, D. K. Chang and C. Zhang, "Autonomous Tracking Using a Swarm of UAVs: A Constrained Multi-Agent Reinforcement Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 69, no. 11, pp. 13702-13717, Nov. 2020, doi: 10.1109/TVT.2020.3023733.

[6] R. O. Schmidt, "A New Approach to Geometry of Range Difference Location," in IEEE Transactions on Aerospace and Electronic Systems, vol. AES-8, no. 6, pp. 821-835, Nov. 1972, doi: 10.1109/TAES.1972.309614.

[7] C. Peters and M. A. Thornton, "Reducing RF Emitter Localization Error in Urban Environments with Geometry Adaptive UAS Arrays," 2024, submitted as publication.

[8] Radiocommunication Sector of International Telecommunication Union, "Propagation data and prediction methods for the planning of short-range outdoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz", Rec. ITU-R P. 1411–12 ITU Recommendation, Aug. 2023

[9] W. E. Hoover and U. S., “Algorithms for confidence circles and ellipses,” United States National Ocean Service Office of Charting and Geodetic Services, Tech. Rep., 1984, nOAA technical report NOS 107 C&GS 3. [Online] Available: https://repository.library.noaa.gov/view/noaa/23141.