COOPERATIVE UNMANNED AERIAL SYSTEM (UAS) GEOLOCATION OF EMITTERS

Approved by:

Mitchell A. Thornton
Electrical and Computer Engineering
Dissertation Committee Chairperson

Jennifer Dworak
Electrical and Computer Engineering

Eric Larson
Electrical and Computer Engineering

Himanshu Thapliyal
Electrical and Computer Engineering

Darrell Young
Electrical and Computer Engineering

COOPERATIVE UNMANNED AERIAL SYSTEM (UAS) GEOLOCATION OF EMITTERS

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Electrical Engineering

by

Christopher Peters

M.S., Systems Engineering, Johns Hopkins University
M.S., Physics, University of Mississippi
B.S., Physics, University of Mississippi
B.S., Mathematics, University of Mississippi

May 16, 2026

ACKNOWLEDGMENTS

This page is usually included but is not required. If included, it goes here ONLY IF YOU ALSO HAVE A DEDICATION PAGE. If you do not have a dedication page, the acknowledgment page comes directly before the text—as the last preliminary page.

Thanks.

Peters, Christopher M.S., Systems Engineering, Johns Hopkins University, 2019
M.S., Physics, University of Mississippi, 2009
B.S., Physics, University of Mississippi, 2007
B.S., Mathematics, University of Mississippi, 2007

Cooperative Unmanned Aerial System (UAS) Geolocation of Emitters

Advisor: Mitchell A. Thornton

Doctor of Philosophy conferred May 16, 2026

Dissertation completed February 25, 2026

A collection of unmanned aerial systems (UAS) can be networked as a cooperative wireless sensor array to geolocate an unknown-location RF emitter using time-based measurements. In operation, however, environmental multipath and hardware errors in sensor positioning and timing can degrade emitter localization accuracy and limit the practicality of single-snapshot solutions. This dissertation evaluates time-of-arrival and time-difference-of-arrival (TOA/TDOA) geolocation for cooperative UAS arrays under realistic error sources and develops geometry-control strategies that actively reduce localization uncertainty through iterative UAS repositioning.

This work studies the Location On a Conic Axis (LOCA) method for emitter localization. Using Monte Carlo simulations with hardware error models and environment-induced excess delay, including multipath statistics consistent with ITU-R P.1411, results show that LOCA remains close to the Cramér–Rao lower bound and benefits consistently from increased sensor redundancy. Across the evaluated mission volumes and error budgets, the most significant improvement occurs when increasing from five sensors, the minimum required for 3D localization, to six sensors, with diminishing returns observed beyond approximately eight sensors.

A primary contribution of this dissertation is the use of a subset-based LOCA solution cloud to provide dispersion-based convergence metrics without requiring knowledge of the true emitter location. These uncertainty metrics are then used to guide reinforcement-

learning-based geometry controllers that dynamically reposition the UAS array between localization snapshots to reduce uncertainty and drive the solution cloud toward a compact and consistent region. In comparative policy experiments, active geometry adaptation reduces localization uncertainty within a small number of repositioning steps and achieves substantial reductions in localization error relative to static sensing.

Finally, the learned geometry-control behavior is validated in an EXata digital-twin environment that includes Rician fading and adverse weather. The EXata results demonstrate stable convergence and faster refinement than the baseline simulation, supporting the feasibility of adaptive geometry control for cooperative UAS geolocation under more realistic propagation conditions. Collectively, these results show that for small cooperative UAS arrays operating in dense multipath environments, adaptive geometry becomes a primary mechanism for efficient convergence on an unknown emitter location.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This Dedication page is *optional*.

1. If you are not including a Dedication page, place the Acknowledgment page here. Else the Acknowledgment page should be placed before the Abstract section.

2. If the Acknowledgment page (generated using the `makeackpage` option of the LaTeX template) is placed here, then it will be automatically numbered and included in the Table of Contents.

3. If this page is used for Dedication (generated using the `makededicationpage` option of the LaTeX template), then the LaTeX template will not number this page and will not include it in the Table of Contents.

4. Read the comments in the `frontpages/preliminary.tex` file for more information.

Chapter 1
INTRODUCTION

1.1 Background

In modern sensing systems across multiple domains, networked "unmanned aerial systems" (UAS) are becoming more common [1] for autonomous missions. A collection of UAS with the capability to establish a "wireless ad hoc network" (WANET) results in a cooperative wireless sensor array with a dynamic geometry reconfiguration capability that can be deployed to measure the position of an RF emitter of interest [2,3]. As an example application, a search and rescue mission might require the ability to obtain the geolocation estimate of an emergency transponder beacon whose location was previously unknown [3].

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 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 [4,5]. This motivates the use of a cooperative wireless independently-mobile UAS sensor array since it can dynamically reconfigure its array geometry to improve its localization estimate [6].

The agility and adaptability of a UAS network configured as a cooperative wireless sensor array also offers additional desirable characteristics. When the array is comprised of relatively small UAS, search areas that are not possible to navigate with larger fixed-wing aircraft or satellite-based systems, such as confined spaces within a building, underground tunnels, caves, or beneath a foliage canopy, can be practically accessed by the UAS array. Also, the UAS-based sensor array can operate in hazardous areas such as hostile and radiation- or chemically-contaminated spaces that would otherwise not be accessible for a system requiring human operators aboard.

Although the UAS-based wireless sensor array provides significant advantages in comparison to fixed array systems that may require the presence of human operators, there are technical challenges in designing such a system. A primary motivating factor for choosing a time-based emitter location methodology is the challenge in implementing phase coherence among the array receivers [7]. When each of the multiple individual sensor apertures is coupled to a different RF receiver, the apertures are not typically phase coherent since synchronization of the receivers' local oscillators (LO) is difficult to achieve and maintain. The carrying capacity of each UAS can limit the overall allowed payload weight and may require the use of lightweight and less effective onboard assets. Space and weight limitations also restrict the power capacity of onboard batteries, thereby reducing the amount of time that the array can operate before a recharging cycle is required.

In view of these advantages and challenges, the implemented wireless sensor array emitter geolocation techniques should depend on simple and low-power computations while also completing as quickly as possible. Certain other methods of emitter geolocation, such as those that depend upon received signal phase differences or other coherency requirements, such as phase interferometric or cyclostationary approaches, are often impractical to implement. These observations motivate us to consider time-based emitter location approaches that are well-known to have sensitivity to the array geometry and can thus take advantage of the dynamic reconfiguration capability of the array.

"Time of Arrival" and "Time Difference of Arrival" (TOA/TDOA) emitter location methods depend upon solutions of analytical geometry equations in two- or three-dimensions that are defined by the relative positions of the sensor locations and the emitter of interest. In such approaches, emitter location accuracy can be very sensitive to the overall geometry of the array in relation to the true emitter location. The range of the emitter with respect to the sensors is obtained as the product of the timing measurements with the signal propagation speed, typically, the speed of light, c. Due to the large magnitude of electromagnetic signal propagation speed, large changes in the emitter location estimates can occur due to relatively small sensor timing errors or localization errors. For this reason, we are particularly concerned with the effect of these two error sources in a UAS array.

While the local UAS clock sources may be fairly stable and of high frequency through the use of small atomic clocks and accurate frequency multiplication circuitry, the challenge of clock signal synchronization among the array sensors can nevertheless induce timing inaccuracies. Clearly, high-frequency local clock sources are desired to provide as much resolution as possible, and low-drift and jitter are equally important to reduce emitter location estimates due to local timing source inaccuracies.

1.2 Problem Statement and Hypothesis

The use of a cooperative UAS array to find an unknown-location emitter is constrained both by local errors, which are those inherent in the UAS hardware, and by environmental errors like multipath that further distort the emitter signal information. This localization system must overcome these challenges to accurately determine the emitter location.

Our hypothesis is that if a cooperative UAS array utilizing time-based localization methods can intelligently adapt its geometry, then it can efficiently reduce localization error in complex multipath environments and quickly converge on an accurate localization solution for an unknown-location emitter.

Therefore, the primary contribution for this dissertation is a geometry controller that dynamically repositions the UAS array to improve localization performance. The methods include adaptive feedback to the array geometry based on measurement heuristics and machine-learning-based controllers, and this study evaluates the performance of the controller in Monte Carlo simulations and the EXata digital-twin environment.

1.3 System Architecture

In the cooperative UAS scenario described here, the UAS positions may be known via the use of the relatively accurate Global Positioning System (GPS) or other navigational systems; however, the UAS comprising the sensor array are individually hovering during the time that TOA/TDOA values are obtained and thus induce a localization error source we term as "hover drift." Other small sensor positioning errors due to variations in environmental factors, such as wind gusts, can cause further degradation in sensor localization. Furthermore, in some environments, GPS may be unavailable requiring an alternative sensor localization approach.

We are motivated to consider the UAS-based cooperative wireless array since the accuracy of emitter location estimates can be partially mitigated using emerging sensor and subsystem technologies that bound the error sources. Specifically, in terms of sensor localization error, we consider the case wherein each UAS incorporates an on-board LiDAR subsystem that permits fine-grained measurements of sensor positioning, such as that described in [8]. Likewise, in terms of local timing source errors, the emergence of small portable "Chip Scale Atomic Clocks" (CSAC) can provide a basis for a local timing source that is of high resolution and offers low drift and jitter rates [9], [10]. The assumed hardware architecture and representative subsystem bounds used in this study are described in Section 3.3.

The observations regarding sensor location and local clock source inaccuracies, combined with the use of emerging and recent technology advances, motivates us to analyze the sensitivity and resultant accuracy of the emitter location estimates assuming the presence of the aforementioned technology within the UAS sensor array. The representative architecture of the hardware studied in this work is shown in Fig. 1.1.

This dissertation focuses on the geolocation of a single, stationary RF emitter using this cooperative UAS array. Extensions to multiple simultaneous emitters or to moving targets are outside the research scope, however, they are appropriate for future studies. Similarly, inter-UAS collision avoidance and obstacle avoidance are assumed to be handled by the autonomy of the UAS system and can rely on the onboard LiDAR for continuous physical geometry feedback. Instead, the geometry controller operates at a higher level to provide navigational comamnds that maintain sufficient UAS separation under nominal operating conditions. Last, while the controller developed in this work is implemented in a quantitative simulation and in digital twin model of both the environment and the hardware, the physical implementation of the system is left as an opportunity for future development.

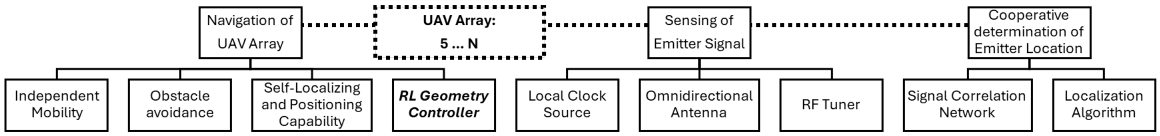

Figure 1.1: A representative functional block diagram of the UAS emitter localization system.

1.4 Summary of Following Chapters

In Chapter 2 [11], we summarize RF emitter geolocation approaches with an emphasis on TOA/TDOA-based methods. This background provides context for the algorithm choices in this study and motivates the use of time-based observables that are compatible with lightweight, non-phase-coherent UAS payloads.

In Chapter 3, we extend the Location On a Conic Axis (LOCA) algorithm to the use case of the cooperative UAS array comprised of at least five sensors and we describe the generation of LOCA solution clouds for three-dimensional localization. We define estimate refinement approaches based on the measurement uncertainty metrics of spherical and elliptical error probable (SEP and EEP). The UAS hardware implementation and subsystem performance bounds are summarized in Section 3.3 and are referenced later when defining error models and simulation parameters.

In Chapter 4, we analyze and quantify the impact of hardware and environmental error sources on time-based emitter geolocation approaches. First [11], we define local hardware error models and compute the Cramér–Rao lower bound (CRLB) for localization performance. Next, we study atmospheric loss and multipath effects and develop an accumulated error model to incorporate and quantify these effects in sensor measurements. Our simulations evaluate the accuracy of emitter location estimates versus environmental impacts and our results provide insight into the sensor quantities needed to mitigate localization error in complex multipath settings.

In Chapter 5, we develop and evaluate geometry-control strategies that dynamically reposition the UAS array to improve localization performance. We compare controllers that are based on measurement heuristics and utilize machine learning to study the cumulative impact of hardware and environmental errors and ways to mitigate these errors. We verify our performance with the EXata digital-twin environment, and we analyze the performance of our controller under various environmental constraints.

In Chapter 6, we provide a summary of the final architecture and evaluate its performance with a variety of test conditions.

In Chapter 7, we summarize the primary findings and discuss opportunities to extend the system into different use cases.

Chapter 2

BACKGROUND THEORY

2.1 Geolocation Approaches

This section summarizes RF emitter geolocation approaches and motivates the selection of the algorithm for the studies that follow. Geolocation techniques are described in this study with geometric interpretations under idealized conditions, including omnidirectional antennas, isotropic channels, and constant free-space propagation speed c. We consider non-ideal conditions, such as dense multipath and hardware-induced errors, in a later section.

The underlying geometric principle supporting time-based location is the spherical wavefront given by the constant speed of light c. In the ideal case, a time-of-arrival (TOA) measurement t_i is related to the position of the i^{th} sensor $s_i = (x_i, y_i, z_i)$ and emitter position $p = (x, y, z)$ as

$$t_i = \frac{1}{c}\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}. \tag{2.1}$$

A time-difference-of-arrival (TDOA) measurement τ_{ij} corresponding to two sensors s_i and s_j is the difference of their TOA measurements. TDOA is defined relative to a reference sensor, s_j as

$$\tau_{ij} = t_i - t_j. \tag{2.2}$$

2.1.1 Hyperbolic Ranging

A single TDOA measurement τ_{ij} constrains the emitter to a locus of constant range-difference between s_i and s_j, which forms a hyperbola in the two-dimensional (2D) plane containing the sensors and emitter and a hyperboloid in three dimensions (3D). Multiple distinct sensor pairs yield multiple hyperboloids, and in the ideal, noiseless case, the emitter location is found at their intersection. However, in practice, the cumulative errors in sensor clocks and uncertainties in sensor positions will prevent an exact intersection, therefore the location is obtained by minimizing the error over a set of TDOA measurements through weighted least-squares solutions [4].

2.1.2 Lateration-based Geolocation

Lateration-based techniques estimate a set of emitter ranges from multiple TOA measurements. Each range estimate corresponds geometrically to the radius of a circle (2D) or sphere (3D) centered at a sensor, and the emitter location is obtained by intersecting these circles or spheres. Because ranges are obtained from TOA, these techniques are commonly referred to as TOA, circular, or spherical approaches.

In the ideal case, two circles in 2D or two spheres in 3D generally intersect at zero, one, or two points; therefore, additional measurements are required to resolve ambiguities and improve performance quality. In 3D, at least four TOA constraints are commonly used to obtain an unambiguous solution in the presence of noise, and multilateration ($N > 3$) remains a widely studied topic for wireless localization [12–15].

Squaring both sides of (2.1), expanding, and rearranging yields

$$c^2t_i^2 - x_i^2 - y_i^2 - z_i^2 = x^2 + y^2 + z^2 - 2x_ix - 2y_iy - 2z_iz. \tag{2.3}$$

As shown in [16], the system for n sensors can be expressed in matrix form $\mathbf{Ax} = \mathbf{b}$:

$$\begin{pmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -2x_n & -2y_n & -2z_n \end{pmatrix} \begin{pmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c^2t_1^2 - x_1^2 - y_1^2 - z_1^2 \\ c^2t_2^2 - x_2^2 - y_2^2 - z_2^2 \\ \vdots \\ c^2t_n^2 - x_n^2 - y_n^2 - z_n^2 \end{pmatrix}. \tag{2.4}$$

A common least-squares estimate uses the inverse,

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}, \tag{2.5}$$

and utilizes additional constraints, such as $x_0 = x^2 + y^2 + z^2$, for ambiguity resolution.

TDOA-based Lateration Lateration can also be posed in terms of TDOA by selecting a reference sensor s_1 and forming $\tau_i = t_i - t_1$ for $i = 2, \ldots, n$ [16–18]. Using $t_i = t_1 + \tau_i$ and

$ct_1 = \sqrt{x^2 + y^2 + z^2}$, the linear equations can formed as:

$$\begin{pmatrix} 2c\tau_2 & -2(x_1 + x_2) & -2(y_1 + y_2) & -2(z_1 + z_2) \\ \vdots & \vdots & \vdots & \vdots \\ 2c\tau_n & -2(x_1 + x_n) & -2(y_1 + y_n) & -2(z_1 + z_n) \end{pmatrix} \begin{pmatrix} \sqrt{x^2 + y^2 + z^2} \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (c\tau_2)^2 - \|s_1\|^2 - \|s_2\|^2 \\ \vdots \\ (c\tau_n)^2 - \|s_1\|^2 - \|s_n\|^2 \end{pmatrix}, \tag{2.6}$$

where $\|s_i\|^2 = x_i^2 + y_i^2 + z_i^2$. This system is solved using the same classes of least-squares/recursive methods as (2.4)–(2.5).

2.1.3 Location On a Conic Axis

Schmidt [19], [20] developed an approach known as "Location On a Conic Axis" (LOCA) that uses the TDOA of sensor triads to find a straight line corresponding to a conic section axis. TDOA values are converted to range differences that define conic sections wherein the emitter is located at one of the foci and is known to be at a point on a line that is coincident with the conic major axis. In the case of LOCA, the sensor locations are points on the conic and the foci define the emitter location, which is the opposite scenario of hyperbolic geolocation. All conic sections may be considered as defining two foci. In the case of an ellipse or hyperbola, the two foci occupy different and distinct locations. In the case of a circle, the two foci may be considered as being co-located at the same point, and in the case of a parabola, one of the two foci may be considered as having a location that is infinitely distant from the other. Fig. 2.1 contains a diagram of the situation wherein the sensors lie along an ellipse with the emitter being positioned at one of the two distinct focal points.

If the sensor array comprises a total of three elements, the emitter location is computed as a conic section focal point. Since hyperbolas and ellipses have two foci, an ambiguous emitter location point results, and some means must be used for determining which foci corresponds to the actual emitter location estimate. When the sensor array comprises four or more sensors, the emitter location can be computed by considering the intersections of all pairs of lines that define the unique conic axes. Typically, a LSE approach is employed to estimate the emitter location based upon all points formed by intersecting conic axis lines.

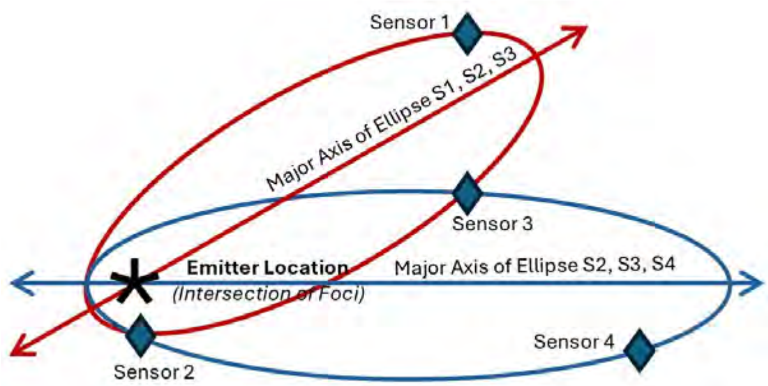

Figure 2.1: LOCA Positioning finds the Emitter at the focus of an ellipse formed by the range differences of the three Sensors to the Emitter.

The conic axis is derived using (2.1) in the form of the range of sensor s_i to the emitter,

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}. \tag{2.7}$$

We define the difference in two ranges of the emitter with respect to sensors s_i and s_j as Δr_{ij},

$$\Delta r_{ij} = r_j - r_i, \tag{2.8}$$

and the absolute range, a_i, of each sensor, s_i, with respect to the origin of the coordinate system origin as

$$a_i = \sqrt{x_i^2 + y_i^2 + z_i^2}. \tag{2.9}$$

We further define Σ as the sum of the range differences as

$$\Sigma_{ijk} = \Delta r_{ij} + \Delta r_{jk} + \Delta r_{ki}. \tag{2.10}$$

Using a triad of sensors, (s_1, s_2, s_3), the conic axis can be calculated using the range differences as expressed in (2.8).

In three-dimensions, the underlying geometry is that of quadrics instead of conics, and the resulting (2.11) is expressed in the form of the plane equation, $Ax + By + Cz = D$:

$$\begin{aligned}(x_1\Delta r_{23} + x_2\Delta r_{31} + x_3(\Delta r_{12} - \Sigma_{ijk}))x+\\ (y_1\Delta r_{23} + y_2\Delta r_{31} + y_3(\Delta r_{12} - \Sigma_{ijk}))y+\\ (z_1\Delta r_{23} + z_2\Delta r_{31} + z_3(\Delta r_{12} - \Sigma_{ijk}))z =\\ \frac{1}{2}(\Delta r_{12}\Delta r_{23}\Delta r_{31} + a_1^2\Delta r_{23} + a_2^2\Delta r_{31} + a_3^2(\Delta r_{12} - \Sigma_{ijk}))\end{aligned}. \tag{2.11}$$

2.1.4 Additional Geolocation Approaches

While timing-based methods are the primary focus of this dissertation, several other technique families exist. In this study, the justification for not using these following methods is primarily driven by the UAS payload and networking constraints summarized in Section 3.3. Accordingly, we emphasize timing-based location methods that are refined by aggregating results from cooperative measurements across a geometrically-diverse sensor array.

2.1.4.1 Angle of Arrival (AOA) / DOA Triangulation

Angle of Arrival (AOA) or Direction of Arrival (DOA) localization is commonly implemented as a geolocation technique where sensor groups estimate an emitter direction, via beamforming or methods such as MUSIC or ESPRIT [21, 22], followed by a fusion of these bearing measurements into an emitter location estimate [7, 23].

A sensor s_i produces a measured azimuth ϕ_i that is the true bearing from s_i to the emitter location p plus error [4]:

$$\phi_i = \arctan\frac{y - y_i}{x - x_i} + n_i. \tag{2.12}$$

Collecting all bearings into $\boldsymbol{\phi}$ and assuming approximately Gaussian bearing errors with covariance N, the emitter location estimate is typically determined through weighted least-squares approaches [4]:

$$\widehat{p} = \arg\min_p \big(\boldsymbol{\phi} - f(p)\big)^T N^{-1} \big(\boldsymbol{\phi} - f(p)\big), \tag{2.13}$$

where $f(p)$ is the vector of predicted bearings. This minimization is nonlinear and is usually solved iteratively [4]. For the 3D case, azimuth and elevation bearings are required.

As with other estimators, with many sensors a solution cloud can be generated in a single measurement snapshot by repeating this fusion over multiple sensor subsets. For this study, this approach alone is not suitable because each sensor is assumed to have a single omnidirectional antenna, versus a calibrated multi-element antenna array. The cooperative use of multiple antennas evolves this method into an interferometric approach. However, even with a multi-antenna architecture, bearing-based localization is highly sensitive to UAS attitude which could further bias DOA accuracy.

2.1.4.2 Frequency Difference of Arrival (FDOA)

Frequency Difference of Arrival (FDOA) measures differential Doppler shifts using the relative motion of sensors and can be used for localization using DOA triangulation techniques. For carrier frequency f_c and sensors i and j with velocities v_i and v_j, a Doppler measurement is expressed as

$$f_i - f_j = \frac{f_c}{c}(v_i \cos\phi_i - v_j \cos\phi_j), \tag{2.14}$$

with $\phi_{i,j}$ representing the bearings to from the emitter to sensors i and j. [4, 24]

FDOA requires a precise estimate of f_c, so it is most useful in locating narrowband emitters [4]. FDOA can improve performance in poor geometry, however, multipath introduces additional Doppler components and Doppler spread that impact the frequency shift. Further, performance depends on the stability of the oscillator as well as accurate knowledge of the sensor velocity. Practical FDOA requires disciplined frequency references across sensors to determine inter-UAS drift. FDOA is not considered for a primary geolocation approach in this study due to these factors.

2.1.4.3 Phase Interferometry

Phase interferometry is a DOA estimation technique that determines emitter bearing by measuring relative phase differences of the received signal measurements across an array of antennas. For baseline array vector b_{ij} from antenna i to j and unit direction vector $u(\theta, \phi)$, the phase differences are modeled as

$$\Delta\phi_{ij} \approx \frac{2\pi}{\lambda} b_{ij}^T u(\theta, \phi) + \phi_{0,ij} \pmod{2\pi}, \tag{2.15}$$

where $\phi_{0,ij}$ accounts for hardware phase offsets and calibration errors for each sensor. Accurate interferometry requires phase-coherent receiver channels and array calibration to reduce ambiguities arising from the 2π periodicity [7]. Dense multipath environments distort the effective wavefront and can bias the measured $\Delta\phi_{ij}$. In this study, the system does not implement phase–coherent sensors or attempt interferometric DOA estimation, but instead utilizes time synchronization across UAS elements to enable TDOA-based localization.

2.1.4.4 Power-Based Methods

Power-based emitter location approaches determine signal bearing from received signal strength (RSS) using a propagation model. A general formulation considers an emitter transmitting a power of P_0 at a distance d from the sensor:

$$RSS(d) \approx P_0 - 10\eta \log_{10}\left(\frac{d}{\delta_0}\right) + X_\sigma, \tag{2.16}$$

where η is the path-loss exponent and shadowing is modeled by X_σ for a reference distance δ_0 [25, 26]. Pattern-based methods consider direction-dependent antenna gains to refine RSS measurements. These methods do not require sensor synchronization, but in dense multipath, RSS is strongly impacted by shadowing and small-scale fading and is further affected by unknown emitter EIRP and attitude. Site-specific calibration is used to account for these impacts, yet because of the multipath the resulting location estimates can yield large, environment-dependent uncertainty. Therefore, RSS and pattern methods are not used as the primary geolocation approach in this study.

2.1.4.5 Hybrid Fusion, Sequential Tracking, and Centralized Direct Methods

Hybrid fusion combines several measurement approaches, for example TDOA, FDOA, and AOA, into a single estimator using maximum-likelihood or other similar methods (2.13) [4]. While hybrid fusion typically combines pre-extracted features like TDOA or AOA, centralized direct methods go further by jointly processing raw or near-raw signals.

When multiple time steps or sensor "snapshots" are available, sequential Bayesian filters propagate a state model and update it with geolocation measurements,

$$x_{k+1} = f(x_k, v_k), \qquad y_k = h(x_k) + e_k, \tag{2.17}$$

with unknown (hidden) state x_k, observation y_k, k sample number, v_k stochastic noise process, e_k additive noise [27]. Hence, these systems integrate information over time and leverage the motion of the UAS to improve geometry [27, 28]. For this study, this filtering is used as a second-stage refinement to track the estimated location across snapshots. However, sequential filters do not, by themselves, resolve errors like the impacts from non-line of sight measurements; in those cases, the filter may inadvertently and consistently converge on an incorrect localization result unless outliers are appropriately handled at the measurement front-end. Therefore, these techniques should be used in conjunction with other approaches to protect against these cases and improve localization accuracy.

Centralized direct maximum-likelihood localization methods, such as Direct Position Determination (DPD), utilize data from multiple sensors as a way to estimate emitter location without intermediate parameter estimation to find results in a single-step approximation. DPD formulates a position-dependent likelihood directly from the received signals [29]:

$$\widehat{p} = \arg\max_{p \in \mathcal{P}} \; \mathcal{L}(\{y_\ell(t)\}_\ell \, ; \, p) \, , \tag{2.18}$$

where $\widehat{p}$ is the emitter location estimate and $\mathcal{L}(\cdot)$ is derived from a measurement model that assumes Gaussian noise [29]. DPD can achieve near-optimal performance at low SNR by preserving information that is effectively lost in two-step processing. Ideal settings are generally required for DPD, as well as a very accurate channel model. In this work, dense multipath produces strong model mismatch unless the propagation channel is modeled in detail, which reduces the effectiveness of DPD for unknown emitters. Further, the lightweight network architecture in this study is assumed unable to support the significant sampling and statistical analysis required for DPD, which is typically performed with continuous streaming of IQ data. Thus, DPD is not adopted in this study in favor of timing-based methods.

2.1.4.6 *Cyclostationary Feature Exploitation*

Cyclostationary processing leverages periodic signal conditions, such as symbol timing and periodic framing, in man-made waveforms using the cyclic autocorrelation function (CAF) [30]

$$R_x^{\alpha}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t+\tau/2)\, x^*(t-\tau/2)\, e^{-j2\pi\alpha t}\, dt, \tag{2.19}$$

which extracts correlations at specific cycle frequencies α (which are typically multiples or fractions of baud rate), along with related spectral correlation measures. In geolocation, cyclostationary methods are used to enhance TOA/TDOA estimation by reducing interference that does not share the signal-of-interest cycle features [31, 32]. Cyclostationarity has additionally been incorporated into DPD formulations by combining the cyclic AOA and TDOA for a single-step grid search and direct localization estimate [33]. Naturally, these techniques require the emitter to have cyclostationary features, and while this is likely for most cases, it is not a guaranteed condition. CDPD generally requires time-aligned baseband data or the exchange of statistics at a central processing node, and this is followed by a multi-dimensional search or an iterative estimate optimization over candidate positions. This joint processing increases computation and communications relative to two-step approaches that localize using time-based features, like TDOA, and can therefore be challenging to scale in distributed UAS swarms. As a front-end signal pre-processor, when the cycle frequency of the emitter can be reliably identified and sufficient observation time is available, cyclostationary processing can improve measurement integrity in interference-limited and low-SNR conditions by suppressing non-coincident interferers and reducing outlier rates, improving the performance of geometry-based localization, assuming the sensors have computing bandwidth for this pre-processing. Because the focus for this work was instead towards optimizing sensor geometry with an assumption of time-synchronized and correlatable signal data across sensors, cyclostationary processing was not implemented.

2.1.4.7 *Model-Based Localization*

Fingerprinting treats the environment as a propagation model by matching measurements to a pre-collected radio map database. Given a feature r such as RSS, channel state information, or other features, an estimator is

$$\widehat{p} = \arg\min_{p \in \mathcal{P}} \ \|r - \overline{r}(p)\|_W^2, \tag{2.20}$$

that associates measurements to the best-matching map locations [34, 35]. Fingerprinting can exploit multipath as a distinct signal feature rather than treating it as an error.

A separate but related class of approaches attempts to exploit multipath by introducing an environment model, such as a 3D map, and treating propagation paths as geometric constraints. An extracted delay and angle for path m at sensor i is modeled as

$$\tau_{i,m} \approx \frac{1}{c}\, \ell_{i,m}(p; \mathcal{M}), \tag{2.21}$$

where $\ell_{i,m}$ is the predicted path length from candidate emitter position p to sensor i under map $\mathcal{M}$ and the reflection model [36].

While fingerprinting relies on empirical databases and model-based methods on geometric/physical predictions, both leverage site-specific priors to turn multipath into an asset. However, the performance of these techniques depends critically on map density, temporal stability, feature repeatability, correct path association, and sufficient fidelity of $\mathcal{M}$. Therefore, small modeling errors can induce systematic bias and create spurious modes in the location likelihood [37].

For UAS geolocation of unknown emitters, constructing and maintaining a radio map or detailed environment model over a dense urban area is typically infeasible, and environmental changes can rapidly invalidate the database or model. Because these approaches require extensive, location-tagged calibration data, environmental stationarity, and significant characterization of the target location, all of which are not available for unknown emitters and changing urban channels, they are not considered for this study. The approach in this study uses statistical models to enable emitter localization rather than relying on prior characterization of the target environment.

2.2 Machine Learning Approaches

Recent work has applied machine learning (ML) strategies across UAS arrays for robotic path planning, measured and tracked emitter signals, and target localization [38–44]. Other investigations [1, 45, 46] have incorporated RSS and TDOA measurements into UAS-based machine learning approaches. In dense multipath and urban settings, classical triangulation and static placements yield unstable localization estimates and slow convergence [47]. Because UAS are independently mobile, they have the ability to actively shape their geometry during a localization event and this repositioning enables more accurate localization [6]. These efforts motivate a learning-based approach to exploit geometry-aware localization algorithms to control a cooperative UAS array towards accurate emitter localization.

2.2.1 Reinforcement-Learning Controllers for Geometry-Adaptive Localization

We consider, in a machine learning context, a multi-agent system evolving in discrete steps $k = 0, 1, \dots$ with a goal of cooperatively locating an emitter with iteratively lower localization error. Each agent $i \in \{1, \dots, N\}$ observes o_i^k and selects an action a_i^k; the action produces rewards r_i^k and the controller transitions the system to the next step. The controller rewards behavior that improves localization accuracy in this reinforcement learning (RL) setting. The observation vector for this study includes UAS geometry, features from the signal measurement such as TDOA and RSS, and uncertainty metrics derived from the cooperative localization approach.

Because an emitter would be at an unknown location in a real deployment, the performance of the system must be determined without any dependence on the true emitter location. We therefore utilize geometry-driven metrics produced by the cooperative localization algorithm to evaluate the convergence of the system on an accurate solution to guide learning. The specific uncertainty metrics and convergence criteria used in this dissertation are defined in the cooperative localization discussion in Chapter 3 (see Section 3.2.2). The error models and parameter settings used throughout training and evaluation are summarized in Chapter 4 (see Table 4.3). During simulation-based training, the true emitter position is available for computing diagnostics and optional auxiliary reward terms; however, the deployed controller and primary convergence logic rely only on the uncertainty metrics produced by the cooperative localization algorithm, since the true location is not available in operation.

2.2.2 Deep Q-Learning and DQN-Based Geometry Control

The architecture in this study builds on Q-learning localization approaches [39–42] with extensions to a continuous state space [43, 44] to formulate a deep Q-network (DQN) [48] with a discrete action set. In this dissertation, DQN is treated as an independent, off-policy, value-based controller where agents learn from a replay buffer and stabilize temporal-difference (TD) learning with a slowly-updated target network using a soft update rate τ, with discount factor γ.

Each UAS drone agent independently trains a DQN to determine new locations that cooperatively improve localization accuracy. The system begins by measuring signal information at an initial UAS geometry to create an initial observation and state comprised of UAS geometry, features from the signal measurement such as TDOA and RSS, and uncertainty metrics derived from the cooperative localization approach.

At step k, agent i observes a state (feature) vector s_{ik} and executes an action a_{ik}, receiving reward r_{ik} and transitioning to s_{ik+1}. Q-learning seeks the optimal action-value function $Q^\star(s, a)$ that satisfies the Bellman optimality equation [49]. In DQN [48], a neural network $Q_\theta(s, a)$ approximates $Q^\star$ from replayed transitions. For a sampled transition (s_k, a_k, r_k, s_{k+1}), the one-step TD target is

$$y_k = r_k + (1 - \mathrm{done}_k)\,\gamma \max_{a'} Q_{\theta^-}(s_{k+1}, a'), \tag{2.22}$$

where θ^- denotes the target network, $\gamma \in (0, 1)$ is a discount factor, and $\mathrm{done}_k \in 0, 1$ indicates episode termination. The TD error is

$$\delta_k = y_k - Q_\theta(s_k, a_k), \tag{2.23}$$

and parameters are updated to minimize a regression loss such as MSE or Huber [48].

In our application, the DQN structure provides the foundation for learning policies for determining a geometry that maximizes the localization accuracy. In a later chapter (Chapter 5), we describe the specific state features, action set, and training curriculum that are implemented to achieve the desired system performance.

The actions are selected from a discrete set of bounded displacement vectors across the measurement space. During training, action exploration is encouraged with an annealing (ϵ)-greedy policy [49]:

$$a_i^k = \begin{cases} a_{i,\text{greedy}}^k & \text{with probability } 1 - \epsilon_k, \\ a_{i,\text{random}}^k & \text{with probability } \epsilon_k, \end{cases} \tag{2.24}$$

where ϵ_k decays throughout training to guarantee exploration and escape poor local geometries, while later converging to geometry-improving moves. In evaluation, we set $\epsilon_k = 0$ to ensure a pure greedy action selection.

We reward states that produce geometries yielding improved localization performance, and the reward function combines relevant state features and metrics (p_i) along with tuned weights (w_i)

$$r \;=\; w_1 p_1 \;+\; w_2 p_2 \;+\; w_3 p_3. \tag{2.25}$$

2.2.3 Alternate Learning Algorithms

Prior studies have explored using TDOA or RSS as features in RL-based geometry control, and these approaches are categorized as either independent/decentralized or cooperative/-centralized multi-agent controllers [3, 38, 41, 42, 46, 50]. There is limited literature, however, in comparing these controller families under consistent sensing, motion, and noise models. For our study, the cooperative localization method is identical across all controller variants so that only the geometry-control policy changes. This approach allows us to evaluate the effect of independent versus cooperative control on localization performance.

Independent RL controllers include value-based learners such as DQN and methods such as asynchronous advantage actor–critic (A3C) [51]. In multi-agent architectures, independent learners face a challenge of non-stationarity because each agent updates its policy without accounting for the concurrent updates of the others. This non-stationarity refers to the difference in the learning environment seen by any one agent, because as the remaining agents learn and adjust their behaviors over time, the state transitions and rewards associated with a given action can diverge across the agents, making the environment appear time-varying.

Cooperative multi-agent RL methods, such as Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [52] and Multi-Agent Proximal Policy Optimization (MAPPO) [53], utilize centralized training with decentralized execution (CTDE) to mitigate this non-stationarity found in independent learning. In CTDE, decentralized actors utilize local observations for state decisions, while a centralized critic or value function stabilizes learning updates during training using the collective information from all agents. Multi-agent approaches can outperform independent learners for their mission objectives when agent coordination is required, such as in UAS swarms [3, 38, 42]. Therefore, these controller approaches offer significant opportunities for cooperative localization over independent learning approaches.

In this geometry-control setting, algorithms can be distinguished by being on- or off-policy. Off-policy algorithms update from a replay buffer of previously collected state—action-–reward transitions which are often drawn from earlier rollouts or short trajectories of sequential geometry-control steps. This replay-based update allows the controller to reuse past experience and can reduce the number of new episodes or steps required for training. On-policy algorithms update only from rollouts generated by the current policy, which can be more stable to optimize, but typically requires collecting fresh rollouts as the policy evolves.

We categorize the controller families evaluated in this study by their independent or cooperative natures, visualized in Table 2.1. DQN and A3C/A2C are independent learners, while MADDPG and MAPPO are CTDE methods. DQN and MADDPG are trained off-policy using replay buffers and TD updates; in these methods, the discount factor γ sets the effective planning horizon by weighting future geometry improvements relative to immediate reward. Because the TD target depends on a slowly moving function approximator, these off-policy methods also use target networks updated with a soft-update rate $\tau \ll 1$ so that the target values change gradually and stabilize learning from replayed experience. Soft target updates ($\tau \ll 1$) reduce oscillations by preventing the TD target from changing as quickly as the online network. In contrast, A3C/A2C and MAPPO are trained on-policy using finite rollouts (or trajectory segments) of length L, which are used to form advantage estimates A^k and compute policy updates.

Table 2.1: Controllers organized along two axes: train-time centralization (rows) and data regime (columns).

	Off-policy	**On-policy**
Independent	**DQN**	**A3C/A2C**
CTDE	**MADDPG**	**MAPPO**

In addition to learned controllers, we consider a policy-free heuristic-only approach where no policy is trained or applied; the system instead executes greedy actions that maximize an immediate reward and reduce uncertainty, similar to [6]. This baseline is valuable in analyzing potential RL solutions to verify whether the system can exploit domain-specific knowledge from outside the RL model and whether there is an advantage to utilizing RL in the application [54].

2.2.3.1 Multi-Agent Deep Deterministic Policy Gradient

Multi-Agent Deep Deterministic Policy Gradient (MADDPG) is a CTDE actor–critic method with decentralized actors and centralized critics [52]. For agent i, the actor $\pi_{\theta_i}(a_i \mid o_i)$ conditions on local observation, while the critic $Q_{\phi_i}(o, a)$ conditions on joint observation and joint action. This centralized critic reduces the apparent non-stationarity during training by explicitly modeling the dependence of rewards and transitions on the joint behavior.

Because this study uses discrete displacement actions, a common adaptation is to relax discrete sampling with the Gumbel–Softmax (Concrete) estimator [55, 56]. Given logits $z \in \mathbb{R}^{|\mathcal{A}|}$ and i.i.d. Gumbel noise $g_k = -\log(-\log u_k)$ with $u_k \sim \text{Uniform}(0, 1)$, the relaxed action is

$$y_k = \frac{\exp\left((z_k + g_k)/\tau_g\right)}{\sum_j \exp\left((z_j + g_j)/\tau_g\right)}, \tag{2.26}$$

which approaches a one-hot sample as $\tau_g \to 0$.

Critics are trained by minimizing a TD regression,

$$\mathcal{L}_{C,i}(\phi_i) = \mathbb{E}\left[\left(Q_{\phi_i}(o^k, a^k) - \left(r_i^k + (1 - \text{done}_k)\gamma Q_{\phi_i^-}(o^{k+1}, a^{k+1})\right)\right)^2\right], \tag{2.27}$$

and actors are updated to maximize the centralized critic value (optionally with entropy regularization),

$$\max_{\theta_i} \mathbb{E}\big[Q_{\phi_i}(o^k, a^k)\big] + \beta\, \mathbb{E}\big[\mathcal{H}(\pi_{\theta_i}(\cdot \mid o_i^k))\big]. \tag{2.28}$$

2.2.3.2 Advantage Actor–Critic

Asynchronous Advantage Actor–Critic (A3C) improves scalability by updating shared parameters from multiple actor-learners [51]. A3C is based on the policy-gradient theorem with a learned value baseline. For agent i, the policy gradient is weighted by an advantage estimate A_i^k:

$$\nabla_{\theta_i} J_i \approx \mathbb{E}\big[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i^k \mid o_i^k)\ A_i^k\big]. \tag{2.29}$$

Advantages are commonly computed using Generalized Advantage Estimation (GAE) [57]:

$$\delta_i^k = r_i^k + \gamma V_{\psi_i}(o_i^{k+1}) - V_{\psi_i}(o_i^k), \qquad A_i^k = \sum_{l \geq 0} (\gamma\lambda)^l\, \delta_i^{k+l}, \tag{2.30}$$

with return target $R_i^k = A_i^k + V_{\psi_i}(o_i^k)$. The combined actor–critic loss is

$$\mathcal{L}_i(\theta_i, \psi_i) = -\mathbb{E}\big[A_i^k \log \pi_{\theta_i}(a_i^k \mid o_i^k)\big] + c_v\, \mathbb{E}\big[(V_{\psi_i}(o_i^k) - R_i^k)^2\big] - c_H\, \mathbb{E}\big[\mathcal{H}(\pi_{\theta_i}(\cdot \mid o_i^k))\big]. \tag{2.31}$$

2.2.3.3 Multi Agent Proximal Policy Optimization

Proximal Policy Optimization (PPO) stabilizes policy-gradient updates by constraining the change from a behavior policy $\pi_{\theta_{\text{old}}}$ used to collect rollouts [58]. The probability ratio is

$$\rho_k(\theta) = \frac{\pi_\theta(a_i^k \mid o_i^k)}{\pi_{\theta_{\text{old}}}(a_i^k \mid o_i^k)}. \tag{2.32}$$

PPO maximizes the clipped surrogate objective

$$\mathcal{L}_{\text{clip}}(\theta) = \mathbb{E}\big[\min\big(\rho_k(\theta)A^k,\ \text{clip}(\rho_k(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\, A^k\big)\big]\,, \tag{2.33}$$

where A^k is an advantage estimate. Here, ϵ_{clip} is the PPO clipping parameter, and not the ϵ_k used in ϵ-greedy exploration in Eq. (2.24).

Multi-Agent PPO (MAPPO) extends PPO to cooperative multi-agent settings, commonly with a parameter-shared actor for homogeneous agents and a centralized value function to stabilize advantage estimates during training [53]. Parameter sharing improves learning stability by pooling experience into a shared policy. A typical MAPPO loss function combines the PPO objective with value regression and entropy regularization:

$$\mathcal{L}(\theta, \psi) = -\mathcal{L}_{\text{clip}}(\theta) + c_v\, \mathbb{E}\big[(V_\psi(o^k) - R^k)^2\big] - c_H\, \mathbb{E}\big[\mathcal{H}(\pi_\theta(\cdot \mid o_i^k))\big]\,. \tag{2.34}$$

Advantages A^k are commonly computed using Generalized Advantage Estimation (GAE) [57] from a value baseline. The corresponding return target is $R^k = A^k + V_\psi(o^k)$, which is used in the value regression term of Eq. (2.34). In cooperative settings, V_ψ is often formed from a centralized value function that conditions on joint information available during training.

Chapter 3

LOCA APPLICATION FOR UAS

Summary of use of LOCA for this Study LOCA was selected for this study for three primary reasons:

1. First, it utilizes TDOA, thereby avoiding absolute transmit-time dependence while remaining compatible with distributed UAS receivers that can timestamp and correlate signals. These timing-derived observables can be exchanged across sensors as compact features to avoid wideband IQ sharing.

2. Second, it scales with swarm size by producing redundant measurements from many triads. This enables improved localization estimation in the presence of noise.

3. Third, the resulting solution distribution provides a convergence measure for adaptive geometry control and the basis of the reinforcement-learning-based motion reward strategies evaluated in later chapters.

Therefore, in our study, we consider LOCA in the three-dimensional case, requiring four sensors plus a fifth sensor for ambiguity resolution between the two conic focal points for a lower bound minimum requirement of five UAS sensors. The use of additional sensors leads to an increasing number of geolocation estimates; however, as quantified in Chapter 4, using UAS swarm sizes larger than eight provides diminishing improvement in terms of emitter location convergence rate [11], [59]. Because LOCA expresses TDOA constraints along a conic axis jointly defined by sensor triads, rather than independent single-reference hyperbolas for each sensor pair, the geometry is well-suited to distributed and highly–mobile receiver networks.

3.1 LOCA Extension to Four or More Sensors

In the 2D case, when only three sensors are utilized, an ambiguity can arise when the resulting conic is an ellipse or hyperbola since only one of the two different focal points is the actual position of the emitter. If the conic is a circle or parabola, no ambiguity results. The particular form of the conic resulting from the relative positioning of the sensors and emitter can be determined by computing its eccentricity, e_{cc}. If $e_{cc} = 0$ (circle) or $e_{cc} = 1$ (parabola), no ambiguity results. If $0 < e_{cc} < 1$, the conic is an ellipse and if $e_{cc} > 1$, the conic is a hyperbola. It is noted that the case of all sensors lying along a line can be viewed as a conic section with both foci at infinity and is a degenerate case that is incompatible with the LOCA method. With more than three properly located sensors, ambiguities can be resolved mathematically by including an additional sensor, such that a second independent triad can be formed to compute a second conic, thus allowing intersections of the axes of the two conics to be calculated [20]. For the two-dimensional case, two triads (four sensors) are necessary to find this intersection, whereas for the three-dimensional case, since (2.11) results in the equation of a plane, three triads (at least five sensors) are necessary to find the intersection point.

In matrix form, all combinations of triads from $\mathcal{S}$ can be represented from (2.11) as:

$$\begin{pmatrix} A_{123} & B_{123} & C_{123} \\ A_{124} & B_{124} & C_{124} \\ \vdots & \vdots & \vdots \\ A_{ijk} & B_{ijk} & C_{ijk} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \approx \begin{pmatrix} D_{123} \\ D_{124} \\ \vdots \\ D_{ijk} \\ \vdots \end{pmatrix}. \tag{3.1}$$

Subset intersection via numerically stable least squares. Consider the set of five sensors needed for a 3D emitter location estimate as a sensor subset $\mathcal{S}$ of size ℓ. For each sensor subset $\mathcal{S}$, stacking all triads from (3.1) yields an overdetermined system of equations, $A\mathbf{e} \approx b$, to which the (x, y, z) solution is the estimated location of the emitter $\widehat{\mathbf{e}}$. This system can be resolved with a regularized least squares approach. To perform this computation, each of the A_{ijk}, B_{ijk}, C_{ijk}, D_{ijk} elements in are divided by their respective $\sqrt{A_{ijk}^2 + B_{ijk}^2 + C_{ijk}^2}$ prior to solving:

$$
\begin{pmatrix}
\frac{A_{123}}{\sqrt{A_{123}^2+B_{123}^2+C_{123}^2}} & \frac{B_{123}}{\sqrt{A_{123}^2+B_{123}^2+C_{123}^2}} & \frac{C_{123}}{\sqrt{A_{123}^2+B_{123}^2+C_{123}^2}} \\
\frac{A_{124}}{\sqrt{A_{124}^2+B_{124}^2+C_{124}^2}} & \frac{B_{124}}{\sqrt{A_{124}^2+B_{124}^2+C_{124}^2}} & \frac{C_{124}}{\sqrt{A_{124}^2+B_{124}^2+C_{124}^2}} \\
\vdots & \vdots & \vdots \\
\frac{A_{ijk}}{\sqrt{A_{ijk}^2+B_{ijk}^2+C_{ijk}^2}} & \frac{B_{ijk}}{\sqrt{A_{ijk}^2+B_{ijk}^2+C_{ijk}^2}} & \frac{C_{ijk}}{\sqrt{A_{ijk}^2+B_{ijk}^2+C_{ijk}^2}} \\
\vdots & \vdots & \vdots
\end{pmatrix}
\begin{pmatrix} x \\ y \\ z \end{pmatrix} \approx
\begin{pmatrix}
\frac{D_{123}}{\sqrt{A_{123}^2+B_{123}^2+C_{123}^2}} \\
\frac{D_{124}}{\sqrt{A_{124}^2+B_{124}^2+C_{124}^2}} \\
\vdots \\
\frac{D_{ijk}}{\sqrt{A_{ijk}^2+B_{ijk}^2+C_{ijk}^2}} \\
\vdots
\end{pmatrix}. \tag{3.2}
$$

We compute the $\mathcal{S}$-subset emitter estimate using a numerically stable least-squares approach, with regularization used for ill-conditioned geometries:

$$
\widehat{\mathbf{e}} = \arg\min_{\mathbf{e}\in\mathbb{R}^3} \ \|A\mathbf{e} - b\|_2^2 + \lambda\|\mathbf{e}\|_2^2, \tag{3.3}
$$

where $\lambda \geq 0$ is small (or zero) under well-conditioned geometry.

Full LOCA Algorithm. The full LOCA algorithm is shown below in Algorithm 1. Each set of triads from $\mathcal{S}$ are computed in the LOCA approach using the summary of equations in (3.4) from (2.11):

$$\Delta r_{ij} = c\,(t_j - t_i), \tag{3.4a}$$

$$\Sigma_{ijk} = \Delta r_{ij} + \Delta r_{jk} + \Delta r_{ki}, \tag{3.4b}$$

$$A_r = x_i \Delta r_{jk} + x_j \Delta r_{ki} + x_k(\Delta r_{ij} - \Sigma_{ijk}), \tag{3.4c}$$

$$B_r = y_i \Delta r_{jk} + y_j \Delta r_{ki} + y_k(\Delta r_{ij} - \Sigma_{ijk}), \tag{3.4d}$$

$$C_r = z_i \Delta r_{jk} + z_j \Delta r_{ki} + z_k(\Delta r_{ij} - \Sigma_{ijk}), \tag{3.4e}$$

$$D_r = \tfrac{1}{2}\Big(\Delta r_{ij}\Delta r_{jk}\Delta r_{ki} + a_i^2 \Delta r_{jk} + a_j^2 \Delta r_{ki} + a_k^2(\Delta r_{ij} - \Sigma_{ijk})\Big), \tag{3.4f}$$

$$\delta_r = \max\Big(\sqrt{A_r^2 + B_r^2 + C_r^2},\, 10^{-12}\Big), \tag{3.4g}$$

$$\mathbf{r}_r^{\mathsf{T}} = \Big[A_r \quad B_r \quad C_r\Big]/\delta_r, \qquad b_r = D_r/\delta_r. \tag{3.4h}$$

For each subset $\mathcal{S}$ of size ℓ, we re-index the selected sensors locally to the algorithm as $i = 1, ..., \ell$, so s_i and t_i denote the sensor positions and TOAs within the current subset. Pairwise range differences are computed as $\Delta r_{ij} = c(t_j - t_i)$, where $c = 299\,792\,458$ m/s is the speed of light. Each sensor subset $\mathcal{S}$ yields one emitter estimate $\widehat{\mathbf{e}}_m$; the index m enumerates subsets and thus rows of LOCASoln and tags for that timestep. The per-timestep LOCA median estimate $\widehat{\mathbf{e}}_k$ and its associated uncertainty (derived from the filtered cloud) then serve as the input to downstream fusion and sequential estimation stages.

Algorithm 1: LOCA 3D Solution Cloud via Sensor Subsets and Triad Constraints

Require : Sensor positions $S \in \mathbb{R}^{N\times 3}$, time of arrival $t \in \mathbb{R}^{N}$

Ensure : LOCASoln $\in \mathbb{R}^{M\times 3}$, tags $\in \{0,1\}^{M\times N}$

$c \leftarrow 299\,792\,458\text{m/s}$

$\mathcal{X} \leftarrow [\,]$

$\mathcal{T} \leftarrow [\,]$

$m \leftarrow 0$

for $\ell \leftarrow 5$ **to** N **do**

  foreach $\mathcal{S} \subseteq \{1,\ldots,N\}$ with $|\mathcal{S}| = \ell$ **do**

    $m \leftarrow m + 1$

    for $i \leftarrow 1$ **to** ℓ **do**

      $a_i \leftarrow \sqrt{x_i^2 + y_i^2 + z_i^2}$

    for $i \leftarrow 1$ **to** ℓ **do**

      for $j \leftarrow 1$ **to** ℓ **do**

        $\Delta r_{ij} \leftarrow c(t_j - t_i)$

    $A \leftarrow [\,] \in \mathbb{R}^{0\times 3}$

    $b \leftarrow [\,] \in \mathbb{R}^{0}$

    $r \leftarrow 0$

    foreach triad (i,j,k) with $1 \le i < j < k \le \ell$ **do**

      $r \leftarrow r + 1$

      Compute $(A_r, B_r, C_r, D_r, \delta_r)$ from (i,j,k) using (3.4) Append row $\mathbf{r}_r^{\mathsf{T}}$ and b_r using (3.4)

    $\hat{\mathbf{e}}_m \leftarrow$ regularized least squares solution of $A\mathbf{e} \approx b$ (3.3)

    Append $\hat{\mathbf{e}}_m$ to $\mathcal{X}$

    $\mathbf{t}_m \leftarrow \mathbf{0} \in \{0,1\}^N$

    for $n \leftarrow 1$ **to** N **do**

      if $n \in \mathcal{S}$ **then**

        $(\mathbf{t}_m)_n \leftarrow 1$

    Append $\mathbf{t}_m$ to $\mathcal{T}$

LOCASoln $\leftarrow$ stack($\mathcal{X}$)

tags $\leftarrow$ stack($\mathcal{T}$)

return (LOCASoln, tags)

3.2 Estimate Filtering and Distributions

3.2.1 IQR-based filtering of the LOCA solution cloud.

In an N-sensor system, the LOCA algorithm shown allows combinations of any subset of five sensors to generate a cloud of emitter location estimates.

$$\Lambda = \{\boldsymbol{\lambda}_m\}_{m=1}^{M} \subset \mathbb{R}^3. \tag{3.5}$$

There are $\binom{N}{3}$ triads that can be formed from a set of N sensors. When forming LOCA estimates from all sensor subsets ℓ of size 5, the number of subset-based location estimates for a fixed N-sensor system is $M(n) = \sum_{\ell=5}^{n} \binom{n}{\ell}$. If we sweep the sensor count from $n = 5$ up to $n = N$ (as in Fig. 3.1), the cumulative number of subset solutions grows as

$$I_{\max}(N) = \sum_{n=5}^{N} \sum_{\ell=5}^{n} \binom{n}{\ell}. \tag{3.6}$$

With $N = 5$ sensors, only one $\ell = 5$ subset of sensors exists in a single 3D LOCA estimate, so the solution cloud contains a single estimate per timestep or "snapshot". For six $\ell = \binom{6}{5}$ or more sensors, many subsets exist, yielding a redundant solution cloud that enables statistical filtering. For $N = 8$, this cumulative total is $I_{\max}(8) = 130$. Fig. 3.1 presents an example of how the cumulative number of subset-based LOCA estimates increases with sensor count and illustrates the corresponding growth in estimate-cloud density.

Thus, the system utilizing this algorithm could statistically determine the most probable emitter location and utilize the density of estimates to refine its solution. While a five-sensor configuration is minimally viable, adding at least a sixth sensor enables redundancy and consistency checking across solutions, and it provides a distribution of estimates that can be analyzed for mean or median performance to reduce sensitivity to outliers and unfavorable geometries. Increasing to seven or eight sensors further increases the number of available subset solutions, thus producing a denser and potentially broader estimate cloud, improving the likelihood of well-conditioned subsets and strengthening statistical refinement. Naturally, the expansion is at the cost of additional platform, payload, and coordination complexity.

With a large solution set formed with N-sensors, we initially filter the cloud based on

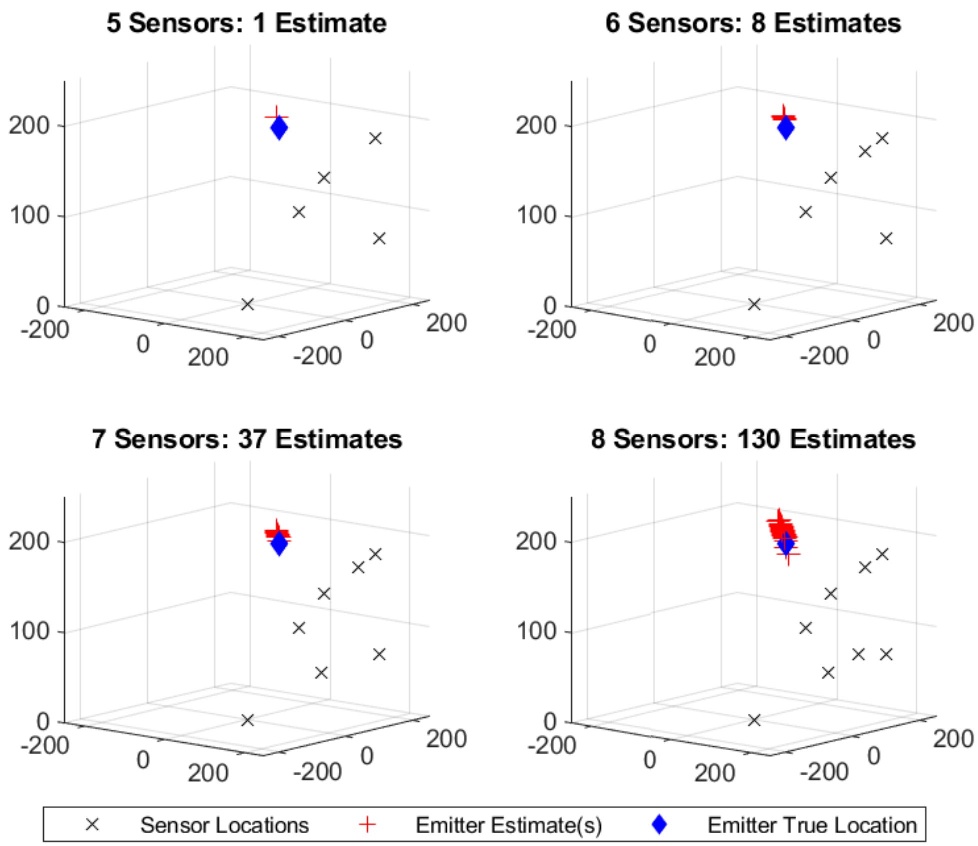

Figure 3.1: Count and placement of LOCA estimates with increasing sensor quantities showing the growth in per-timestep subset estimates (solution-cloud density) with increasing N

the middle 50% interquartile range (IQR) of the solution set and define the emitter estimate $\widehat{\mathbf{e}}$ as a probabilistic centroid given by the median of the filtered LOCA solutions [59].

Let (3.5) denote the set of per-subset LOCA estimates at a given timestep, where $\boldsymbol{\lambda}_m = [x_m \ y_m \ z_m]^\mathsf{T}$. To suppress gross outliers without assuming a parametric distribution, we apply the standard interquartile-range (IQR) rule independently to each coordinate. Define the componentwise quartiles $Q_1 = \text{quantile}(\Lambda, 0.25)$ and $Q_3 = \text{quantile}(\Lambda, 0.75)$, with $\text{IQR} = Q_3 - Q_1$, and bounds

$$L = Q_1 - 1.5 * \text{IQR}, \qquad U = Q_3 + 1.5 * \text{IQR} \tag{3.7}$$

interpreted componentwise. We retain an estimate $\boldsymbol{\lambda}_m$ when $L_x \le x_m \le U_x$, $L_y \le y_m \le U_y$, and $L_z \le z_m \le U_z$, forming the filtered set Λ_f.

The per-timestep LOCA measurement is tracked as the median, and not the mean, of the solution cloud to guard against outliers or asymmetries in the filtered solution cloud:

$$\widehat{\mathbf{e}}_{k,\text{med}} = \text{median}(\Lambda_\mathrm{f}). \tag{3.8}$$

We use this median $\widehat{\mathbf{e}}_{k,\text{med}}$ for reporting and for cloud-derived uncertainty metrics. When an optional sequential tracker is used (Sec. 3.2.3), we additionally form a Gaussian summary (z_k, R_k) for Kalman filtering; this does not change the cloud-based metrics.

3.2.2 Elliptical and Spherical Error Distribution

[1] In an ideal geometry with minimal UAS positioning errors and time of arrival errors, a localization cloud would be tightly clustered with minimal dispersion of $\widehat{e}_i$ estimates from $\widehat{e}_m$. However, these errors are inherent to the hardware design and especially increased in a dense multipath environment where building reflections impact signal propagation. The underlying geometry of LOCA can improve localization accuracy by repositioning the UAS array into a configuration that minimizes the spread of Λ_f [59].

The quality of the localization estimate can be expressed in terms of the spherical error probable (SEP) radius, denoted SEP_{50} for the median case, where the radius defines a spatial

[1] Analysis [60] submitted to the 2026 IEEE Dallas Circuits and Systems Conference.

region containing the estimated location Λ_{f} with probability 0.50 [61]:

$$SEP_{50} \;=\; q_{0.5}\Big\{\|\mathbf{x}_i - \widehat{\mathbf{e}}_{k,\mathrm{med}}\|_2 :\; \mathbf{x}_i \in \Lambda_{\mathrm{f}}\Big\}\,. \tag{3.9}$$

where $q_{0.5}\{\cdot\}$ denotes the 0.5 quantile (median) operator.

Smaller SEP_{50} indicates a tighter cloud and, in this study, is used as a convergence metric for geometry control.

To quantify localization uncertainty, we calculate empirical elliptical error probable (EEP) in azimuth and elevation planes directly from the filtered solution cloud. Following standard confidence-ellipse theory [4], if the in-plane cloud is locally approximated as Gaussian, the probability-p ellipse is given by the quadratic form $(\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}) \leq \chi_2^{2-1}(p)$. The covariance matrix formed from the estimate cloud Λ_{f} is calculated and an eigendecomposition is performed for the azimuth and elevation planes with a small positive regularity term $\varepsilon\mathbf{I}_3$ added to stabilize the result. Let $\lambda_1 \geq \lambda_2$ be the eigenvalues of the in-plane covariance; then the EEP semiaxes are [62]

$$a = \sqrt{\chi_2^{2-1}(p)\,\lambda_1}, \qquad b = \sqrt{\chi_2^{2-1}(p)\,\lambda_2}. \tag{3.10}$$

Azimuth $(a_{\mathrm{az}}, b_{\mathrm{az}})$ is calculated in the horizontal x–y plane. Elevation $(a_{\mathrm{el}}, b_{\mathrm{el}})$ is calculated by the plane spanning the dominant horizontal uncertainty direction and the vertical axis to provide a meaningful 2D slice for elevation error analysis.

A circularized in-plane radius, which is a CEP-like summary in terms of azimuth r_{az} and elevation r_{el}, is obtained from the EEP semiaxes using the NASA convention [62]:

$$r_{\mathrm{EEP}} \approx 0.75\,\sqrt{a^2 + b^2}. \tag{3.11}$$

This probability depends on ellipse eccentricity, and in this study, these approximated circularized radii, in terms of Azimuth EEP and Elevation EEP, are utilized as the primary metrics for the geometry convergence.

The algorithm to compute the EEP and SEP from the estimate cloud is presented in Algorithm 2, generalized from the 50% confidence p used in this study.

Algorithm 2: Azimuth and Elevation EEP Radii and 3D SEP from a 3D Estimate Cloud

Require : Emitter Location Estimate Cloud $\Lambda_{\mathrm{f}} = \{\mathbf{x}_i\}_{i=1}^{N>1}$, $\mathbf{x}_i \in \mathbb{R}^3$; confidence level $p \in (0, 1)$
Ensure : $(a_{\mathrm{az}}, b_{\mathrm{az}}, r_{\mathrm{az}}, a_{\mathrm{el}}, b_{\mathrm{el}}, r_{\mathrm{el}}, \mathrm{SEP})$

Center and 3D estimate covariance
$\widehat{\mathbf{e}} \leftarrow \mathrm{median}(\{\mathbf{x}_i\}_{i=1}^{N})$
$\tilde{\mathbf{x}}_i \leftarrow \mathbf{x}_i - \widehat{\mathbf{e}},\ \forall i$
$\boldsymbol{\Sigma} \leftarrow \frac{1}{N-1} \sum_{i=1}^{N} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^{\mathsf{T}} + \varepsilon \mathbf{I}_3$
$q \leftarrow \chi_2^{2\,-1}(p)$

Azimuth (horizontal x–y plane)
$\boldsymbol{\Sigma}_{xy} \leftarrow \boldsymbol{\Sigma}(1{:}2,\ 1{:}2)$
Compute eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ of $\boldsymbol{\Sigma}_{xy}$
$a_{\mathrm{az}} \leftarrow \sqrt{q\,\lambda_1}, \quad b_{\mathrm{az}} \leftarrow \sqrt{q\,\lambda_2}$
$r_{\mathrm{az}} \leftarrow 0.75\,\sqrt{a_{\mathrm{az}}^2 + b_{\mathrm{az}}^2}$

Elevation plane (spanned by dominant horizontal direction and $\hat{\mathbf{z}}$)
Let $\mathbf{u} \in \mathbb{R}^2$ be the unit eigenvector of $\boldsymbol{\Sigma}_{xy}$ associated with λ_1
$\mathbf{h} \leftarrow [\,u_1\ \ u_2\ \ 0\,]^{\mathsf{T}}, \quad \mathbf{h} \leftarrow \mathbf{h}/\|\mathbf{h}\|$
$\hat{\mathbf{z}} \leftarrow [\,0\ \ 0\ \ 1\,]^{\mathsf{T}}$
$\mathbf{T} \leftarrow [\,\mathbf{h}\ \ \hat{\mathbf{z}}\,] \in \mathbb{R}^{3\times 2}$
$\boldsymbol{\Sigma}_{\mathrm{el}} \leftarrow \mathbf{T}^{\mathsf{T}} \boldsymbol{\Sigma} \mathbf{T} \in \mathbb{R}^{2\times 2}$
Compute eigenvalues $\mu_1 \geq \mu_2 \geq 0$ of $\boldsymbol{\Sigma}_{\mathrm{el}}$
$a_{\mathrm{el}} \leftarrow \sqrt{q\,\mu_1}, \quad b_{\mathrm{el}} \leftarrow \sqrt{q\,\mu_2}$
$r_{\mathrm{el}} \leftarrow 0.75\,\sqrt{a_{\mathrm{el}}^2 + b_{\mathrm{el}}^2}$

3D SEP (empirical from cloud radii)
$r_i \leftarrow \|\mathbf{x}_i - \widehat{\mathbf{e}}\|_2,\ \forall i$
$\mathrm{SEP} \leftarrow q_p\big(\{r_i\}_{i=1}^{N}\big)$

return $(a_{\mathrm{az}}, b_{\mathrm{az}}, r_{\mathrm{az}}, a_{\mathrm{el}}, b_{\mathrm{el}}, r_{\mathrm{el}}, \mathrm{SEP})$

The cloud median and dispersion measures defined in this section provide geometry-driven metrics for convergence and reward shaping in later controllers (Chapter 5). When a temporally smooth track is helpful for visualization or evaluation, we additionally summarize the cloud as an approximate Gaussian measurement (z_k, R_k) and apply a standard position-only Kalman filter, described next.

3.2.3 Fusion and Sequential Estimation

At each snapshot k, LOCA produces a solution cloud of emitter locations, $\Lambda_k = \{\boldsymbol{\lambda}_q\}_{q=1}^{Q}$, where $\boldsymbol{\lambda}_q \in \mathbb{R}^3$. For subsequent processing we summarize this cloud by an approximate Gaussian measurement (z_k, R_k), where z_k is a variance-aware centroid and R_k is a corresponding measurement covariance.

Each candidate $\boldsymbol{\lambda}_q$ is scored by its agreement with the current timing-derived TDOA measurements using all sensor pairs (Chapter 4). Let the measured pairwise TDOA at snapshot k be $\Delta t_{ij,k}$ for all $1 \leq i < j \leq N$, and stack them in a fixed order into

$$\mathbf{d}_k \;=\; \big[\Delta t_{12,k}\ \Delta t_{13,k}\ \cdots\ \Delta t_{(N-1)N,k}\big]^{\mathsf{T}} \in \mathbb{R}^P, \qquad P = \tbinom{N}{2}. \tag{3.12}$$

For a candidate location $\boldsymbol{\lambda}_q$, we form the predicted all-pairs TDOA vector

$$\widehat{\Delta t}_{ij}(\boldsymbol{\lambda}_q) \;=\; \frac{\|\boldsymbol{\lambda}_q - \mathbf{s}_i\| - \|\boldsymbol{\lambda}_q - \mathbf{s}_j\|}{c}, \qquad \widehat{\mathbf{d}}(\boldsymbol{\lambda}_q) = \big[\widehat{\Delta t}_{12}(\boldsymbol{\lambda}_q)\ \cdots\ \widehat{\Delta t}_{(N-1)N}(\boldsymbol{\lambda}_q)\big]^{\mathsf{T}}, \tag{3.13}$$

where $\mathbf{s}_i$ is the ith sensor position and c is propagation speed.

Using the modeled per-pair timing variances (Chapter 4), we take a diagonal pairwise timing covariance

$$\Sigma_{t,k} \;=\; \mathrm{diag}\big(\sigma^2_{12,k}, \sigma^2_{13,k}, \ldots, \sigma^2_{(N-1)N,k}\big), \tag{3.14}$$

and compute a variance-aware residual

$$r^2_{k,q} \;=\; \Big(\mathbf{d}_k - \widehat{\mathbf{d}}(\boldsymbol{\lambda}_q)\Big)^{\mathsf{T}} \Sigma^{-1}_{t,k} \Big(\mathbf{d}_k - \widehat{\mathbf{d}}(\boldsymbol{\lambda}_q)\Big) \;=\; \sum_{i<j} \frac{\big(\Delta t_{ij,k} - \widehat{\Delta t}_{ij}(\boldsymbol{\lambda}_q)\big)^2}{\sigma^2_{ij,k}}. \tag{3.15}$$

This inverse-variance structure ensures that lower-variance TDOA pairs contribute more strongly to the consistency score. Because pairwise residuals can be corrupted by multipath, we convert $r_{k,q}$ into a robust weight using a Tukey bisquare (biweight) rule, following the robust M-estimation treatment in [63]. Let $m_k = \mathrm{median}(\{r_{k,q}\}_{q=1}^{Q})$

$$s_k \;=\; \mathrm{median}\Big(\{|r_{k,q} - m_k|\}_{q=1}^{Q}\Big)\,, \tag{3.16}$$

define $u_{k,q} = r_{k,q}/(\kappa\ \max(s_k, \epsilon))$ with small $\epsilon > 0$ and tuning constant κ, and set

$$\omega_{k,q} \;=\; \begin{cases} \left(1 - u_{k,q}^2\right)^2, & |u_{k,q}| < 1, \\ 0, & |u_{k,q}| \ge 1. \end{cases} \tag{3.17}$$

Candidates inconsistent with the timing model contribute negligibly to the fused measurement.

We smooth the per-snapshot measurements with a position-only Kalman filter (KF) [64]. Because LOCA provides a 3-D position estimate per snapshot, we use a linear KF with $F = I_3$ and $H = I_3$ (random-walk process model and direct position measurement):

$$x_{k+1} = x_k + w_k, \quad w_k \sim \mathcal{N}(0, Q_w), \qquad z_k = x_k + n_k, \quad n_k \sim \mathcal{N}(0, R_k), \tag{3.18}$$

with $Q_w = qI_3$ and q chosen small relative to R_k.

Define the innovation $\nu_k = z_k - x_{k|k-1}$ and innovation covariance $S_k = P_{k|k-1} + R_k$ (since $H = I_3$). The normalized innovation squared (NIS) statistic is

$$\eta_k \;=\; \nu_k^{\mathsf{T}} S_k^{-1} \nu_k, \tag{3.19}$$

which is χ^2-distributed with 3 degrees of freedom under the assumed model.

We use η_k as a consistency check. If a snapshot is inconsistent with the assumed innovation covariance S_k, the KF-updated estimate is rejected and the system continues with the predicted state. The KF update is accepted only when $\eta_k \le \gamma$, where γ is selected from a χ^2 threshold at a chosen gate probability with degrees of freedom equal to the innovation dimension (3 for 3-D position), similar to the approach in [65].

3.3 Hardware Implementation

This dissertation considers a cooperative UAS-based wireless sensor array comprised of UAS equipped with individual antennas, RF receivers, local positioning capabilities, high-resolution timers, WANET interfaces, and computing subsystems. The goal of this section is to document the assumed functional architecture and the representative subsystem performance bounds used throughout the dissertation. These assumptions directly inform the hardware error models (Chapter 4) and provide context for the simulation parameter choices used in the algorithm evaluation and geometry-control studies (Chapter 5). A representative functional block diagram is shown in Fig. 1.1.

UAS payload and networking constraints. Alternative geolocation technique families and their feasibility under payload and networking constraints are discussed in Chapter 2. Consistent with those constraints, each UAS is assumed to utilize a lightweight and non-phase-coherent RF front end, the UAS network links do not continuously stream wideband IQ data, and the emitter waveform is not assumed to be known. Therefore, the architecture focuses on timing-derived observables exchanged as compact features and refined by aggregating results from cooperative measurements across a geometrically-diverse sensor array.

RF receiver and time-stamp generation. Each UAS receives the signal-of-interest using an omnidirectional antenna and a local RF receiver chain that can detect (or correlate against) the signal sufficiently well to assign a time-of-arrival (TOA) time stamp to each snapshot. These TOA features, together with the local sensor position estimate and quality indicators, are the primary observables exchanged across the WANET. This design avoids continuous wideband IQ sharing and instead supports cooperative fusion using compact per-snapshot features, consistent with the overall payload and networking constraints described above.

Local timing source and synchronization. It would be preferable if the local UAS clock sources within the wireless array network could operate in a synchronous manner at precise and equal frequencies with minimal drift to enable phase coherency among the array elements; however, synchronization of the receivers' local oscillators is difficult to achieve and maintain and is not assumed in this study. Even when local UAS clocks are synchronized prior to the mission, the presence of clock drift and jitter affects timing accuracy and therefore degrades time-based localization estimates.

The utilization of a WANET clock synchronization method would be desirable to mitigate clock error. It has been shown that, with a wideband receiver, it is possible to measure propagation time at sub-nanosecond accuracy using off-the-shelf IEEE 802.11 network interface cards [66], [67]. The use of ultrawideband (UWB) has likewise been shown to achieve sub-nanosecond accuracy [68]. Another significant advancement in local timing sources is the emergence of Chip Scale Atomic Clocks (CSAC) that provide highly stable and accurate timing sources with a small form factor [9], [10], [69], [70]. In the hardware model used throughout this dissertation, each UAS is assumed to have a high-resolution local timing source whose baseline error is on the order of 10 ns (chosen conservatively) but which can be improved toward the nanosecond regime with a CSAC and supporting synchronization methods.

Self-positioning, hover drift, and representative bounds. In the cooperative UAS scenario described here, the UAS positions may be known via the relatively accurate Global Positioning System (GPS) or other navigational systems; however, the UAS comprising the sensor array are individually hovering during the time that TOA/TDOA values are obtained and thus induce a localization error source we term as hover drift. Other small sensor positioning errors due to variations in environmental factors, such as wind gusts, can cause further degradation in sensor localization. Furthermore, in some environments, GPS may be unavailable requiring an alternative sensor localization approach.

We assume that each UAS can bound its self-positioning error using an on-board ranging/self-localization subsystem, such as LiDAR-based techniques [8]. Texas Instruments has developed a compact LiDAR solution [71] that permits a ranging accuracy of 1 cm at 100 m, which is a significant improvement over GPS [72]. For this reason, the baseline self-positioning error used in this dissertation is bounded to the centimeter regime (2 cm in the nominal case), and the impact of degraded positioning accuracy is studied explicitly in the error-source analysis (Chapter 4).

WANET interface and cooperative computation. The UAS are assumed to form a wireless ad hoc network (WANET) that supports the exchange of compact observables (time stamps, range-difference features, sensor position estimates, and associated error parameters) and supports command and control messages for geometry adaptation. This communication layer enables (i) cooperative fusion across many sensor subsets to generate LOCA solution clouds and related uncertainty metrics, and (ii) closed-loop geometry control policies that reposition the array to reduce localization uncertainty. The architecture therefore supports a distributed sensing capability with low-bandwidth feature exchange rather than centralized wideband RF data aggregation.

Chapter 4

EVALUATION OF ERROR SOURCES IN GEOLOCATION WITH COOPERATIVE UAS ARRAYS

This chapter describes the dominant error sources in a cooperative UAS-based wireless sensor array comprised of UAS equipped with individual antennas, RF receivers, local positioning capabilities, high-resolution timers, WANET interfaces and computing subsystems. The hardware error analysis was published in [11] and is summarized in Section 4.1. The environmental and accumulated error analyses are summarized in Section 4.2. The chapter concludes with simulations that quantify the impact of these error sources on localization performance.

The formulation and results of this chapter were published and presented at the 2023 IEEE Aerospace Conference [11] and at the 2025 IEEE Systems Conference [59].

4.1 Analysis of Geolocation Hardware Error Sources

The formulation and results of this section were published and presented at the 2023 IEEE Aerospace Conference [11].

4.1.1 Position and Timing Error

The location of each sensor i is modeled as its true location: $\mathbf{s}_i = [x_i, y_i, z_i]^\mathsf{T} \in \mathbb{R}^3$. Each UAS operates with measurement inaccuracies associated with sensor location (e.g., hover drift and onboard navigation uncertainty). Thus, a better model of the location of each sensor is to consider the coordinates as an estimated position $\widehat{\mathbf{s}}_i = (\widehat{x}_i, \widehat{y}_i, \widehat{z}_i)$, that include error terms, $(\epsilon_x, \epsilon_y, \epsilon_z)$, as given in (4.1):

$$\widehat{\mathbf{s}}_i = \mathbf{s}_i + \boldsymbol{\epsilon}_{s,i}, \qquad \boldsymbol{\epsilon}_{s,i} = [\epsilon_{xi}, \epsilon_{yi}, \epsilon_{zi}]^\mathsf{T}, \tag{4.1}$$

where $\boldsymbol{\epsilon}_{s,i}$ is treated as zero-mean with scenario-dependent covariance.

Assuming the signal propagation velocity, c, is constant in a particular environment, the emitter range is dependent on the accuracy of the emitter-to-sensor propagation time. Representative timing-source and synchronization assumptions (and corresponding subsystem bounds) are summarized in Section 3.3. The sensor TOA measurement can be modeled as the true TOA, t_i, with a corresponding error term, ϵ_{ti}, as

$$\widehat{t_i} = t_i + \epsilon_{ti}. \tag{4.2}$$

It is likewise assumed that the detected and received RF transmissions of each sensor are at sufficiently high-power levels such that variations in antenna gain patterns are negligible and that all sensors receive transmitted emitter signals at sufficiently high Signal-to-Noise (SNR) levels. For these reasons, we primarily focus on sensor localization, local measurement timing errors, and multipath delay effects to localization accuracy.

Another significant source of inaccuracy is related to the ability of each sensor to accurately identify the same instantaneously received signal features when obtaining a TOA measurement by assigning a local time stamp to the same portion of the received signal by each sensor array element. This issue has been addressed in depth in past work: for communications signals, previous studies have resulted in acceptable methods for correlating continuous signals across spatially separated sensors [73], [74], [75], [76], [77]; whereas for pulsed signals, rising or falling edges are typically used [78], [79] to determine TOA signal time "stamps". It is assumed that this “time stamping” error source is relatively small in comparison to the errors from unsynchronized local UAS timing sources across the UAS network. That is, it is assumed that any timing errors due to misaligned time stamping of received signal artifacts among individual sensor array elements is much less than the local UAS clock period. Thus, inaccuracies due to this error source may be considered as being contained within the timing error models employed in our analyses.

4.1.2 Cramér–Rao Lower Bound for Sensor Position and Clock Timing Errors

In consideration of the dominant error sources due to sensor localization caused by UAS hover drift and the fact that local clock sources with errors are present within each UAS, we model the cooperative wireless sensor array network and simulate the resulting emitter location accuracy with these error sources present using different timing-based emitter location algorithms. Specifically, we consider both multilateration and LOCA emitter location techniques in our models.

To provide a unified basis in evaluating the emitter location accuracy of the sensor array, we also derive and compute the Cramér–Rao Lower Bound (CRLB) to evaluate the theoretically possible emitter location accuracy due to varying amounts of the dominant sources of error and measurement inaccuracy.

The CRLB is a measure of the minimal variance that can be achieved by an unbiased estimator. The positions of the UAS are known to a certain accuracy characterized by the standard deviations of the location estimates, $(\sigma_{xi}, \sigma_{yi}, \sigma_{zi})$. Likewise, the local timing errors within each UAS are modeled with a standard deviation statistic denoted as σ_{ti}. The measurement errors are modeled as Gaussian random variables (RV), following the approaches in [4], [5], [80], [81]with a likelihood function of:

$$p(\widehat{\tau}_i|\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2\sigma_i^2}(\widehat{\tau}_i - \tau_i(\mathbf{x}))^2}. \tag{4.3}$$

With respect to TOA measurements, we assume that the variance of a single measurement is dependent on the variance σ_i^2 of the single sensor s_i. For TDOA approaches, [81] and others show that the use of the common reference or anchor sensor, s_1, correlates the measurements so that the variance of a single measurement is the sum of the variance of the two sensors, $\sigma_i^2 + \sigma_1^2$, as characterized by the corresponding $\Big\{N-1\Big\} \times \Big\{N-1\Big\}$ covariance matrix:

$$\mathbf{R}_{TDOA} = \begin{pmatrix} \sigma_2^2 + \sigma_1^2 & \sigma_1^2 & \dots & \sigma_1^2 \\ \sigma_1^2 & \sigma_3^2 + \sigma_1^2 & \dots & \sigma_1^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^2 & \sigma_1^2 & \dots & \sigma_N^2 + \sigma_1^2 \end{pmatrix}. \tag{4.4}$$

The CRLB is determined by calculating the inverse of the Fisher Information Matrix (FIM), $\mathbf{J}$, expressed as

$$\mathbf{J} = \mathrm{E}\big[\nabla_{\mathbf{x}} \ln p(\widehat{\boldsymbol{\tau}} \mid \mathbf{x})\ \nabla_{\mathbf{x}}^{\mathsf{T}} \ln p(\widehat{\boldsymbol{\tau}} \mid \mathbf{x})\big], \qquad \mathrm{Cov}(\widehat{\mathbf{x}}) \succeq \mathbf{J}^{-1}, \qquad \sigma_{x_i}^2 \triangleq \mathrm{Var}(\widehat{x}_i) \geq [\mathbf{J}^{-1}]_{ii}. \tag{4.5}$$

E[•] denotes the expected value operator of a RV. Under the assumption that the errors are modeled with a Gaussian RV, a constant covariance matrix results [81]. The elements of the FIM can be determined as

$$\mathbf{J}_{ij} = \left(\frac{\partial\tau(\mathbf{x})}{\partial x_i}\right)^T \mathbf{R}_{TDOA}^{-1} \left(\frac{\partial\tau(\mathbf{x})}{\partial x_j}\right),$$

$$\frac{\partial\tau(\mathbf{x})}{\partial x_i} = \begin{pmatrix} \frac{\partial\tau_2(\mathbf{x})}{\partial x_i} \\ \frac{\partial\tau_3(\mathbf{x})}{\partial x_i} \\ \vdots \\ \frac{\partial\tau_N(\mathbf{x})}{\partial x_i} \end{pmatrix}. \tag{4.6}$$

For receiver localization errors, the FIM becomes

$$\mathbf{J}_\tau = \begin{pmatrix} \frac{\partial\tau_2}{\partial x} & \frac{\partial\tau_3}{\partial x} & \dots & \frac{\partial\tau_N}{\partial x} \\ \frac{\partial\tau_2}{\partial y} & \frac{\partial\tau_3}{\partial y} & \dots & \frac{\partial\tau_N}{\partial y} \\ \frac{\partial\tau_2}{\partial z} & \frac{\partial\tau_3}{\partial z} & \dots & \frac{\partial\tau_N}{\partial z} \end{pmatrix} \mathbf{R}_{TDOA}^{-1} \begin{pmatrix} \frac{\partial\tau_2}{\partial x} & \frac{\partial\tau_2}{\partial y} & \frac{\partial\tau_2}{\partial z} \\ \frac{\partial\tau_3}{\partial x} & \frac{\partial\tau_3}{\partial y} & \frac{\partial\tau_3}{\partial z} \\ \vdots & \vdots & \vdots \\ \frac{\partial\tau_N}{\partial x} & \frac{\partial\tau_N}{\partial y} & \frac{\partial\tau_N}{\partial z} \end{pmatrix}^T. \tag{4.7}$$

The FIM can be evaluated in the temporal (time) domain or in the spatial (range) domain depending on the error terms of the covariance matrix. The conversion to the range domain is obtained by multiplying $\mathbf{R}_{TDOA}^{-1}$ with the square of the signal propagation speed, c^2. In the following analysis, the receiver location error contributions are analyzed in the range domain using $r_i = ct_i$ and (2.7).

For timing errors, [5], [68], [81], identified the primary contributing factor for calculating the CRLB of a TDOA estimate to be the bandwidth, B, of the received signal and the received signal SNR_i at sensor s_i, via the relationship:

$$\sigma_i^2 \geq \frac{c^2}{B^2 * SNR_i}. \tag{4.8}$$

Following [81], we consider the factors carrier frequency, bandwidth, integration time, propagation velocity, etc. to be incorporated into the constant, a. Note that the received signal SNR_i at the i^th sensor element is not considered within the constant, a, in our analysis

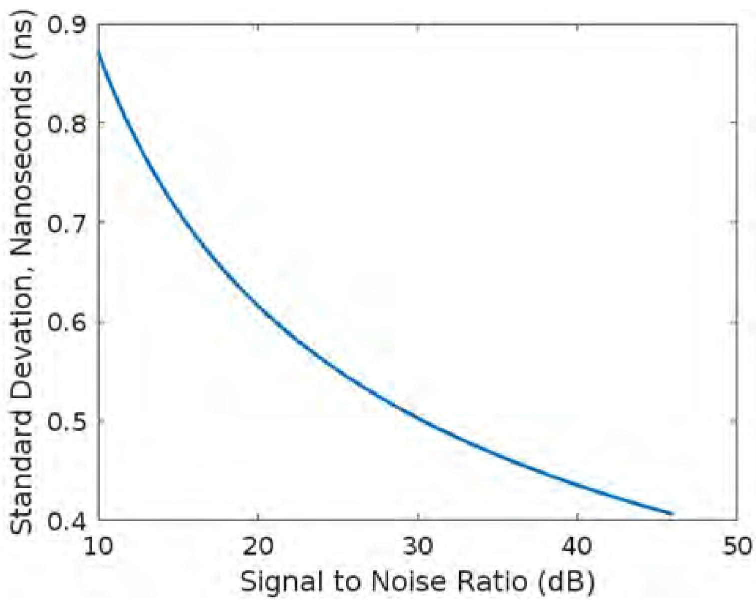

Figure 4.1: Standard Deviation of local timing inaccuracy as a function of SNR.

since we assume that SNR_i exceeds a threshold value. That is, we assume that the received signal is at a sufficiently high SNR to enable emitter location estimation such that a need not comprise a contribution due to SNR_i. Because the SNR_i of the emitter at s_i is inversely proportional to the range by $\frac{1}{r^2}$, an ideal lower threshold for the distance, r_0, for an optimal SNR_0 is used to determine the SNR_i, by:

$$SNR_i = SNR_0 \frac{r_0^2}{r_i^2}. \tag{4.9}$$

The associated variance is then:

$$\sigma_i^2(r) \geq \frac{a}{SNR_0} * \frac{r_i^2}{r_0^2}. \tag{4.10}$$

As an example of this effect, we assume a signal bandwidth, B, and corresponding SNR_0 that result in a worst case, or maximum value, for the standard deviation, σ, to be approximately 1 ns at the maximum range. Fig. 4.1 shows the improvement of σ for increasing SNR where the minimum σ is determined by the SNR at r_0.

Following [81], because the variance is range-dependent and no longer constant, its term in the likelihood function, and therefore the FIM, is not reduced. The TDOA sensor pair $\sigma_i^2 + \sigma_1^2$ variance term is:

$$\sigma_i^2 + \sigma_1^2 = \frac{a(r_i^2 + r_1^2)}{r_0^2 SNR_0}, \tag{4.11}$$

with a standard deviation, σ_{i1}, given by

$$\sigma_{i1} = \sqrt{\sigma_i^2 + \sigma_1^2} = \frac{\sqrt{a(r_i^2 + r_1^2)}}{r_0\sqrt{SNR_0}}. \tag{4.12}$$

The FIM in (4.5) takes the form

$$\mathbf{J}_{ij} = \left(\tfrac{\partial \tau(\mathbf{x})}{\partial x_i}\right)^T \mathbf{R}(\mathbf{x})^{-1} \frac{\partial \tau(\mathbf{x})}{\partial x_j} + \frac{1}{2} tr\left(\mathbf{R}(\mathbf{x})^{-1}\left(\tfrac{\partial \sigma_{i1}}{\partial x_i}\right)^T \mathbf{R}(\mathbf{x})^{-1} \tfrac{\partial \sigma_{i1}}{\partial x_j}\right), \tag{4.13}$$

with a parameter-dependent FIM Jacobian of

$$\frac{\partial \sigma_{i1}}{\partial x} = \frac{\sqrt{a}(\mathbf{r}_i + \mathbf{r}_1)}{r_0\sqrt{SNR_0(r_i^2 + r_1^2)}}. \tag{4.14}$$

Using the FIMs in (4.7) and (4.13), as substituted into (4.5), allows for the computation of the TDOA CRLB that encompasses inaccuracies due to array element localization error and local timing variances, respectively. Considering the LOCA emitter location technique, the analysis considers a combination of sensor triads instead of using a single reference or anchor sensor; thus, the covariance matrix in (4.4) evolves to the cases for considering the individual sensor triads. There is an expected processing gain as the number of sensors is increased since the result is dependent on the intersection of all the resulting planes. Our analysis is based on a least squares method to reduce the TDOA and LOCA results; however, many approaches exist to improve the TDOA accuracy through other processing methods or specialized geometries [82].

4.2 Analysis of Environmental Error Sources

The formulation and results of this section were published and presented at the 2025 IEEE Systems Conference [59].

In the previous section, we explored the impact to emitter localization accuracy from effects of sensor positioning errors and time-of-arrival measurement errors with two geolocation algorithms. We assumed the environmental effects, such as atmospheric refraction and multipath scattering, were negligible. However, it is necessary to understand the impact of the environment to the accuracy of the emitter localization, especially when operating in a dense urban environment. This investigation is particularly motivated by the fact that the UAS collection has the capability to operate in locations not accessible by conventional fixed-wing sensor array hosts including dense urban environments, building interiors, under tree canopies, etc. where effects due to multipath and the environment can be severe.

The assumption of constant signal propagation velocity neglects atmospheric refraction effects and small-scale fading due to Rayleigh scattering and the presence of multipath, both of which will affect location accuracies. Multipath is of particular concern when the sensor array is deployed within a dense urban environment that may consist of large metallic structures such as the steel frames of large buildings. The UAS-based wireless system considered here is assumed to be in the far-field but at relatively short operational ranges to the emitters, thus, the multipath effects are assumed to be significant contributors among this group. The incorporation of dynamically updated signal propagation speed and the use of models that predict multipath effects such as a Rician propagation model could be optionally included in the sensor array if sufficient WANET bandwidth and a reliable means for estimating the signal propagation characteristics were present. In this study, we assume that these resources are not present, so that the impact of multipath is realized in the localization process.

Because the received signal in urban environments is subject to multipath and atmospheric impacts, we are expanding on our previous research to include these effects in comparing the performance of geolocation algorithms for cooperative UAS systems. In this study, we simulate these environmental impacts using stochastically-derived urban propagation loss models, specifically those in ITU-R 1411-12 [83], that was revised just prior to the composition of this study. The goal in this research is to locate an emitter with minimal measurements and computations, and the current analysis seeks to identify and quantify the environmental error sources in emitter geolocation.

4.2.1 Path Loss Models

In a sensing scenario where the RF transmitting power of the emitter is unknown, a given sensor s_i of $i = 1, \ldots, N$ sensors will receive a power level that is the equivalent isotropically radiated power (EIRP) of the emitter reduced by the loss in that sensor $L_s i$ and environmental path loss from the emitter to the sensor. With an omnidirectional antenna, this is the received signal strength, RSS_i, and can be expressed as (4.15):

$$RSS_i = EIRP - L_{si} - L_P, \tag{4.15}$$

where L_P is the total environmental path loss from the emitter to the sensor. For the purposes of this effort, we assume that L_{si} is known and therefore can be ignored.

Determining L_P requires an understanding of the scattering mechanisms present in the propagation environment. These mechanisms begin with the free space path loss (L_{fsi}), and then include contributions from the moisture in the atmosphere (L_{atm}) such as rain or fog, and contributions from local scatterers and multipath fading such as the landscape, roads, buildings, etc (L_{ref}). The path loss is defined by the total contribution of each of these sources:

$$L_P\,(dB) \;=\; L_{fsi} + \; L_{ref} + \; L_{atm}, \tag{4.16}$$

where L_{fsi} is the free space path loss, L_{atm} is the path loss due to rain, fog, etc., and L_{ref} is the path loss from local scatterers including multipath contributors.

4.2.1.1 Free Space Path Loss

In a scattering-free environment, free space RF propagation L_{fsi} for sensor s_i follows a path loss defined by:

$$L_{fsi}\,(dB) = 20\log_{10}\left(\frac{4\pi R_i f}{c}\right), \tag{4.17}$$

where R_i is the distance from the emitter to sensor s_i, f is the transmitting frequency of the RF signal, and c is the speed of light.

For a given L_P the estimated distance $\hat{R}_i$ between a sensor and emitter pair could be found by transforming the free space equation into terms of R_i, and the signal transmit time across the path can be found by dividing $\hat{R}_i$ by c

$$\hat{t}_i = \frac{\hat{R}_i}{c} = \frac{1}{4\pi f} 10^{L_P/20}. \tag{4.18}$$

This is a limited approximation for our context, because in the presence of scatterers and real-world sensor dynamics, the magnitude of L_P will be greater than L_{fsi}, leading to inaccuracies in $\hat{R}_i$. Further, in our application, this requires exact knowledge of L_P, referenced from an emitter with unknown power, so excluding all other losses and system gains, a direct range measurement from L_P is not feasibly known.

4.2.1.2 Urban Environment Path Loss

In [84], path loss profiles are surveyed for a variety of UAS-to-emitter scenarios. In this text it was found for a complex environment with a number of scatterers, such as an urban area, a dual-slope path loss model provides a best fit. This type of model has a path loss profile that changes as R_i increases.

The International Telecommunication Union (ITU) Recommendation P.1411-12 [83] formulates a stochastic dual-slope path loss model of height-dependent scenarios for short range path loss in urban environments. In Section 4 of [83], the effects of streets and buildings are examined with a single-bounce geometry to provide expected path loss in both line-of-sight (LoS) and non-line-of-sight (NLoS) conditions between the transmitter and receiver. The model considers transmitters and receivers operating below- and above-building rooftops, assuming that building heights follow a Rayleigh distribution. Further, the model parameters of average building height, average street and building widths, and average number of buildings in given land area, can be modified for the type of physical operating environment: urban high-rise, urban low-rise, suburban, and residential. Finally, for each scenario, both a generalized statistical model and a site-specific model are provided for simulations utilizing 3-dimensional urban profiles.

The median loss for a given scenario is found by:

$$L_M\left(dB\right) = 10\alpha log_{10}\left(R_i\right) + \beta + 10\gamma log_{10}\left(f\right) + L_\sigma, \tag{4.19}$$

Table 4.1: Coefficients from the ITU-R P.1411-12 for calculating propagation loss for a variety of urban scenarios.

Frequency (GHz)	Range (m)	Environment	LoS / NLoS	Propagation height	α	β	γ	σ
0.8-8.2	5-660	Urban high-rise,Urban low-rise,Suburban	LoS	Below Rooftop	2.12	29.2	2.11	5.06
0.8-82	30-715	Urban high-rise	NLoS	Below Rooftop	4.00	10.2	2.36	7.60
0.8-73	30-170	Residential	NLoS	Below Rooftop	3.01	18.8	2.07	3.07
2.2-73	55-1200	Urban high-rise,Urban low-rise,Suburban	LoS	Above Rooftop	2.29	28.6	1.96	3.48
2.2-66.5	260-1200	Urban high-rise	NLoS	Above Rooftop	4.39	-6.27	2.30	6.89

where α is a coefficient associated with the increase of basic transmission loss with distance, β is a coefficient associated with the offset value of the basic transmission loss, γ is a coefficient associated with the increase of the basic transmission loss with frequency, and L_σ is a zero mean Gaussian random variable, $\mathcal{N}(0, \sigma^2)$, with standard deviation, $\sigma\,(dB)$, defined for specific scenarios. For NLoS urban environments, the model states that there will be an excess loss with respect to free-space that will not exceed

$$NLoS_{excess}\,(dB) = 10log_{10}\left(10^{0.1A} + 1\right), \tag{4.20}$$

where A is a Gaussian random variable, $N\left(\mu, \sigma^2\right)$, with σ defined above and mean given by:

$$\mu\,(dB) = L_M - L_{fsi}. \tag{4.21}$$

From [83], Table 4.1 provides the values of the above coefficients $(\alpha, \beta\ ,\ \gamma, \mu\ , \sigma)$ that are relevant to the simulations herein. For the propagation height, "Below Rooftop" is a scenario where both the transmitter and receiver are below the height of the roofs of the buildings in the range, whereas "Above Rooftop" refers to the scenario where either of the transmitter or receiver are located above the rooftop. In the "Above Rooftop" case, it is assumed the link can be described with L_{fsi}.

Because the model statistically formulates loss profiles, shadowing effects of buildings are incorporated into the coefficients such as L_σ, so the likelihood of shadowing and therefore the associated loss increases more than L_{fsi} with increasing R_i. The model claims validity for frequencies of 300 MHz to 100 GHz and the coefficients in Table 4.1 are valid for the ranges specified. However, to ensure that this model would be effective over the full ranges in this simulation, the coefficient values were extrapolated to distances approximately 15% outside the specified ranges.

For the analysis of environmental pathloss in this chapter, MATLAB is used to model the signal interactions; however, MATLAB does not offer ITU-R P.1411 in its model library, so the models were programmed with a custom implementation.

4.2.1.3 Atmospheric Path Loss

As a signal travels through the atmosphere, it is attenuated according to the geometry and density of moisture, such as rain or water vapor (fog), in its path. The ITU Recommendation P.838-3 [85] offers an empirically-derived signal attenuation model for determining the impact of rainfall on signal attenuation using a statistical rain rate to determine the signal loss at a specific frequency over a specified range. Further, the ITU Recommendation P.676-10 [86] considers ambient temperature, pressure, and atmospheric water vapor density to determine the signal loss at a specific frequency over a specified range. The MATLAB Communications Toolbox [87] provides, among other capabilities, implementations of the above water vapor model and the rain models, extended with the work in [88], for atmospheric path loss, and those models are used in the simulations in this study. In Fig. 4.2, the additional path loss in free space is shown using these atmospheric models. Modeling a "severe" condition for a 5 GHz signal, it can be seen that rain has the dominant effect, however, the total atmospheric loss is only approximately 0.5 dB at 1 km for a 5 GHz signal.

Other atmospheric models were not considered because they fell outside the parameters of this study. For example, models for fog are generally valid for frequencies greater than 10 GHz, while this analysis considers a 5 GHz signal with the assumption that the UAS array is likely to be processing 5G New Radio signals [89]. Further, ionospheric and tropospheric models consider a signal propagating in an altitude much greater than the 500 m height that is considered in this analysis.

4.2.2 Multipath Fading

The reflections of a signal against local scatterers will cause multipath fading depending on the geometry and characteristics of the environment. Two fading models, Rician and Rayleigh, are widely used to statistically model fading for LoS and NLoS conditions, respec-

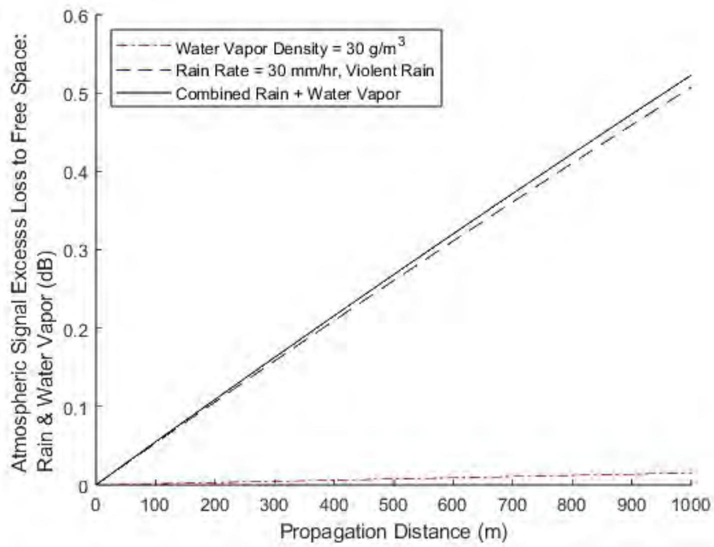

Figure 4.2: Impact of atmospheric attenuation at 5 GHz in ranges <1 km.

tively. Multipath fading is modeled according to the path loss profile of the propagation channel, the path delay between the primary and multipath components, and the Rician K-factor which is the ratio of the dominant channel component to the sum of all other components. A Rician distribution is usually appropriate to model the small-scale amplitude fading formed from an airborne UAS system to a stationary ground-based emitter [84].

To compute the multipath impact, the K-factor is environmental- and scenario-dependent. Values of 28 dB have been found to be appropriate for a UAS mission in an urban area [90], [91]. For this analysis, the MATLAB Communications Toolbox [87] was utilized to generate the multipath fading characteristics, using the empirical path delay spread and path loss models from [83].

Path delay spread impacts signal timing and is determined with scenario-specific coefficients. Section 5 of [83], provides the delay spread characteristics that were developed empirically at several frequencies and geometries. For the "Below Rooftop" scenario described previously, the delay spread impact in different geometries and environments is accounted for using Gaussian random variables, $N(\alpha_s, \sigma_s)$, with mean and standard deviation determined by coefficients C_α, γ_α, C_σ, and γ_σ defined for different scenarios of the model:

Table 4.2: Coefficients from the ITU-R P.1411-12 for calculating RMS delay spread for a variety of urban scenarios.

Frequency range (GHz)	Distance range (m)	Urban environment	A	B	C_α	γ_α	C_σ	γ_σ
3.35-15.75	50-400	Below Roof-top	-	-	23	0.26	5.5	0.35
3.65-3.75	100-1000	Over Roof-top	0.031	2.091	-	-	-	-
1.92-1.98, 2.11-2.17	100-1000	Over Roof-top	0.038	2.3	-	-	-	-

$$\alpha_s\left(ns\right) = C_\alpha {R_i}^{\gamma_\alpha} \tag{4.22}$$

$$\sigma_s\left(ns\right) = C_\sigma {R_i}^{\gamma_\sigma}. \tag{4.23}$$

For the "Above Rooftop" scenario, the median "RMS Delay Spread" is determined with scenario-specific coefficients by:

$$S_u\left(ns\right) = e^{(A \bullet L_M + B)}. \tag{4.24}$$

The relevant coefficients for the delay spread characteristics are summarized in Table 4.2. In the case where the distances or the frequencies did not correspond to the ones used in this simulation, the coefficients were extrapolated based on the parameters provided in [83].

A representation of the implementation of these delay spread profiles over a fixed distance is shown in the following figures. In Fig. 4.3, it is seen that the mean delay spread for the Below Rooftops condition exceeds the delay spread for the Above Rooftops condition, but due to the characteristics of the scattering environment, the difference in the delay spread could vary significantly.

In Fig. 4.4, the mean delay spreads for the Above and Below Rooftop conditions are compared to the signal delay value determined by (4.18). The distance from the emitter to the sensor is increased to 1 km, and the path loss is found both by the free space path loss and L_b determined by [83]. By using (4.18), the median signal strength in an urban environment could be perceived to be from an emitter that is much further away than the true position.

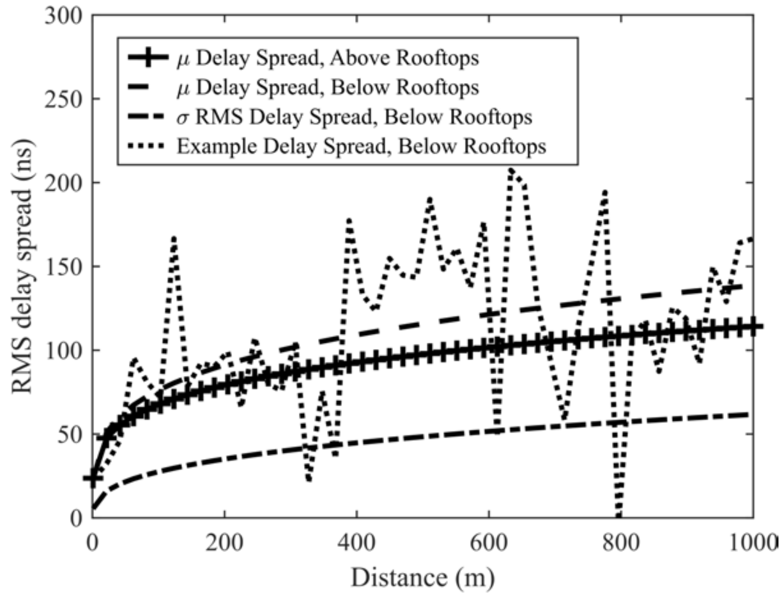

Figure 4.3: Comparison of delay spread profiles in height-specific geometries.

Therefore, the use of the time delay of the signal to sensors in various positions could resolve this uncertainty and potentially inform characteristics of the scattering environment.

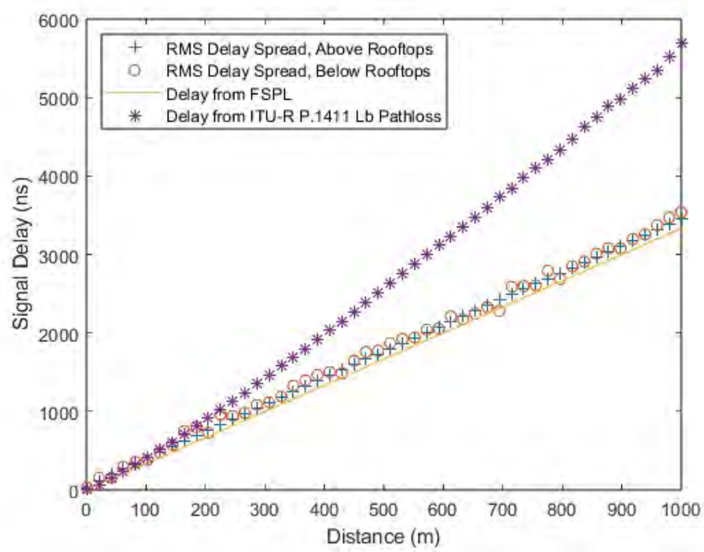

Figure 4.4: Comparison of Delay Spread against Path Loss Delay.

4.3 Accumulated Error Models for Sensor Measurements

4.3.1 Range-like Measurement Model Weighting

To remain compatible with timing-based observables while keeping the KF measurement fusion step lightweight, we treat each sensor's time-of-arrival estimate as a range-like measurement for weighting and consistency checks. In the simulation, TOA samples are generated with a common transmit-time origin. In a real system with unknown transmit time, the same weighting can be formed using differences relative to a reference sensor.

For sensor $i \in \{1, \ldots, N\}$ at snapshot k, define the debiased range-like measurement

$$r_{k,i} \triangleq c\,\widehat{t}_{k,i} - b_{\mathrm{env}}(R_i^0), \qquad R_i^0 \triangleq \|\mathbf{s}_i - \widehat{\mathbf{e}}_{k,\mathrm{med}}\|, \tag{4.25}$$

where $b_{\mathrm{env}}(R)$ is the modeled mean excess-delay bias (Sec. 4.3.3) evaluated at the nominal link range computed at the cloud center $\widehat{\mathbf{e}}_{k,\mathrm{med}}$.

The corresponding variance used for inverse-variance weighting is modeled as

$$\sigma_{r,k,i}^2 = c^2\sigma_{t,k,i}^2 + \sigma_{\mathrm{env}}^2(R_i^0) + \sigma_{\mathrm{pos}}^2(\mathbf{s}_i, \widehat{\mathbf{e}}_{k,\mathrm{med}}), \tag{4.26}$$

where $\sigma_{t,k,i}$ represents timing uncertainty, $\sigma^2_{\text{env}}(\cdot)$ represents environment-induced excess-delay variance in (4.31), and $\sigma^2_{\text{pos}}(\cdot)$ represents the sensor-position-induced range variance (approximated using an isotropic per-sensor position error model).

4.3.2 Linearized Information Covariance Floor for Fusion

Measurement covariances can become overconfident under poor geometry or strong multipath. To prevent unrealistically small R_k, we compute a linearized information floor using the range-like measurement model evaluated at the fused centroid z_k.

Define the range measurement function

$$g_i(\mathbf{e}) \triangleq \|\mathbf{s}_i - \mathbf{e}\|, \qquad i = 1, \ldots, N.$$

The Jacobian row at z_k is

$$\mathbf{h}_{k,i}^{\mathsf{T}} = \left.\frac{\partial g_i(\mathbf{e})}{\partial \mathbf{e}}\right|_{\mathbf{e}=z_k} = -\frac{(\mathbf{s}_i - z_k)^{\mathsf{T}}}{\|\mathbf{s}_i - z_k\|}, \qquad i = 1, \ldots, N, \tag{4.27}$$

and the stacked sensitivity matrix is $H_k \in \mathbb{R}^{N\times 3}$,

$$H_k = \begin{bmatrix} \mathbf{h}_{k,1}^{\mathsf{T}} \\ \vdots \\ \mathbf{h}_{k,N}^{\mathsf{T}} \end{bmatrix}.$$

Using the modeled variances from (4.26), form the diagonal covariance $R_{r,k} \in \mathbb{R}^{N\times N}$:

$$R_{r,k} = \operatorname{diag}\left(\sigma^2_{r,k,1}, \ldots, \sigma^2_{r,k,N}\right).$$

The local information-like matrix and its inverse define a geometry- and noise-dependent covariance scale:

$$F_k = H_k^{\mathsf{T}} R_{r,k}^{-1} H_k + \epsilon I_3, \qquad P_k^{(\text{lin})} = F_k^{-1}, \tag{4.28}$$

where $\epsilon > 0$ prevents numerical singularity.

We enforce a conservative measurement covariance floor,

$$R_k \succeq P_k^{(\mathrm{lin})}, \tag{4.29}$$

by applying diagonal clamping so that the stabilized R_k is not more optimistic than the linearized information floor. This provides a simple safeguard against numerical overconfidence when the cloud covariance collapses due to poor geometry or transient multipath [92].

4.3.3 Environment-Induced Excess-Delay Statistics for Fusion

The fusion and covariance-weighting steps require a bias and variance model for environment-induced delay. We represent the excess propagation delay on link i (relative to free-space) as a random variable $\tau_{\mathrm{env},i}$ whose statistics depend on the operating scenario (LoS/NLoS, rooftop condition) and on the link geometry.

Using the ITU-R P.1411 path loss and RMS delay spread models, we generate Monte Carlo realizations of the excess delay for each link over the range of distances used in simulation. For a given nominal link range $R_i^0 = \|\mathbf{s}_i - \widehat{\mathbf{e}}_{k,\mathrm{med}}\|$, we estimate the mean and variance of the excess delay,

$$\mu_\tau(R_i^0) \triangleq \mathrm{E}\left[\tau_{\mathrm{env},i} \mid R_i^0\right], \qquad \sigma_\tau^2(R_i^0) \triangleq \mathrm{Var}\left[\tau_{\mathrm{env},i} \mid R_i^0\right], \tag{4.30}$$

where $\sigma_\tau(R)$ is driven by the scenario-dependent RMS delay spread relationships in (4.22)–(4.24).

We map these statistics into the range domain via multiplication by the propagation speed c:

$$b_{\mathrm{env}}(R_i^0) \triangleq c\,\mu_\tau(R_i^0), \qquad \sigma_{\mathrm{env}}^2(R_i^0) \triangleq c^2\,\sigma_\tau^2(R_i^0). \tag{4.31}$$

In practice we evaluate these terms at the nominal ranges computed from the cloud center $\widehat{\mathbf{e}}_{k,\mathrm{med}}$ and treat them as known model inputs during fusion and filtering.

4.4 Error Simulations

The evaluations in this section were published and presented at the 2023 IEEE Aerospace Conference [11] and at the 2025 IEEE Systems Conference [59].

4.4.1 Simulation Approach

For this analysis, the emitter-sensor system was modeled in MATLAB, where sensor networks of fixed-location UAS are used to determine the location of a small, stationary emitter transmitting a 5 GHz signal from an omnidirectional antenna with known signal characteristics. The sensor and emitter locations were randomized, following a uniform distribution, in a simulated urban environment that is approximately 500 m wide, long, and high, with the sensors within the far-field of the emitter at an operational range ∼200 m from the emitter. We assume all sensors receive the emitter signal with sufficient SNR, and the simulations varied both the number of sensors and the locations of the sensors and emitter.

The UAS system was modeled with a variety of sensor quantities, with at least five sensors to resolve ambiguities in three dimensions, and each UAS followed the architecture in Fig. 1.1, including an omnidirectional antenna and parameter sharing capability across an ad hoc network. Each sensor utilizes a LiDAR-based UAS self-positioning system that has an accuracy of 2 cm, and a local clock source with a baseline accuracy of 10 ns (chosen conservatively) that is varied in these trials. The local UAS clocks are assumed synchronized prior to the mission and the UAS systems include a local sensor to determine their position with respect to the other UAS sensors.

The self-positioning accuracy is supported by products such as Texas Instruments compact LiDAR [71] permitting a ranging accuracy of 1 cm at 100 m, which is better than GPS [72]. Likewise, the 10 ns local timing source error is supported by small portable Chip Scale Atomic Clocks (CSAC) that provide a basis for high resolution local timing sources with low drift and jitter rates [9, 10] with $\leq$ 1 ns accuracy. Even without the presence of timing source synchronization methods, the use of a CSAC-based local timing source allows for a low-latency free-running source that is synchronized at pre-mission time only, and it is therefore a reasonable approach to enable accurate emitter location estimates over limited mission durations. Localization and timing errors are incorporated in a manner consistent with the assumptions that result in (4.1) and (4.2).

For the environmental error analysis, the simulations were modeled for both an urban low-rise and an urban high-rise scenario, and the path loss profiles were modeled in MATLAB using the methodology in ITU-R P.1411 [83]. Because the localization algorithms are time-based, the TOA of the signal to each sensor was augmented by ITU-R P.1411 models for estimated path loss and RMS delay spread. For atmospheric scatterers, the Crane [88] model assumed a maximum rain rate of 30 mm/hr and a maximum water vapor of 30 g/m^3, each of which can be considered "worst-case" weather conditions. We use Monte Carlo simulations to characterize the impact of these errors on the performance of the algorithm. The emitter location was determined using the LOCA algorithm for each Monte Carlo result.

The corresponding error parameter distributions used to augment the measurements are summarized in Table 4.3.

Table 4.3: Error parameters for localization model.

Error Parameter	**Error Distribution**
Self-positioning error (m)	$\varepsilon_U \sim \mathcal{N}(0,\, 0.02)$
Local clock inaccuracy (ns)	$\varepsilon_c \sim \mathcal{N}(0,\, \sigma_t + 10)$
Multipath delay spread (ns)	$\varepsilon_m \sim 0.5{*}\mathcal{N}\big(23{R_i}^{0.26},\, 5.5{R_i}^{0.35}\big)$, (Table 4.2)
Time of arrival error (ns)	$\varepsilon_{t,i} = \varepsilon_m + \varepsilon_c$
Scattering loss (dB)	$\varepsilon_P \sim \mathcal{N}(3,\, 5)$

These parameter distributions are used throughout the simulation studies and controller evaluations in Chapter 5.

In these simulations, time-difference measurements (TDOA) are the primary observables used by both multilateration and LOCA to estimate the emitter location, utilizing the algorithms in (2.4) and (3.2).

For each Monte Carlo trial n, we define the localization error magnitude as the Euclidean distance $d_n = \|\hat{\mathbf{e}}_n - \mathbf{e}_n\|_2$ (m). Across a set of M trials, we report either (i) the mean Euclidean error $\mu_d = \frac{1}{M}\sum_{n=1}^{M} d_n$ or (ii) the RMS Euclidean error (RMSE) $\mathrm{RMSE}_d = \sqrt{\frac{1}{M}\sum_{n=1}^{M} d_n^2}$. Because $d_n \geq 0$, $\mathrm{RMSE}_d \geq \mu_d$, with RMSE more sensitive to occasional large-error trials. These simulation results are then compared to the CRLB results to obtain an estimate of overall performance.

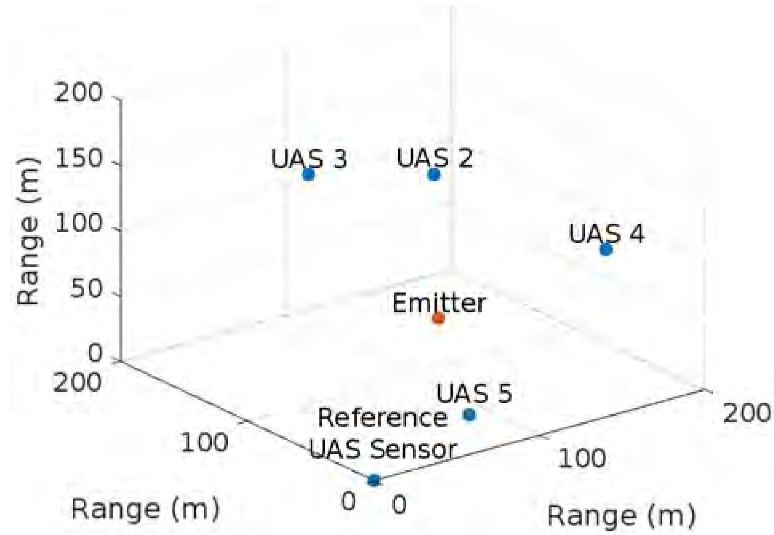

Figure 4.5: Example sensor geometry for five UAS sensors.

For TDOA-related analyses, the reference or anchor sensor is always placed at the origin, as shown in Fig. 4.5 for a geometry of five sensors.

While a primary motivation for deploying a cooperative and wireless sensor array to locate emitters is due to the ability to dynamically reconfigure the array geometry and to improve emitter location accuracy through subsequent measurements, we choose to consider the case where only a single emitter location estimate is performed at each simulated time instance to capture worst-case behavior. Iteratively updating the emitter location estimate by including subsequent measurements will greatly improve system accuracy, particularly when new geometries are used that enhance the new field-of-view (FOV) in relation to the previous FOV.

4.4.2 Simulation Results

4.4.2.1 Error-Free Performance Comparison of Multilateration and LOCA Approaches

For an analysis of error-free performance, the true location of the emitter is randomly selected for 100 unique geometries. The number of receivers varies from 5 to 25 to show the improvement in emitter location with increasing array size. We use Monte Carlo simulations with 10,000 runs (i.e., 100 emitter geometries with 100 sensor geometries each) for our analy-

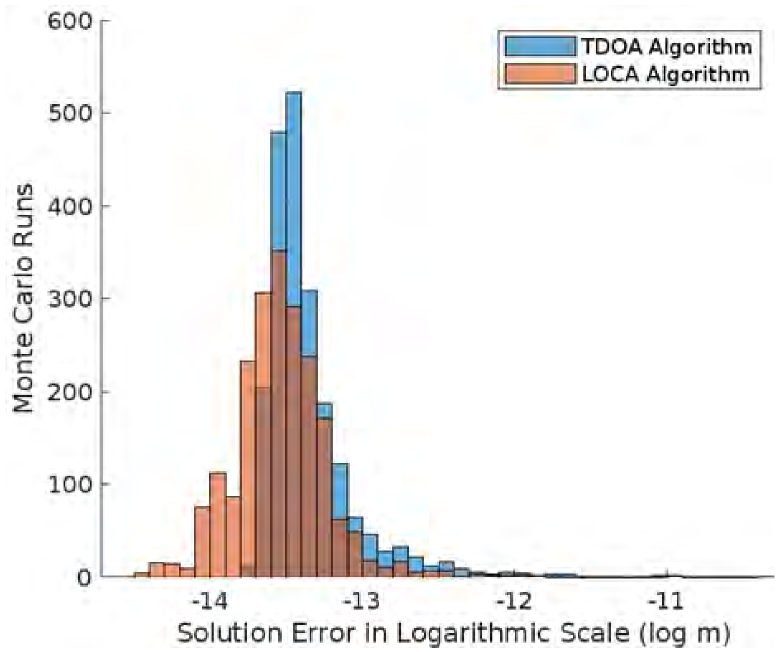

Figure 4.6: Histogram of RMS error of TDOA and LOCA algorithms with no induced error.

sis. The same geometry, both in terms of quantities and locations, of each run was processed with both the LOCA algorithm and a nominal TDOA (multilateration) algorithm to avoid geometry bias in the comparisons of the two emitter location techniques. The distribution of the localization performance for the error-free solutions for each of the TDOA and LOCA algorithms is provided in Fig. 4.6 and, as is expected, Fig. 4.7 indicates that the error values improve in a roughly logarithmic manner with increasing sensor quantities. Although it is certainly dependent upon the specific mission objectives, it is generally observed that array sizes of five to ten sensors should be sufficient for most applications.

4.4.2.2 Performance in the Presence of Local Sensor Timing and Positioning Errors

In the following simulations, sensor hardware errors are induced to quantify their impacts on localization performance in terms of RMS Euclidean error across trials. The sensor position inaccuracy is varied versus increasing sensor array size using the signal parameters discussed in 4.4. Fig. 4.8 illustrates the impact of an increasing sensor positioning error and shows how the error of the emitter location estimate increases as sensor positioning error

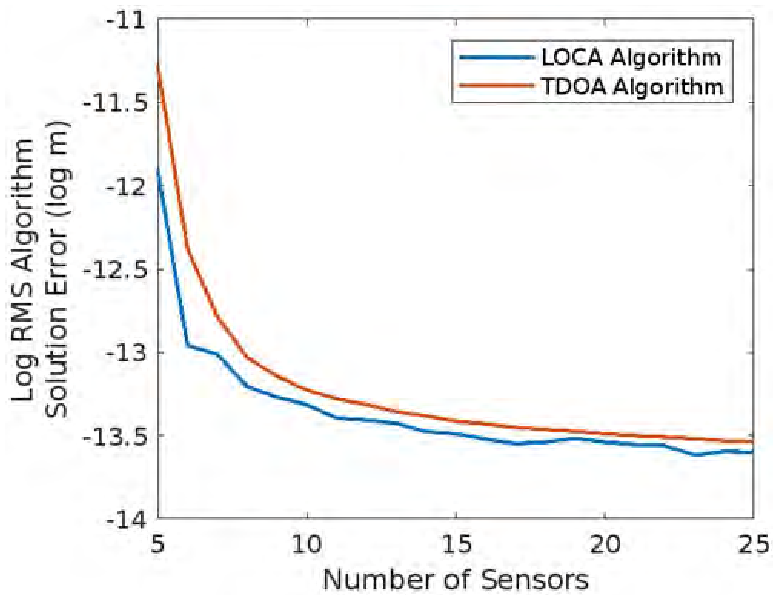

Figure 4.7: RMS error of TDOA and LOCA algorithms with no induced error for increasing sensor quantities.

increases.

If the sensors are localized with an intra-array method such as LiDAR, then the localization error has an upper-bound to the accuracy from that sensor positioning method, which is 2 cm in this case. The availability of practical LiDAR ranging subsystems within individual sensor elements greatly impacts the viability and practicality of using a collection of UAS to serve as a cooperative wireless sensor array for the purpose of estimating an RF emitter location since it provides an upper bound on array element localization error. Fig. 4.9 contains simulation results in the form of RMS emitter localization error due to sensor positioning errors, on a logarithmic scale, versus varying numbers of sensor array elements for a given array geometry.

For the multilateration technique (referred to as "TDOA Location" in Fig. 4.9), the overall absolute error in emitter location range is much higher than the CRLB, while the same error for LOCA is much closer to the CRLB. As expected, the impact of sensor positioning decreased with more sensors and as also expected, the RMS emitter location range error varies and depends upon array geometry. Although a variety of different randomly-

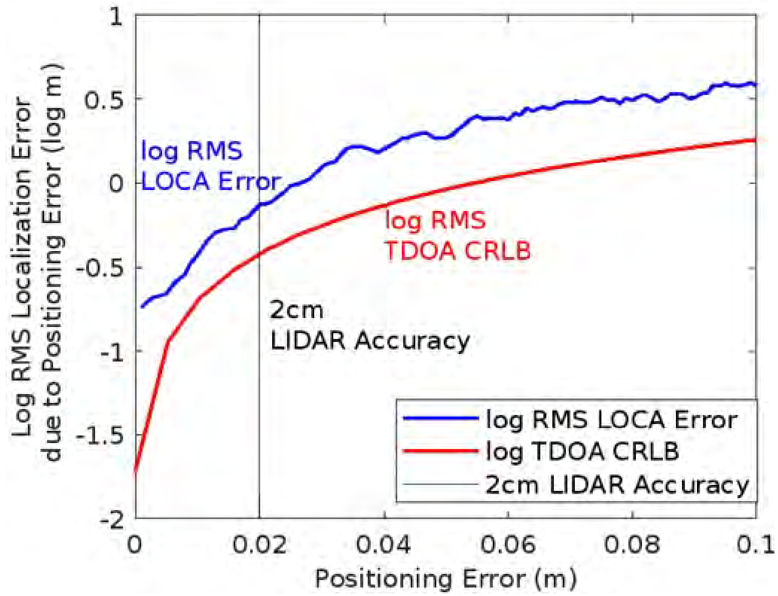

Figure 4.8: Log RMS localization error of LOCA algorithm with increasing sensor positioning error, upper-bounded by in-network sensor positioning accuracy.

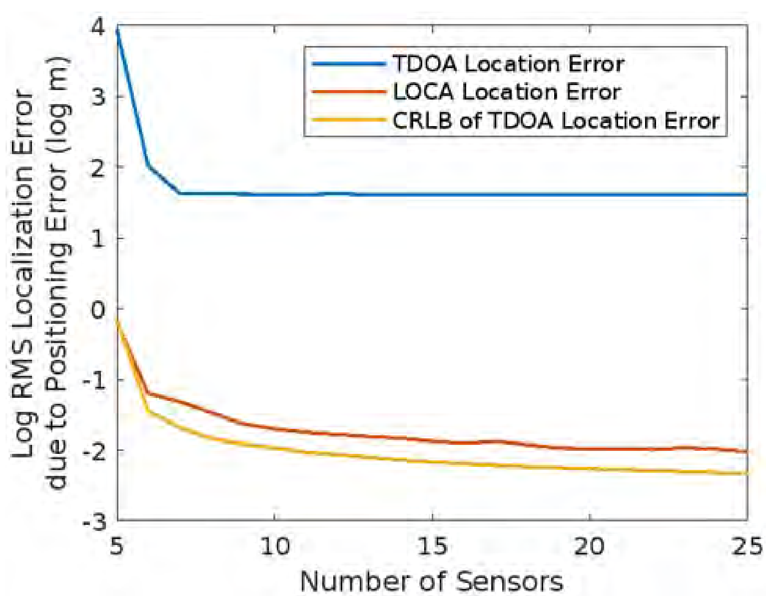

Figure 4.9: RMS error of TDOA and LOCA algorithms with induced sensor positioning error for increasing sensor quantities.

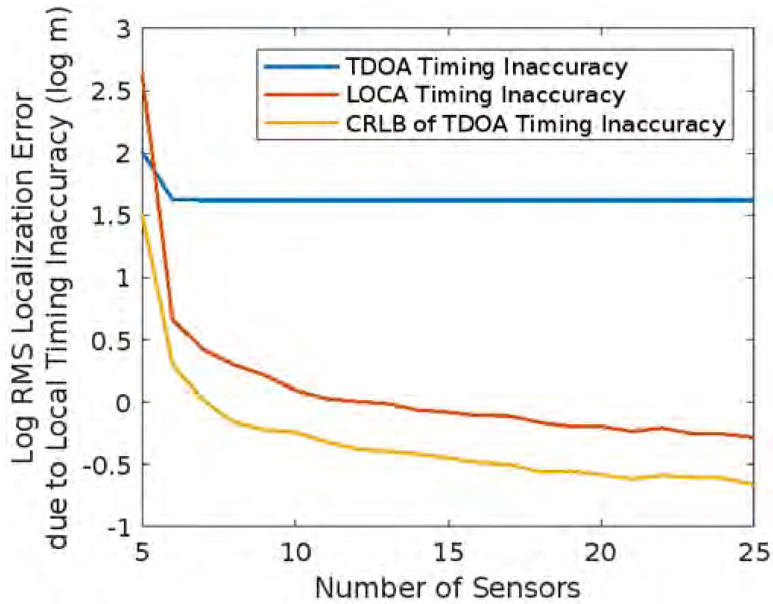

Figure 4.10: RMS error of TDOA and LOCA algorithms with induced local timing inaccuracy for increasing sensor quantities.

selected array geometries are analyzed, only a single representative example is included in this chapter. There is a noticeable accuracy improvement for LOCA as the number of sensors increases, whereas for the multilateration, or TDOA algorithm, the induced errors diminish the processing gains found with increasing sensor quantities when the array size exceeds eight sensors.

The local sensor timing error is also simulated for increasing numbers of array elements using the same geometries and signal parameters as discussed in 4.4. Fig. 4.10 contains RMS emitter location error, on a logarithmic scale, due to local sensor timing offsets versus an increasing sensor array size.

Consistent with the results due to sensor localization inaccuracies, the multilateration, or TDOA, error is significantly higher than the CRLB, whereas the LOCA error is significantly closer to the CRLB. Again, there is an accuracy improvement with LOCA as the number of sensors increases, whereas the TDOA error remains nearly constant. Therefore, we reach the same conclusion that, although highly dependent upon mission parameters, an array size in excess of eight sensors provides largely diminishing increases in emitter location accuracy.

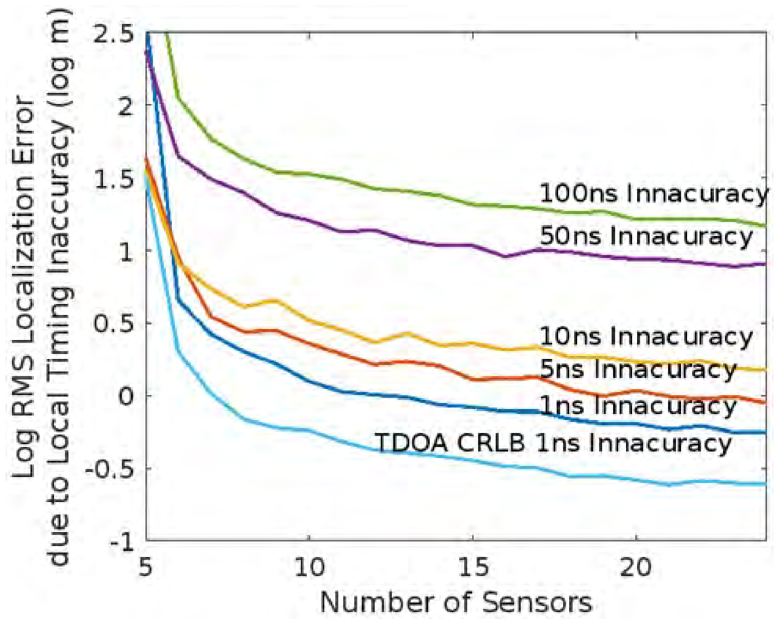

Figure 4.11: RMS error of LOCA algorithms with varying induced local timing inaccuracy for increasing sensor quantities.

We sweep values of local sensor element timing inaccuracies from 1 ns to 100 ns and show their impact on emitter location accuracy using the LOCA method, with simulation results shown in Fig. 4.11.

The cooperative use of increasing numbers of sensors in the UAS network can overcome the local timing inaccuracies to a certain extent if the previously assumed 1 ns bound on timing errors is not achievable in a geometry. As shown in Fig. 4.1, as the overall mean range of the UAS array position to that of the true emitter location decreases, the received signal SNR_i at each sensor increases. If available, the utilization of array clock synchronization methods decreases the emitter location error, and the emitter location resolution will increase. In this case, since local timing source synchronization is likely an expensive operation, a larger array size with less accurate timing sources can be used to obtain an initial emitter location estimate. Next, a timing source synchronization operation among a smaller subset of array elements can be used to obtain a subsequent emitter location estimate that is more accurate and that could potentially be acquired in a closer range to the emitter position. Furthermore, the initial emitter location estimate could be used to select the subset of array elements to be synchronized, a new geometry for those elements, and even the choice of an alternative emitter location algorithm to be used for subsequent measurements.

4.4.2.3 Performance in the Presence of Environmental Errors

The study of induced error is extended to external environment impacts including atmospheric scattering and multipath from an urban environment. Environment-induced excess delay (modeled using ITU-R P.1411 [83] RMS delay spread and associated statistics) can be treated as an equivalent timing error, which allows the environmental error contributions to be compared against the time-based Cramér–Rao Lower Bound (CRLB) and the individual hardware error sources, that is, the timing inaccuracies and positioning error, reported in [11].

The mean localization error was evaluated for multiple sensor quantities over 1000 randomized sensor/emitter geometries in urban high-rise and urban low-rise environments and with atmospheric scattering. For each geometry, we performed 500 Monte Carlo channel realizations (per ITU-R P.1411 excess delay [83]) to capture varying environmental performance, and computed the mean error over those trials. As shown in Fig. 4.12, atmospheric scattering alone produces substantially lower localization error than urban multipath and is on the same order as a system with approximately 1 ns local timing inaccuracy. In an urban low-rise multipath environment, the system appears to converge to similar accuracy as a system with 10 ns timing inaccuracy, while the urban high-rise multipath case yields higher error than the low-rise scenario but still performs better than that of a system with 50 ns timing inaccuracy. From this initial analysis, Fig. 4.12 also indicates that increasing the number of sensors significantly reduces the impact of environmental effects, but that the accuracy gains diminish beyond nine sensors. This trend is consistent with [11] (see Section 4.4.2.2), where increasing array size under local clock inaccuracy and sensor positioning errors provided limited additional benefit beyond approximately eight sensors; therefore, the remainder of this study considers a system size of no more than nine sensors.

4.4.2.4 Accumulation of Errors

We accumulate the impact of environmental scatterers with the hardware error models from [11] in our performance evaluation of the cooperative UAS. For TDOA localization

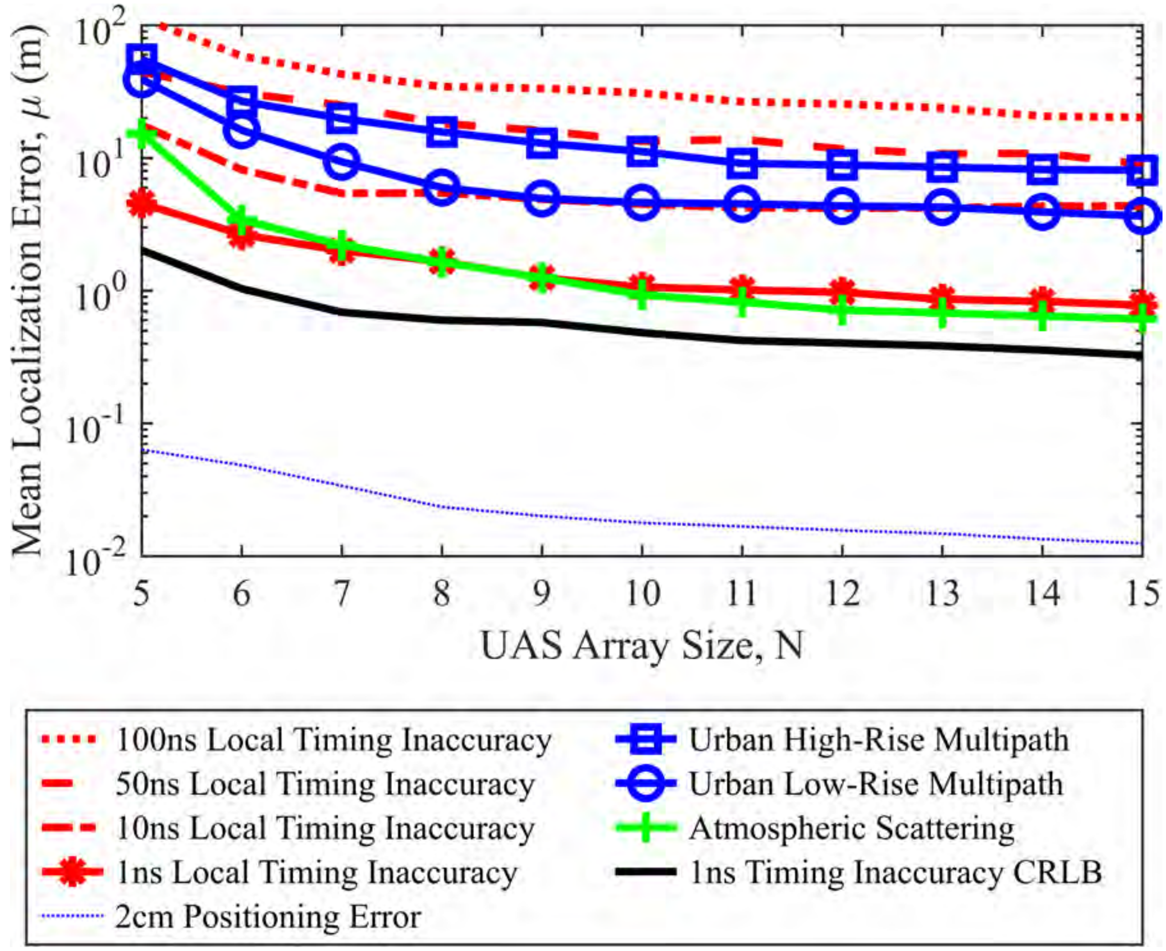

Figure 4.12: Localization Error due to hardware errors and environmental scatterers with up to 15 Sensors.

algorithms, it is necessary to quantify the error contribution from UAS sensor position estimates and clock synchronization offsets that could cause TDOA errors when correlating a time-stamped signal.

We model the effective TOA at sensor i as a sum of nominal free-space TOA and additive error processes:

$$\widehat{t_i} = t_i + \epsilon_{t,i}^{(\mathrm{clk})} + \tau_{\mathrm{atm},i} + \tau_{\mathrm{mp},i}, \tag{4.32}$$

where $\epsilon_{t,i}^{(\mathrm{clk})}$ captures local clock error, and $\tau_{\mathrm{atm},i}$ and $\tau_{\mathrm{mp},i}$ capture atmosphere-induced and multipath-induced excess delay, respectively. In the range domain,

$$\widehat{r_i} = c\,\widehat{t_i}. \tag{4.33}$$

Accordingly, the range-difference measurement (relative to reference sensor 1) becomes

$$\Delta r_{i1} = (r_i - r_1) + c\Big[(\epsilon_{t,i}^{(\mathrm{clk})} - \epsilon_{t,1}^{(\mathrm{clk})}) + (\tau_{\mathrm{atm},i} - \tau_{\mathrm{atm},1}) + (\tau_{\mathrm{mp},i} - \tau_{\mathrm{mp},1})\Big]. \tag{4.34}$$

Sensor position error is modeled by $\widehat{\mathbf{s}}_i = \mathbf{s}_i + \boldsymbol{\epsilon}_{s,i}$ as in (4.1).

With the hardware and environmental error contributions formulated and with the theoretical lower bound assessed, we use this information to quantify the impact of these errors in emitter localization accuracy.

In Fig. 4.13, we accumulate the environmental and hardware errors for both urban multipath settings and present the localization error distribution of per-trial Euclidean errors for random emitter positions and random initial, i.e., static, sensor geometries over different UAS array sizes N. For both urban settings, the mean (μ) and median (med) localization error improve as N, with the most significant improvement occurring from $N = 5$ to $N = 6$. In an urban setting, and especially in a high-rise environment, the inclusion of a sixth sensor significantly mitigates the errors introduced by the scatterers. In addition to the improvement in mean localization accuracy, the distributions also show an increasing density of high-accuracy outcomes as sensor quantity grows, reflecting the combinatorial advantages of the LOCA algorithm.

While the above analysis was a 500 m volume with sensors within ∼250 m of the emitter,

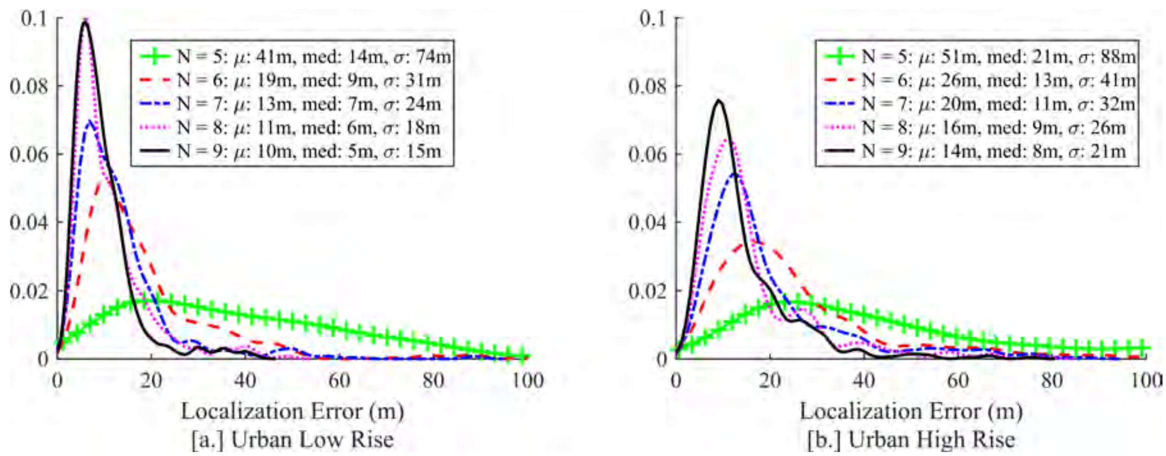

Figure 4.13: Distribution of localization error with accumulated hardware and environmental errors and increasing sensor quantity N in [a.] urban low-rise and [b.] urban high-rise settings.

even in this relatively short range, the performance suffers. For the remainder of this analysis, we expand the range to 2000 m horizontally and maintain 500 m vertically. In this longer range, the performance degrades significantly, as shown in Fig. 4.14, from increased multipath scattering and longer signal transit times. We are thereby motivated to develop a capability to improve the localization accuracy.

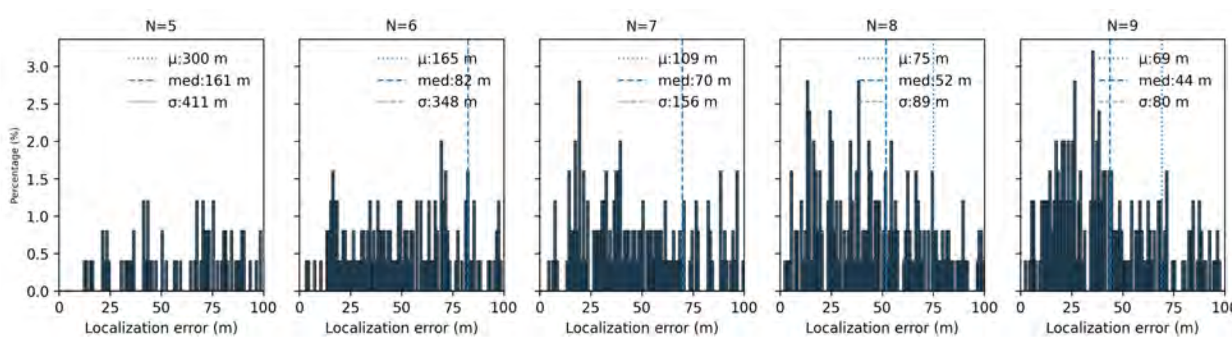

Figure 4.14: Distribution of localization error with accumulated hardware and environmental errors and increasing sensor quantity N in large dense multipath environment.

Chapter 5

OPTIMIZING EMITTER LOCALIZATION ACCURACY WITH UAS GEOMETRY CONTROLLER

5.1 Motivation

The previous chapters established the LOCA localization pipeline for cooperative UAS arrays and quantified the dominant hardware and environmental error sources that limit practical performance in dense urban settings. A key conclusion from the accumulated error analysis is that the array geometry is not a static design choice: in a multipath-dominated environment, adaptive changes in relative sensor placement can shift the estimator from an ill-conditioned regime (wide solution clouds and large SEP_{50}) to a well-conditioned regime where redundant triads reinforce a consistent solution.

Accordingly, this chapter focuses on geometry-adaptive localization, where the UAS network repositions between localization snapshots to reduce uncertainty and drive the solution cloud toward a compact, consistent region with minimal ambiguities. In considering an ideal geometry, we inform the new sensor geometry by exploiting the conic and quadric theory inherent in LOCA. Since the LOCA algorithm views the sensor-emitter geometry as an analysis of a conic section in 2D and a quadric in 3D, we consider the focal properties of the conic and quadric. In 2D, both a circle and a parabola contain one foci, and their corresponding quadrics, the sphere and paraboloid, also contain one foci. Therefore, a geometry that places the sensors along the surface of a sphere or paraboloid with the emitter at the focus could generate an ideal geometry for LOCA, since this geometry will result in exactly one focal point.

Further, and typically for geolocation algorithms [61], the circular and spherical error probable (CEP, SEP) is utilized as a summary figure of merit to define the likelihood that a calculated emitter geolocation is equivalent to the true geolocation in the presence of errors.

We consider the target CEP and SEP as a guiding factor to how tolerable is the system for a growing magnitude of error. To take advantage of the combinatorial aspects of multiple sensors, we determine the SEP by utilizing the LOCA algorithm for subsets of the UAS up to the full UAS array. As the emitter estimates are determined, a corresponding SEP is generated that provides a confidence region for the emitter location. By using both the full UAS array and subsets of the UAS for different localization estimates, we realize the combinatorial advantages of the distributed UAS as provided by LOCA.

Additionally, as the sensors converge on the emitter, the RSS will increase at each sensor. The combination of the improved RSS, the emitter location SEP, and the idealized geometry introduces a novelty in our method and differentiates our approach from previous work that focuses solely on increasing RSS. Weighting both the RSS and the SEP emitter location estimate as part of the ML objective function augments the maximum reward, and we study both this approach with the idealized geometry determination and compare it to the RSS-only approach to evaluate the benefits to the overall system "time-to-target convergence efficiency".

5.1.1 Repositioning Concept

To conceptualize the advantages of repositioning, in Figure 5.1, we illustrate an initial repositioning scheme in that utilizes both cooperative UAS emitter localization estimates and individual UAS element emitter received signal strength (RSS). This scheme was analyzed, in a simulated small environment for 1000 trials of random sensor and emitter positions, to show the initial effectiveness of UAS repositioning on improving the localization perforamnce in a dense scattering environment. Once the emitter location estimate was initially determined, a rudimentary repositioning approach was applied to the sensor locations. In this approach, the three sensors with the simulated highest-magnitude emitter RSS were relocated to a position 50% closer to the LOCA-estimated emitter position. The simulation repeated the approach with the new sensor locations to iteratively reposition the sensors, continuing to choose the sensors with the highest magnitude RSS, until three repositioning iterations were performed. The results shown in Fig. 5.1 are in terms of the mean accumulated localization error for the iterative searches with 1, 10, and 100 ns timing inaccuracy for both the urban low-rise and

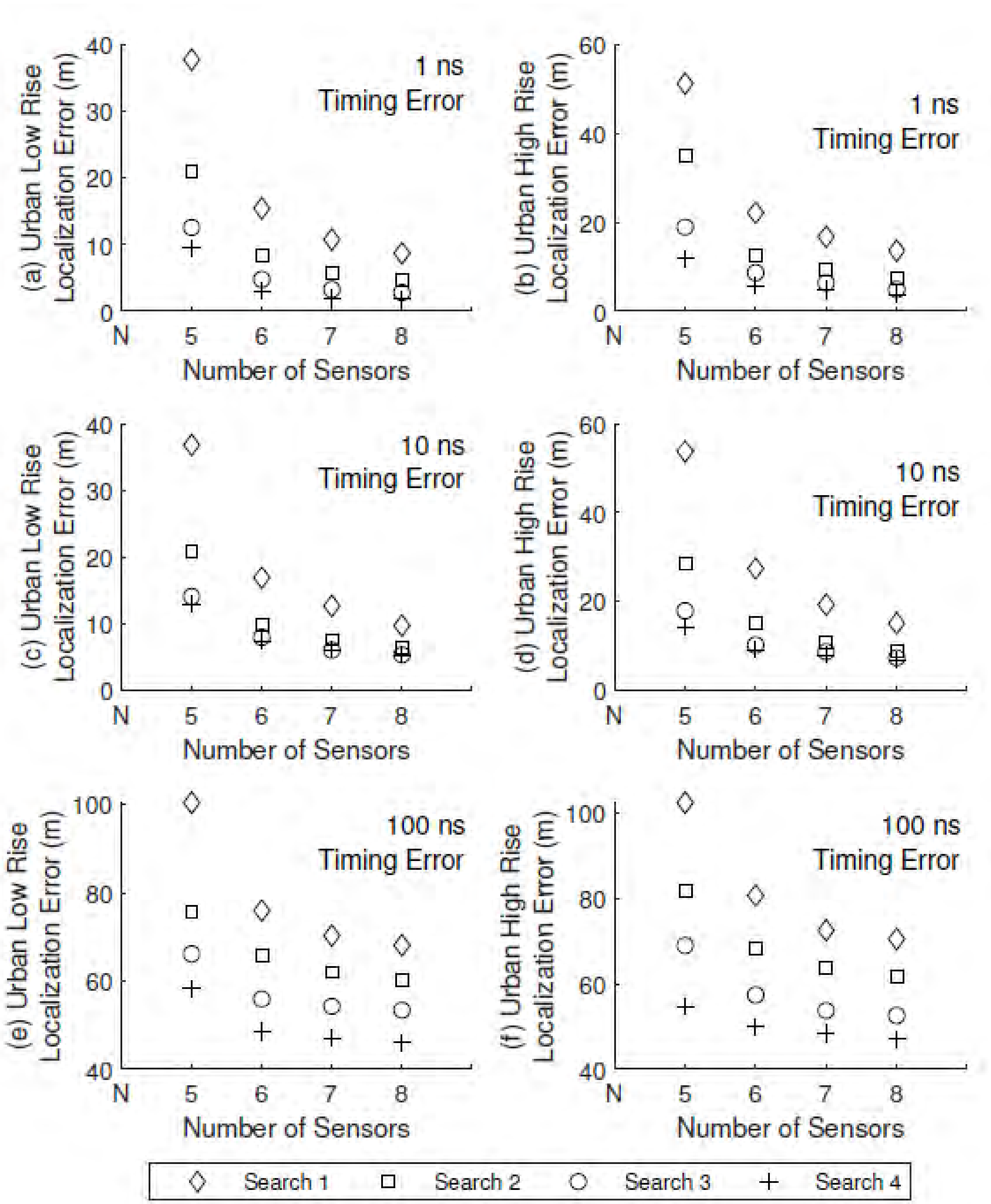

Figure 5.1: Mean accumulated localization error in iterative search events of a stationary 5 GHz emitter for varying UAS array size with [a., b.] 1 ns local timing inaccuracy, [c., d.] 10 ns local timing inaccuracy, and [e., f.] 100 ns local timing inaccuracy in [a., b., c.] urban low-rise and [d., e., f.] urban high-rise multipath environments.

urban high-rise environments and for five to eight sensors. As the search algorithm adaptively repositions the sensor array informed by the emitter location estimate and highest RSS, the accumulated localization error decreases for all sensor and environmental cases. Each iteration shows approximately 40% improvement in the localization error for 1 ns timing inaccuracy. Similar analysis for the 10 ns case shows 28% improvement, while 100 ns sees 16% improvement. Thus, an informed and adaptive UAS repositioning scheme significantly improves localization accuracy.

5.1.2 Autonomy Considerations

We consider that in a real mission, the UAS array is fully autonomous. Autonomy is defined using the NIST Levels of Autonomy [93], also known as the Autonomy Levels for Unmanned Systems (ALFUS). ALFUS considers three metrics to determining the autonomy of a given system: the complexity of the mission, the complexity of the environment, and the level of operator independence. Each of these metrics are scored to determine an overall autonomy level. The assumption in the autonomy of the system presented here is that the path planning decisions are both guided by obstacle avoidance and by the outcome of the geometry optimization algorithm. Without optimizing the geometry through dynamic reconfiguration, the UAS is highly dependent on the operator to determine where to perform the searches, which minimizes the effectiveness of the mission and limits the adaptation of the UAS. Further, the system is susceptible to decisions that are influenced by multipath errors in the environment which provides a high solution/possibility ratio to the emitter location. Utilizing the guidance in the ALFUS framework, without dynamic reconfiguration, the system would likely score a 5 of 10 for levels of autonomy. However, utilizing a dynamic reconfiguration approach would improve all three metrics. First, the human reliance is mitigated with the ability to automatically determine subsequent geometries, so that an operator can now provide strategic goals to the UAS instead of individual waypoints. While the conditions of the environment do not change, the UAS is less influenced by the complexities of the environment as the individual sensors move towards improved RSS, so that the solution/possibility ratio is improved as is the margin for error with the optimized geometry. Finally, the dynamic reconfiguration improves the adaptability of the UAS to the mission scenario to include real-time decision making based on sensor feedback from improved location estimates. With an improvement to each metric, the ALFUS score increases to 8 of 10. A visualization of the three-metric scoring mechanism with ALFUS is presented in Fig. 5.2, where we show the improved scores of the system with the dynamic reconfiguration ability compared to the system without the dynamic reconfiguration ability.

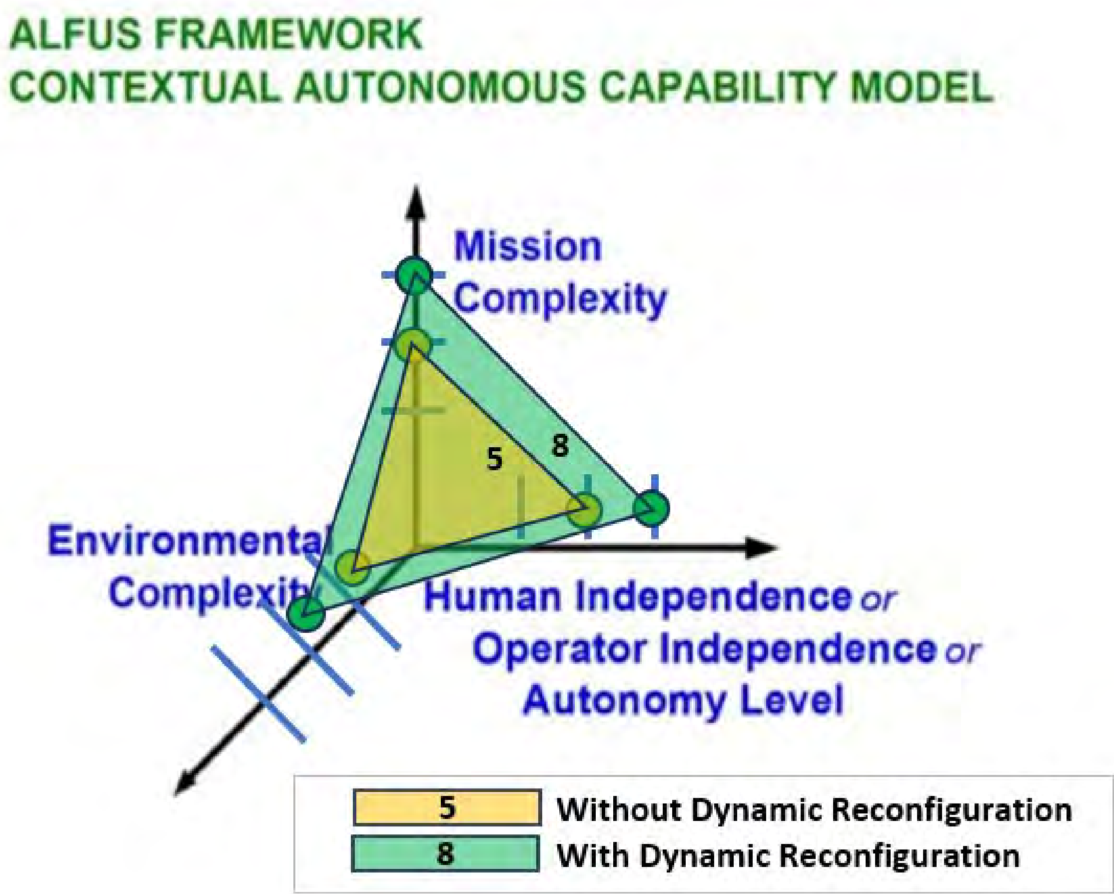

Figure 5.2: ALFUS Framework shows three-axis improvement to autonomous capability with the introduction of Dynamic Reconfiguration.

Chapter Outline This chapter provides the results of evaluations performed in the development of the geometry controller architecture:

1. We provide **motivation and setup** (this section and Section 5.2) for the evaluations, including pertinent background information for the controller evaluation.

2. We evaluate a **heuristic baseline** geometry controller (Section 5.3) that provides an interpretable reference and exposes the main failure modes in multipath. Results were published and presented at the 2025 IEEE Systems Conference [59].

3. We evaluate a **deep Q-network (DQN)** geometry controller, including a shared consensus policy that allows the policy to be extended to any quantity of homogeneous sensors (Section 5.4). Results were published and presented at the 2025 Asilomar Signals, Systems, and Computers Conference [94].

4. We perform **multi-algorithm RL comparison** across DQN, MADDPG, A3C, MAPPO against to the heuristic controller under matched interfaces and stopping criteria (Section 5.5). Results were originally submitted for publication at the 2026 ICASSP conference (not accepted) and have been recently resubmitted for publication at the 2026 Asilomar Signals and Systems Conference [95].

5. We compare our **Python simulation against the EXata digital-twin** for the DQN controller (Section 5.6). Results were accepted for publication and presented at the 2026 IEEE Aerospace Conference [96].

The models that are presented reference the LOCA estimator, solution-cloud filtering, EEP/CEP/SEP definitions, and the ML architectures from Chapter 3, and we reference the hardware and environmental error models from Chapter 4.

5.2 Comparative Evaluation Considerations

Across all controllers considered in this chapter, we consider an iterative refinement of a 3-D emitter estimate using a cooperative UAS array. At snapshot k, LOCA produces a filtered solution cloud Λ_k and a fused estimate $\widehat{E}_k$ with an associated uncertainty metric (e.g., SEP_{50}). The controller selects a movement action for the sensors, the array repositions, and

LOCA is re-run at snapshot $k+1$ using the updated geometry. The loop repeats until the convergence targets are met or the step budget is exhausted.

5.2.1 Controller interface

Each controller (heuristic or learned) operates on a common observation/action interface. The observation vector is intentionally composed of quantities available from estimator outputs and from the known sensor geometry, without assuming AoA capability or phase-coherent receivers. Actions are parameterized as bounded 3-D movement vectors for each sensor.

Episodes terminate when the convergence targets are met (for example, when $SEP_{50} \leq 2\,\text{m}$), or when a maximum step budget is reached. Unless otherwise specified, localization error sources (clock/position errors and environmental multipath) follow the models and parameter choices in Chapter 4.

5.2.2 Performance metrics

Performance is evaluated using: (i) final localization error $\|\widehat{E} - E\|_2$, (ii) SEP_{50} from the solution cloud, (iii) total movement distance (by all UAS elements for a localization episode), (iv) total steps to convergence (number of "snapshots" an repositioning steps), and (v) success rate (fraction of trials meeting the convergence targets within the step budget). Reported statistics are computed over Monte Carlo runs with randomized initial sensor and emitter locations.

5.3 Heuristic Geometry-control

The formulation and results of this effort were published and presented at the 2025 IEEE Systems Conference [59].

5.3.1 Background

In consideration of UAS repositioning methods, the authors in [41, 45] optimize the geometric configuration of the UAS to improve the accuracy of the localization based on the RSS. We implement the notion of this RSS-based approach and extend it by quantifying the capability with a robust environmental error model. However, these RSS methods and related ones either rely on pre-defined optimized geometries that do not consider the dynamics of a complex RF signal environment [2, 45], they are computationally complex utilizing low-rank optimization and trajectory planning [2, 97], or they consider only RSS [2, 41], which may not be useful in geography-constrained situations.

The authors of [2] utilize RSS and add a complex algorithm to perform collaborative processing within the UAS to improve the inter-UAS positioning estimates. They combine these measurements with angle of arrival data and implement super multidimensional scaling with patch dividing/merging techniques. In a similar level of complexity, the authors of [97] perform a dynamic repositioning approach utilizing semidefinite programming and alternating convex optimization to determine the emitter location and the UAS iterative geometry via pre-determined waypoints. Further, approaches such as [97] and [98] consider trajectory planning with a single UAS and do not consider a cooperative UAS array.

While effective, many methods like these are computationally demanding and require physical convergence on the emitter. Our study examines similar approaches, yet we differentiate our work by showing that utilizing a localization algorithm while remaining at a distant range and incorporating real-time environmental feedback to inform iterative updates to the sensor geometry provides a continuous refinement of localization estimates.

5.3.2 Heuristic action rules

A heuristic controller provides an interpretable baseline that requires minimal training data and computational expense, which is attractive for UAS payload constraints. Second, its failure modes help motivate the need for learning-based control in multipath-dominated settings.

The heuristic controller considered here uses the current solution-cloud geometry to choose movement actions. At each snapshot, a repositioning scheme is applied to the sensor locations, where a sensor subarray corresponding to a certain suboptimal condition is relocated to a new position and the target position is subsequently estimated. The approach continues with the new sensor locations to iteratively reposition the array. By only repositioning the subset of the UAS array that corresponds to the suboptimal conditions, we adapt the geometry to the environment and expect to improve the overall localization accuracy. We consider as a performance metric the reduction of the SEP radius for each iterative repositioning step. Two methods are based on geometrically converging a subarray on the centroid to reduce the range-based error, while one method is based on randomly repositioning a subarray to potentially remove a NLoS condition. The last method finds an equal range to the centroid for a subarray to reduce the TDOA errors, the goal of which is to exploit the inherent geometry of the LOCA algorithm to find optimal positions without geometrically converging on the emitter. These repositioning schemes are as described below for each localization step:

I. RSS Convergence:

1. Determine the emitter estimate centroid and the range from each sensor to the centroid, R_i, using (10).
2. Identify the three sensors, s_i, s_j, s_k, with the lowest RSS, such that $RSS_i \leq RSS_j \leq RSS_k \leq \ldots \leq RSS_n$.
3. Reposition s_i, s_j, s_k to a location 25% closer to the centroid.

II. Best Estimate Random:

1. Determine the emitter estimate centroid and the range from each sensor to the centroid, R_i, using (10).
2. Consider from (10) the contribution of each s_i to the set of emitter estimates (130 estimates for the 8 UAS system).
3. Next, determine the median absolute deviation from the centroid for each estimate and associate that deviation with each sensor used for that estimate.
4. Compute the total deviation for each sensor as a sum of all deviations associated with that sensor.
5. Identify s_i, s_j, s_k with the highest deviation.
6. Reposition the subarray $s_i, s_j, s_k$to a random position.

III. Best Estimate Convergence:

1. Perform method in {II} through step {5}.
2. After identifying s_i, s_j, s_k, reposition the subarray to a location 25% closer to the centroid.

IV. Best Estimate Equidistant:

1. Perform method in {II} through step {5}.
2. After identifying s_i, s_j, s_k, reposition the subarray to a location that is the mean range of s_i, s_j, s_k to the centroid.

5.3.3 Baseline performance in dense-urban multipath

After each initial localization, the UAS repositioning approaches {I}-{IV} were independently modeled for 25 steps. At each step, the localization estimate was repeated using the described MATLAB simulation environment to inform the accuracy improvement in dynamic repositioning.

Although the added sensors provide increasing localization accuracy in the presence of environmental scatterers, we are seeking to maximize this accuracy for all sensor quantities by exploiting the mobility of the UAS network and utilizing the cooperative aspect of the UAS array via emitter localization at the subarray-level. Therefore, we consider the repositioning approaches and the mitigation of the accumulated environmental errors with adaptive UAS geometries. Further, we analyze reducing the SEP radius as a metric to quantify improvements to the localization accuracy with successive geometries, with the assumption that these metrics are correlated. Fig. 5.3 shows the SEP propagation for the case where no hardware or environmental errors are present. In this case, the RSS and the equidistant approaches continually improve the SEP, while the methods of random repositioning and best estimate convergence have a minor impact to the SEP radius & localization error. For the RSS method, this is expected as all elements eventually migrate towards the target and ensure their TDOA, and thus their solution offset, is minimized. The best estimate convergence approach remains static because, in an error-free case, the localization estimates completely converge regardless of sensor position, while the TDOA will not necessarily converge as in the case of the equidistant approach.

We apply our repositioning methods in a simulated scenario representative of an urban scattering environment with a 10 ns local timing inaccuracy and 2 cm UAS positioning error with 6- and 9-element UAS arrays. The dynamic repositioning results presented in Fig. 5.4 show the SEP radius and localization error improvement normalized to the initial

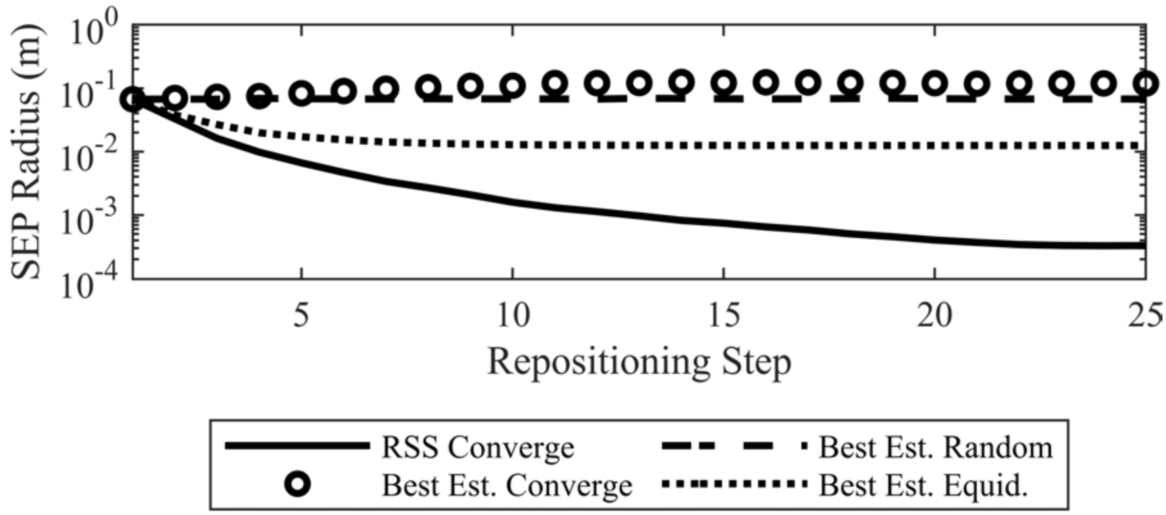

Figure 5.3: Spherical error probable (SEP) radius with iterative geometries in error-free environment by repositioning subarray with: 1) lowest RSS towards centroid, 2) largest LOCA deviation to random location, 3) largest LOCA deviation towards centroid, and 4) largest LOCA deviation to equidistant range.

SEP radius and localization error for each scenario. In general, as the search algorithm repositions the sensor array, the SEP radius and localization error decrease, except for the random repositioning approach, which shows no improvement. The RSS method offers the greatest improvement, which is a direct result of the path loss models from [83], such that a smaller propagation distance results in a lower error term compared to the overall signal strength. However, the equidistant approach, which does not rely on physical convergence like RSS, provides a similar reduction in SEP radius and localization error. Further, while the 6-element UAS array offers an overall higher initial localization error than larger arrays as seen in Fig. 4.13, it experiences a consistently greater improvement to the SEP radius with an adaptive geometry compared to the 9-element UAS case.

The adaptive geometry approach enables the system to minimize environmental and hardware errors in its localization performance irrespective of the array size, with a greater impact of improvement with smaller arrays. Further, the equidistant approach offers an alternative to methods that converge on the emitter location, allowing missions to improve their localization accuracy at standoff distances.

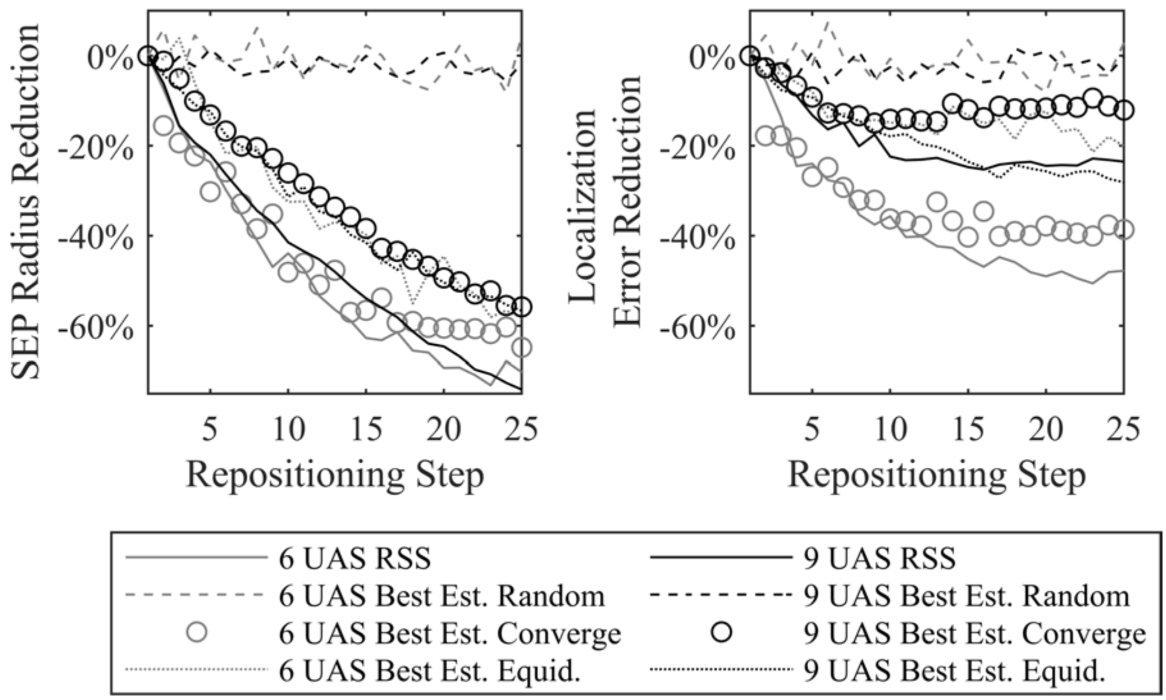

Figure 5.4: Reduction in spherical error probable (SEP) radius and localization error as a percentage of initial SEP and localization error for 6-sensor and 9-sensor UAS arrays, in presence of environmental and hardware errors by repositioning subarray with: 1) lowest RSS towards centroid, 2) largest LOCA deviation to random location, 3) largest LOCA deviation towards centroid, and 4) largest LOCA deviation to equidistant range.

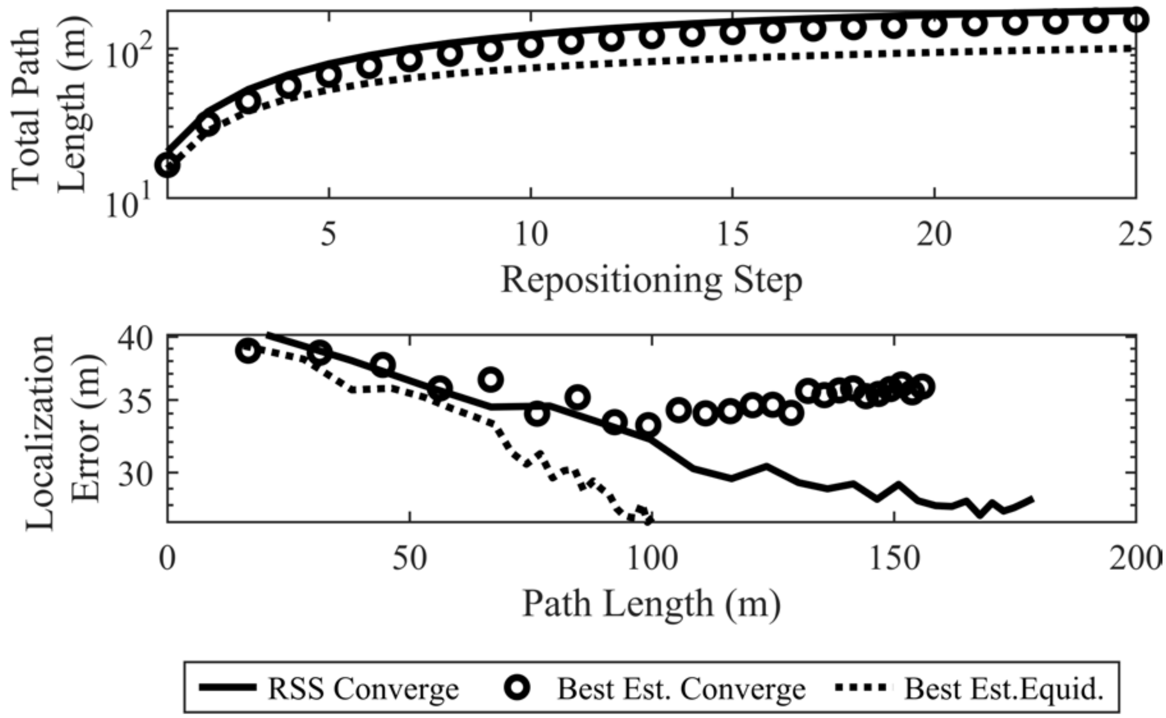

Figure 5.5: Comparison of total path length traveled by each UAS and the associated localization error reduction for repositioning subarray with: 1) lowest RSS towards centroid, 2) largest LOCA deviation towards centroid, and 3) largest LOCA deviation to equidistant range.

Last, we consider mission effectiveness by evaluating localization error as related to the total distance traveled by the UAS for each method. In Fig. 5.5, we see that for the equidistant method, the UAS travels approximately half the distance of the RSS method to achieve an equivalent localization performance, resulting in a reduced usage of system resources and mission time. The best estimate convergence method has a plateau of performance after traveling the same length as the equidistant method; however, it does not improve its localization accuracy to the same magnitude as the other approaches.

5.3.4 Evaluation Summary

The simulation and analysis in this investigation informs the viability of a reconfigurable cooperative wireless sensor array for emitter localization in an urban environment. Informed repositioning of the UAS sensor array results in improved accuracy in localization without requiring physical convergence on the emitter, providing more mission flexibility than RSS-based methods. The sensor array utilizes the LOCA algorithm to generate target location estimates with its distributed subarrays and with the full array, enhancing its flexibility as a cooperative emitter localization system. The localization error can be minimized with a small quantity of sensors and a low-complexity localization and optimization algorithm, reducing the overall system cost without sacrificing performance. While a 5-sensor cooperative UAS array is minimally suitable for emitter localization with this method, increasing the array size with up to nine sensors significantly enhances accuracy and enables more dynamic missions at both the array- and subarray-level.

Subsequent sections we explore the use of machine learning methods to optimize UAS sensor positions, driven by these findings that dynamically reconfiguring the array allows for improved emitter localization accuracy.

5.4 Deep Q-network geometry control

The formulation and results of this effort were published and presented at the 2025 Asilomar Signals, Systems, and Computers Conference [94].

5.4.1 Background UAS DQN Approaches

We study a fully decentralized [48, 99] multi-agent deep Q-learning (DQN) framework, utilizing a time difference of arrival (TDOA) algorithm, where UAS agents independently optimize their policies to promote geometries that minimize localization error in complex multipath environments. After training, we perform a pairwise comparison analysis of the policies learned by each agent to find a single representative "consensus" policy [100] that can be applied to all agents for system operation. While other approaches optimize policies during training or reach consensus at run-time [101], the consensus representative policy method offers a simple architecture that is scalable to larger drone network sizes without additional training or cross-agent learning parameter sharing.

Reward Methodology We evaluate three methods to inform the UAS geometry, utilizing both the SEP (defined in Chapter 3, Section 3.2.2) determined by the LOCA algorithm and the RSS. The success of each method is measured by its reduction of median localization error, which is only captured as a metric, and is not used as a parameter in the training, since the true target location is unknown in practice. We consider three reward methods: RSS Optimization, SEP Optimization, and combined RSS & SEP Optimization.

RSS Optimization: The reward is maximized when the aggregate RSS at a given timestamp is greater than the previous step, as in [43], promoting the UAS geometry to physically converge on the emitter.

SEP Optimization: The reward is maximized when SEP decreases [59] relative to a previous timestamp, so that the geometry is optimized towards $sep_{\min}$. This case does not require geometric convergence on the emitter and gives the system the freedom to optimize the geometry according to the localization algorithm.

RSS & SEP Optimization: The reward is maximized by equally weighting geometry that both reduces SEP and increases RSS.

Our geometry-optimizing localization algorithm is summarized in Algorithm 3.

5.4.2 Shared (consensus) policy for homogeneous sensors

Because the DQN architecture is comprised of independent agents and because the LOCA algorithm supports individual sensors operating toward an optimized geometry, the policies learned by one agent could potentially be applied to another. Therefore, after training a small set of UAS drone elements, we analyze learned policies to generate a consensus representative policy. The consensus policy [100], [101] approach assesses the individual policies for similarity using pairwise mean absolute differences and Pearson correlation to identify a single representative policy that is most similar to all agent policies. We apply this representative policy to all agents, replacing the independent policies learned during training. This consensus policy approach allows us to scale the UAS system to any number of elements without additional training. The approach is enabled both by using a network of UAS that have the same hardware architecture and by the use of a localization algorithm that is optimized by a distributed sensor geometry [59]. This approach preserves a distributed execution model while allowing centralized training with shared experience across sensors.

5.4.3 Training configuration

The simulated system was first evaluated with the 8-UAS case with the individual (non-consensus) policies developed during the training. The evaluation utilized the same simula-

Algorithm 3: Optimizing UAS Geometry for Localization

Require : Emitter location E (known only in simulation for diagnostics); initial UAS locations U_0; max steps $n_{\max}$; SEP threshold $SEP_{\min}$; discount γ

Ensure : Final estimate $\widehat{e}_n$ and (simulation-only) localization error ξ

Initialize online Q-network $Q_\theta(s, a)$ and target network $Q_{\theta^-}(s, a)$ [48]
Initialize replay memory $\mathcal{D}$
Choose reward condition: (i) RSS optimization, (ii) SEP optimization, or (iii) both
Estimate $\widehat{E}_0$ and SEP_0 using LOCA
$s_0 \leftarrow \{U_0,\ TDOA_0,\ SEP_0,\ RSS_0\}$
$done \leftarrow$ **false**; $n \leftarrow 0$
while ***not*** *done* **do**
    Action selection:
    With probability ϵ_n, select a random displacement a_n; otherwise select $a_n \leftarrow \arg\max_a Q_\theta(s_n, a)$
    Execute motion & sensing:
    $U_{n+1} \leftarrow U_n + a_n$
    Measure $TDOA_{n+1}$ and RSS_{n+1}
    Localization & uncertainty:
    Run LOCA to obtain candidate solutions $\widehat{E}_{n+1}$
    $\widehat{e}_{n+1} \leftarrow \text{median}(\widehat{E}_{n+1})$
    Compute SEP_{n+1}
    $s_{n+1} \leftarrow \{U_{n+1},\ TDOA_{n+1},\ SEP_{n+1},\ RSS_{n+1}\}$
    Reward:
    if *reward condition is met* **then**
        $r_n \leftarrow +1$
    else
        $r_n \leftarrow -1$
    if $n + 1 \geq n_{\max}$ ***or*** $SEP_{n+1} \leq SEP_{\min}$ **then**
        $done \leftarrow$ **true**
    DQN update (training):
    Store $(s_n, a_n, r_n, s_{n+1}, done)$ in replay $\mathcal{D}$
    Sample a minibatch from $\mathcal{D}$
    $y_n \leftarrow r_n + (1 - done)\,\gamma \max_{a'} Q_{\theta^-}(s_{n+1}, a')$
    Minimize $(y_n - Q_\theta(s_n, a_n))^2$ over the minibatch (MSE or Huber)
    Periodically update θ^- from θ (hard copy or soft update)
    $n \leftarrow n + 1$
$\xi \leftarrow \|E - \widehat{e}_n\|_2$

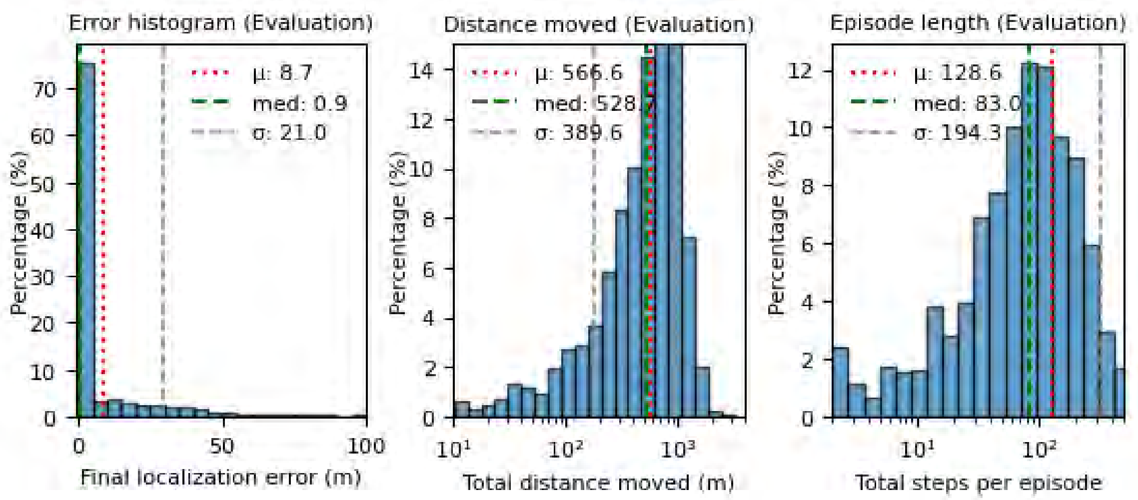

Figure 5.6: Evaluation metrics for the UAS geometry optimizer with per-agent policies, rewarded by a combination of SEP and RSS. Performance shows the final localization error, the average distance moved per drone, and the number of repositioning steps to convergence across all evaluation episodes for the given policy approach.

tion environment as training, including hardware errors and multipath effects, with different emitter locations and initial UAS starting positions. Next, the consensus policy was created and applied to the simulated system, again using the same simulation environment as the training with the same emitter locations and initial UAS starting locations as the independent policy evaluation. For the 8-UAS case, for each of the independent and consensus approaches, we conducted 1500 evaluation episodes. We then scaled the consensus policy to larger UAS swarms by applying the same policy to each additional UAS in the network. For the 9-12 UAS swarm array size cases, we evaluated fewer trials than the 8-UAS case due to faster convergence.

For the baseline 8-UAS configuration, the localization performance metrics for the independently trained policies (Fig. 5.6) and the learned consensus policy (Fig. 5.7) are nearly identical in terms of localization error, per- drone distance moved, and the number of repositioning steps. These results indicate that the consensus policy preserves both convergence behavior and motion cost observed during training. Therefore, for this architecture, the policy-selection procedure does not degrade performance and a single representative policy

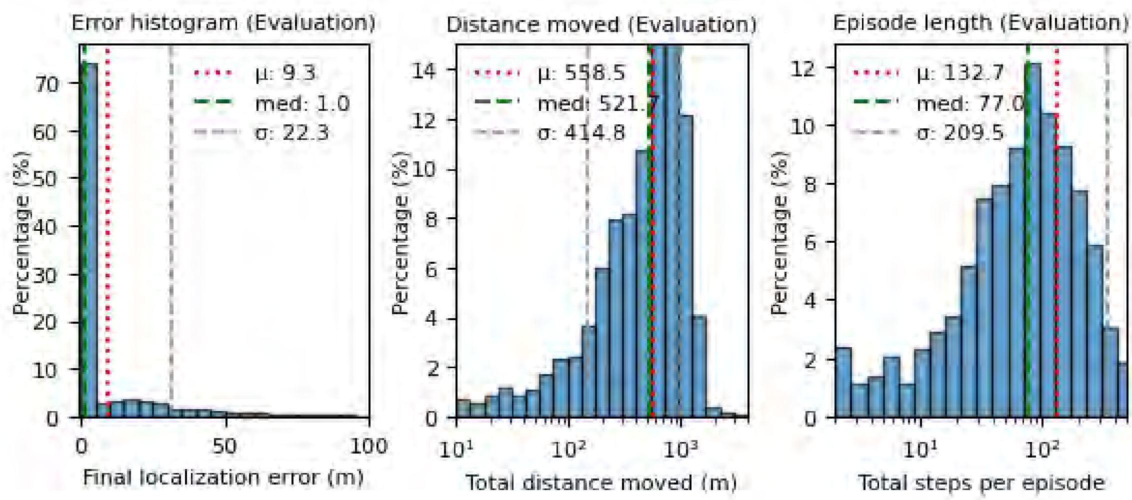

Figure 5.7: Evaluation metrics for the UAS geometry optimizer with single consensus policy applied to all agents, rewarded by a combination of SEP and RSS. Performance shows the final localization error, the average distance moved per drone, and the number of repositioning steps to convergence across all evaluation episodes for the given policy approach. Metrics are comparable to per-agent policy results.

can effectively replace individually trained agent policies.

After evaluating the 8-UAS case, we evaluated the 9-12 UAS cases. Table 5.1 provides the metrics for the individual policies and the shared consensus policy evaluated on the 8 UAS system, and it provides the metrics for scaling the UAS array size under the shared consensus policy. As the swarm increased from 8 to 12 UAS (Table 5.1), the mean localization error drastically reduced from 8 m to nearly 0 m (>99.9% reduction), and the standard deviation reduced from 21 m to 0 m, which is an expected performance improvement with increasing UAS array size [59]. At the same time, the mean per-drone distance moved decreases from 567 m for 8 UAS to only 7.2 m for 12 UAS, and the mean repositioning steps to convergence is reduced from 129 to 2. The intermediate 9–11 UAS cases follow the same trend, with agents simultaneously reducing localization error, tightening the spread of outcomes, and traveling shorter distances for convergence. The behavior shows that the approach becomes increasingly advantageous as additional agents are added to the swarm because the consensus policy approach allows scaling the UAS size without retraining agents.

Table 5.1: Shared consensus policy performance evaluation with increasing UAS quantity.

UAS Array Size	**Final Localization Error (m)**			**Per-Drone Distance Moved (m)**			**System Repositioning Steps**		
	Mean (μ)	**Median** (med)	**Std. Dev.** (σ)	μ	med	σ	μ	med	σ
8 (*individual policy*)	9.3	1.0	22.3	559	522	415	133	77	210
8 (*consensus policy*)	8.7	0.9	21.0	567	529	390	129	83	194
9	0.6	0.0	2.5	221	108	287	36	12	64
10	0.1	0.0	0.3	69	1.6	160	10	2	21
11	0.02	0.0	0.1	19.1	1.5	66	3	2	8
12	0.01	0.0	0.1	7.2	1.5	39	2	2	4

The evolution of localization performance over episode steps Fig. 5.8 further highlights the benefit of scaling the array. For all swarm sizes, the system reduces localization error as the geometry is adapted, but larger swarms converge faster and more reliably. Whereas the 8-UAS case often requires tens to over 100 steps to reach a low-error configuration, the 11- and 12-UAS cases typically converge within a few steps, with much lower residual error due to improved geometric diversity and larger volume of localization estimates from LOCA-based TDOA measurements.

The three reward structures (RSS-only, SEP-only, and combined RSS+SEP) also show distinct behaviors. The RSS approach encourages physical convergence on the emitter and can achieve low error, but often requires more motion and steps because it is unaware of malformed geometries that degrade TDOA localization. The SEP-only approach focuses on improving LOCA geometry and often achieves low errors without clustering around the emitter. The mixed RSS+SEP reward provides the best trade-off, consistently reaching the smallest mean error with few steps and limited per-drone distance. Together with the consensus policy scaling results, these results show that combining a geometry-aware metric (SEP) with a physically intuitive metric (RSS) produces swarm behaviors that are both mission-efficient and tolerant to multipath conditions.

Conclusion The agents trained independently and naturally converged to similar behavioral policies due to the shared optimization approach. Through decentralized learning

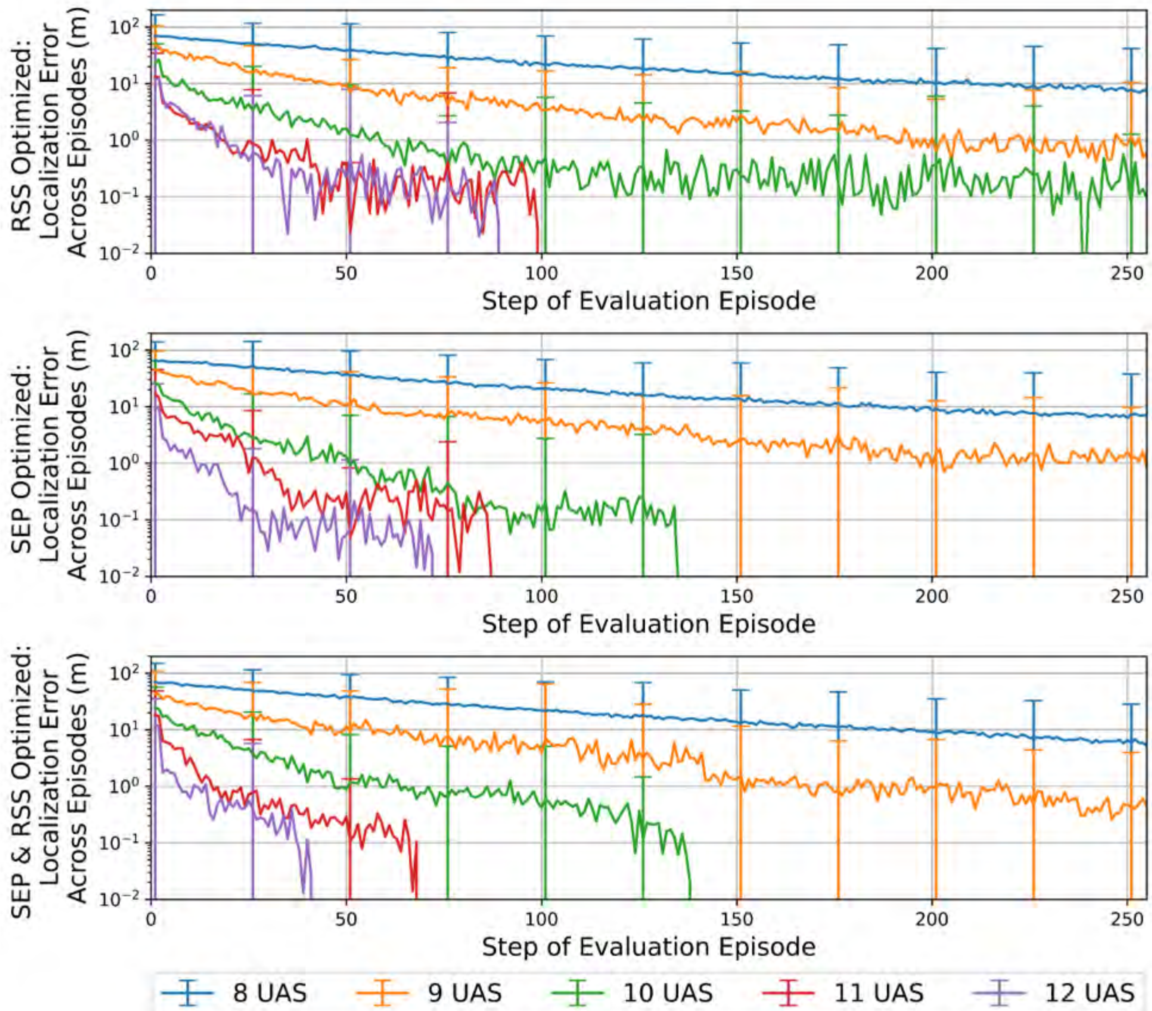

Figure 5.8: Comparison of mean localization error with increasing size of UAS "drone swarm" and different optimization reward approaches: RSS Optimization (top), SEP Optimization (Middle), RSS & SEP Optimization (bottom). Results find minimal error and fewest steps to convergence with 12 drones in a mixed RSS+SEP reward structure.

and consensus policy selection, this work enables accurate cooperative localization of emitters with simplified scalability to larger drone networks. The consensus policy maintains performance for the original 8-UAS configuration while providing substantial gains when additional UAS are added, reducing mean localization error by more than two orders of magnitude with fewer steps and less per-drone motion.

In the next section, we extend the controller to alternate learning methods, followed by the incorporation of more robust communication models using a digital twin. As an additional goal, we seek fewer steps to convergence by performing additional reward shaping.

5.5 Comparison to alternative reinforcement-learning controllers

While DQN provides a strong baseline for discrete-action control, multi-agent variants may better exploit the structure of cooperative sensing problems. To test this hypothesis, we compare DQN against MADDPG, A3C, and MAPPO under a matched observation/action interface, identical stopping criteria, and the same error-model conditions. The intent is not to exhaustively tune each algorithm, but to measure whether additional coordination mechanisms (centralized critics, actor-critic training, or proximal policy updates) yield consistent gains for geometry control.

MADDPG [52] extends deterministic policy gradients to cooperative multi-agent settings using centralized training and decentralized execution, which can improve coordination but introduces sensitivity to hyperparameters and credit assignment. A3C [51] uses parallel actor-learners to stabilize value estimates and improve sample efficiency in on-policy settings. MAPPO [53] extends PPO to multi-agent problems under a centralized value function, which often improves stability for cooperative tasks with shared reward.

We compare the RL controllers (DQN, MADDPG, A3C, MAPPO) to a policy-free heuristic baseline we call "SEP-Greedy" (SEPG), which greedily selects actions minimizing SEP_{50} at each step. We view the greedy controller as a domain heuristic that could guide or initialize learning in hybrid schemes [54]. The heuristic baseline corresponds to a low-compute, fully distributed option (no training), while the learning-based approaches trade offline training cost for improved online decision quality and the potential to scale across sensor counts via shared-parameter policies.

Table 5.2: Summary of controller architectures; parameters: target-network soft-update rate τ, discount factor γ, rollout length L.

Controller	Policy / Value Nets	Train-time Centralization	Policy Type	Data Regime	Action Sampling
SEPG	N/A	N/A	Heuristic: min current SEP	N/A	arg min over discrete moves
DQN	Per-agent Q-net + target networks (2× dense ReLU with L2 & dropout; Huber loss).	None (independent learners).	Off-policy, value-based	Replay buffer; soft target update $\tau = 0.005$; discount $\gamma = 0.99$.	ε-greedy
A3C	Per-agent actor (logits) + critic (value).	None (independent learners).	On-policy, actor-critic	Short rollouts $L = 32$; entropy bonus.	Categorical sampling
MADDPG	Per-agent actor; centralized critic over joint (observations, actions) + targets.	CTDE (critic centralized, actors decentralized).	Off-policy	Joint replay; $\tau = 0.005$; $\gamma = 0.99$; entropy bonus.	Straight-through Gumbel–Softmax
MAPPO	Shared actor; centralized value over concatenated observations.	CTDE (central value).	On-policy, PPO	Rollout $L = 128$; clip=0.2; 4 epochs; minibatch size 1024.	Categorical sampling

The architecture of each policy is summarized by Table 5.2.

5.5.1 Algorithm Overview

The algorithm studied is a generalization of Algorithm 3 applied to all controllers. As a geometry-shaping improvement from Section 5.4 and to reduce the number of steps to localization convergence, we include additional algorithm features to coerce the sensors to a spherical pattern around the emitter. The spherical pattern is supported by the results in Section 5.3 that showed ideal performance when the sensors were equidistant to the emitter.

The RL observation includes normalized relative geometry, per-agent RSS and TDOA features, and geometry statistics to track the equidistance from the sensors to the emitter. Actions are bounded 3-D displacements, with a first step to determine a "sphere" pattern around the initial localization estimate of the emitter and translates the sensors to that location. Rewards combine improvement in uncertainty (SEP_{50}) with a per-step penalty to reduce time-to-localize; during training only, an improvement in localization error, $\varepsilon_x = \|x - \widehat{x}\|$, contributes to rewards.

For all cases, at each snapshot or repositioning step, the LOCA cloud Λ is combined with sensor jitter and multipath variance-aware weights to yield a measurement (z_k, R_k) [92].

During evaluation and execution, we optionally apply a standard position-only Kalman filter (KF) [64] as a temporal smoother on the per-snapshot fused measurements. We use a random-walk model with $F = I_3$ and direct position measurement $H = I_3$. A normalized innovation squared (NIS) statistic is computed as a consistency check; when a snapshot is strongly inconsistent under a chosen χ^2 threshold, the KF output is not used for downstream control decisions. During training, the KF is disabled so agents learn directly from (z_k, R_k).

The geometry-refined emitter localization system is summarized by Algorithm 4.

Algorithm 4: Emitter localization with UAS geometry controllers

Require : UAS starts $\{s_i^0\}_{i=1}^N$; unknown emitter $x \in \mathbb{R}^3$; initial estimate z_0; SEP goal ς_{SEP}; horizon $K_{\max}$; training flag; noise models $\varepsilon_{U,i}, \varepsilon_{t,ij}, \varepsilon_{P,i}$; controller $\mathcal{C} \in \{\mathrm{SEPG}, \mathrm{DQN}, \mathrm{A3C}, \mathrm{MADDPG}, \mathrm{MAPPO}\}$; per-step penalty κ; weights α, β

Ensure : $\overline{x}_k$, covariance P, achieved SEP_{50}, total steps, total distance

Initialize: $k \leftarrow 0$; $x_{0|0} \leftarrow z_0$; $P_{0|0} \leftarrow 10^6 I_3$; $\overline{x}_0 \leftarrow z_0$; $P \leftarrow P_{0|0}$; $D_{\mathrm{tot}} \leftarrow 0$

while $k < K_{\max}$ ***and*** $SEP_{50} > \varsigma_{\mathrm{SEP}}$ ***and*** *not plateau* **do**

  1) TOA and RSS sensing:
  Draw $\Delta t_{ij}, RSS_i$ from x
  $\widehat{s}_i \leftarrow s_i^k + \varepsilon_{U,i}$, $\widehat{\Delta t}_{ij} \leftarrow \Delta t_{ij} + \varepsilon_{t,ij}$, $\widehat{RSS}_i \leftarrow RSS_i + \varepsilon_{P,i}$
  2) LOCA cloud & outlier rejection:
  Form cloud Λ using Algorithm 1
  $\Lambda \leftarrow \{\lambda_m\}_{m=1}^M = \mathrm{LOCA}(\{\widehat{s}_i\}, \{\widehat{\Delta t}_{ij}\})$
  Prune IQR outliers and geometric outliers via RSS–TOA residuals $\rho_m = \sum_i \left|\widehat{RSS}_i - \widetilde{RSS}_i(\lambda_m)\right|$
  3) Cloud statistics: $\widehat{x} \leftarrow \mathrm{median}(\Lambda)$, $\quad SEP_{50} \leftarrow q_{0.5}\{\|\lambda_m - \widehat{x}\|\}$
  4) Variance-aware fusion:
  $\sigma_i^2 \leftarrow \sigma_{\mathrm{env}}^2(\widehat{x}, i) + \sigma_{\mathrm{hw}}^2(i)$
  $\omega_m \propto \left(\sum_i a_i(\widehat{x}, \lambda_m)\sigma_i^2\right)^{-1}$
  $z_k \leftarrow \dfrac{\sum_m \omega_m \lambda_m}{\sum_m \omega_m}, \quad R_k \leftarrow \dfrac{\sum_m \omega_m (\lambda_m - z_k)(\lambda_m - z_k)^{\mathsf{T}}}{\sum_m \omega_m}$
  5) Position-only KF (during evaluation only):
  if *training* **then**
    $\overline{x}_k \leftarrow z_k, \quad P \leftarrow R_k$
  else
    Kalman Filter: Apply the position-only KF recursion on (z_k, R_k) (random-walk $F = H = I_3$) during evaluation.
    6) Progress signals: Track SEP_{50} and if training $\varepsilon_x(k) = \|x - \widehat{x}\|$; plateau guard on SEP_{50}
    7) Compute State: $\phi_k = \left\{\frac{\widehat{s}_i - \overline{x}_k}{\|\widehat{s}_i - \overline{x}_k\|},\ \widehat{RSS}_i,\ \widehat{\Delta t}_{ij},\ P,\ SEP_{50}\right\}$
    8) Compute Action:
    if $k = 0$ **then**
      $r_{\mathrm{sph}} \leftarrow \frac{1}{N}\sum_i \|s_i^0 - \overline{x}_0\|, \quad a_i^0 \leftarrow \mathrm{clip}\left(\overline{x}_0 + r_{\mathrm{sph}} \frac{s_i^0 - \overline{x}_0}{\|s_i^0 - \overline{x}_0\|} - s_i^0\right)$
    else
      if $\mathcal{C} = SEPG$ **then**
        $a_i^k \leftarrow \arg\min_{a \in \mathcal{A}} SEP_{50}$(after applying a)
      else
        $a_i^k \leftarrow \pi_{\mathcal{C}}(\phi_k), \quad a_i^k \in \mathbb{R}^3$ bounded; step size $\propto \|\widehat{s}_i - \overline{x}_k\|$
    9) Compute Reward: $r_k = \alpha\,\Delta SEP_{50} + \beta\,\Delta\|x - \overline{x}_k\| - \kappa$, with $\beta = 0$ (evaluation)
    (If training and $\mathcal{C}$ learnable: update policy using $(\phi_k, a^k, r_k, \phi_{k+1})$)
    10) Advance & bookkeeping:
    $s_i^{k+1} \leftarrow s_i^k + a_i^k$; $D_{\mathrm{tot}} \leftarrow D_{\mathrm{tot}} + \sum_i \|a_i^k\|$; $k \leftarrow k + 1$

return $\overline{x}_k$, P, SEP_{50}, *total steps* k, *total distance* D_{tot}

5.5.2 Comparative results

We assess the impact of each RL controller on cooperative emitter localization in UAS swarms in terms of SEP_{50} radius reduction, localization error reduction, and error interquartile range (IQR) across steps of execution episodes. The IQR is a measurement of the 25^{th}-75^{th} percentile of localization error, representing the typical performance of the controller.

Experiments use the Python PettingZoo/Gymnasium [102] with eight identical UAS agents operating in a 3-D, 2 km urban volume. Emitter and UAS initial positions are randomized per episode, with measurement noise following the position, time, and multipath terms discussed previously. Each policy, implemented via the architecture in Table 5.2, is trained for 1000 episodes and evaluated over 1500 Monte Carlo episodes per method. A position-only Kalman filter is applied only at evaluation. Table 5.3 reports the final metric (final localization error, steps to convergence, median per-UAS path length, final SEP, wall-clock time). Fig. 5.9 shows a) SEP_{50} vs step, b) percent error reduction, and c) error IQR.

All methods reduce the SEP_{50} within 5-10 repositioning steps (Fig. 5.9.a)), demonstrating the advantage of active geometry control over static sensing. By step 10 (Fig. 5.9.b), most RL policies achieve >60% error reduction, whereas MAPPO and SEPG are closer to 50%; SEPG subsequently plateaus while MAPPO slowly improves. Final errors reflect these trends (Table 5.3), A3C, DQN, and MADDPG reach ∼16-17 m, SEP/MAPPO remain ∼24-25 m.

Uncertainty IQR (Fig. 5.9.c) is broader for SEPG and MAPPO (∼30–40 m) than for the

Table 5.3: Model performance during policy evaluation.

Parameters	**SEPG**	**DQN**	**A3C**	**MADDPG**	**MAPPO**
Final localization error (m)	24	17	16	17	25
Error reduction	38%	61%	61%	59%	34%
Steps to convergence	11	8	12	7	19
Distance moved per UAS (m)	349	312	316	431	850
Final SEP (m)	6	5	5	5	6
Episode clock time (s)	5.1	3.0	3.9	3.0	6.0

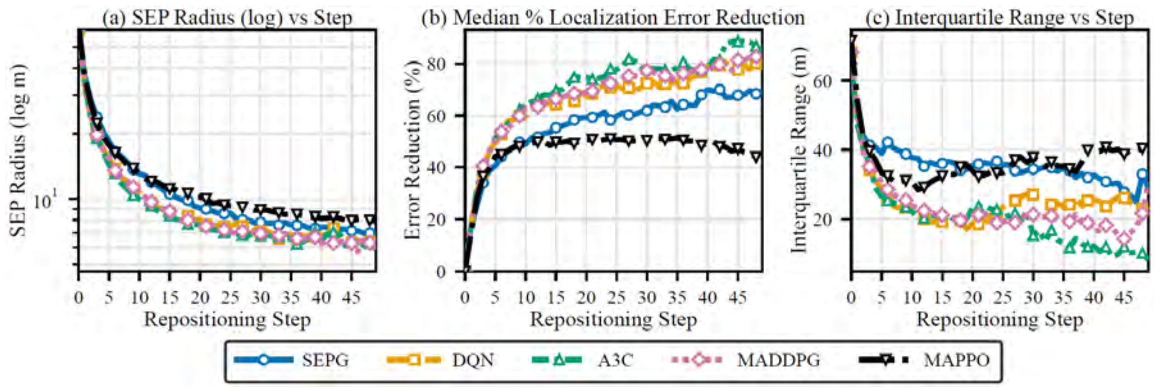

Figure 5.9: a) Comparison of each policy: a) SEP radius convergence across episodes; b) relative improvement in localization accuracy across repositioning steps; c) uncertainty in localization estimates across episodes.

RL methods (∼20 m). A3C and MADDPG show the most stable spread across episodes, suggesting more consistent performance from randomized initial conditions under identical sensing and motion settings. Path length further reflects how each controller trades coordination for effort: DQN and A3C are comparatively conservative (∼315 m per UAS), MADDPG expends more distance to converge quickly (∼431 m over 7 steps), and MAPPO travels the farthest (∼850 m). Episode wall-clock time is ∼3 s for DQN and MADDPG and ∼6 s for MAPPO.

The relative performance follows each algorithm's update mechanics and use of information. A3C updates on short, on-policy rollouts with entropy regularization producing steady geometry improvements. MADDPG utilizes a centralized critic that stabilizes joint moves. MAPPO uses a shared-actor PPO with longer rollouts and partial observability slows early learning and widens IQR. DQN had competitive final errors at convergence but was less uniform mid-trajectory. Because SEPG minimizes instantaneous SEP and lacks long-term strategy, it reduces error early but plateaus when local geometry stops improving.

In these conditions, A3C and MADDPG provide the best balance of speed, precision, and stability, supporting the advantage of RL-based geometry control to mitigate the impact of environmental and hardware contributions.

5.5.3 Conclusion

The comparable performance of DQN and MADDPG suggests that, in this localization architecture, the primary coordination burden is imposed by the cooperative estimator rather than by the learning algorithm. Although LOCA produces triad-based subset estimates, the final filtered solution cloud and its uncertainty statistics improve most when the full sensor geometry evolves toward configurations that generate consistent subset solutions. This creates an aligned objective across agents: each UAS can improve global localization performance by making locally-selected moves that reduce cloud inconsistency and promote a globally favorable geometry. As a result, decentralized independent learning can approach the performance of CTDE methods, since the reward landscape provides a strong, shared geometric signal and does not require finely coupled complementary actions for credit assignment. In contrast, on-policy actor–critic baselines (MAPPO/A3C) are more sensitive to reward variance and tuning under the same sensing and motion constraints, which can reduce their practical performance in this setting.

We showed that the independent policies and control in DQN effectively localized the unknown location emitter, and we achieved significantly fewer convergence steps by coercing the sensors to the equidistant geometry around the emitter. In the next section, we will continue evaluating the DQN controller with geometry-shaping metrics in the EXata digital-twin environment.

5.6 Python simulation versus EXata digital-twin verification

The controller experiments above were performed in a Python environment to enable large-scale Monte Carlo evaluation and iterative development. However, because multipath statistics and network effects can be sensitive to modeling assumptions, we also validate the key trends in a higher-fidelity EXata digital-twin environment.

5.6.1 EXata Overview

EXata [103] is a network modeling software application that generates real-time digital twins of operational network topologies, including protocol behavior and environment-driven channel effects such as terrain and weather. The simulator models data movement across the network stack with high fidelity, reports live network statistics, and can interact with external applications for closed-loop analysis. For this study, EXata is primarily valuable because its wireless channel and topology modeling—particularly multipath impacts on signal strength and path delay—provides a stronger verification reference than the simplified propagation and noise models implemented in our Python environment.

EXata is available in the Southern Methodist University (SMU) Cyber Autonomy Range (CAR) facility [104], with sufficient computing resources, and it supports both urban city models (reflectors/scatterers) and the injection of interferers operating near the band of interest. It also supports hardware-in-the-loop evaluation with real radios when needed. We therefore use EXata to verify our system behavior via a digital twin in lieu of flight trials, which are costly, time-consuming, and introduce risk to prototype hardware.

Our results include many test scenarios with varying simulation parameters allowing our prototype to be exercised much more broadly than if a physical test range were used. Our digital twins are implemented at the physical layer with physics simulation engines for RF propagation, fading, scattering, kinematic dynamics with gravity field interactions, and weather effects using physical layer accurate protocols including 5G and IEEE 802.11 standards. The system twin includes dominant sensor hardware error sources such as UAS local clock inaccuracies and UAS positioning drift. The environmental twin is implemented using Keysight's EXata high fidelity urban environment within the CAR. With these realistic constraints, our functional evaluations confirm the system's low localization error and SEP under degraded RF conditions, demonstrating its viability for aerospace and other applications.

5.6.2 Analysis Approach

Although we can use a variety of RF transmitter frequencies in our digital twin environment, the results described here use 5 GHz. At each measurement snapshot or step, UAS elements acquire TDOA and RSS measurements, form LOCA-based solution clouds, and fuse results with variance-aware weighting. Intra-swarm communication is achieved with an ad hoc wireless network optimized to minimize transmit power and with extremely short message frames. A controller UAS is elected within the swarm that receives measurement data, performs computations, and issues repositioning commands. The localization estimates are performed in 2-D and 3-D, depending on the dominant geometry characteristics, and the controller determines a new geometry based on the metrics from the localization result. Because we develop a policy with a DQN, the approach runs in two phases: training within the digital twin environment, where true emitter location information is utilized to guide the improved array geometry, and evaluation, where the true emitter location is unknown, but a Kalman filter is used to refine the localization estimates. Our system performance is evaluated by the localization error and uncertainty (framed by SEP_{50}), the number of repositioning steps required to converge on the emitter, and the distances the UAS elements travel across a localization episode. These metrics serve as operationally-relevant performance parameters providing a measure of system effectiveness in its ability to accurately, rapidly, and efficiently localize an unknown-location emitter.

After initial TDOA and RSS measurements, the 2-D or 3-D LOCA solution cloud Λ is fused with the sensor jitter and multipath variance to yield the solution set (z_k, R_k). We apply the variance-fused measurements to a position-only KF [64] to improve localization accuracy during evaluation and generate the smoothed estimate $\overline{E}_k$. During training, the KF is disabled so agents learn directly from the noisier per-snapshot estimates; during evaluation, the KF smoothing primarily reduces snapshot-to-snapshot jitter.

The DQN controller UAS receives normalized geometry information, per agent features, and estimate statistics, utilizing two fully-connected layers with dropout regularization and an experience replay buffer [48]. Actions are 3-D displacements, scaled adaptively by current SEP_{50} radius to ensure stability across mission phases. The reward function combines localization improvements, geometry formation, and a path penalty to shape the policy towards a pseudo-optimal geometry.

At each repositioning step (k), each UAS element determines a state vector in Eq. 5.1:

$$\phi_k = \left[\widehat{\delta}_i,\ \widehat{RSS}_i, \widehat{\tau}_{i,j}, P, SEP_{50}\right] \tag{5.1}$$

where $\widehat{\delta}_i = \frac{\widehat{s}_i - \overline{E}_k}{\left\|\widehat{s}_i - \overline{E}_k\right\|}$ is a normalized UAS-emitter displacement, and additionally includes RSS, TDOA measurements, SEP_{50}, and the state error covariance P. In our architecture, the RSS is provided only as context and is not included in the reward function. The agents are trained to improve geometry for emitter localization, not to maximize RSS as in prior power-seeking approaches, which can produce degenerate TDOA geometries. However, because RSS correlates with signal quality and multipath conditions, exposing it in the state allows the policy to use it as a contextual indicator when making geometry decisions.

Each agent selects a bounded displacement vector:

$$a_i^k = \eta \widehat{\delta}_i \in \mathbb{R}^3 \tag{5.2}$$

including the normalized UAS-emitter displacement direction δ, and η which is an adaptive step size scaled to current SEP. During training, action exploration to the optimal geometry is further encouraged with an annealing ϵ-greedy policy.

The action selection is additionally guided by a 2-D-to-3-D phase strategy to optimize the LOCA geometry. Initial measurements utilize a 2-D localization step to reduce the circular error probable (CEP) area in an azimuthal plane and to coerce an array geometry into a ring-like or spherical formation around the estimated emitter location which is optimal for the LOCA methodology. Once the initial geometry is formed, the elevation estimate of the emitter location is refined through 3-D localization by leveraging array geometry diversity though evenly-spaced column lattice formations to improve SEP position and volume.

The reward function combines localization improvement, geometry shape, and convergence efficiency:

$$r_k = w_1 \Delta SEP_{50} + w_2 \Delta \left\| \mathbf{E} - \overline{E}_k \right\| + w_3 \widehat{\delta}_i - \kappa \tag{5.3}$$

where (w_1, w_2, w_3) are tuned weights, $\kappa > 0$ is a per-step penalty, and $w_2 > 0$ only during model training. The geometry shape reward encourages the cooperative UAS team to form an equidistant shape lattice around the emitter, similar to the best-performing heuristic approach identified in [59]. The architecture stops iterating when SEP_{50} meets a user-specified goal, and smaller SEP probabilities can easily be accommodated through the change of a parameter. The RL controller algorithm is shown in Algorithm 5, using the parameters defined in Table 8.1.

Algorithm 5: Emitter localization with UAS geometry control

Require : Initial UAS positions $\{s_i^0\}_{i=1}^N$; SEP target ς_{SEP}; initial estimate z_0; noise models $\varepsilon_{U,i}$, $\varepsilon_{t,ij}$, $\varepsilon_{P,i}$; training flag

Ensure : Final estimate $\overline{E}_k$, covariance P, achieved SEP_{50}, total steps, total distance

Initialize: $k \leftarrow 0, E \in \mathbb{R}^3 unknown, x_{0|0} \leftarrow z_0, P_{0|0} \leftarrow 10^6 I_3$

while $SEP_{50} > \varsigma_{\text{SEP}}$ ***and not*** *plateau* **do**

  Capture Signal Measurements:

  Draw τ_{ij} and RSS_i from E

  $\widehat{s}_i \leftarrow s_i^k + \varepsilon_{U,i}$

  $\widehat{\tau}_{ij} \leftarrow \tau_{ij} + \varepsilon_{t,ij}$

  $\widehat{RSS}_i \leftarrow RSS_i + \varepsilon_{P,i}$

  LOCA solution cloud: Form cloud Λ using Algorithm 1

  Prune outliers using (3.7)

  Cloud statistics: Compute $\widehat{E}$ and SEP_{50} using (3.8) and (3.9)

  Kalman Filter: Apply the position-only KF on (z_k, R_z) (random-walk $F = H = I_3$) during evaluation to generate $\overline{E}_k$ from $\widehat{E}$.

  RL geometry control:

  Build state using (5.1) ; select bounded action $\{a_i^k\}$ using (5.2)

  Enforce elevation diversity; compute reward using (5.3) ; update policy

  Advance geometry: $s_i^{k+1} \leftarrow s_i^k + a_i^k; k \leftarrow k+1$

return $\overline{E}_k$, P, SEP_{50}, *total steps k, total distance traveled*

5.6.3 Python Simulation Overview

The system was simulated using the Actor-Environment Cycle Environment class of PettingZoo in Python which was developed to simulate multi-agent environments with an observation space defined by Box from Gym [102]. We conducted 1000 Monte Carlo runs over a 4000 m×4000 m×500 m urban volume populated with Rayleigh-distributed scatterers following ITU-R P.1411 recommendations [83]. Emitters broadcasting a 5 GHz signal were randomly placed within the environment. An eight-element UAS swarm is deployed with randomized initial positions.

UAS elements are modeled as independently mobile systems with omnidirectional antennas and precise self-positioning capability. Clock accuracy is simulated as synchronized pre-mission by CSAC oscillators, and TDOA errors were drawn from Gaussian distributions consistent with CSAC-specified jitter parameters. RSS values are perturbed by log-normal shadowing and multipath fading. Measurement noise is injected at each step, with variance values selected to replicate realistic hardware tolerances. Multipath scattering and fading is modeled in accordance with ITU-R P.1411 [83]. Since these values are frequency and elevation-dependent and are stochastically specified, we consider a 5 GHz operating frequency with each UAS located at varying elevations giving primary signal contributions in a mix of LoS and non-LoS conditions such that we reduce the multipath effects by 50%. The above errors follow the methodology in [59] and are detailed in Table 4.3.

RL Training Setup The DQN was trained within our digital twin environment for 1000 episodes of random initial geometries with each episode consisting of up to 250 localization steps. Each episode terminated when the SEP radius fell below a mission-defined threshold (we use 5 m for the results described here) or the system performed 250 repositioning steps. Reward shaping followed the formulations described in Section 3. The resultant policy from the RL training was utilized in the same environment for an initial evaluation of the DQN.

5.6.4 EXata Digital Twin Environment Simulation Overview

The EXata network simulator is integrated with the SMU CAR [104] to create a digital twin of both the UAS swarm and the RF environment. The following elements are included within our digital twin:

(1) Propagation: ITU-R P.1411 multipath propagation and Rician fading with configurable K-factor, simulated urban environment.

(2) Weather effects: rain attenuation modeled at 0.5 dB/km at 5 GHz; atmospheric absorption by oxygen and water vapor following ITU-R P.676 [86].

(3) Hardware errors: UAS hover drift modeled as Gaussian-distributed errors ($\varepsilon_p = 2$ cm); clock jitter drawn from Gaussian distributions with $\varepsilon_t = 10$ ns.

(4) Kinematics: UAS motion followed gravity field dynamics with wind perturbations.

At each snapshot, each UAS collects TDOA and RSS measurements generated by EXata's physical-layer accurate RF models and stores the measurements within an internal EXata system database. These measurements are parsed and passed to the DQN architecture that selects discrete displacement actions for each UAS based on the trained policy. Updated positions are then re-injected into EXata for the next iteration. This process allows for a hardware-in-the-loop ready framework for use in the future to evaluate how the same policy would be executed on the real-world UAS.

Using EXata, we evaluated two scenarios: (i) a Rician-faded urban multipath environment, and (ii) the same environment under degraded weather conditions. These two scenarios used the same set of 200 randomized emitter/UAS initializations and UAS system on-board positioning and clock error conditions. Results were compared with DQN simulations utilizing the same initial geometries and comparable multipath conditions.

5.6.5 Comparison of Simulation Results between Python Environment and EXata Digital Twin

Performance is evaluated by the number of steps required to achieve a target SEP_{50} radius, the error in the final localization estimate from the true target location, and the distance the system traveled to localize the emitter. Each of these performance metrics indicates the effectiveness of the system to accurately and efficiently localize an emitter. We analyze the mean and median performance of these metrics in addition to the performance distribution to understand the stability of the system in different scenarios.

5.6.5.1 Python Simulation Results

Using the policy developed in the RL training, the DQN is evaluated in the same dense multipath simulation environment for a first measure of performance prior to integration with EXata. As shown in Fig. 5.10, the starting SEP_{50} and localization error was ∼80 m. The initial reduction in localization error is due to the 2-D ring formation phase, and further improvement is found through the geometry refinement steps after the 2-D-to-3-D localization transition. The DQN simulation realized an average 75 % reduction in SEP_{50} in 10 steps and ∼50 % error reduction in 20 steps, which outperforms the average performance of the heuristic-based approach [59]. However, considering the large standard deviation for the system error, the performance bounds are weighted by some outliers provided in the distributions shown in Fig. 5.11 and Fig. 5.12. Here, SEP_{50} most often reached a minimum of ∼3 m with ∼5.4 m localization accuracy. In the DQN simulation, this accuracy required 15-25 repositioning steps where each UAS in the system explored around 200 m to find the optimal geometry.

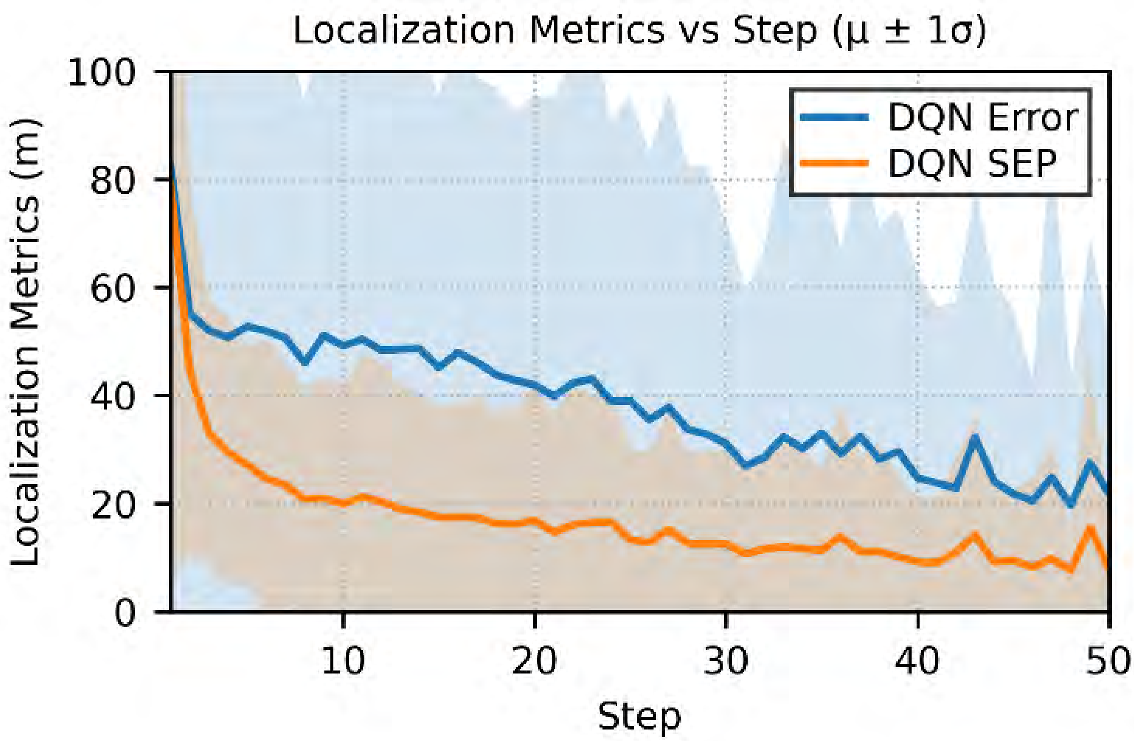

Figure 5.10: Localization error and SEP per repositioning step demonstrates convergence in dense multipath conditions.

While these results demonstrate that the DQN achieves significant error reduction in a controlled setting, the initial simulation model has inherent limits in representing multipath and environmental dynamics. The approach relies on generalized error terms Table 4.3) that inject noise and multipath contributors across trials, but these models do not fully represent all spatial conditions, especially as the UAS elements converge to more optimal geometries and improve their LoS to the emitter. This limitation sometimes made the results appear worse than what would be expected in practice. The conservative modeling was still useful, as it allowed the policy to train under challenging error conditions and respond appropriately when exposed to a range of multipath scenarios. To assess performance under more realistic RF and environmental dynamics, we next evaluate the system using the EXata digital twin.

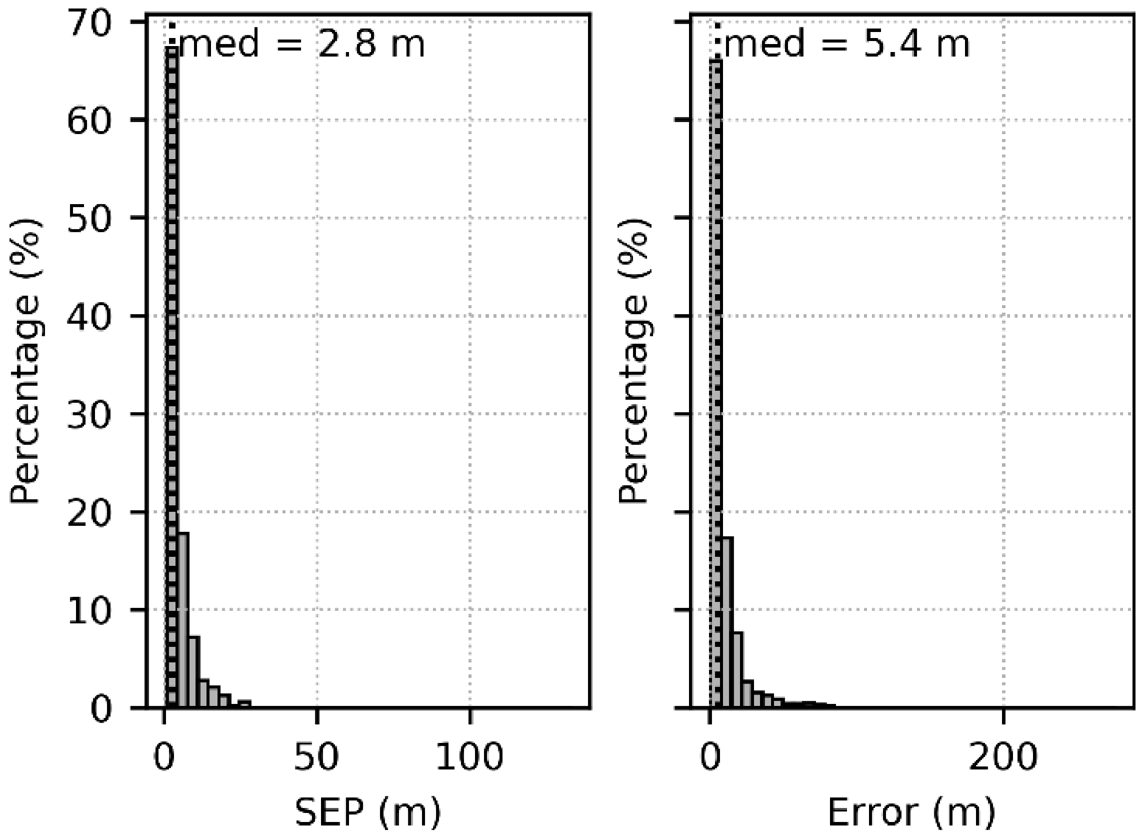

Figure 5.11: Localization metrics and median performance at convergence on emitter for 1000 DQN episodes, SEP radius (m) and localization error (m), show low median error and SEP for majority of evaluation episodes.

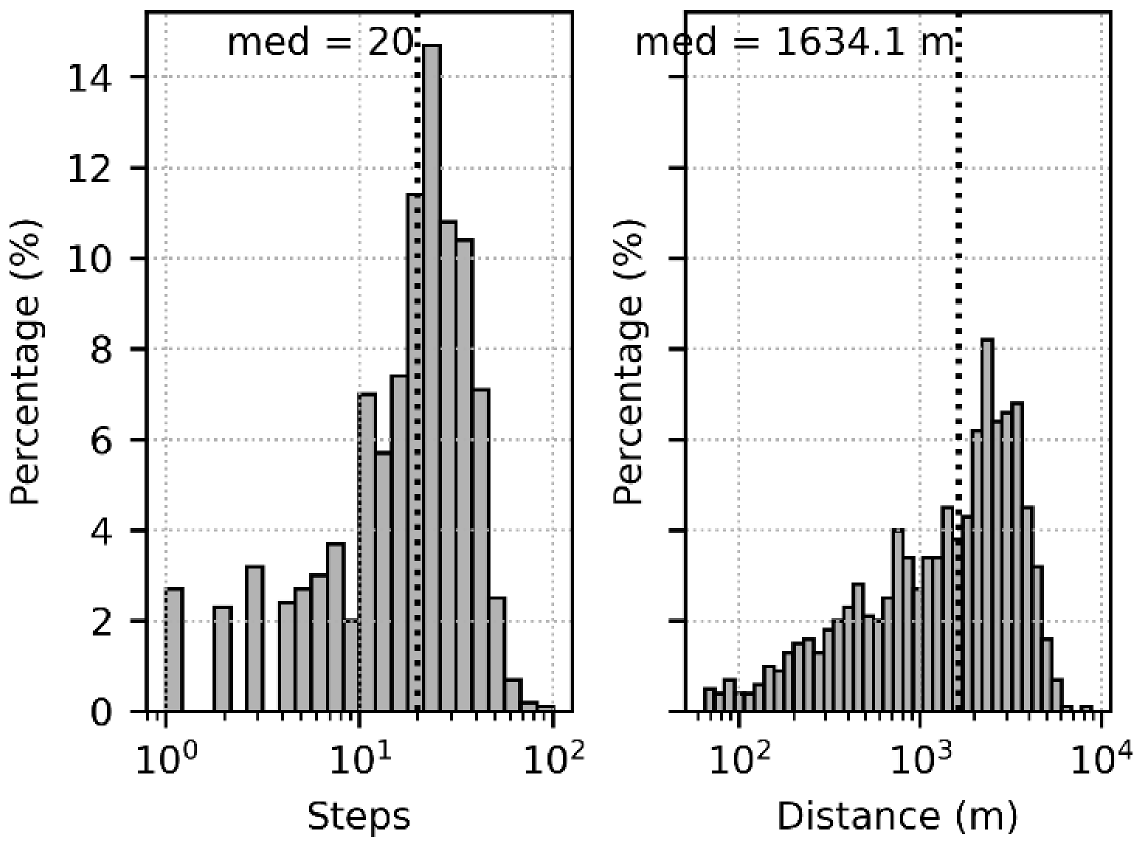

Figure 5.12: Repositioning metrics for 1000 DQN episodes show distance traveled by UAS system (m) and number of repositioning steps, with convergence in approximately 20 steps, traveling 200 m per UAS.

5.6.5.2 *EXata Digital Twin Results*

After evaluating the DQN RL policy in the dense multipath simulation environment, the DQN and policy were integrated into the EXata environment and analyzed under Rician fading with and without the presence of adverse weather conditions. The same initial geometries from the EXata simulations were also provided to the Python DQN simulation and evaluated for comparison. To better model the dominant LoS conditions of the Rician fading model in the Python DQN simulation, we reduced the Multipath Delay Spread factor in Table 4.3 by 50 %. This change in the environment also allowed us to evaluate our policy under different multipath conditions to ensure its suitability in other scenarios.

The primary evaluation metrics for these experiments are summarized in Table 5.4, showing the DQN with more stable performance in the EXata environment, requiring fewer repositioning steps to localize the emitter, and achieving an overall higher localization accuracy. Further, we see that the adverse weather conditions primarily impacted the initial system error, but once the geometry refinement began, the impacts were quickly overcome by the array geometry improvements indicating the high quality of the approach in a variety of weather conditions.

Full metrics are further explored in Fig. 5.13, Fig. 5.14, Fig. 5.15, Fig. 5.16. For the EXata-based evaluations, each approach reached a minimum error in ten steps, while the Python DQN simulation reached approximately double the average error at the ten-step mark.

Table 5.4: Summary metrics for Python DQN simulation (Sim.) vs. EXata digital twin study with Rician fading conditions and adverse weather.

Metric	**EXata Rician**	**EXata Weather**	**Sim.**
Mean final SEP (m)	4.1	4.1	3.9
Mean initial error (m)	18.5	22.7	76
Mean error reduction (%)	81	85	87
Mean step count	2.8	2.8	9.4
Mean system total distance traveled (m)	1544	1588	660

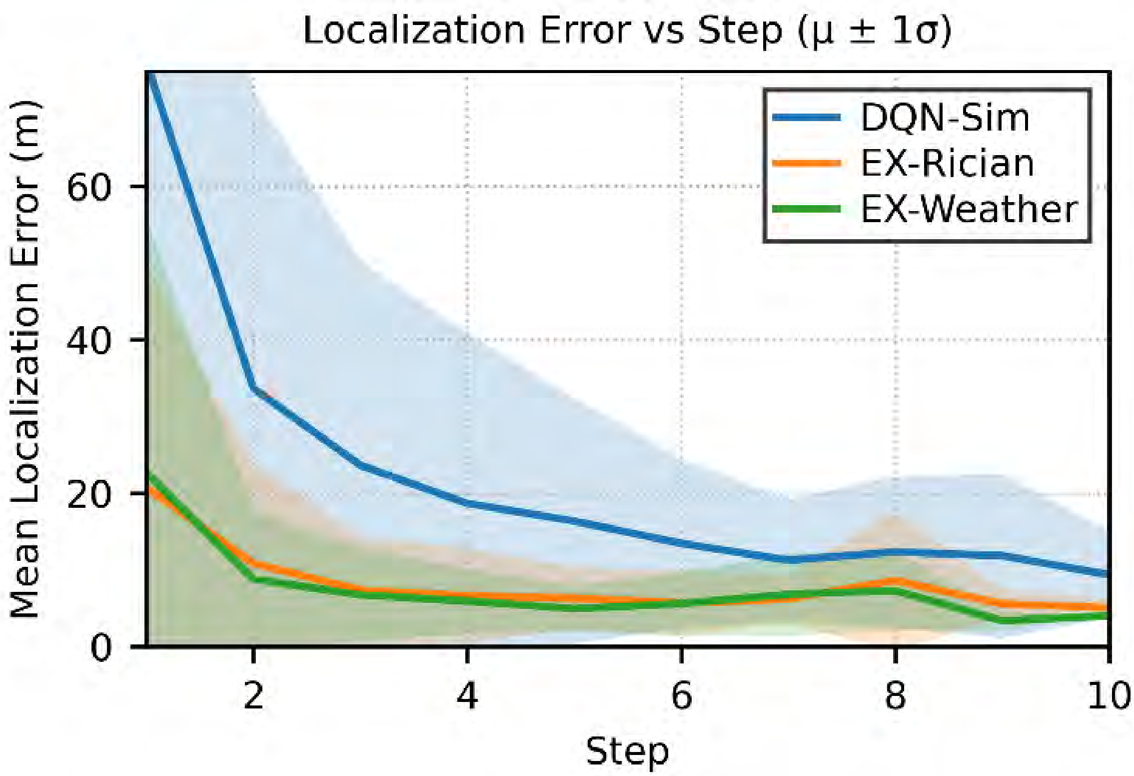

Figure 5.13: Comparison of average and standard deviation of localization error in the first ten reconfiguration steps among the Python simulations and the EXata digital twin trials shows convergence on emitter in approximately ten steps in Rician channel conditions.

To consider the general performance of the system, in Fig. 5.14, Fig. 5.15, Fig. 5.16 we overlay histograms of the primary metrics between the DQN Python simulation with the DQN EXata Rician scenario. In Fig. 5.14, the error response shows a higher distribution of low-valued errors for the EXata trials and a higher spread for the Python DQN simulations, although both had comparable median error in the 2.6 m -4.1 m range. In Fig. 5.15, the typical performance in EXata showed convergence in as few as three repositioning steps, whereas in the Python simulation the system explored more geometries and converged in closer to seven steps. While the system explored over fewer steps in the EXata trials, in Fig. 5.16 we observe the overall distance traveled in the EXata trials for the system is greater than that of the Python simulations.

Fig. 5.17 shows a 2-D visualization of one representative EXata trial. Two agents make

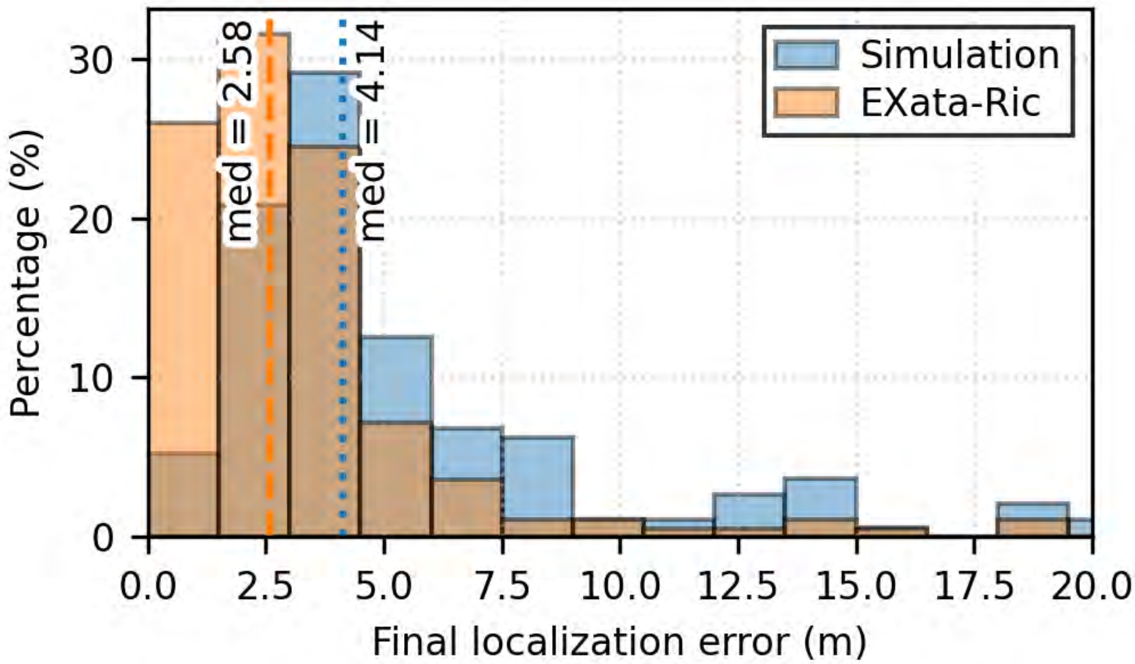

Figure 5.14: Histogram and median comparison of final localization error between Python simulations and EXata digital twin evaluations (<5 % of distribution > 20 m) shows low median error for majority of trials.

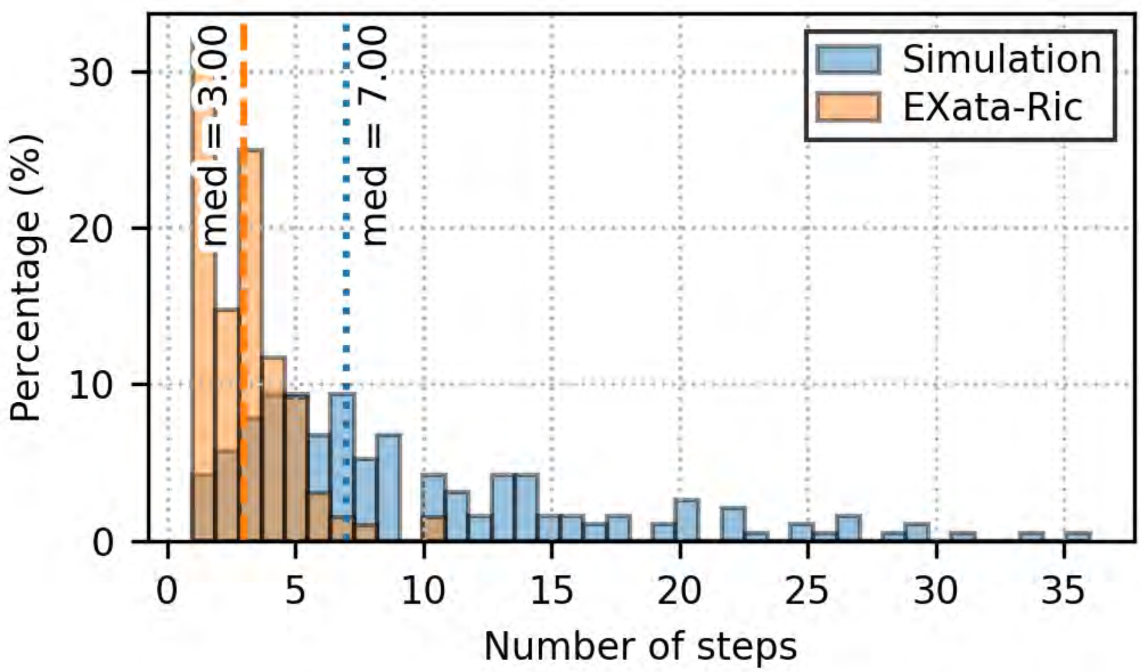

Figure 5.15: Histogram and median comparison of repositioning steps taken per episode between the Python simulations and EXata digital twin models shows majority of trials completing in under ten steps in both evaluation architectures

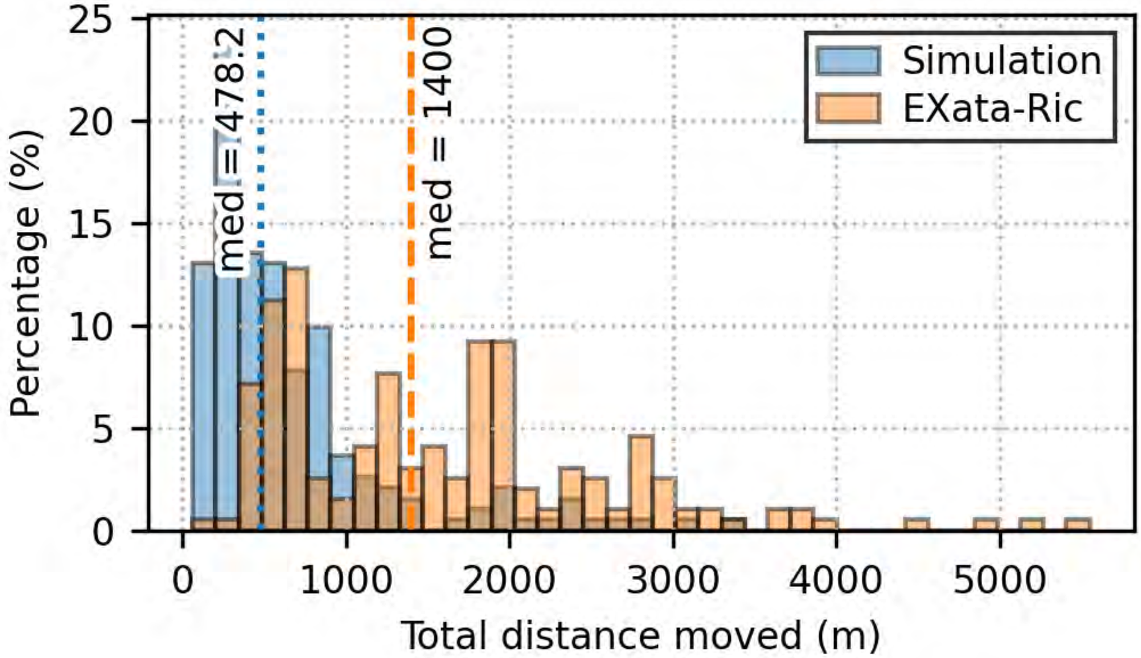

Figure 5.16: Histogram and median comparison of distance traveled by the UAS system between simulation and EXata shows distance traveled was much lower in Python simulation for similar convergence metrics.

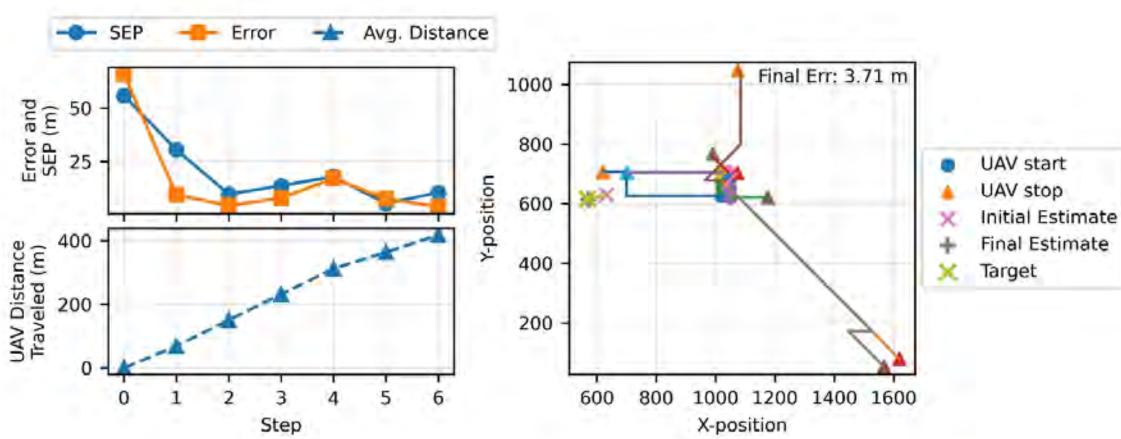

Figure 5.17: Visualization of EXata trial shows path strategy for localization convergence.

a large x-y excursion from their initial positions, which is an expected behavior that is seen consistently in both our DQN simulation and EXata aligned with our two-phase controller. The policy's SEP radius displacement vector guidance moves these agents mostly tangentially to correct the error in the azimuth plane with nominal z-axis change (not shown), and some correction back toward the ring geometry as they improve SEP and ultimately converge on a low localization error.

The EXata trials show an overall improved performance over the simulations, attributed to the robustness of EXata's multipath model versus the more stringent multipath error constraints given to the system in the simulation environment. In the simulation runs, we assume a mix of LoS and non-LoS conditions (with dominant LoS to approximate a Rician condition) but we provide no LoS advantage to the system when the UAS elements converge on the emitter. When the UAS elements are very close to, or even above the emitter, the system can experience pure LoS scenarios that increase the likelihood of an accurate measurement by reducing the error strictly to the clock and positioning hardware errors. In a more robust simulation environment like that provided by the EXata digital twin, when this convergence occurs in the first two or three repositioning steps, the system is more likely to realize a low-noise and low-scattering TDOA measurement and can rapidly converge on the emitter location. The trajectory shown in Fig. 5.17 follows an immediate 50 m drop in error after the first repositioning step. This indicates that the UAS system started with an undesirable geometry, then moved towards an improved geometry nearly immediately while continuing to iterate and refine that geometry for subsequent snapshots. In the Python simulations, the environmental model always assumed multipath error, even in pure LoS cases, reducing simulation realism in comparison to the EXata digital twin evaluations. Therefore, the improved performance observed with EXata is expected and is representative of a realistic operating environment with the presence of more variable dynamics.

Conclusion and Summary We presented a DQN-based framework for emitter localization with cooperative UAS, and we validated its performance in both Python-based simulations and digital twin evaluations. In the initial simulations, the DQN achieved significant error reduction, and demonstrated even better performance in the more realistic EXata trials. With each intelligent repositioning step, the system improves its emitter localization accuracy and is able to converge on the emitter location estimate, typically with very low error and low uncertainty in very few snapshots. In all trials, RSS contributed as a contextual feature in the observation vector but did not dominate agent behavior, confirming that convergence was driven by SEP reduction rather than power maximization. With the performance validation in EXata, the system achieves low localization error in fewer than ten snapshots, even under multipath and adverse weather conditions. These results advance RL-based cooperative localization toward real-world deployment in aerospace and other applications. Although we focus on a passive RF emitter localization application here, DQN-based dynamic array repositioning can be applied to many other applications that depend upon autonomous UAS swarming.

5.7 Summary

This chapter consolidated the geometry-adaptive localization workflow into a single evaluation thread. A heuristic controller provided a low-compute baseline and highlighted failure modes under severe multipath. Learning-based controllers, particularly shared-policy DQN and MADDPG, improved convergence behavior across randomized initial conditions by internalizing the coupling between array geometry and LOCA solution-cloud structure. Finally, a Python-to-EXata comparison verified that the principal performance trends persist in a higher-fidelity digital-twin environment, supporting the use of the Python simulator for large-scale training and evaluation with EXata-based verification for representative scenarios.

Chapter 6

EVALUATION OF FINAL ARCHITECTURE

6.1 Experimental Analysis of System Variables

The initial results of this section were submitted for publication in [60].

6.1.1 Final Architecture using Elliptical Error Probable

The UAS geometries are controlled by a deep Q-network (DQN) with two fully-connected layers, dropout regularization, and an experience replay buffer [48]. The DQN previously only considered action displacements which were a combination of direction with distance scaled by SEP. Instead, the DQN now includes an additional "speed" layer so that the displacement η in Eq. 5.2 is a learned parameter, allowing the system to better learn both direction and distance to travel as the system converges.

The DQN state features observe the previously-defined metrics (s_i, τ_{ij}, Λ_{f}, $\widehat{e}_m$, SEP_{50}, $RSS_{i\ldots N}$). In this architecture, we additionally include EEP_{Az} and EEP_{El} to better shape the geometry during convergence. Further, we track the Euclidean distance δ_i from s_i to $\widehat{e}_m$: $\widehat{\delta}_i = \frac{\widehat{s}_i - \widehat{e}_m}{\|\widehat{s}_i - \widehat{e}_m\|}$, considering the performance of [59] where localization accuracy improved significantly as the UAS found equidistance around the emitter. Altogether this generates the state information in Eq. 6.1:

$$\phi_k = \Big[s_i, \Lambda_{\mathrm{f}}, \widehat{e}_m, \widehat{\delta}_i,\ \widehat{RSS}_i, \tau_{i,j}, SEP_{50}, EEP_{Az}, EEP_{El}, P \Big]. \quad (6.1)$$

The DQN is rewarded by UAS position-based actions that (I.) reduce the cloud error metrics and (II.) stabilize $\widehat{\delta}_i$ so the UAS elements form a pseudo-column or sphere around the emitter, aided by RSS_i. Minimizing the per-axis EEP from the cloud allows the system to optimize its geometry in the field-of-view context and reduce the emitter localization error. Additionally, during training (when the emitter position is known), the network is rewarded by decreases in localization error L_e so that it receives additional features to learn optimized geometries. The reward structure is shown in Eq. 6.2:

$$r_k = w_1 \Delta EEP_{Az} + w_2 \Delta EEP_{El} + w_3 \Delta SEP_{50} + w_4 \Delta \Sigma RSS_i + w_5 \Delta \left\| \mathbf{E} - \widehat{e}_m \right\| + w_6 \widehat{\delta}_i - w_7 \kappa \tag{6.2}$$

where (w_i) are tuned weights, $\kappa > 0$ is a per-step penalty, and $w_5 > 0$ only during model training. The geometry shape reward encourages the cooperative UAS team to form an equidistant shape lattice around the emitter, similar to the best-performing heuristic approach identified in [59]. The architecture stops iterating when each of the shape metrics $EEP_{Az}, EEP_{El}, SEP50$ meets a user-specified goal, and tighter target metrics can easily be accommodated.

Last, it was found during experimentation that this final architecture could effectively perform with only six UAS elements in the network, whereas the previous analyses typically featured eight UAS elements. The reduction to six was aided by the EEP observations and rewards, the equidistance approach, and the separation of direction and distance in the DQN.

The system is trained with six UAS agents and the resultant policy is analyzed for consensus so that we apply the policy to swarm sizes of 7, 8, 9, and 10 UAS elements. Analysis is also performed with 5 UAS elements, however, this swarm size does not benefit from the use of cloud metrics to improve convergence.

While the geometry control policies in this chapter optimize localization-oriented objectives, safety behaviors such as inter-UAS collision avoidance and obstacle avoidance are not included in the reward or action constraints in the present study. In a deployed system, these behaviors should be enforced by additional control mechanisms and are considered beyond the scope of the development of the localization controller.

The evaluated architecture is provided in Algorithm 6, using the parameters defined in Table 8.1.

Algorithm 6: Emitter localization with UAS geometry control using EEP Metrics

Require : Initial UAS positions $\{s_i^0\}_{i=1}^N$; SEP and EEP targets $\varsigma_{\mathrm{Az}}, \varsigma_{\mathrm{El}}, \varsigma_{\mathrm{SEP}}$; initial estimate z_0; noise models $\varepsilon_{U,i}$, $\varepsilon_{t,ij}$, $\varepsilon_{P,i}$; training flag

Ensure : Final estimate $\overline{E}_k$, covariance P, achieved $EEP_{Az}, EEP_{El}, SEP_{50}$, total steps, total distance

Initialize: $k \leftarrow 0, E \in \mathbb{R}^3 unknown, x_{0|0} \leftarrow z_0, P_{0|0} \leftarrow 10^6 I_3$

while $SEP_{50} > \varsigma_{\mathrm{SEP}}$ ***and not*** *plateau* **do**

- **Capture Signal Measurements:**
- Draw τ_{ij} and RSS_i from E
- $\widehat{s}_i \leftarrow s_i^k + \varepsilon_{U,i}$
- $\widehat{\tau}_{ij} \leftarrow \tau_{ij} + \varepsilon_{t,ij}$
- $\widehat{RSS}_i \leftarrow RSS_i + \varepsilon_{P,i}$
- **LOCA solution cloud:** Form cloud Λ using Algorithm 1
- Prune outliers using (3.7)
- **Cloud statistics:** Compute $\widehat{e}_m$ using (3.8) and compute $EEP_{Az}, EEP_{El}, SEP_{50}$ using Algorithm 2
- **Kalman Filter:** Apply the position-only KF on (z_k, R_z) (random-walk $F = H = I_3$) during evaluation to evolve $\widehat{e}_m$ to $\overline{E}_k$.
- **RL geometry control:**
- Build state using (6.1) ; select bounded action $\{a_i^k\}$ using (5.2)
- Enforce elevation diversity; compute reward using (6.2) ; update policy
- **Advance geometry:** $s_i^{k+1} \leftarrow s_i^k + a_i^k; k \leftarrow k+1$

return *total distance traveled, total steps k,* $\overline{E}_k$, P, $EEP_{Az}, EEP_{El}, SEP_{50}$

6.2 Initial Evaluation Results

The system was simulated in a Python environment using the Gym and Pettingzoo modules for the DQN [102, 105]. Stochastic propagation models from [83] were implemented to simulate a dense urban multipath environment. The DQN was trained for 1250 random initial UAS and emitter positions with 6 UAS elements. To minimize repositioning steps, training was gated by annealing SEP_{50}, EEP_{Az}, EEP_{El} and L_e targets, so that as the policy learned macro optimal geometries in early stages of training, it would get refined to a goal of 2 m SEP and EEP. The annealing gates are visualized in Fig. 6.1 showing the achievement of a minimized threshold over training cycles.

The consensus policy was determined and applied for 7, 8, and 10 UAS swarm sizes. We first evaluated the policy across 250 episodes for each of 6-10 UAS swarm sizes and target gates corresponding to the final gates from training, that is, SEP_{50}= 1 m, EEP_{Az}= 2 m, and EEP_{El}= 4 m. The system localization performance, shown as the median with shaded

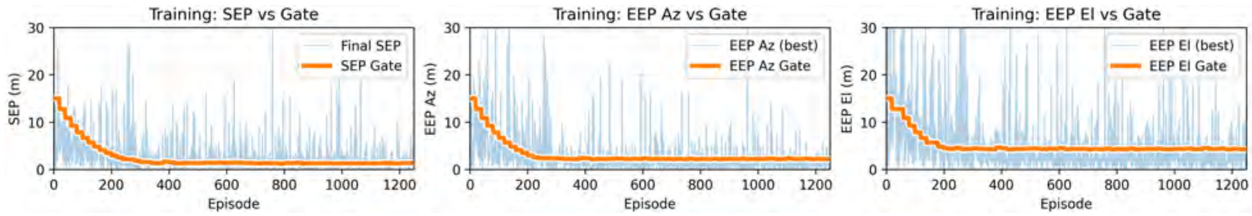

Figure 6.1: DQN geometry-aware cloud metrics targets annealing across training steps, with thresholds achieved: SEP_{50} ~1 m, EEP_{Az} ~2 m, and EEP_{El} ~4 m.

standard deviation at each repositioning step across all episodes per UAS size in Fig. 6.2, improves significantly and converges in as few as five steps for $7+$ UAS swarm sizes. The larger UAS sizes have denser solution clouds, enabling more refined estimates with fewer measurements. The metrics from these evaluations can be found in Fig. 6.3- 6.5.

We next re-evaluated the policy with tighter target gates of SEP_{50}= 1 m, EEP_{Az}= 1.5 m, and EEP_{El}= 2 m. These resulting metrics from these evaluations can be found in Fig. 6.6-6.8. We see in these that the tighter gates enable a lower localization error, while requiring more localization steps to achieve that error.

Overall, these results indicate that geometry-aware DQN control can rapidly converge the cooperative UAS toward optimal localization performance with minimal sensing steps. We depict a nominal performance of an 8-sensor system in Fig. 6.9, showing the initial random positions of the UAS and its convergence on the emitter.

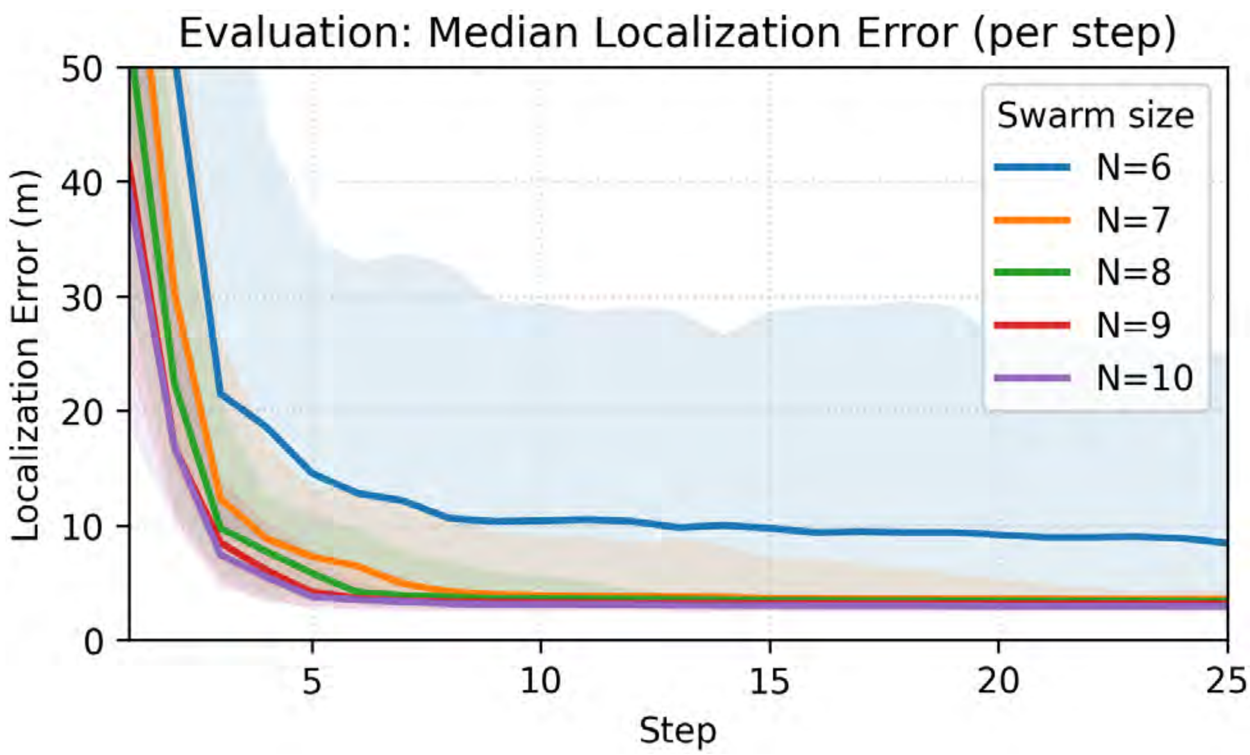

Figure 6.2: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation.

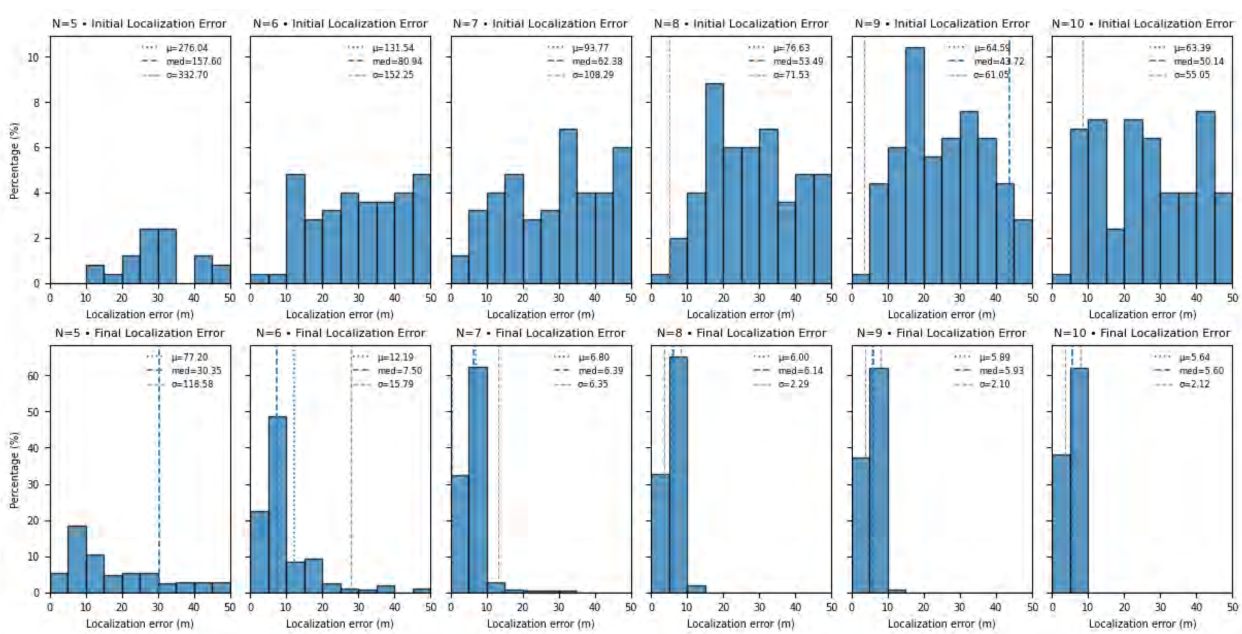

Figure 6.3: DQN final architecture comparison of initial and final localization error for varying swarm sizes.

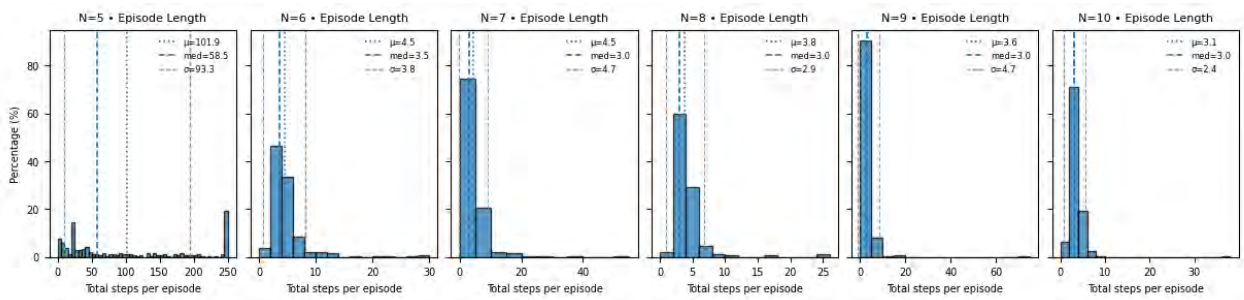

Figure 6.4: DQN final architecture episode length for varying swarm sizes.

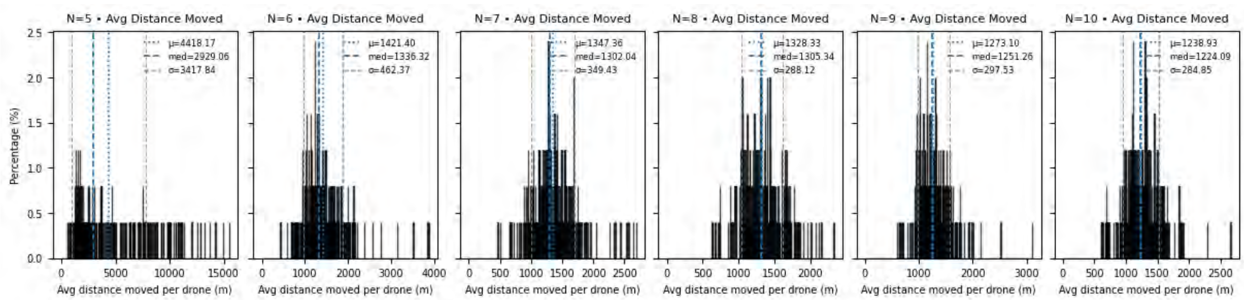

Figure 6.5: DQN final architecture average distance moved for each drone for varying swarm sizes.

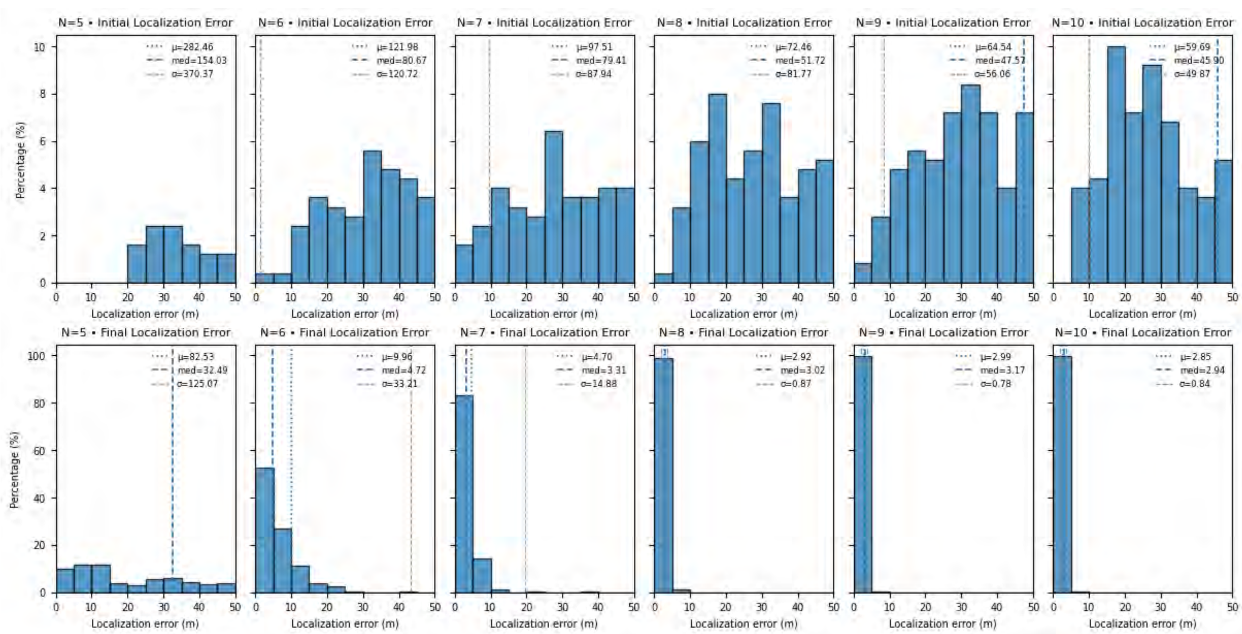

Figure 6.6: DQN final architecture comparison of initial and final localization error for varying swarm sizes, tighter target gates.

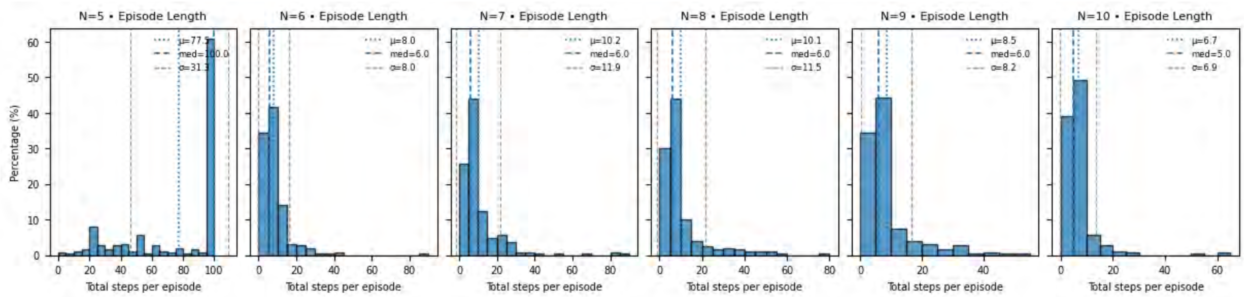

Figure 6.7: DQN final architecture episode length for varying swarm sizes, tighter target gates.

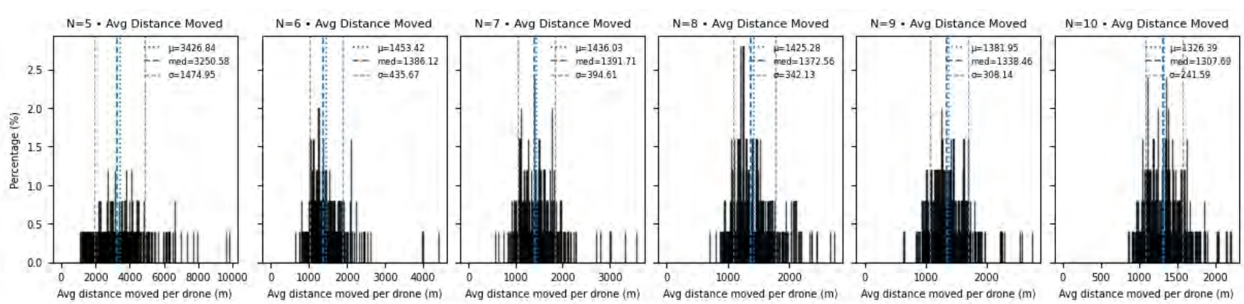

Figure 6.8: DQN final architecture average distance moved for each drone for varying swarm sizes, tighter target gates.

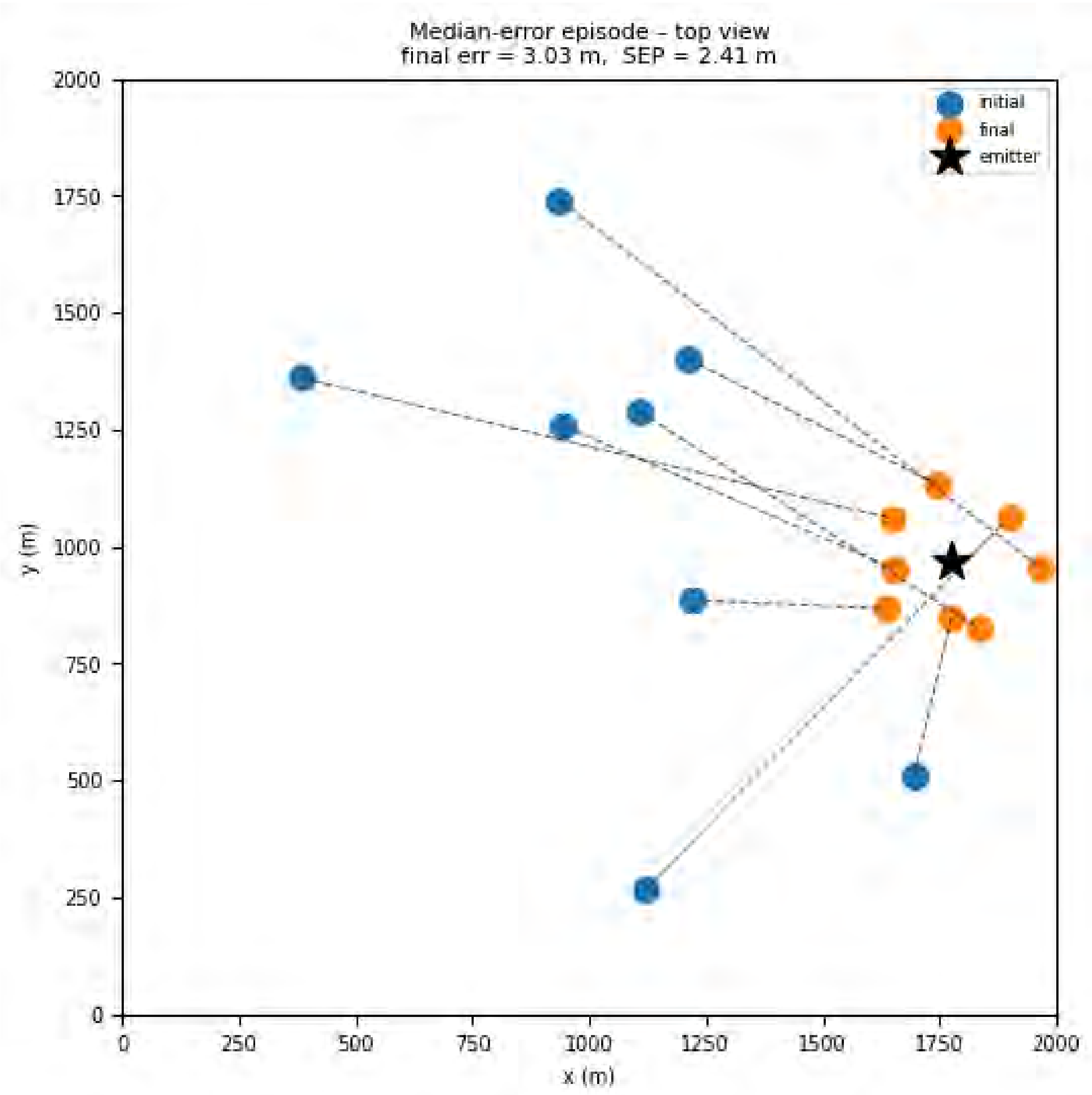

Figure 6.9: Initial and final positions of UAS for a median search event.

6.3 Further Analysis of Architecture

In this section, we vary environmental and system conditions to quantify the localization performance. The performance is quantified primarily by the following metrics: the mean localization error across all trials, and the number of repositioning steps or measurement snapshots required to converge in an accurate localization of the emitter.

Environmental Assumptions Unless otherwise defined, we consider the emitter operating as an omnidirectional transmitter, broadcasting a signal near 5 GHz with a power of 38 dBm (5 W). The setting is a dense multipath urban environment, approximately 2000 m x 2000 m x 500 m, and the emitter is in an unknown (randomized) location in this space. For the Python simulations, we assume a 50 % mix of LoS to non-LoS conditions.

System Assumptions Unless otherwise defined, we consider the UAS to operate with omnidirectional antennas, a LIDAR-aided self-positioning system that maintains 2 cm positioning accuracy, and a CSAC-enabled local clock reference that enables 10 ns timing accuracy. We baseline the performance using six sensors.

We assume the onboard system receivers are tuned to a signal near 5 GHz, have a sensitivity noise floor of -110 dBm and are able to detect emitter signals at 3 dB SNR. The sensors are initially located at a randomized location in the 2000 m x 2000 m x 500 m dense multipath urban environment. At maximum range, free space loss is ∼-116 dB, so the sensors would detect an 38 dBm emitter signal with power near -78 dBm, giving nearly 30 dB margin to the noise floor.

In consideration of the sensitivity, the system localization logic requires a minimum of five sensors to generate an emitter location estimate. If, for example, a 7–sensor system is in operation and at a given time measurement snapshot two of these sensors are receiving a signal below the sensitivity threshold due to low emitter power level, an off-main lobe meausurement with directional antennas, a non-LoS condition, or other adverse conditions impacting signal power, then only those 5 sensors receiving a signal above the threshold will generate a localization measurement.

Further, in most of the evaluation cases, we vary the performance from 5 to 10 UAS, showing the statistical results for each size. As a stopping criteria, we maintain the "tight" target gates of $SEP_{50} = 1\,\text{m}$, $EEP_{Az} = 1.5\,\text{m}$, and $EEP_{El} = 2\,\text{m}$.

Summary of Evaluation Results The summary in Table 6.1 shows several consistent trends in the evaluation results. Relative to the baseline configuration, reducing the array to 5 UAS substantially increases both localization error and episode length, while moving to 6 UAS recovers much of this loss and produces an immediate reduction in steps and error when estimate-distribution metrics are available to shape the geometry updates. Increasing the array to 10 UAS further improves performance, yielding lower error and fewer repositioning steps. Looser target gates reduce the number of steps required for convergence, but this occurs with a higher final localization error, indicating a tradeoff for localization speed–to–accuracy.

The remaining cases show how environment, initial geometry, and sensing assumptions affect convergence behavior. Directional antennas (Sec. 6.3.1) maintain comparable median error but increase error variability and total travel, indicating reduced convergence stability relative to the omnidirectional case. Lower emitter power (0 dBm, Sec. 6.3.2) increases the number of localization steps, and expanding the search area (Sec. 6.3.3.1) from 1.5 mi to 10 mi produces a large increase in episode length and travel distance even when the controller still achieves strong median error reduction from the initial estimate. Collinear starts (Sec. 6.3.3.2) increase steps and destabilize the localization error, whereas grid–like structured starting positions with sufficient spacing improve stability and reduce steps relative to the collinear case. Larger positioning and timing error (Sec. 6.3.4) are overcome with additional localization steps, as are adverse (100% NLoS) propagation conditions (Sec. 6.3.5). Finally, the 5 GHz-trained policy retains useful performance when evaluated at 2.4 GHz and 8 GHz (Sec. 6.3.6), supporting frequency-transfer behavior for the geometry-driven controller, with the 2.4 GHz case requiring substantially more convergence steps and the 8 GHz case remaining closer to the baseline in step count.

In the following analysis, the experimental results are presented as histograms of key performance parameters and as plots of stepwise median localization error, similar to Fig. 6.2.

For the data underlying the median localization error plots, each evaluation episode was stored as a fixed-length stepwise array initialized with NaN values. As the episode advanced, the observed localization error values replaced the corresponding NaN entries, while any steps not reached before termination remained NaN. Thus, early-converging episodes appear

Table 6.1: Summary of performance evaluation results for the system architecture across various test conditions. Values shown are summary statistics across evaluation episodes for the listed comparison case.

Case	Med. Final Loc. Error (m)	σ Final Error (m)	Med. Error Reduction (%)	Med. Episode Length (steps)	Med. Distance per UAS (m)
Baseline	3.0	0.9	94%	6	1373
5 UAS	32.5	125.1	79%	100	3251
6 UAS	4.7	33.2	94%	6	1386
10 UAS	2.9	0.8	94%	5	1308
Looser Target Gates	6.1	2.3	89%	3	1305
Directional Antennas	3.3	12.5	94%	6	1504
0 dBm Emitter	3.4	44.7	95%	7	1484
10 mi Area	3.4	39.0	99%	114	8436
Collinear starts	3.4	92.9	99%	24	1933
Structured starts	3.2	32.4	98%	10	1683
Large Positioning & Timing Error	3.3	1.1	95%	7	1357
25% Mix NLoS-to-LoS	2.9	1.2	90%	3	1290
100% Mix NLoS-to-LoS	3.4	2.7	96%	20	1412
2.4 GHz eval (5 GHz train)	3.4	2.4	96%	16	1429
8 GHz eval (5 GHz train)	2.9	4.9	93%	4	1286
EXata Omnidirectional	4.7	90.6	97%	5	535
EXata Directional	5.8	544.1	98%	8	1227

Notes: The baseline row corresponds to the nominal Sec. 6.2 evaluation conditions. Median Error Reduction (%) is computed as $100\left(1 - \dfrac{\text{median final error}}{\text{median initial error}}\right)$. Rows are reported relative to the baseline configuration unless otherwise indicated in the case label.

as shorter valid trajectories within a common array structure, preserving both the observed data and the point of termination. Because episodes may converge and terminate at different steps, the stepwise performance plots carry each episode's final observed value forward after convergence so that later-step summaries remain representative of the full set of evaluation runs. This approach avoids later-step summaries being computed from only the subset of episodes still active at those steps.

6.3.1 Comparison of Omnidirectional to Directional Antennas

The study had assumed that the sensor on each UAS was omnidirectional, but if the sensor had directivity and favored a gain in a particular orientation, the SNR would be maximized when the antenna pattern peak of the UAS sensor was pointed in the direction of the emitter. Using directional antennas, the system performs an initial measurement, followed by an estimate of the emitter location, then an estimate of the bearing of the emitter to each UAS element. Each UAS element utilizes their individual emitter bearing estimate to rotate in place so that the sensor antenna main lobe is pointed in the location of the emitter. When that occurs, each subsequent measurement is taken with an increased RSS. These evaluations assume some level of polarization matching, as the case where the polarization of the sensor is not matched to that of the emitter (horizontal linear vs vertical linear, for example), additional significant losses in signal power will be present.

To study this approach, a mapping of the antenna pattern of the UAS sensor was developed along with a method to correlate the UAS orientation with respect to the emitter bearing. Once the initial bearing is determined, the pattern is analyzed with a look-up table to correlate the antenna gain in the initial direction with the RSS versus the expected RSS when the maximum gain from the peak of the antenna pattern is aligned to the bearing of the emitter. If the RSS is determined to increase, the UAS then rotates to that new orientation.

Futher, for this directional antenna case, we did not generate new policies for the reinforcement learning architecture; instead, we relied on the policies generated for the omnidirectional antennas. Our assumption for transferring the policy is supported by the deterministic control of the UAS heading towards the peak direction of the antenna pattern. Once the UAS antenna is directed towards the emitter, the directional UAS system can experience a higher magnitude RSS than the omnidirectional case for the same UAS geometry. Therefore, all aspects of the measurement and geometry apply for the directional antenna system.

With the implementation of the antenna pattern, while there is significantly more gain in the main lobe direction over the gain of the omnidirectional antennas, there is also additional loss in the other antenna pattern regions. Because this loss can be significant based on the depth of the antenna pattern nulls, we ensured that the simulation maintained a realistic sensitivity threshold of -110 dBm so that the system would not capture signal measurements

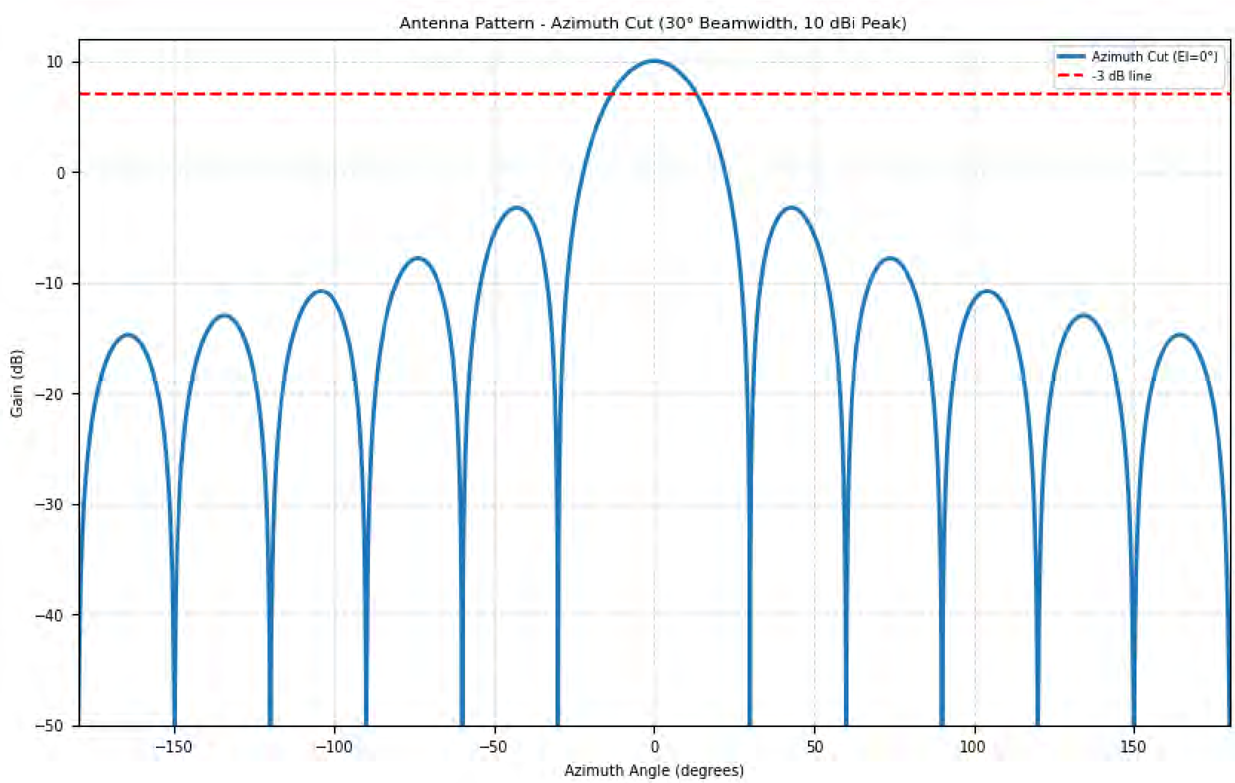

Figure 6.10: Directional antenna pattern used for simulations: 10 dBi gain with 30 ° 3 dB beamwidth.

below the noise floor. Thus, at each measurement snapshot, if a signal measurement for a particular sensor was below the sensitivity threshold, that sensor would not be utilized in the localization estimate. Because the system needs at least five sensors for localization, if fewer than five sensors had sufficient sensitivity, the snapshot would be passed and the system would rely on the previous localization estimate at the new location.

For the initial simulations, we used an antenna pattern based on the *sinc* function with 10 dBi gain and 3 dB beamwidth of 30 °. An example azimuth slice of this pattern is shown in Fig. 6.10, with a similar pattern in elevation.

Simulations Here we show the results of the evaluation of a UAS with directional antennas. In the localization error per step plot below, we see the system converge within the first few snapshots and that array sizes of 7-10 sensors maintain convergence with additional snapshots. However, the 6-sized array does not perform as well and shows excessive divergence at larger step sizes. This is primarily due to the sensitivity threshold when the emitter is not within an acceptable field-of-view of specific sensors reducing the ability of the UAS to use its full sensor set to localize the emitter. The misalignment to the field-of-view is most likely due to an incorrect collective localization estimate and move that placed the emitter off the main lobe of the antenna pattern. In these cases, omnidirectional antennas have an advantage because they are able to sense RSS equally in all directions, providing a greater likelihood of sensing as the UAS converges on the emitter.

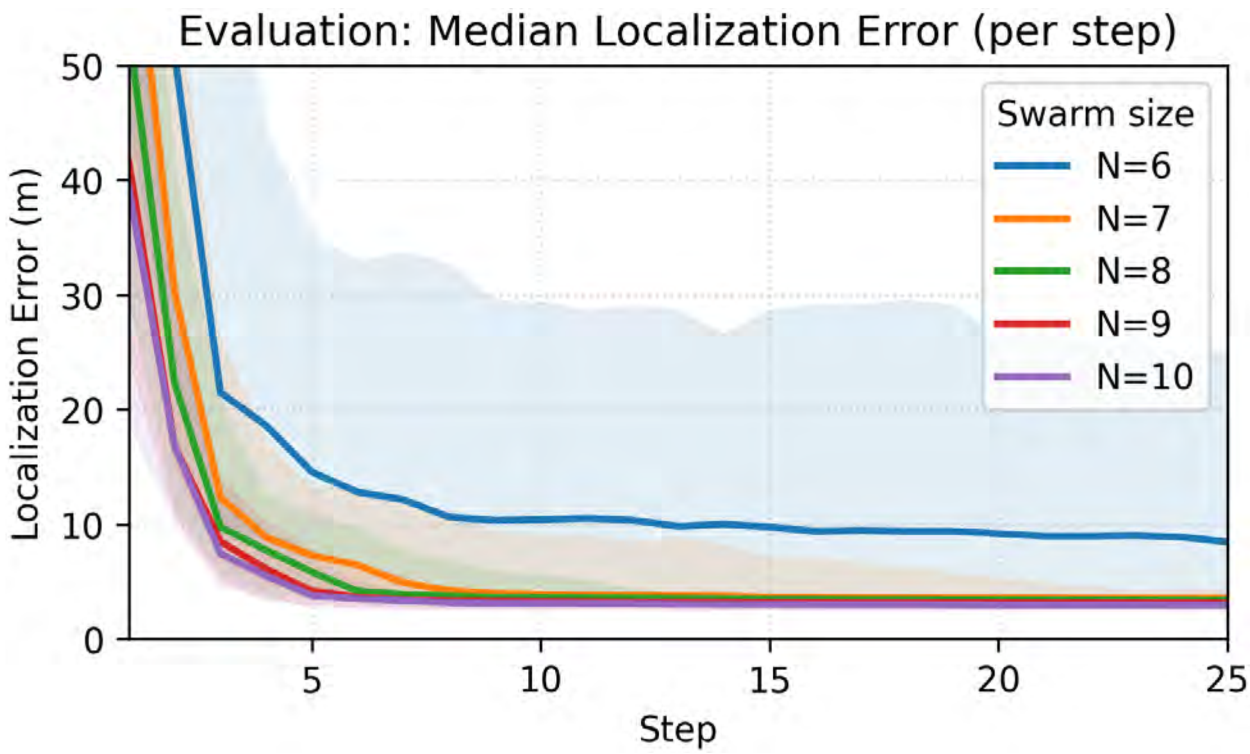

Figure 6.11: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas.

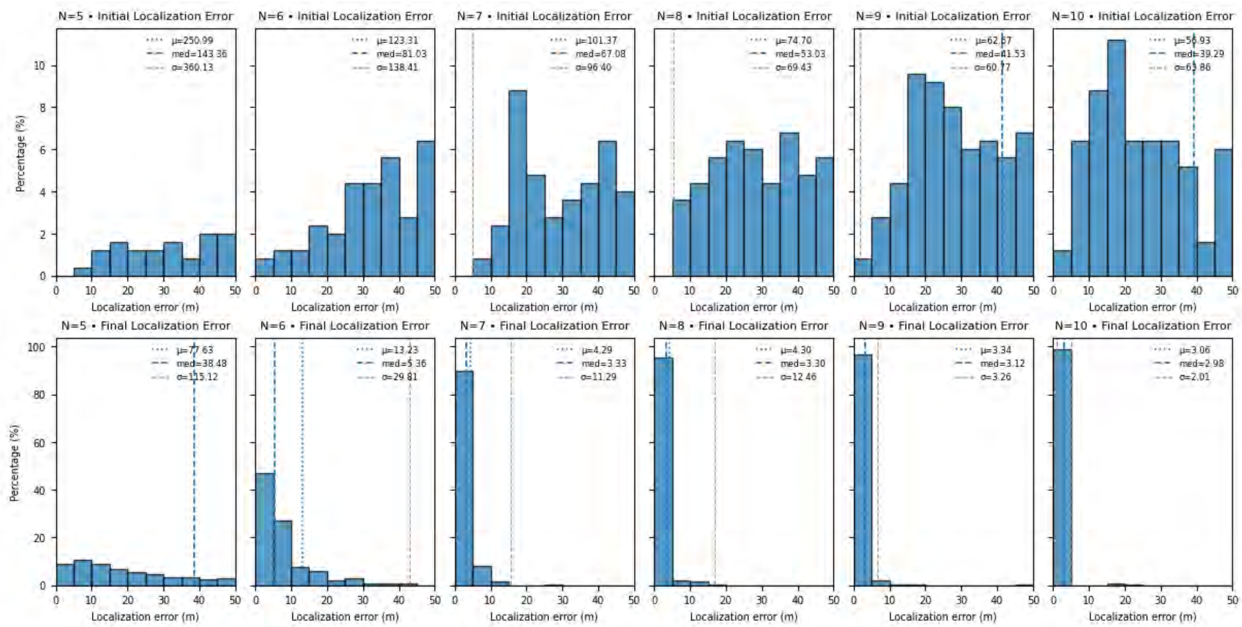

Figure 6.12: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas.

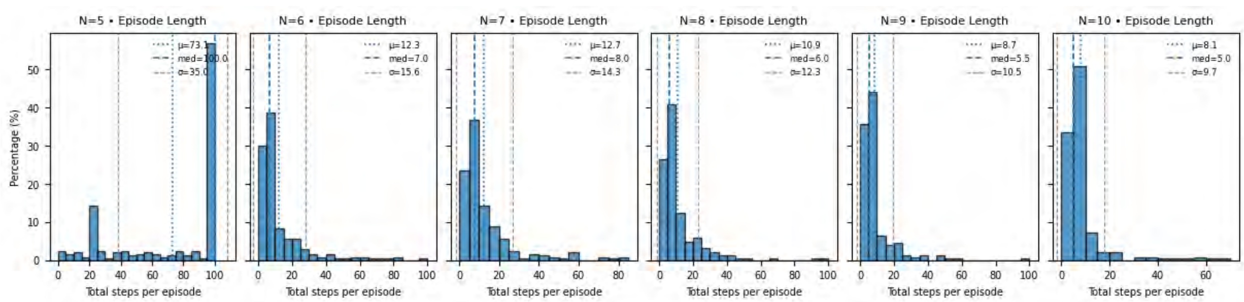

Figure 6.13: DQN final architecture episode length for varying swarm sizes and directional antennas.

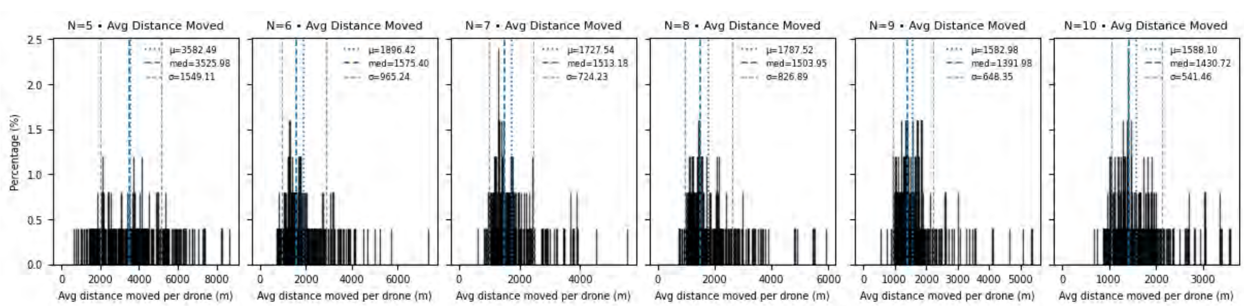

Figure 6.14: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas.

6.3.2 Sensor Receiver Sensitivity / Emitter Power Impacts to Performance

In the system operating environment, depending on the environmental factors, the distance between the sensors and the emitter, and the factors inherent to the individual sensors, the SNR will vary with sensor location and orientation. Here, we consider the impact of the emitter power to the receiver sensitivity for a signal capture.

We have previously assumed a constant output power of the emitter at 30 dBm. We now vary that output power to determine the impacts to the localization performance as the system senses a lower power signal. Wwe maintain the same 2000 m x 2000 m x 500 m range (approximately 1.5 miles at the maximum range) between the emitter and sensor and then vary the emitter power. At maximum range, a 5 GHz 0 dBm signal would be marginally sensed.

6.3.2.1 Omnidirectional 0 dBm

We confirm the degradation of performance with a 0 dBm emitter. The 6-sensor system has the most significant degradation to localization error and episode length, while the 7-9 sensor systems have larger errors primarily seen in the increased standard deviation on all cases with significantly higher mean error for the 7-sensor case.

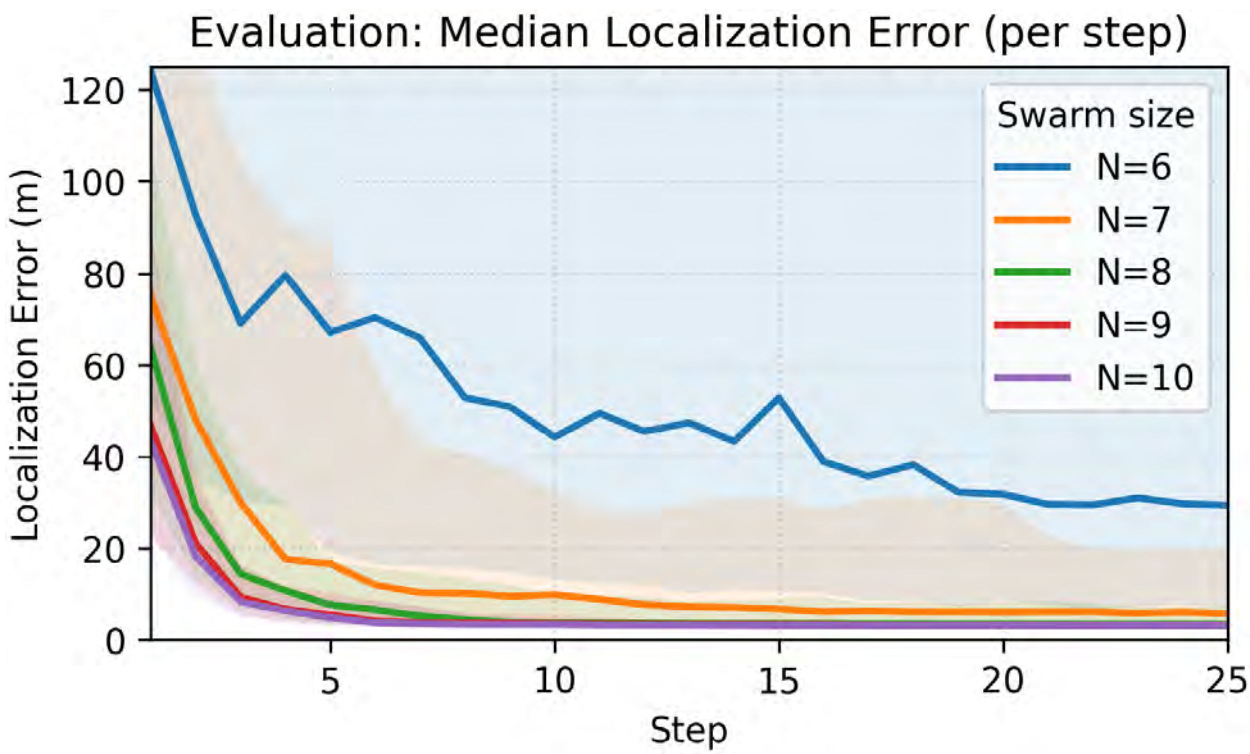

Figure 6.15: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas and emitter power 0 dBm.

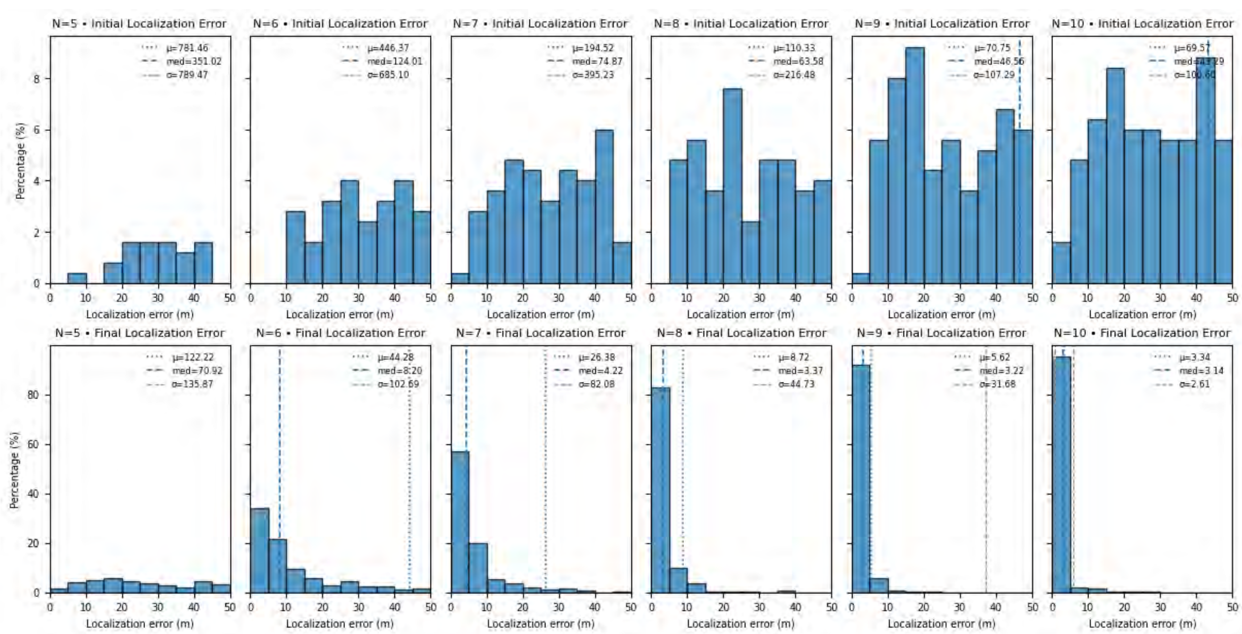

Figure 6.16: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas and emitter power 0 dBm.

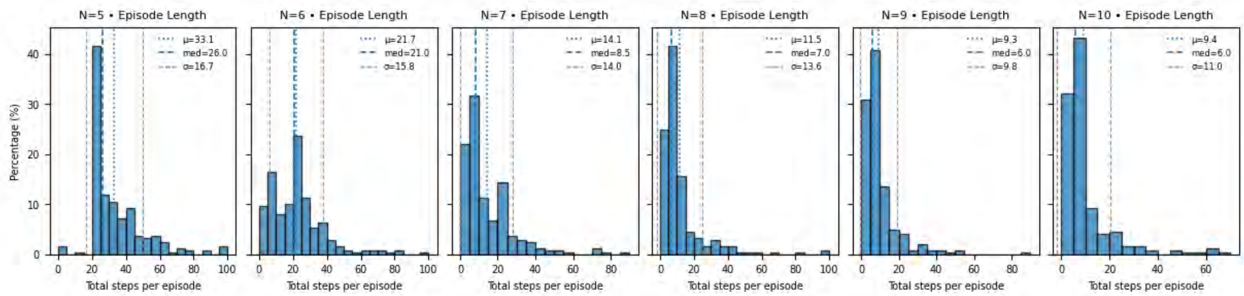

Figure 6.17: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas and emitter power 0 dBm.

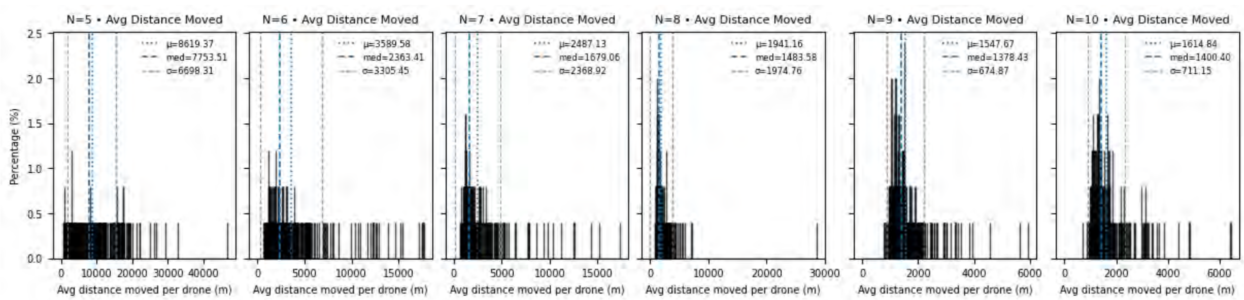

Figure 6.18: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas and emitter power 0 dBm.

6.3.2.2 Directional 0 dBm

For the directional case, we see similar degradation for a 0 dBm emitter signal as we saw with the omnidirectional case. However, because the directioanl system has 10 dBi gain in the main lobe, we additionally consider the localization error per step for -10 dBm emitter case in Fig. 6.20. Here, while some convergence can be seen, it is clear that this power level is not acceptable and the system will not converge.

6.3.3 Geometry Impacts

6.3.3.1 10 miles

First, we maintain a 38 dBm emitter power and extend the range to 12000 m x 12000 m x 500 m (approximately 10 miles at the maximum range). In Fig. 6.24, we see that the system is able to converge, however, the convergence requries signficantly more snapshots.

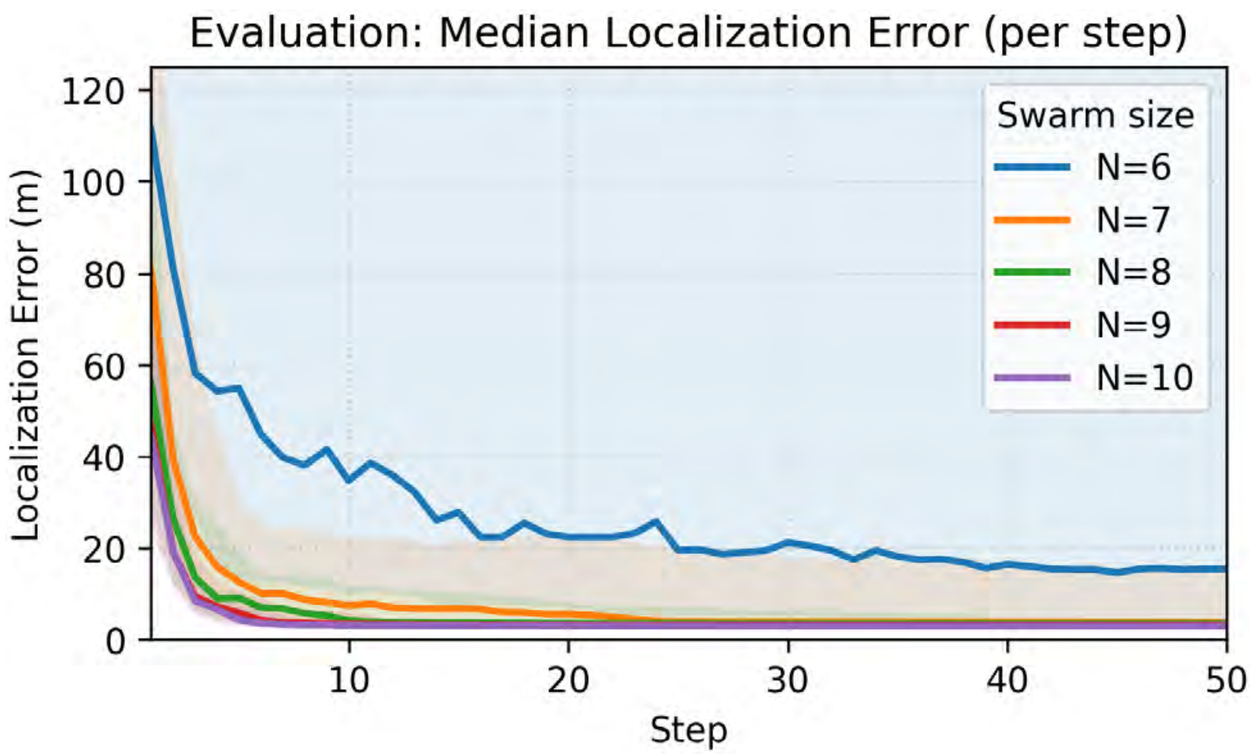

Figure 6.19: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas and emitter power 0 dBm.

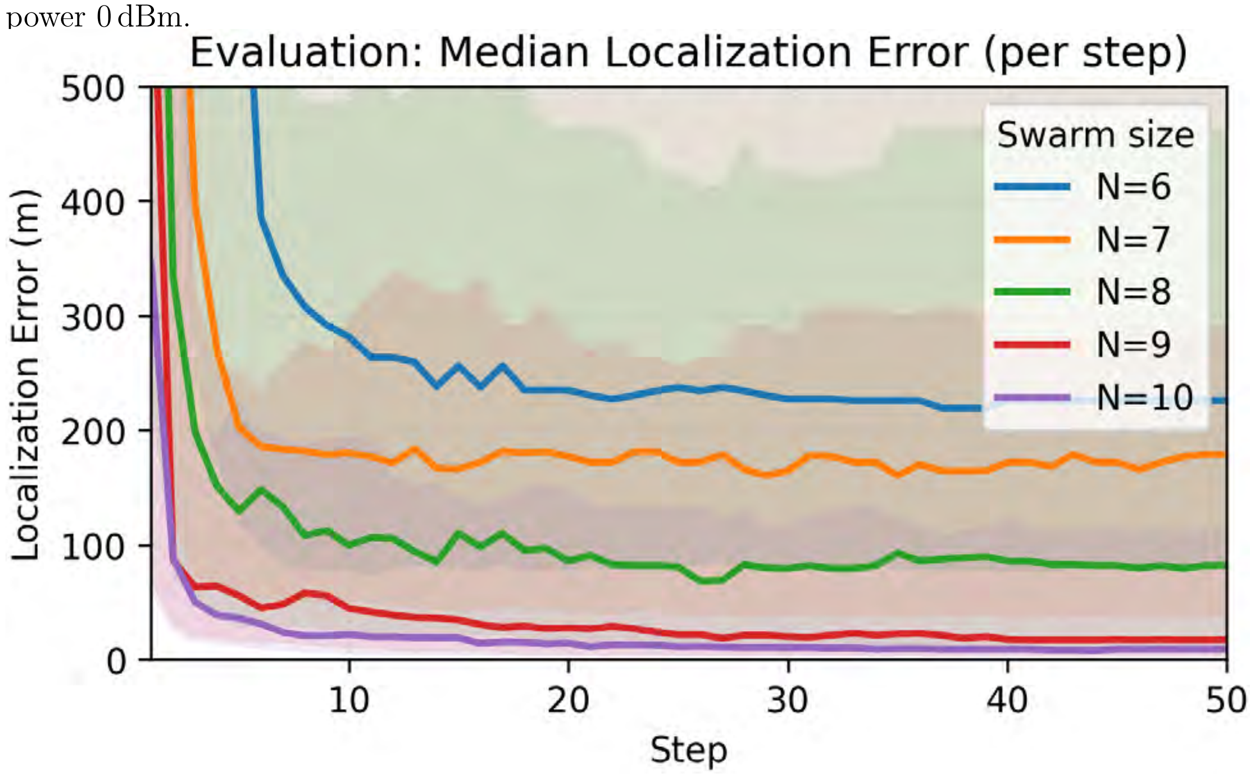

Figure 6.20: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas and emitter power -10 dBm.

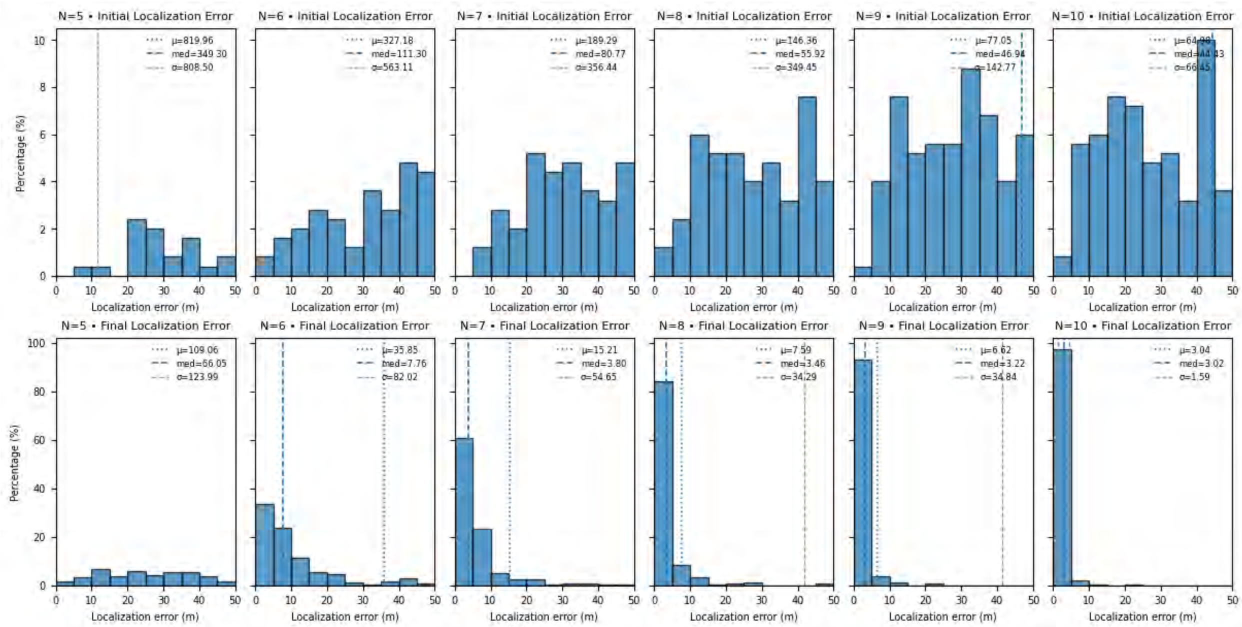

Figure 6.21: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas and emitter power 0 dBm.

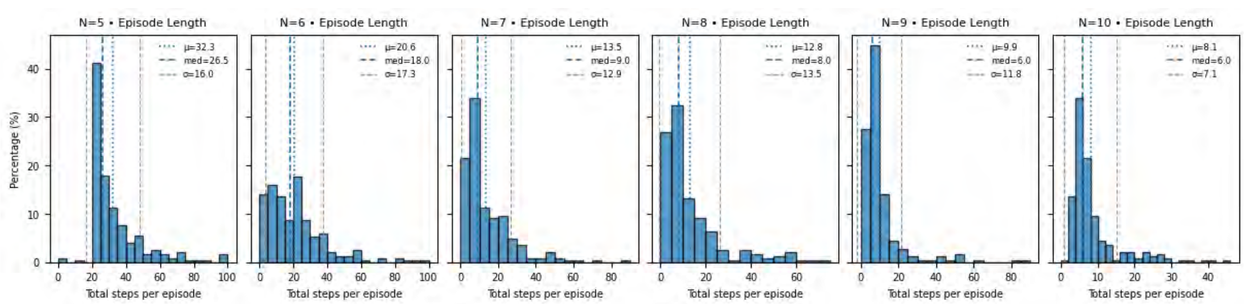

Figure 6.22: DQN final architecture episode length for varying swarm sizes and directional antennas and emitter power 0 dBm.

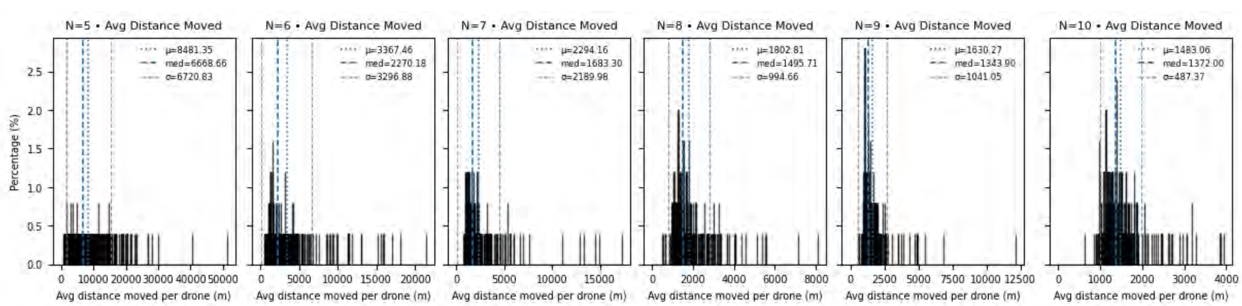

Figure 6.23: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas and emitter power 0 dBm.

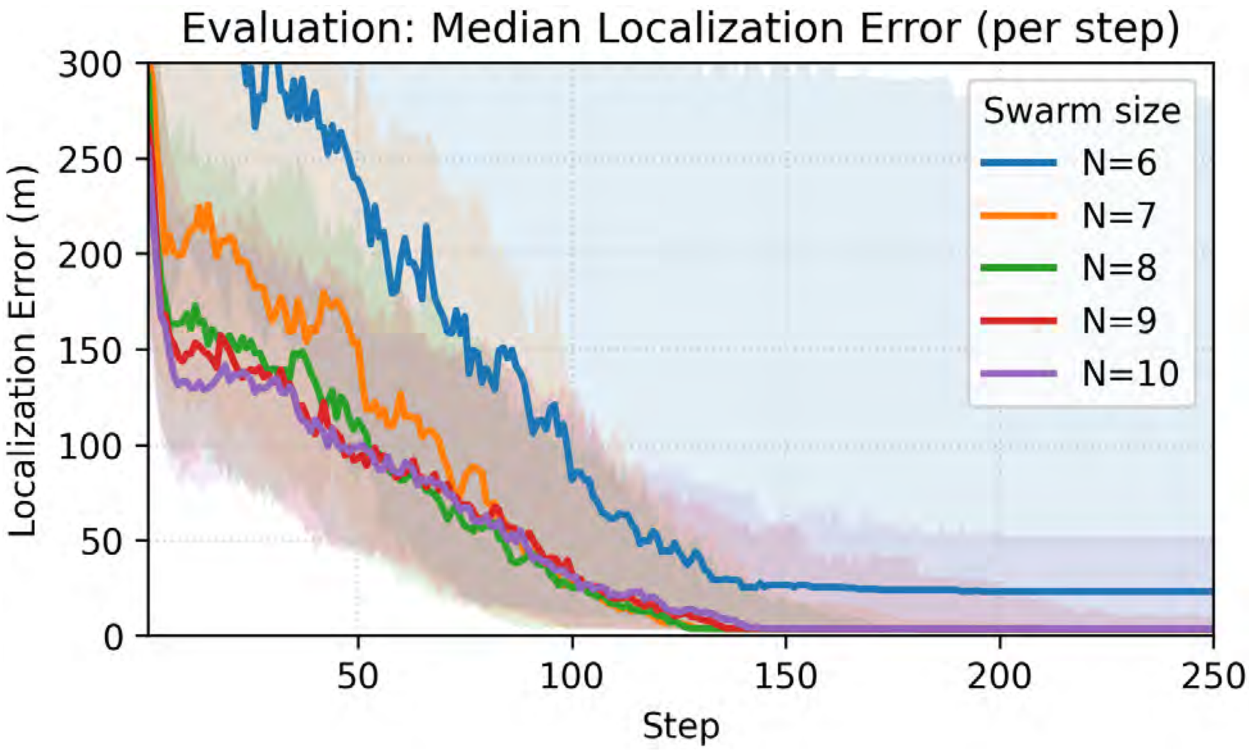

Figure 6.24: Localization error per step with sensors beginning at very long range. System achieves convergence but requires many measurement steps.

Because the system was trained for a much smaller operating region, it is conceivable that the "distance" layer of the DQN policy requires additional, longer range, geometries to train against to enable faster convergence.

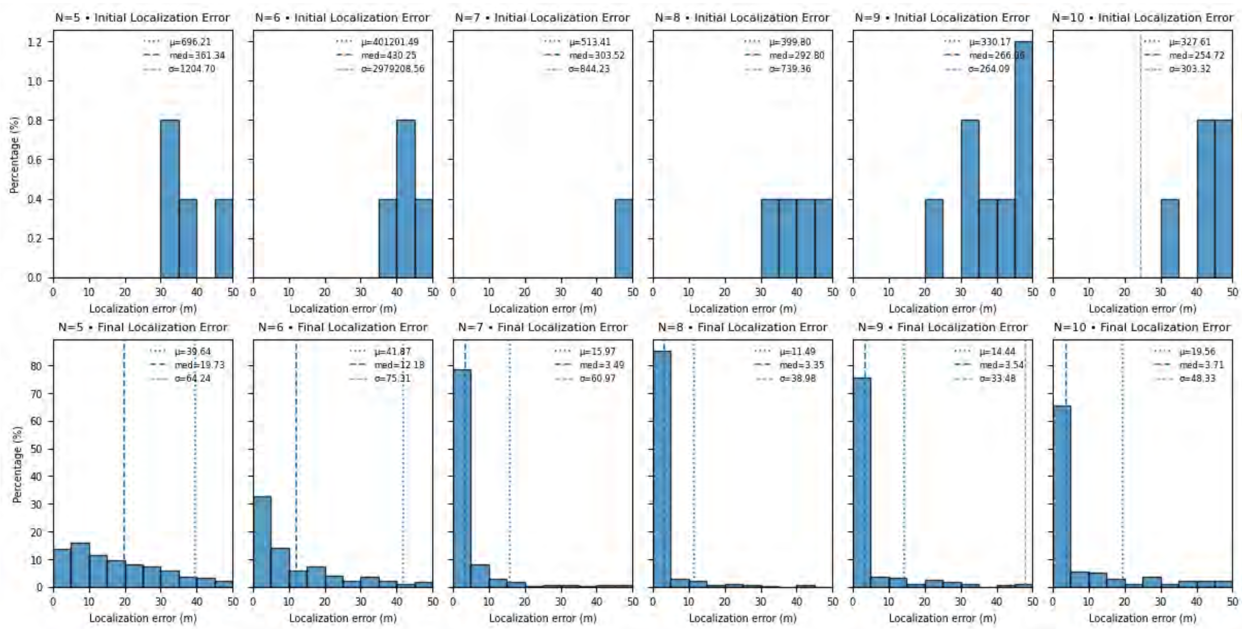

Figure 6.25: Comparison of initial and final localization error for long range performance with varying swarm sizes.

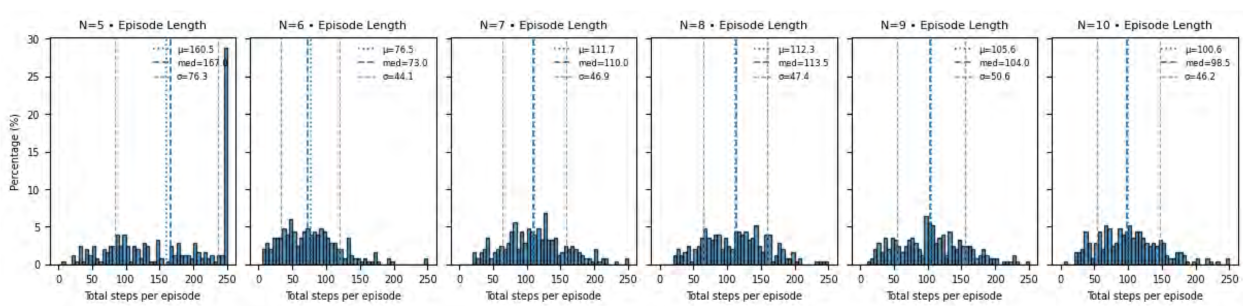

Figure 6.26: Episode lengths for varying swarm sizes and long range localization.

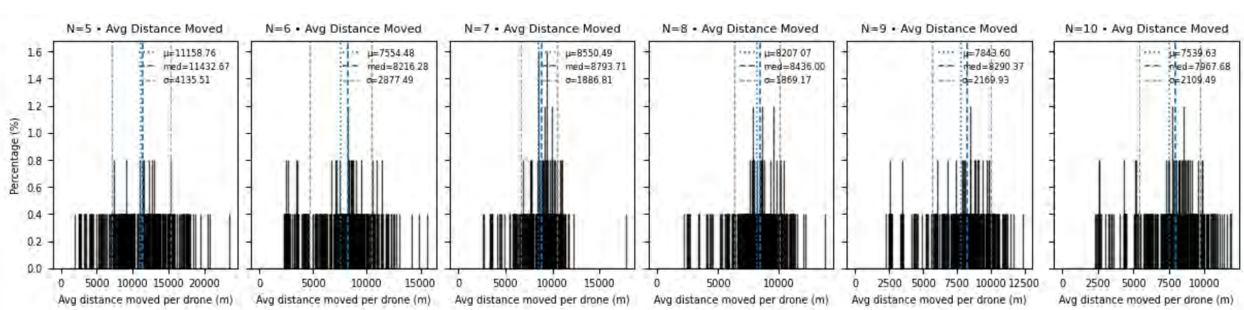

Figure 6.27: Long range localization average distance moved for each drone for varying swarm sizes.

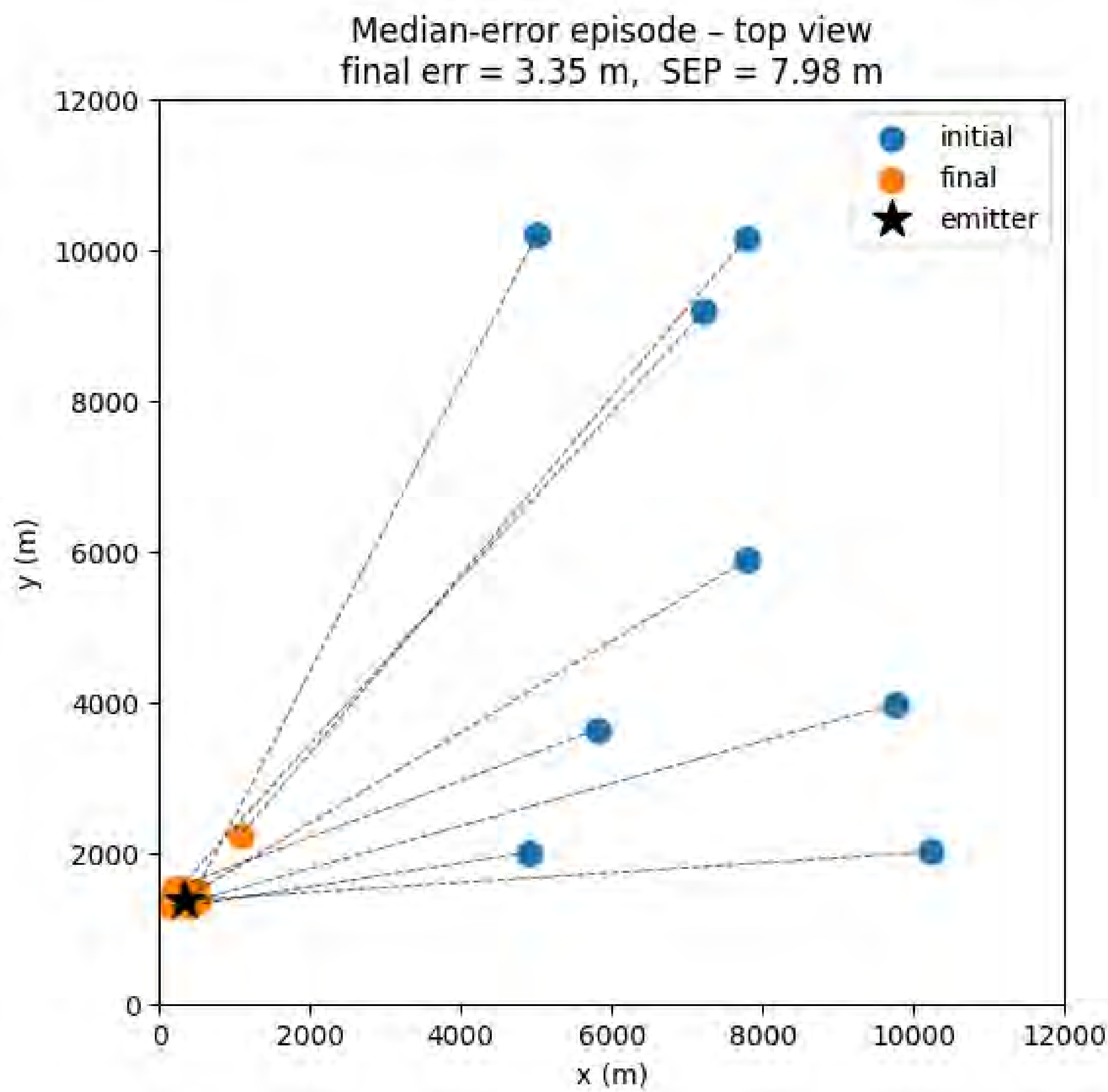

Figure 6.28: Long range localization trajectory for a nominal case.

6.3.3.2 *Non-Random Starting Positions*

Next, we evaluate trials where the initial UAS positions are not random but instead pre-planned configurations. In these cases, the emitter positions are varied for each episode but we maintain the same starting positions across a trial set of episodes.

Non-Ideal Geometries Degenerate starting positions, such as when the elements are tightly clustered (within 10 m in each axis as in Fig.6.29) or are collinear (Fig. 6.31) may lead to low error over time because of the Kalman Filter, however, the SEP metrics do not converge consistently.

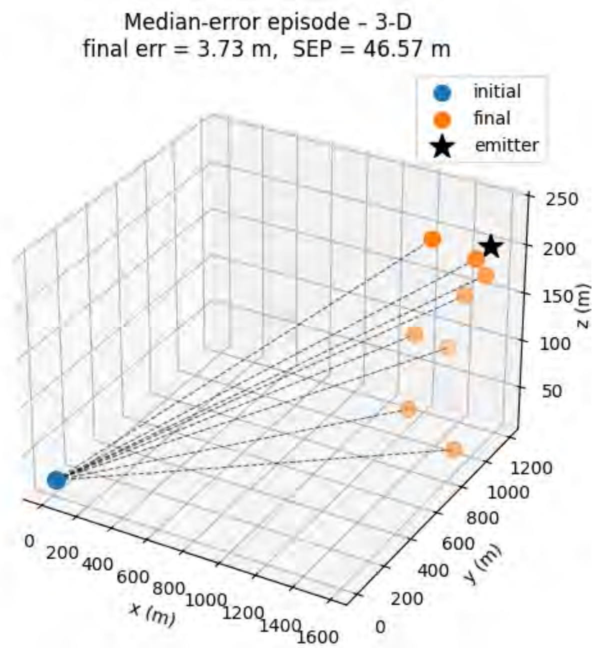

Figure 6.29: Geometry with tightly clustered elements

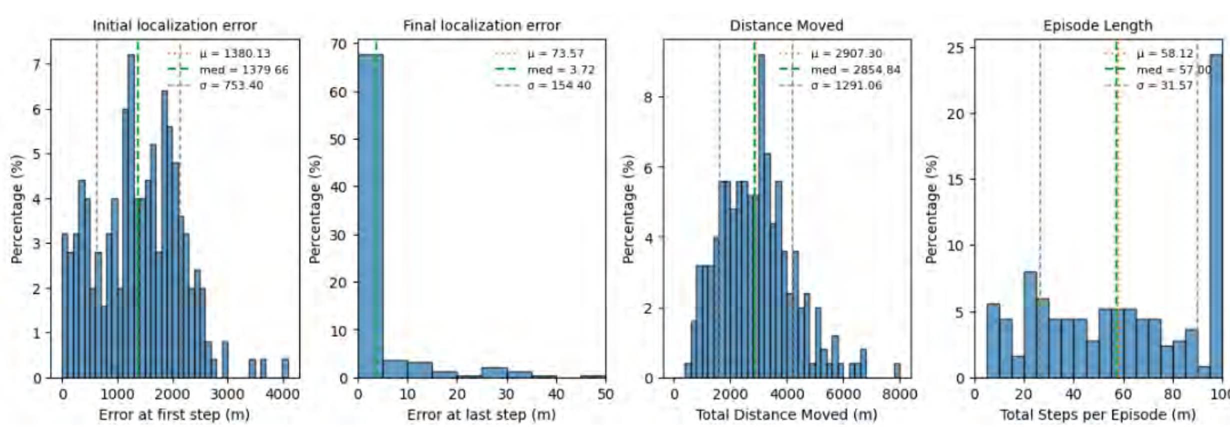

Figure 6.30: Evaluation metrics for trials where UAS begin with tightly clustered elements.

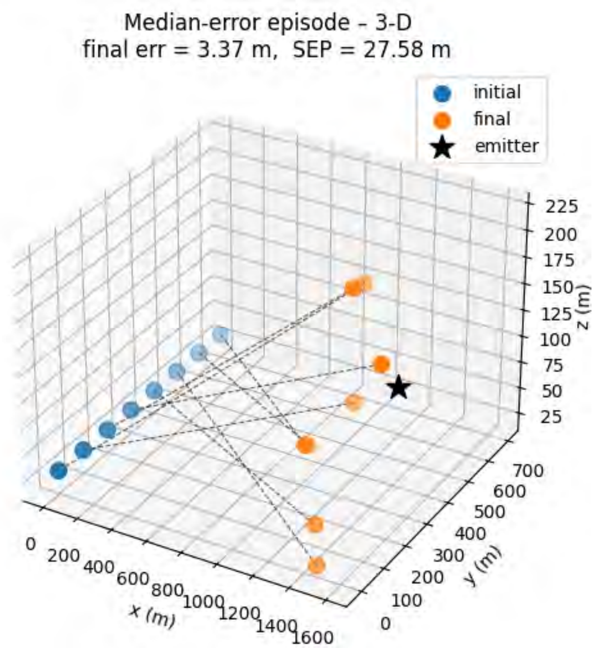

Figure 6.31: Geometry with collinear x and z, 100 m separation in y

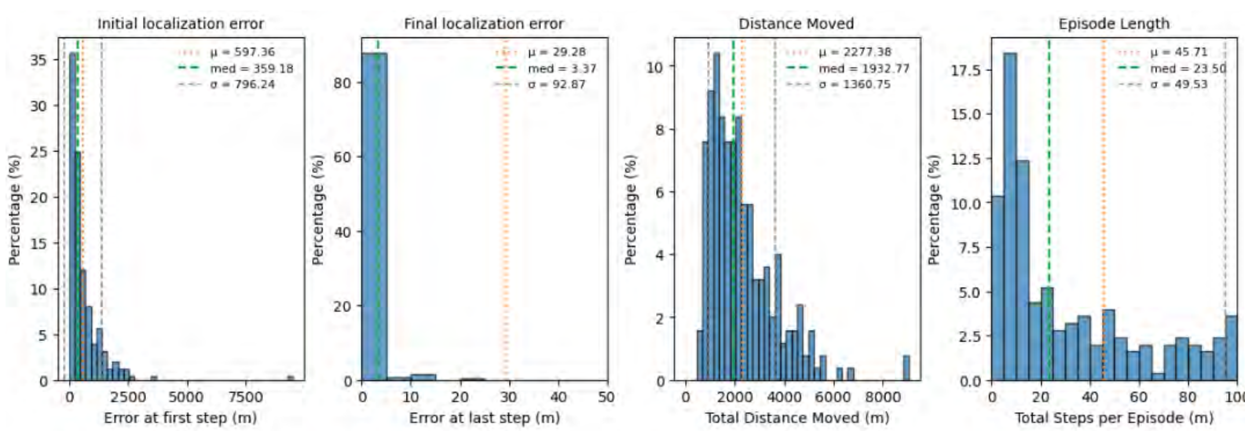

Figure 6.32: Evaluation metrics for trials where UAS begin with collinear x and z, 100 m separation in y.

Spread Cube Provided the sensors have some spatial diversity to start, as seen in Fig.6.33 where the sensors are in a cube with 100 m spread in each axis, nominal performance can be achieved in the typical case. However, as shown in Fig. 6.34, this performance is not guaranteed for sufficient cases.

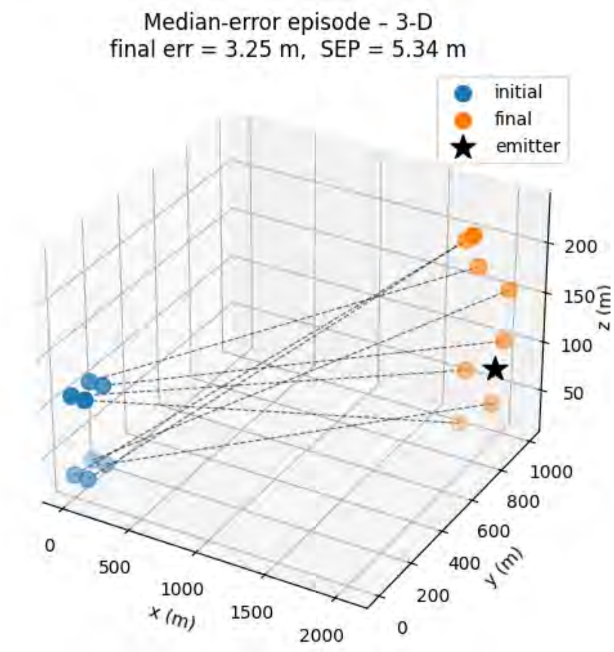

Figure 6.33: Initial and final positions of UAS with spatial diversity in starting position. Nominal performance can be achieved in some cases when sensors have spatial diversity, in this case, 100 m in each axis.

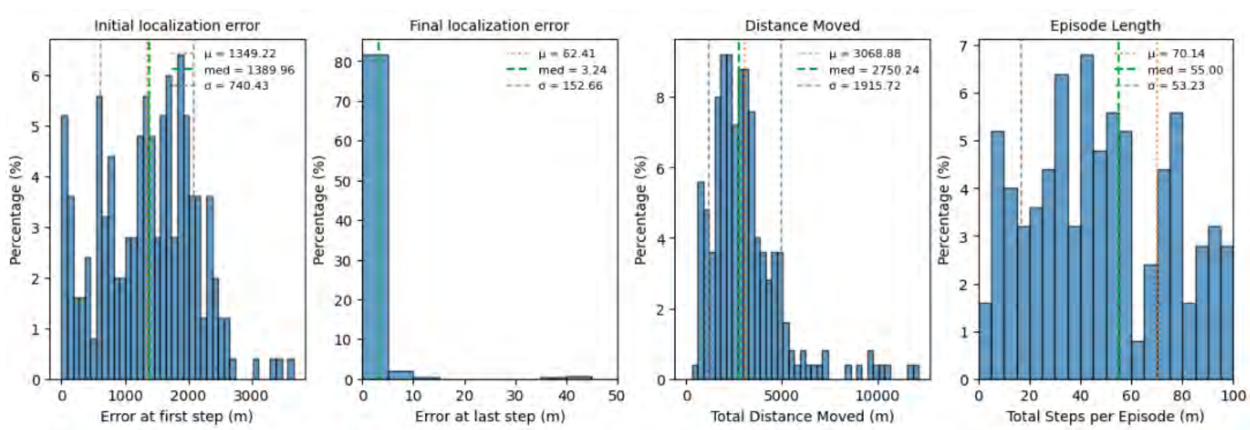

Figure 6.34: Evaluation metrics for trials where UAS begin with spatial diversity in each axis. While nominal performance can be achieved with 100 m separation in each axis, the performance is not typical.

Largely Spread Cube Start In Fig.6.35 the sensors start in a cube geometry with 350 m diversity in the x and y axes and 200 m in the z axis. Nominal performance can be achieved more typically as shown in Fig 6.36. From this performance, we conclude that this spacing should be considered a minimal for nominal performance.

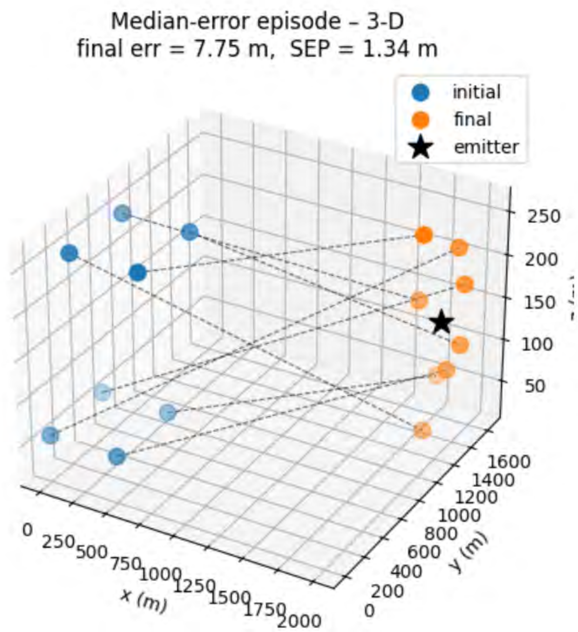

Figure 6.35: Initial and final positions of UAS with sufficiently-spaced starting position. Nominal performance can be achieved in typical cases when sensors are have enough spatial diversity, in this case, 350 m in each axis.

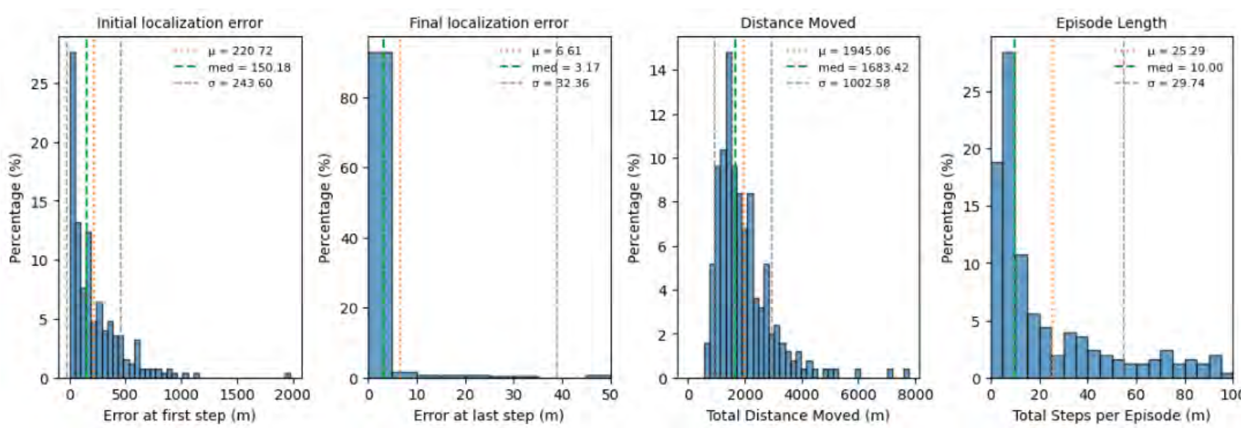

Figure 6.36: Evaluation metrics for trials where UAS begin with sufficient spatial diversity in each axis, which is approximately 350 m in x and y, and 200 m in z. Sufficient spatial diversity provides more consistent typical performance.

6.3.4 Sensor Hardware Impacts to Performance

Our prior assumptions of low positioning error and timing error enabled the system to perform well in the given environment. Here, we modify these errors to determine the sensitivity of the system to a larger magnitude hardware error.

First, we display performance in Fig.6.37 with no hardware error.

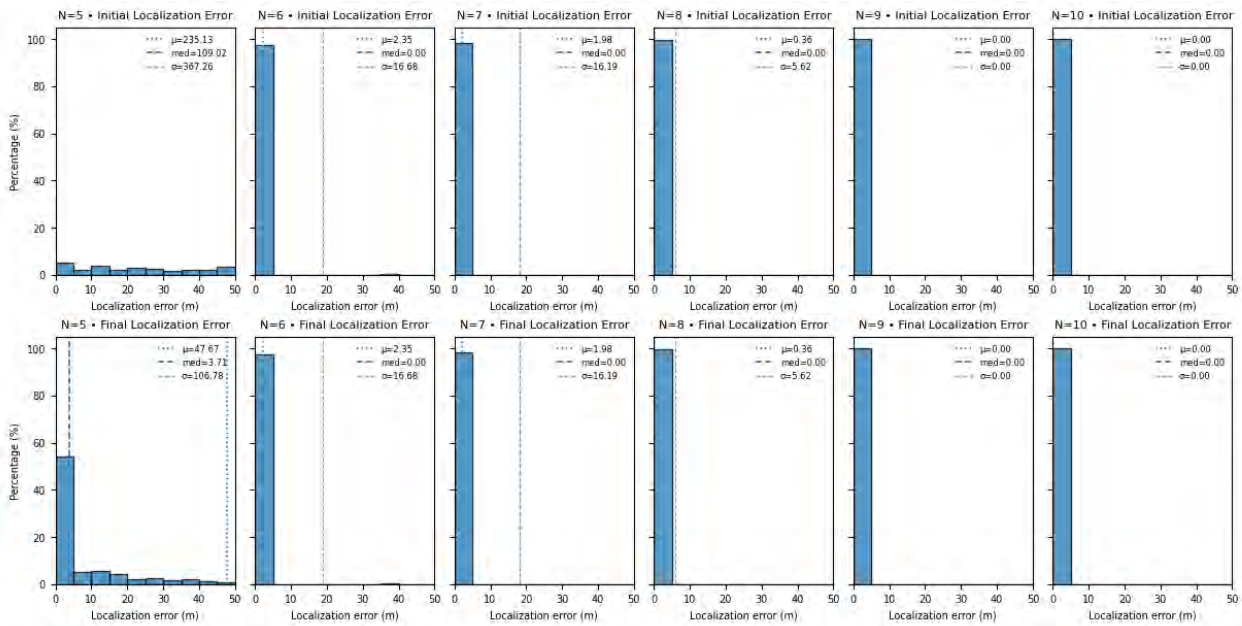

Figure 6.37: Initial and final localization performance with no hardware error.

6.3.4.1 Increased Positioning and Local Timing Error

Prior simulations had 2 cm positioning error and 10 ns local clock timing error. The below simulation increased those errors to 100 cm and 500 ns local timing error.

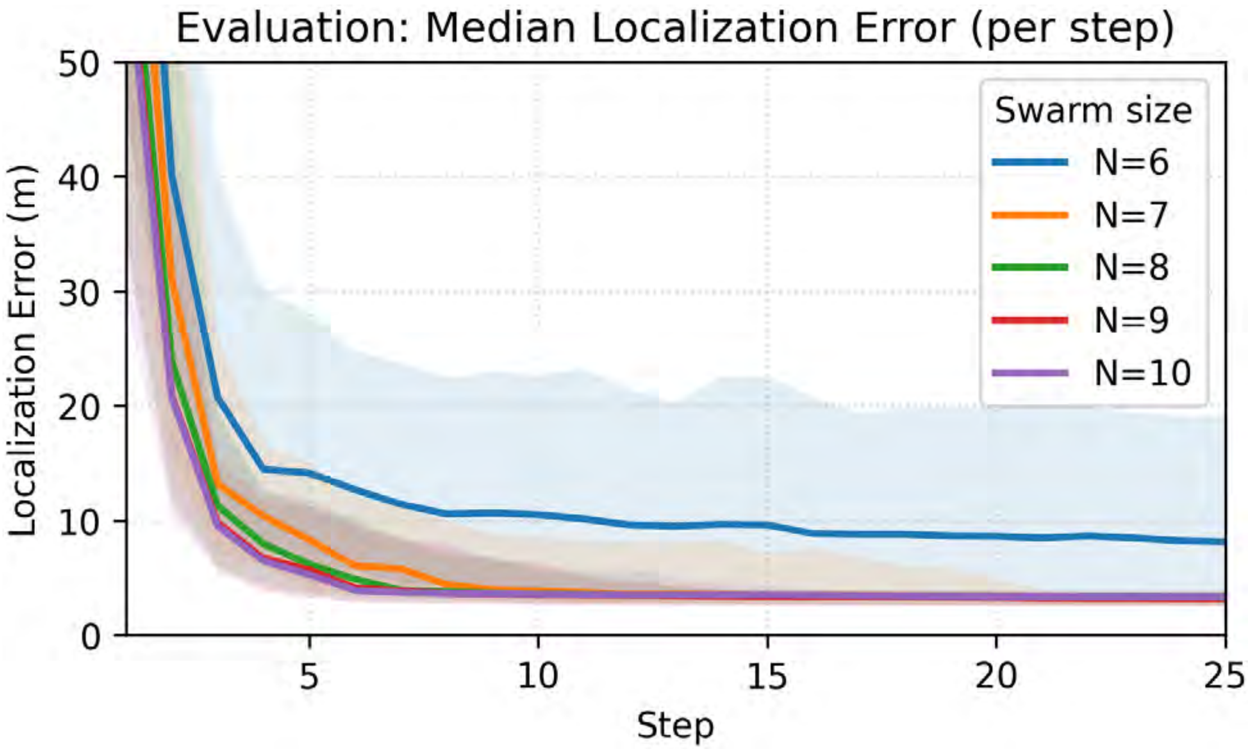

Figure 6.38: Localization performance per step with 500 ns local timing error and 100 cm positioning error.

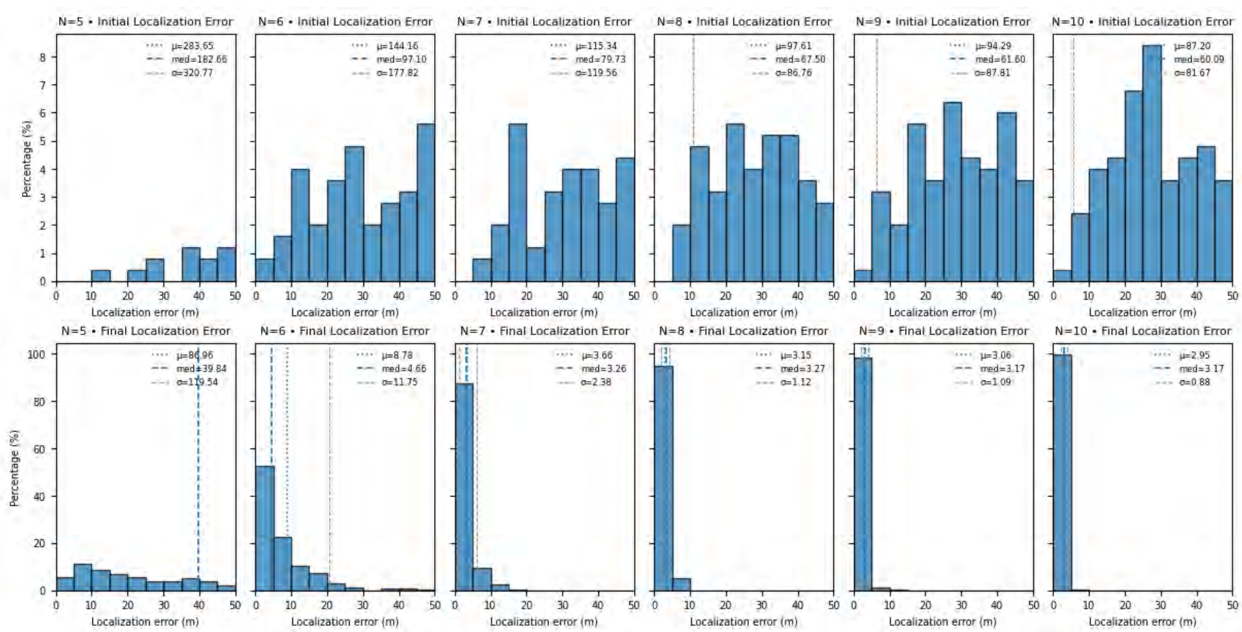

Figure 6.39: Initial and final localization performance comparison with 500 ns local timing error and 100 cm positioning error.

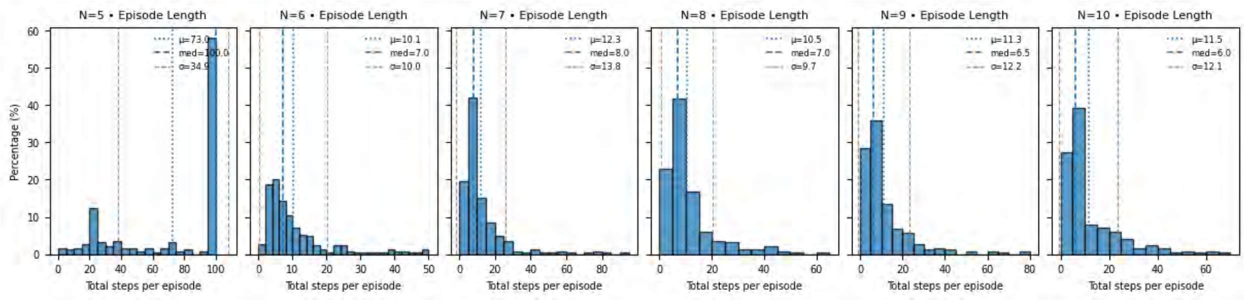

Figure 6.40: Episode length with 500 ns local timing error and 100 cm positioning error.

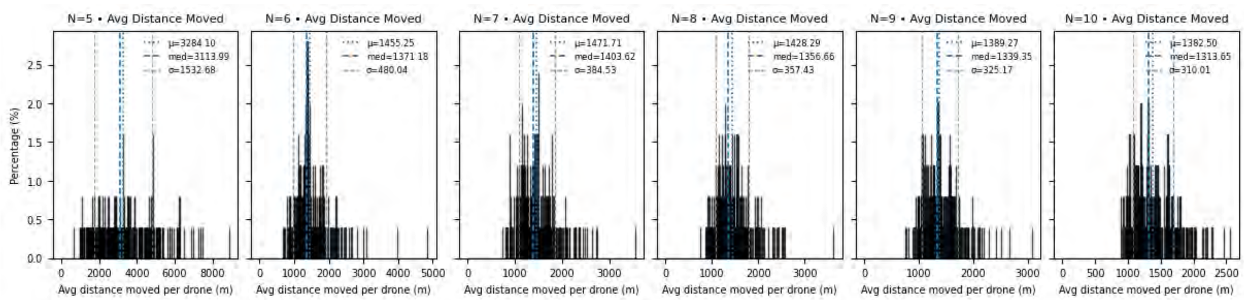

Figure 6.41: Average distance moved with 500 ns local timing error and 100 cm positioning error.

6.3.5 Environment Impacts to Performance

We have previously assumed a 50 % mix of LoS to non-LoS conditions in this multipath environment. In this section, we vary that mix to determine the ability of the system to overcome dense multipath impacts.

Below in Fig.6.42 we baseline the performance with pure LoS performance (0 % mix). This results in limited multipath impact to signal timing measurements, as we see the system nearly immediately converge on the solution.

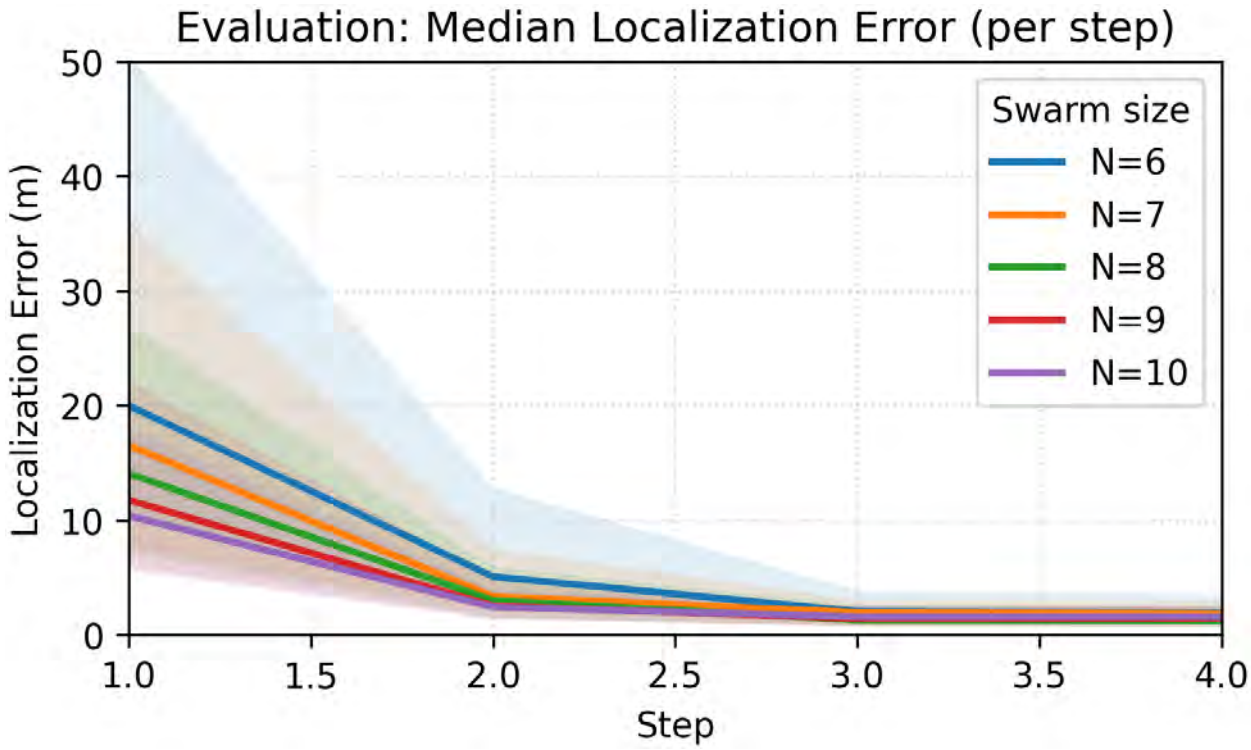

Figure 6.42: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas, pure LoS conditions.

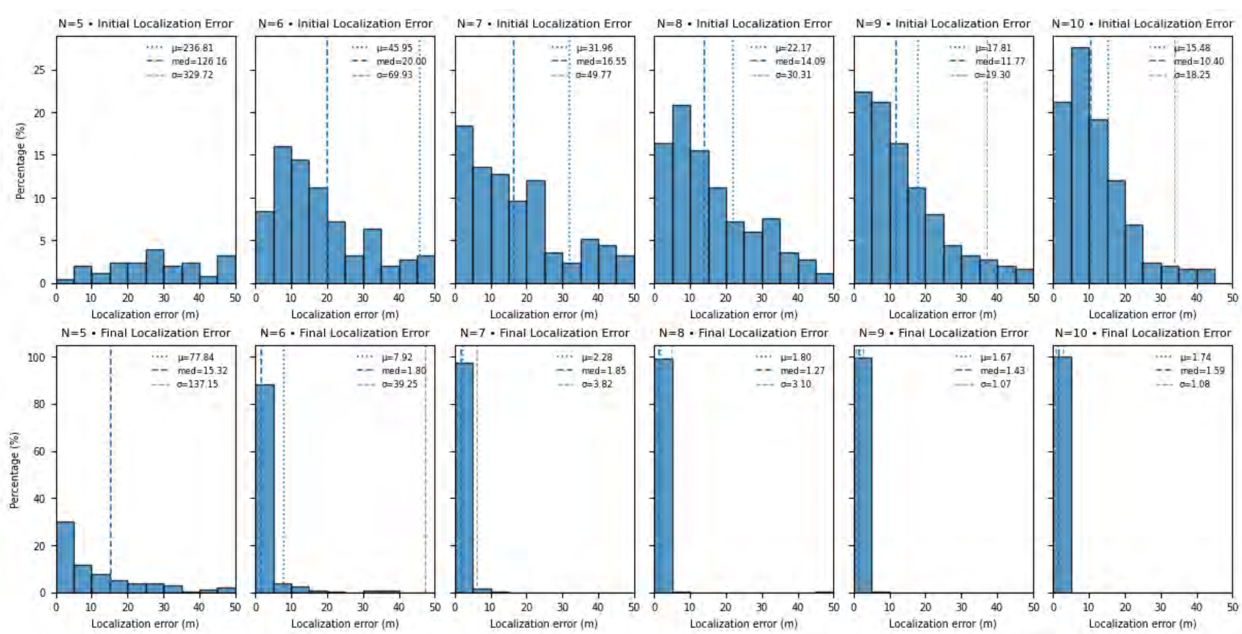

Figure 6.43: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas, pure LoS conditions.

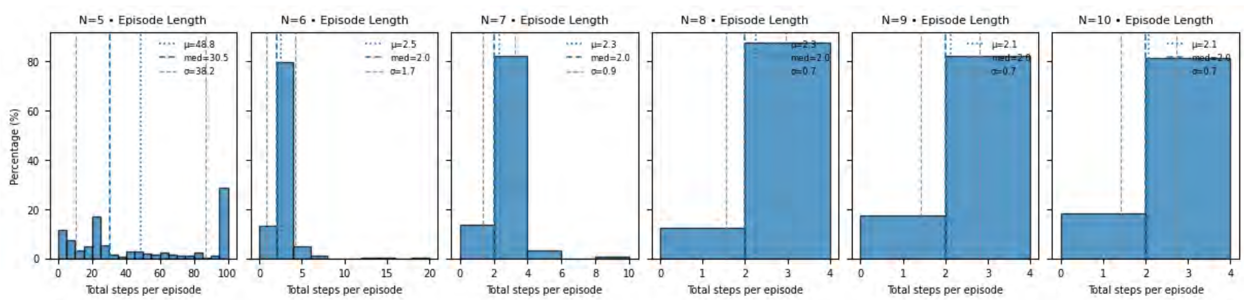

Figure 6.44: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas, pure LoS conditions.

6.3.5.1 25% mix

Omnidirectional Antennas In the case of a 25% mix of NLoS-to-LoS conditions, we see a nominal performance with Omnidirectional antennas. These cases converge quickly to a low localization error.

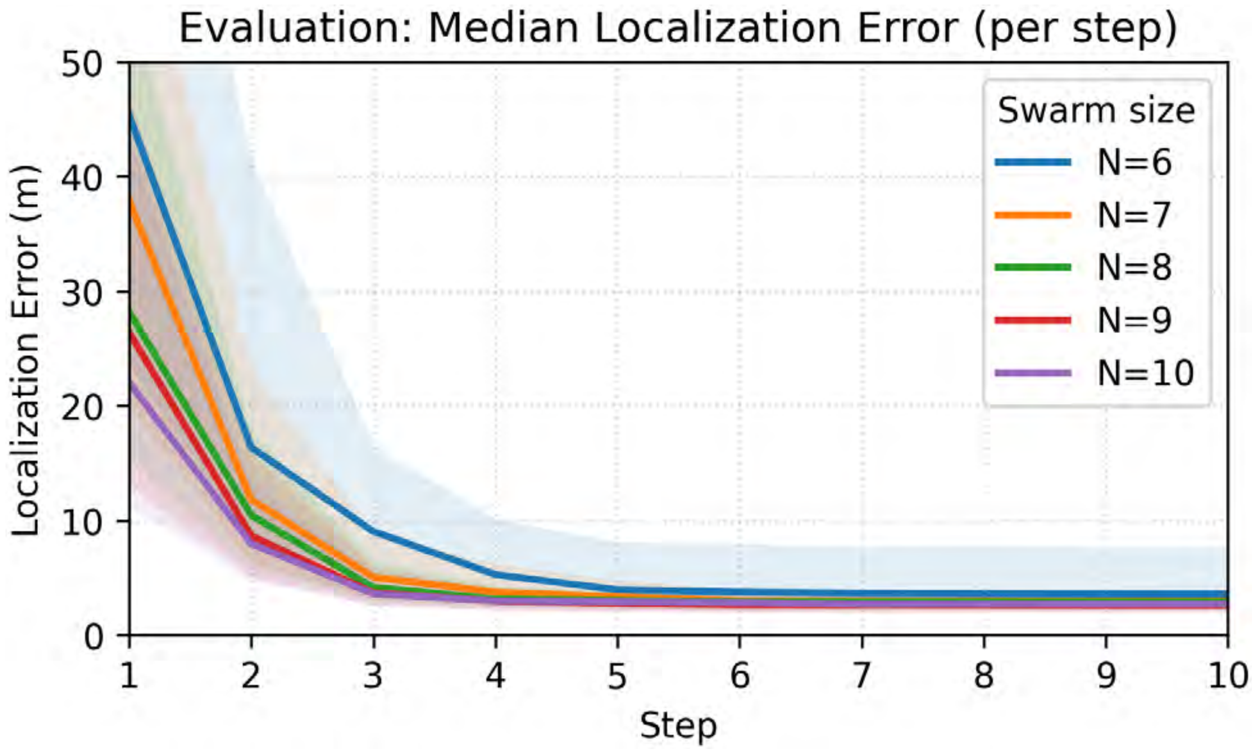

Figure 6.45: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas, 25% Mix NLoS-to-LoS conditions.

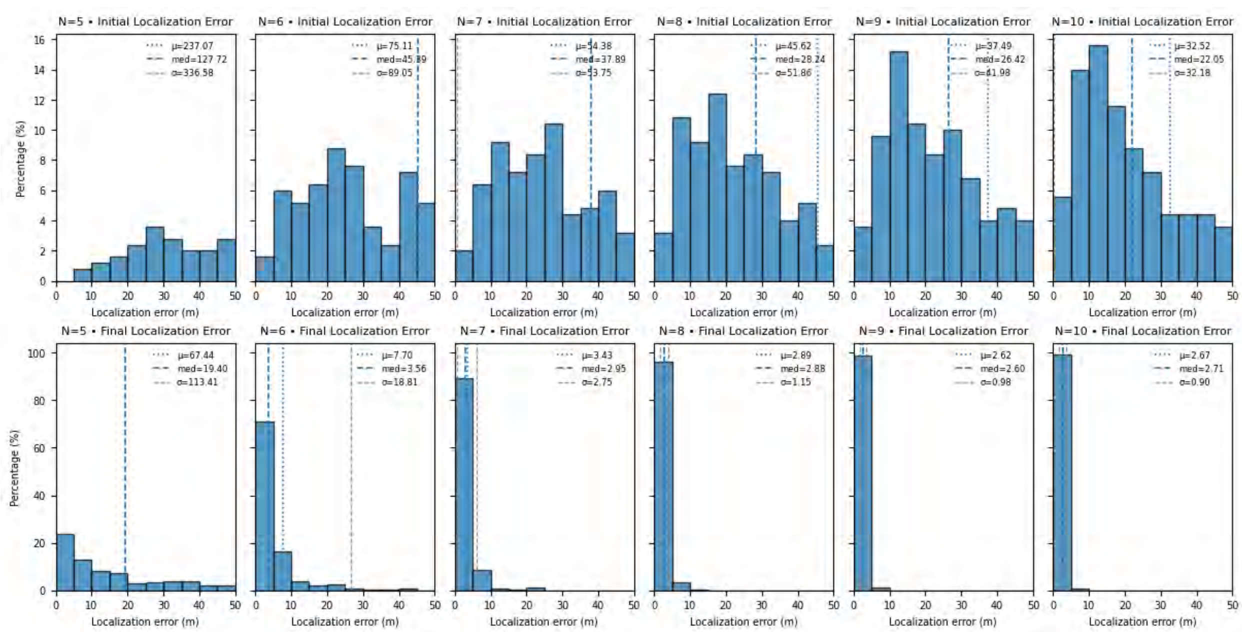

Figure 6.46: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas, 25% Mix NLoS-to-LoS conditions.

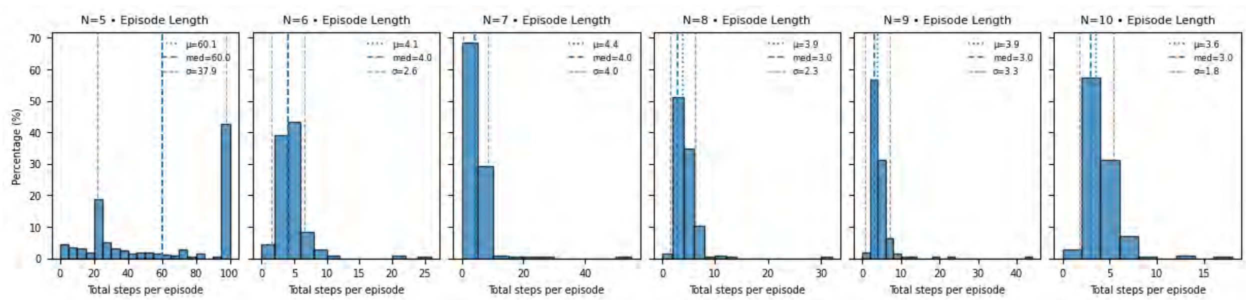

Figure 6.47: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas, 25% Mix NLoS-to-LoS conditions.

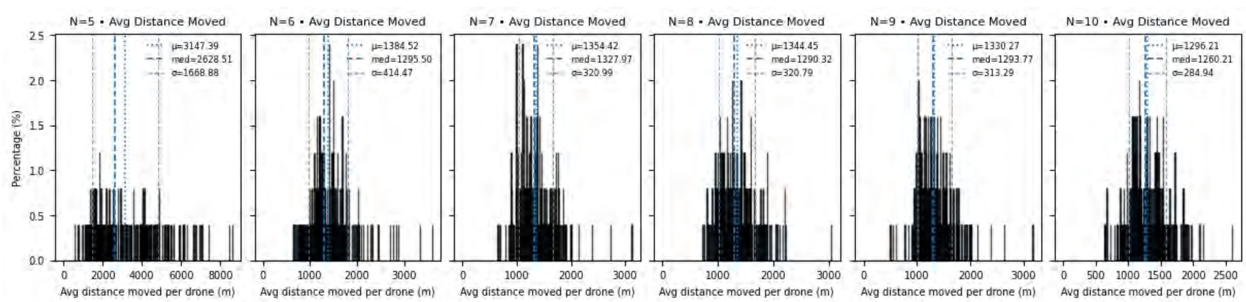

Figure 6.48: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas, 25% Mix NLoS-to-LoS conditions.

Directional Antennas Similar to the omnidirectional case, in a 25% mix of NLoS-to-LoS conditions, we see a nominal highly-accurate and fast-converging performance with Directional antennas.

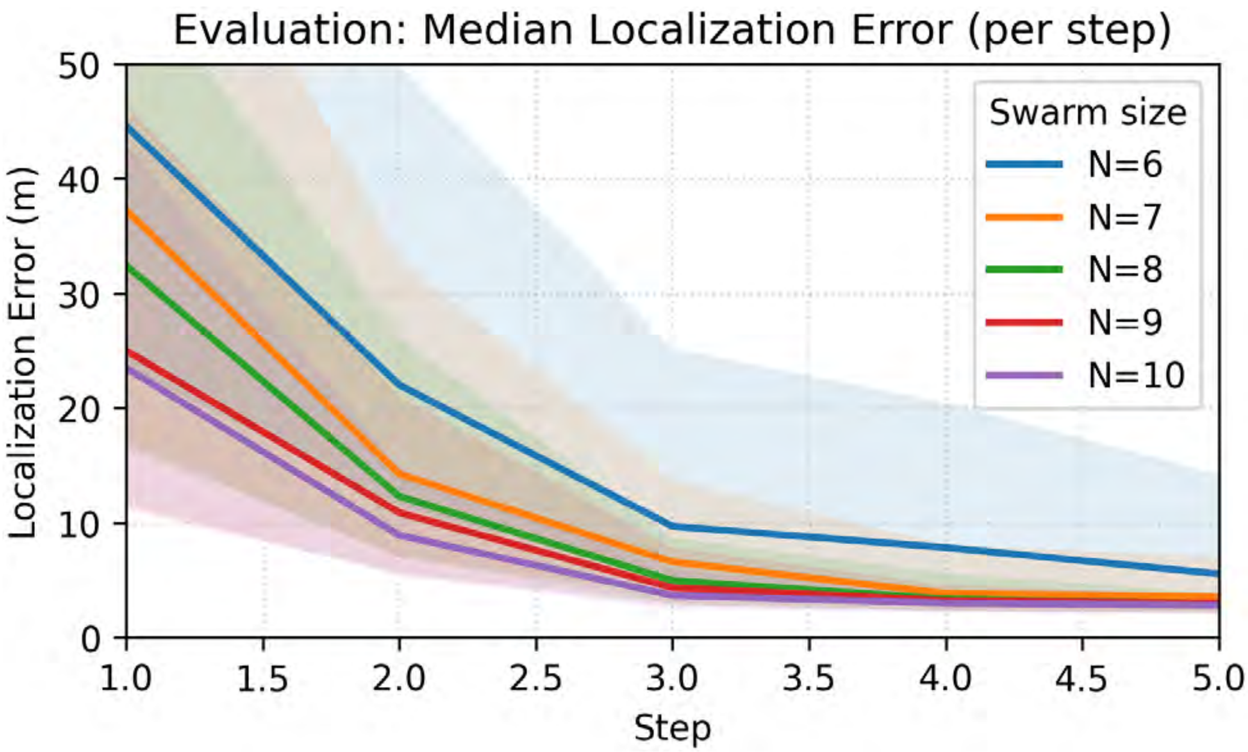

Figure 6.49: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas, 25% Mix NLoS-to-LoS conditions.

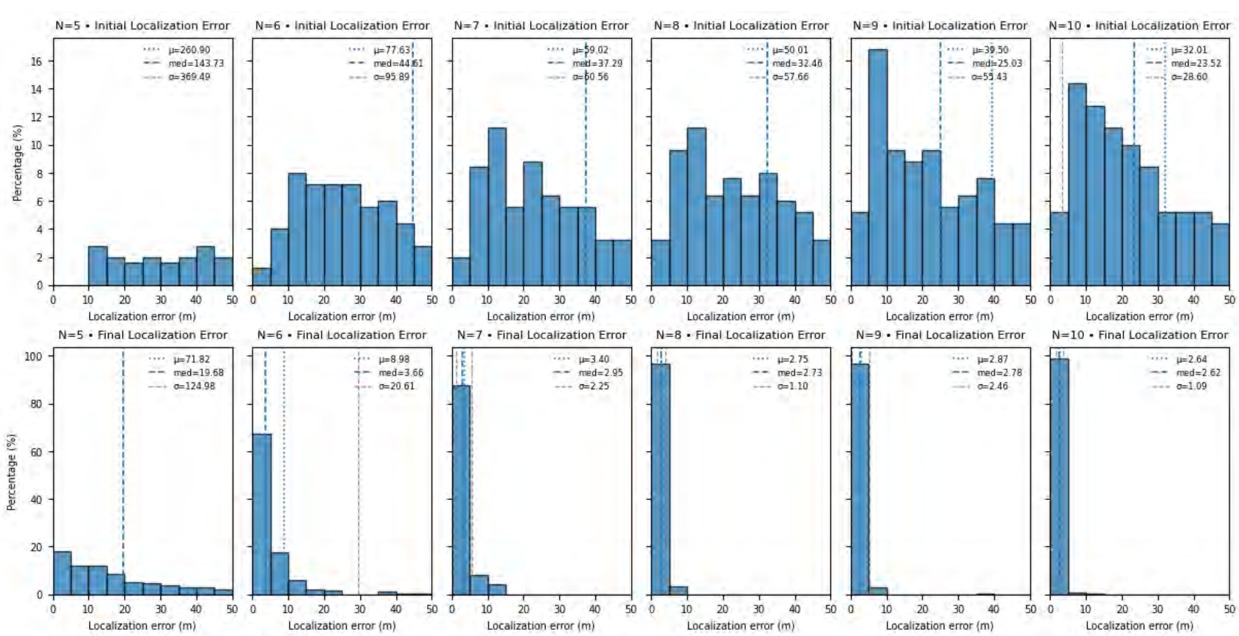

Figure 6.50: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas, 25% Mix NLoS-to-LoS conditions.

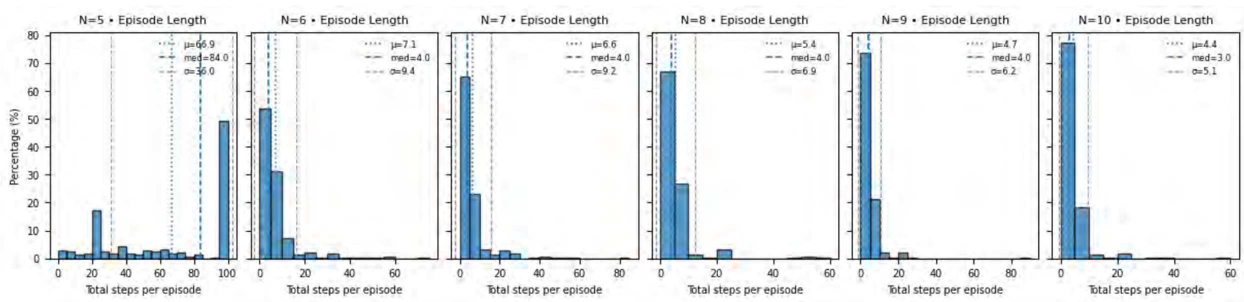

Figure 6.51: DQN final architecture episode length for varying swarm sizes and directional antennas, 25% Mix NLoS-to-LoS conditions.

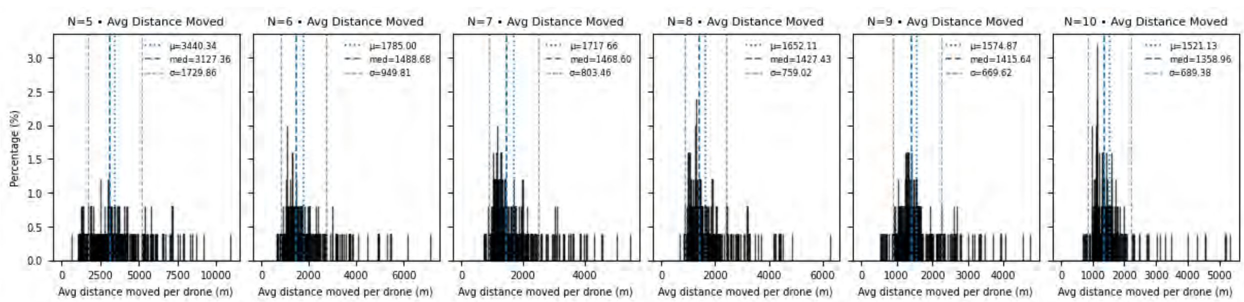

Figure 6.52: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas, 25% Mix NLoS-to-LoS conditions.

6.3.5.2 100% mix

Omnidirectional Antennas As we shift to a 100% mix of NLoS-to-LoS conditions, we see a degradation in nominal performance with Omnidirectional antennas. These cases nominally take 15+ episodes to converge, and the localization error is at least double of the nominal case. This performance is expected as the delay spread impacts increase with increasing NLoS.

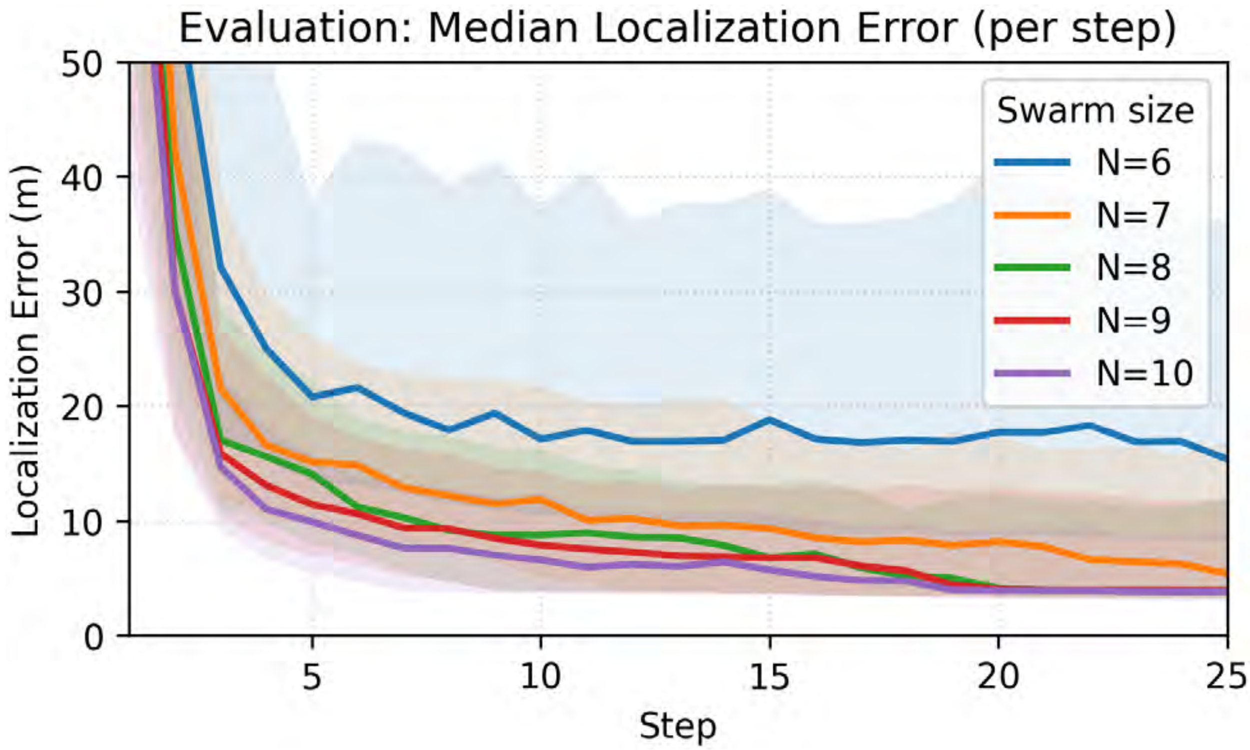

Figure 6.53: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas, 100% Mix NLoS-to-LoS conditions.

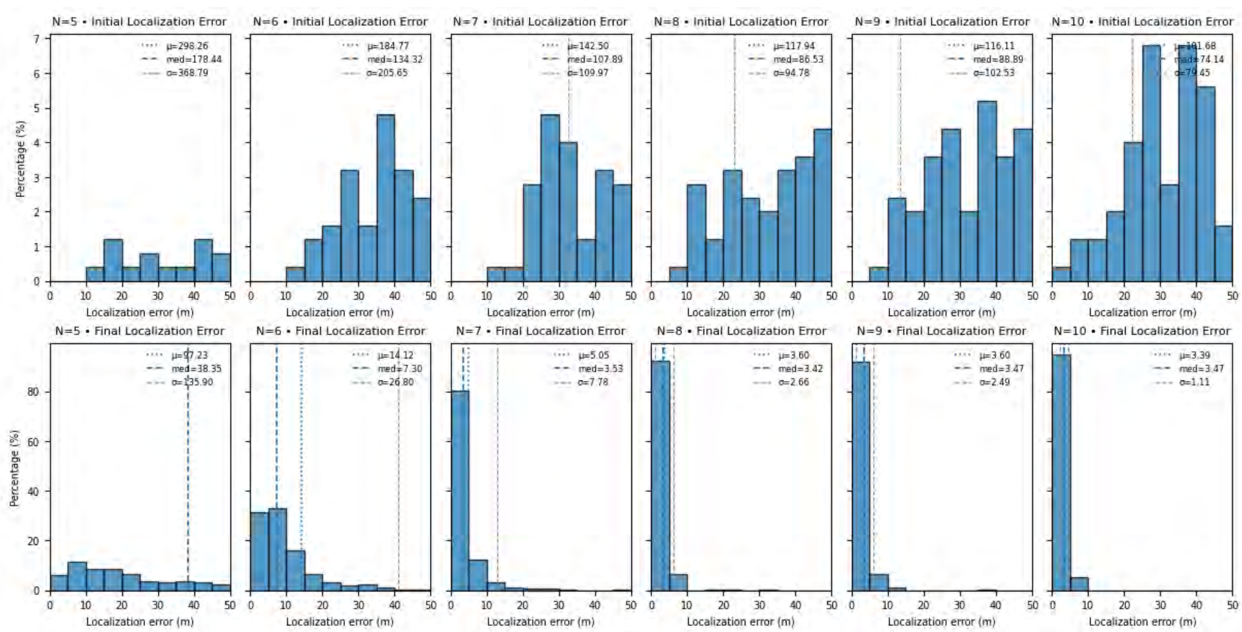

Figure 6.54: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas, 100% Mix NLoS-to-LoS conditions.

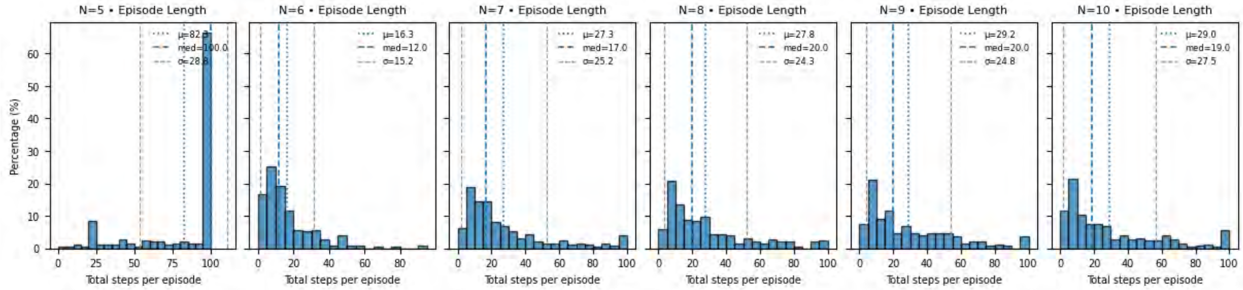

Figure 6.55: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas, 100% Mix NLoS-to-LoS conditions.

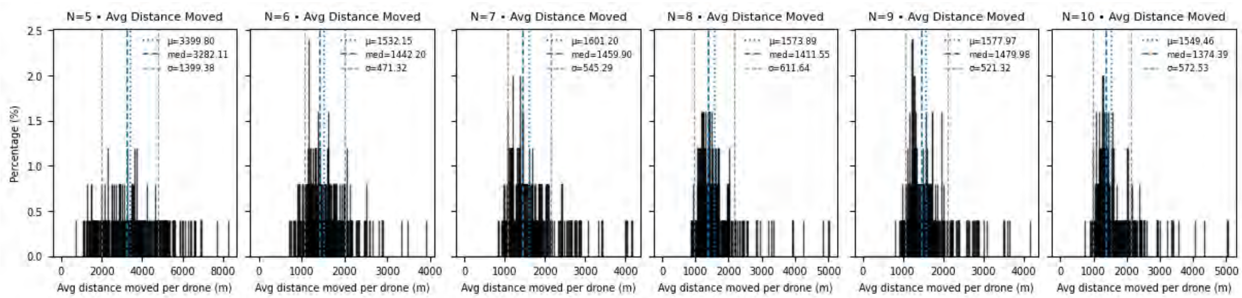

Figure 6.56: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas, 100% Mix NLoS-to-LoS conditions.

Directional Antennas In the 100% mix of NLoS-to-LoS conditions for directional antennas, we see a further degradation over to the localization performance over the omnidirectional case. These cases nominally take 20+ episodes to converge, and the localization error is similar to the omnidirectional case with a high standard deviation.

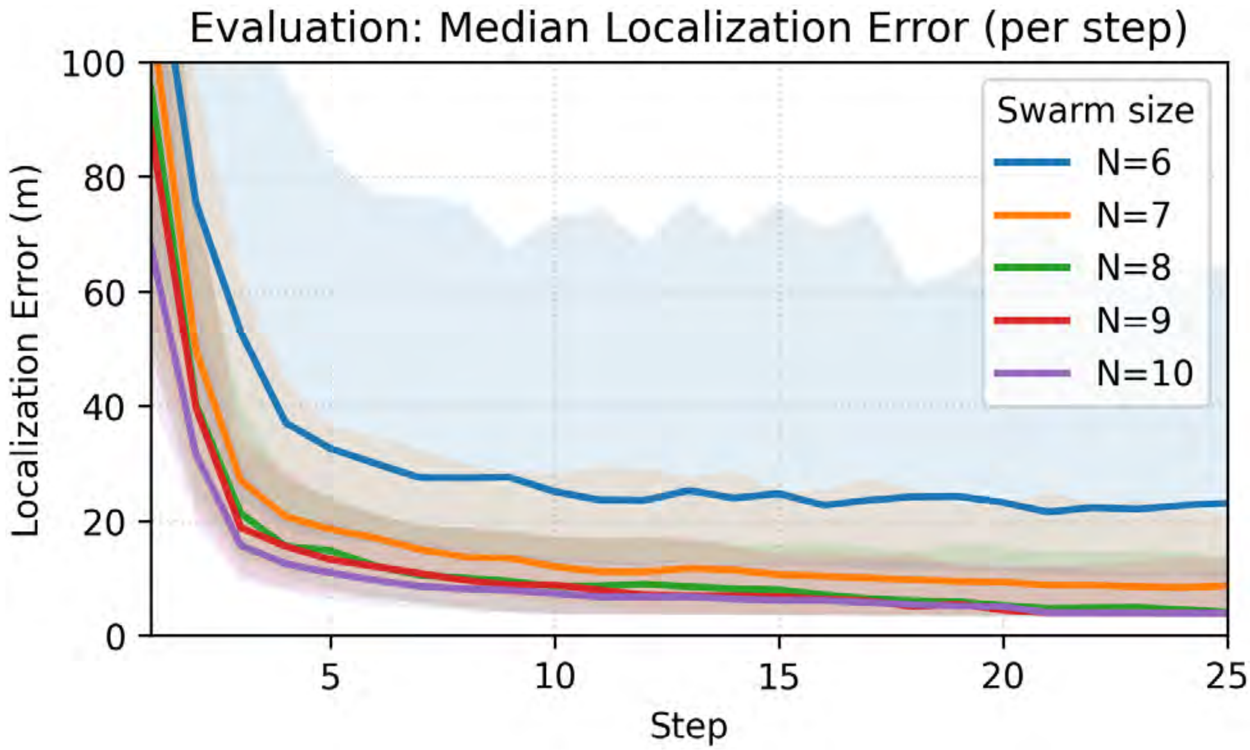

Figure 6.57: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas, 100% Mix NLoS-to-LoS conditions.

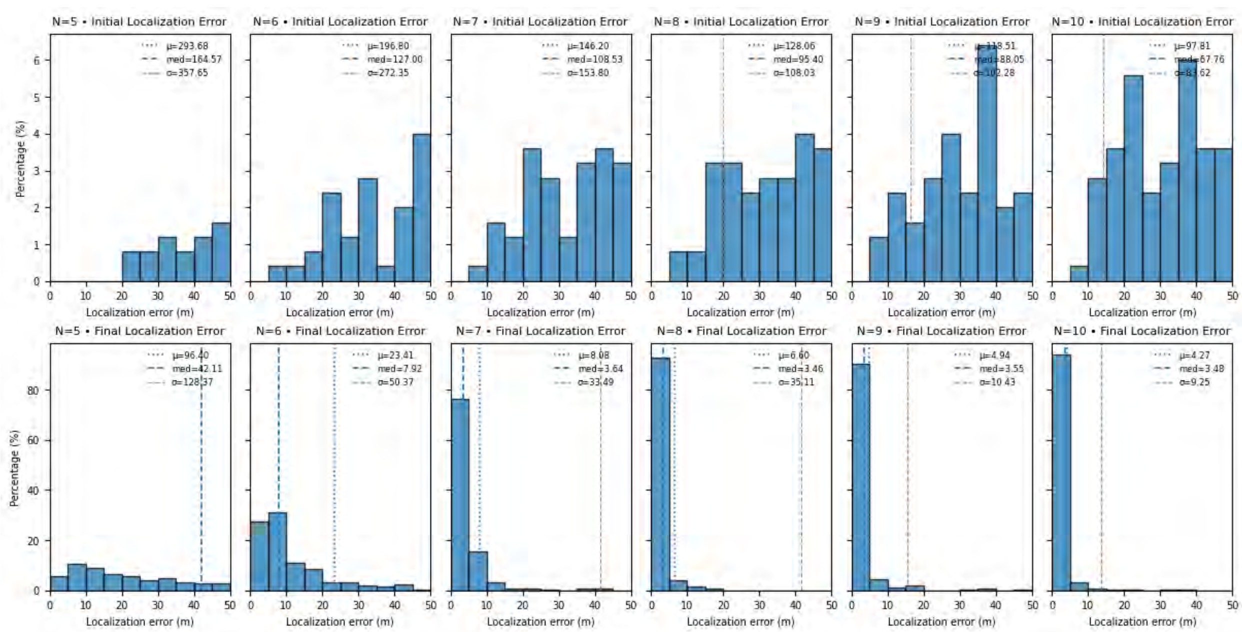

Figure 6.58: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas, 100% Mix NLoS-to-LoS conditions.

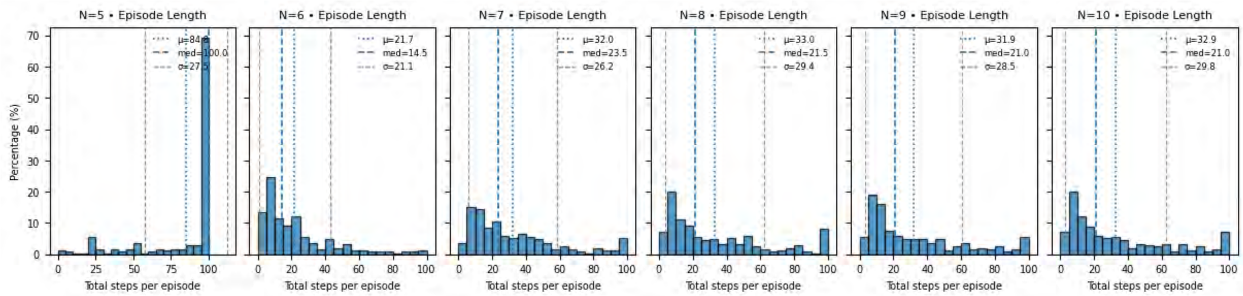

Figure 6.59: DQN final architecture episode length for varying swarm sizes and directional antennas, 100% Mix NLoS-to-LoS conditions.

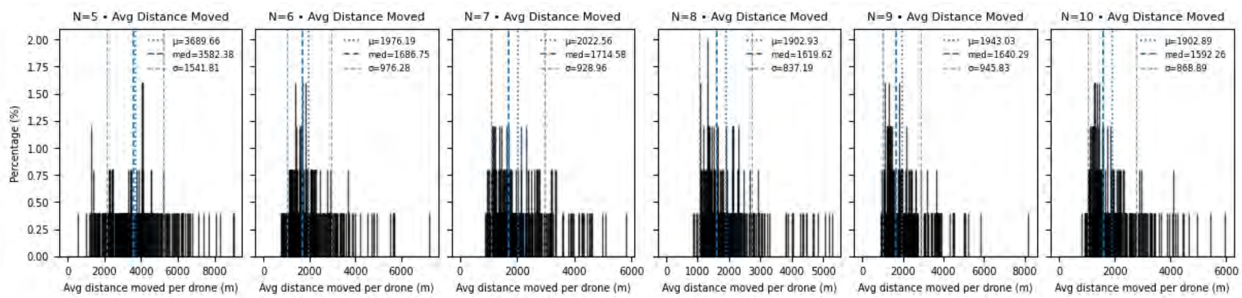

Figure 6.60: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas, 100% Mix NLoS-to-LoS conditions.

6.3.6 Emitter Frequency Impacts to Performance

To evaluate the system in simulation at other frequencies, we modified the implemented error parameters from ITU-R P.1411-12 [83]. The modified frequency changes the path loss and the multipath dynamics, however, because the overall performance of the LOCA algorithm is not frequency dependent, we do not expect to see significant loss in performance with changing frequencies.

6.3.6.1 Train New Policies at Other Frequencies

We first consider the need to train new policies for the system with alternate emitter broadcast frequencies.

2.4 GHz Emitter, Directional Antennas At 2.4 GHz with directional antennas, evaluations showed an increase in the typical localization error at each repositioning step. The localization metrics show both higher localization error and significantly more steps to convergence than the 5 GHz cases.

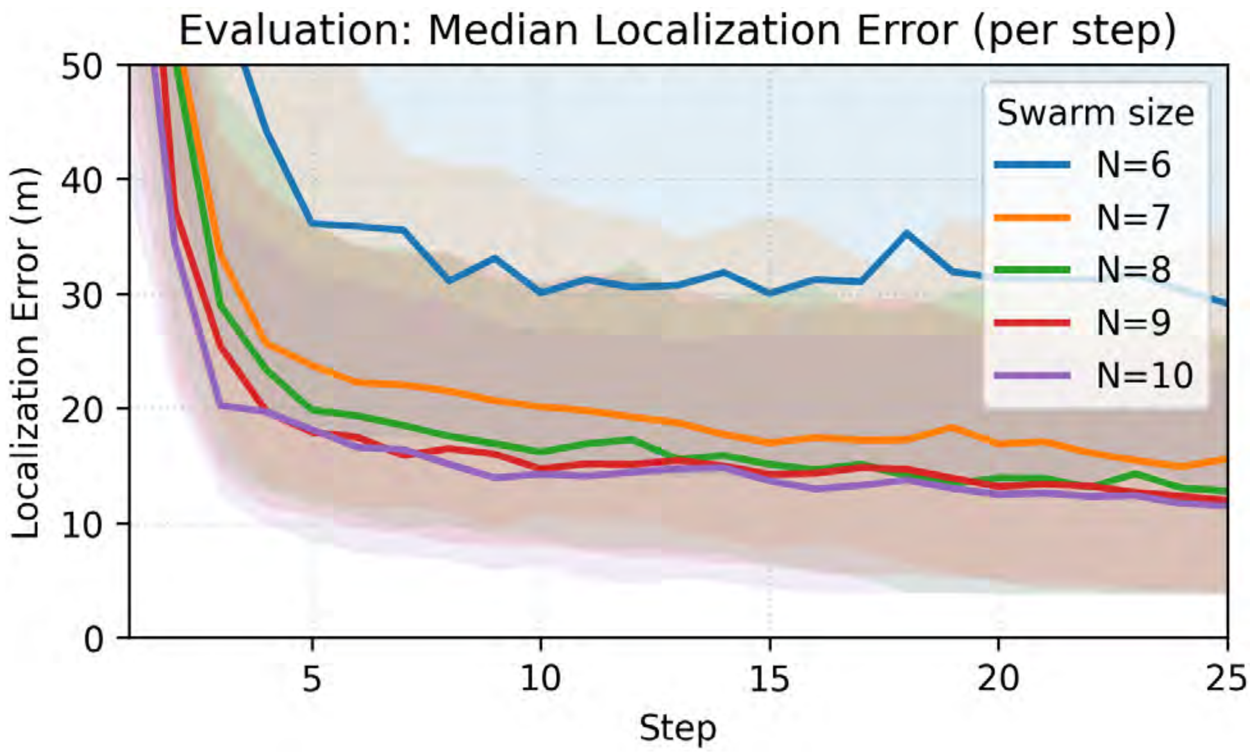

Figure 6.61: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

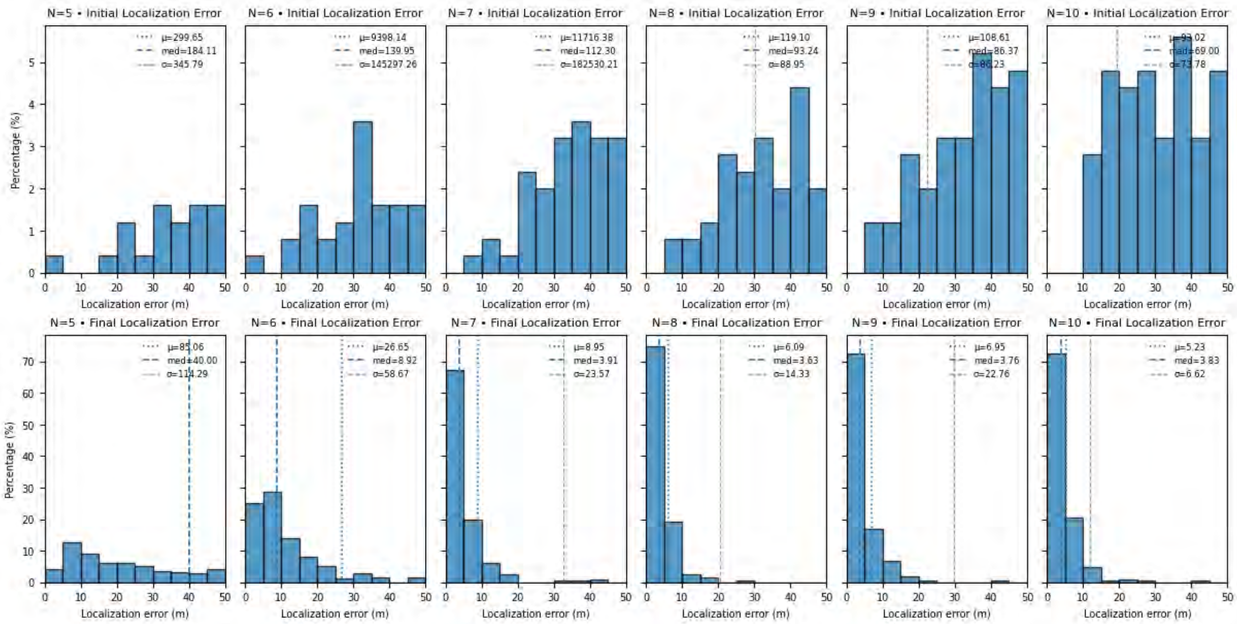

Figure 6.62: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

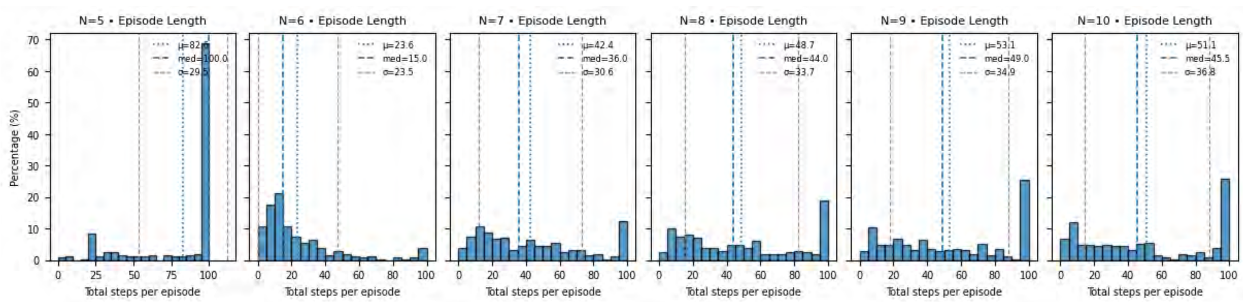

Figure 6.63: DQN final architecture episode length for varying swarm sizes and directional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

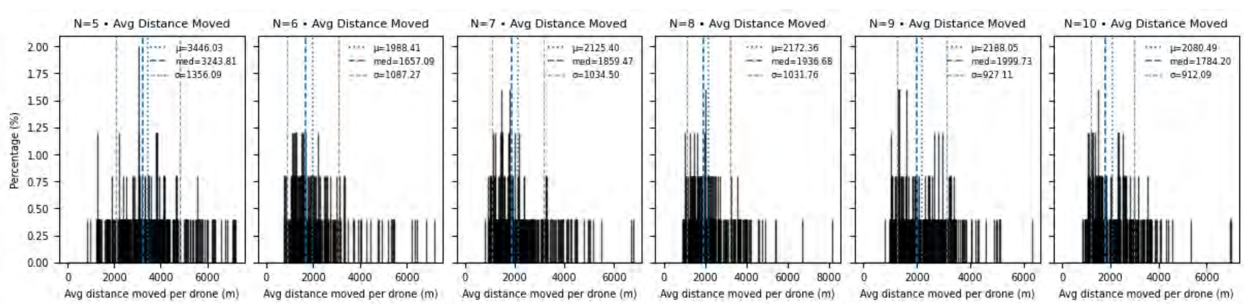

Figure 6.64: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

2.4 GHz Emitter, Omnidirectional Antennas Similar to the directional case, at 2.4 GHz with omnidirectional Antennas, evaluations showed an increase in the typical localization error at each repositioning step over the 5 GHz cases. The steps to convergence are lower than the directional cases, but still significantly higher than the 5 GHz case.

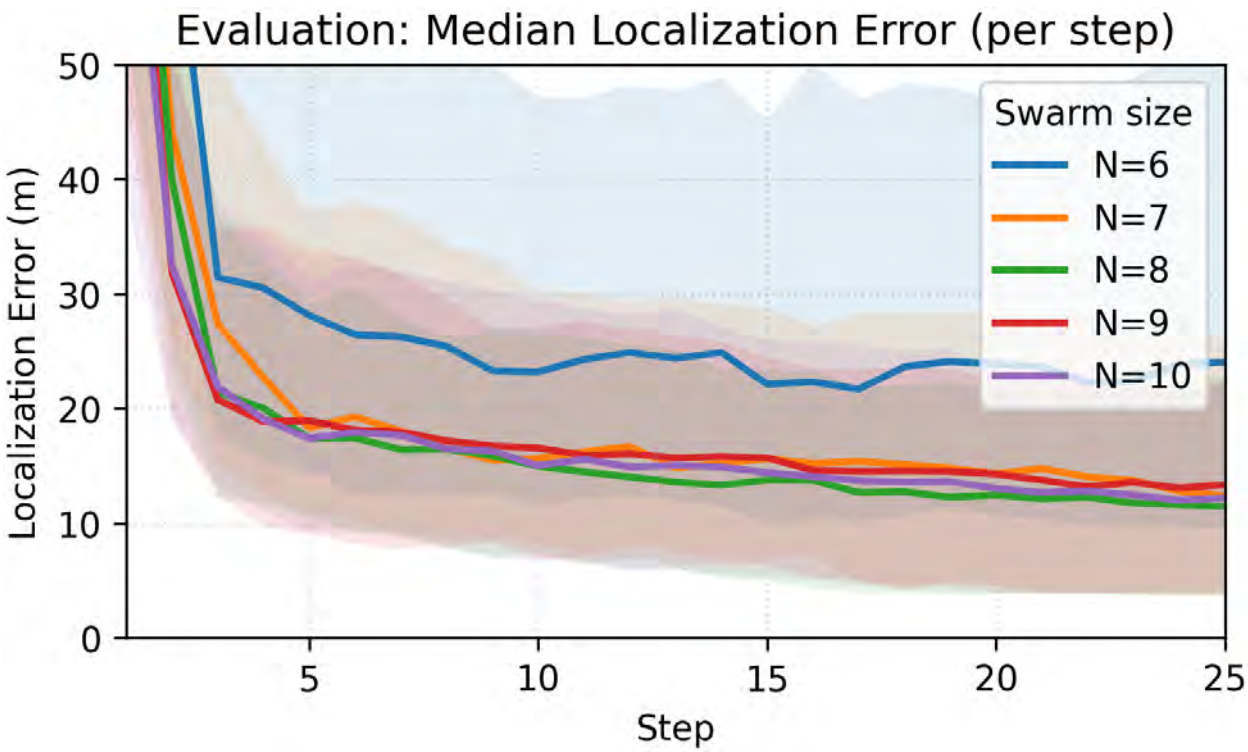

Figure 6.65: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

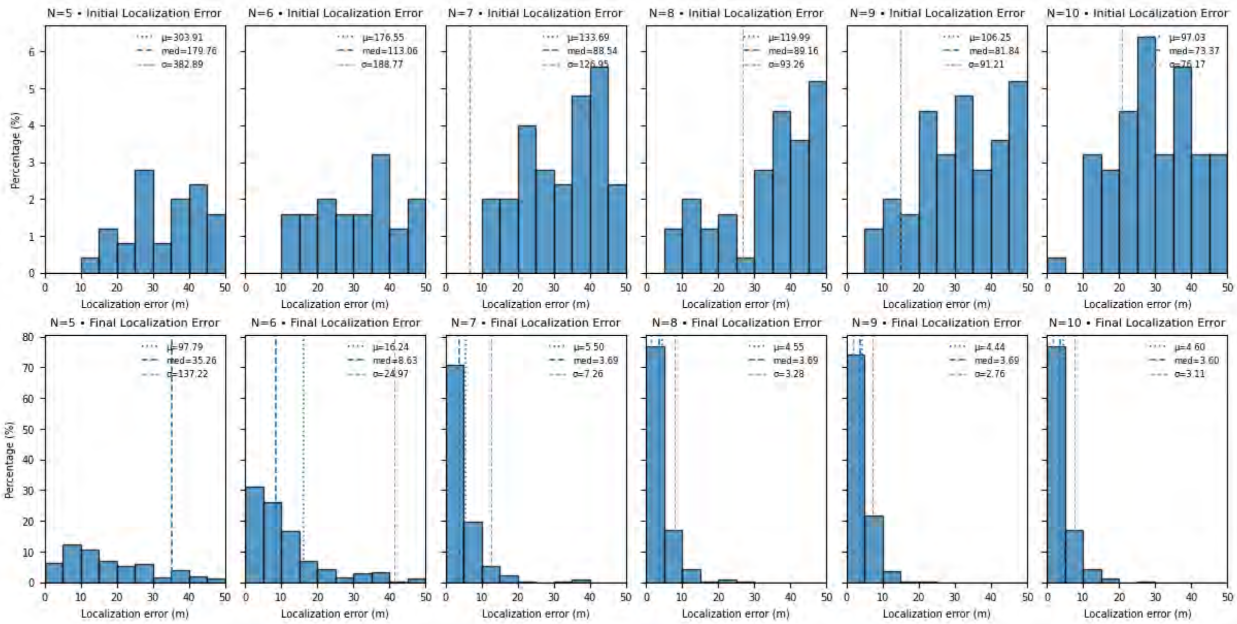

Figure 6.66: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

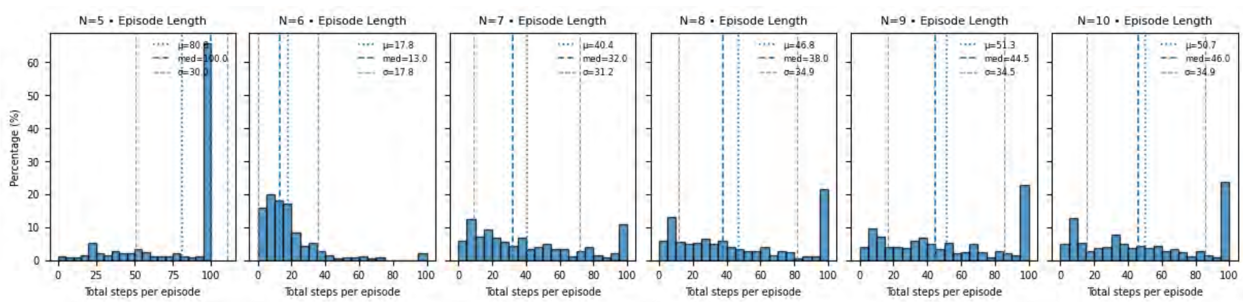

Figure 6.67: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

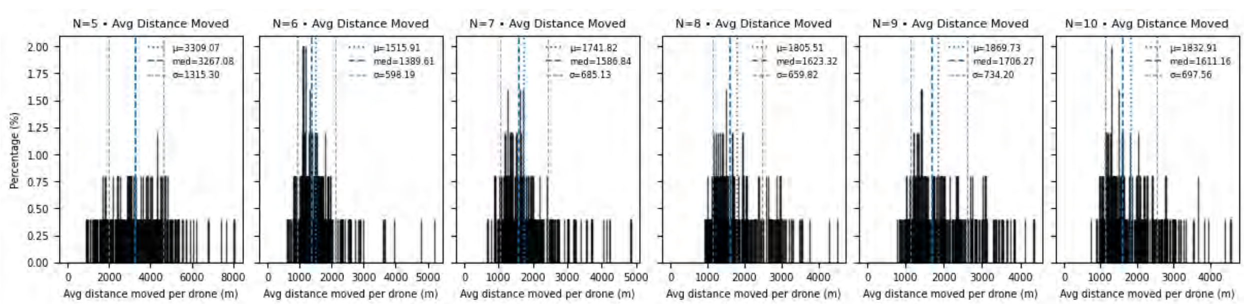

Figure 6.68: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas; system trained for 2.4 GHz and seeking emitter with 2.4 GHz signal.

8 GHz Emitter, Directional Antennas At 8 GHz, the convergence metrics for directional antennas showed slightly lower localization error than the 5 GHz case, however, the system required closer to 10 steps for convergence whereas the 5 GHz case required 4–5 steps. The localization error per step curve in Fig. 6.69 shows convergence for the first few steps followed by divergence, suggesting a significant number of trials converged faster while others required more steps to convergence, however the majority of cases still achieved a low localization error.

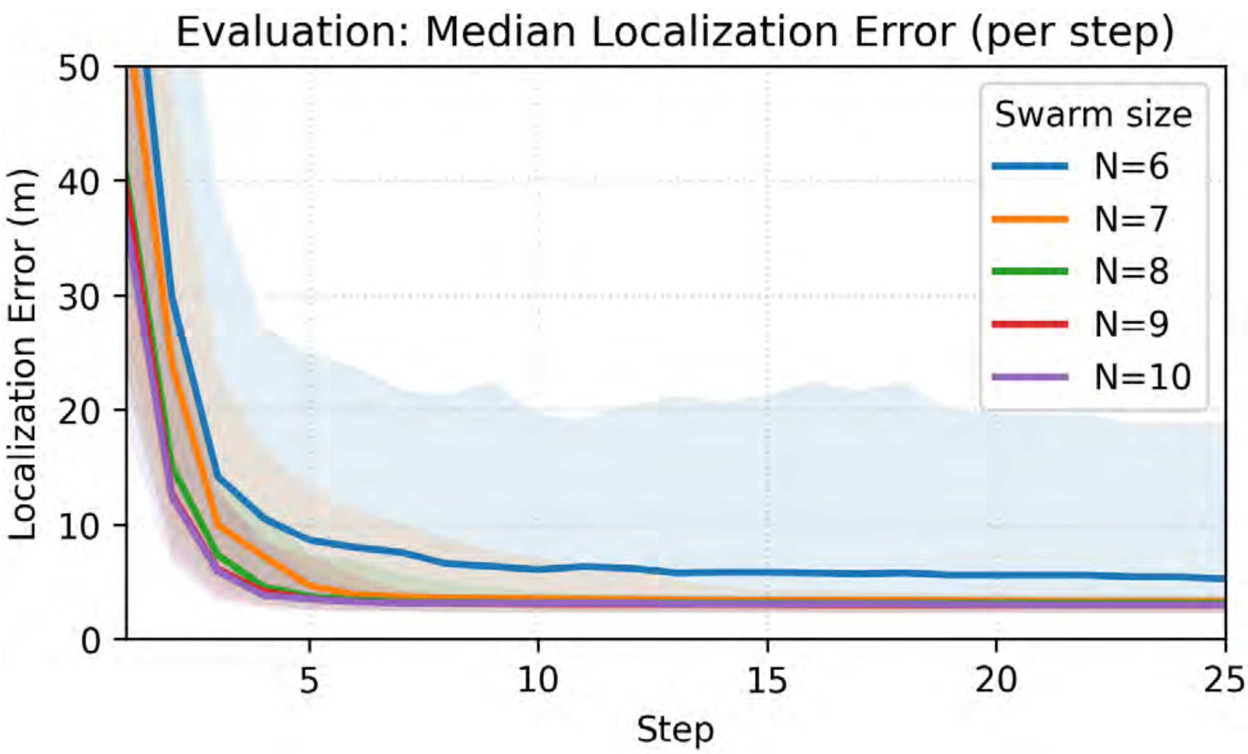

Figure 6.69: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

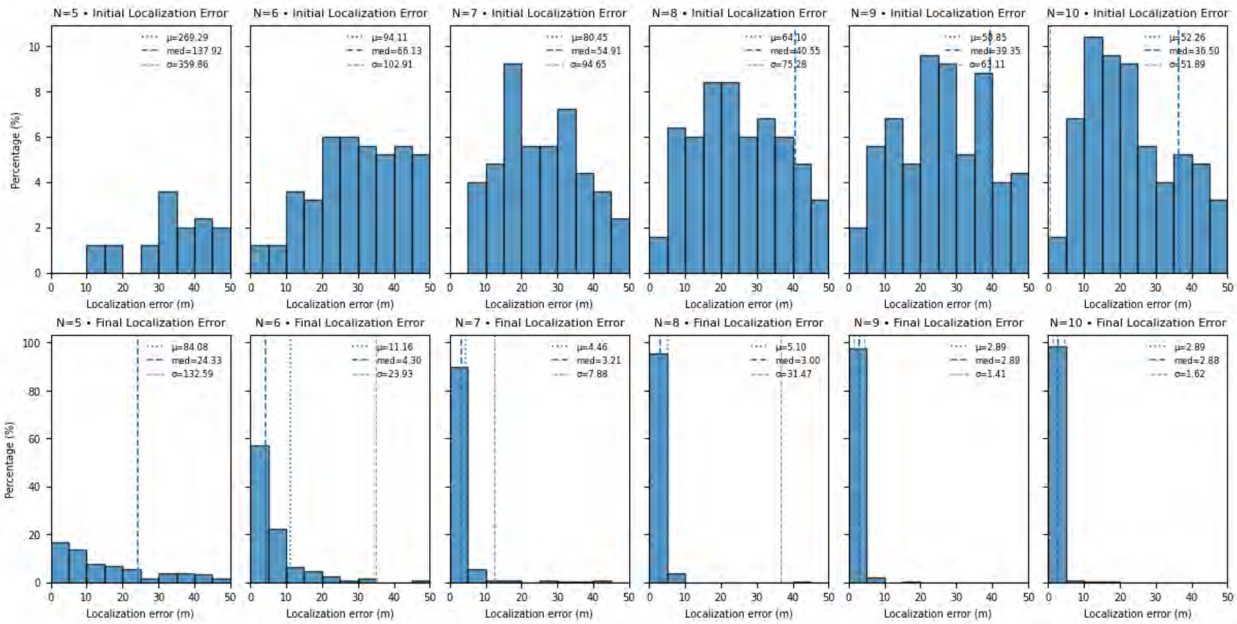

Figure 6.70: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

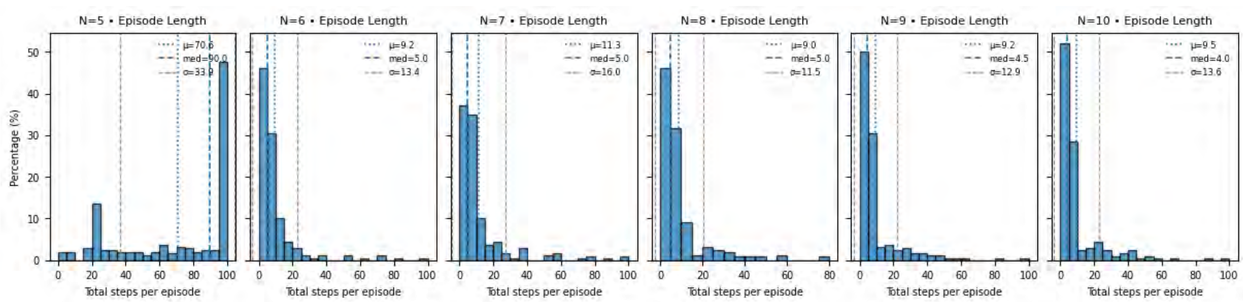

Figure 6.71: DQN final architecture episode length for varying swarm sizes and directional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

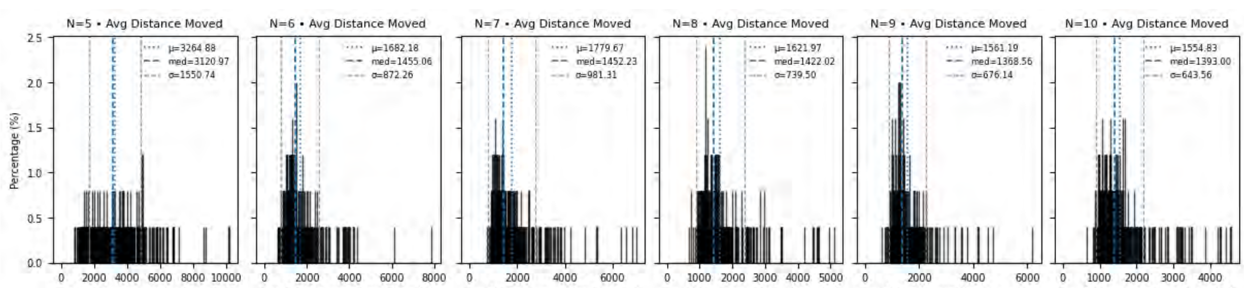

Figure 6.72: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

8 GHz Emitter, Omnidirectional Antennas At 8 GHz, the convergence metrics for the omnidirectional antennas evaluation showed similar performance to the directional case, with slightly fewer episodes to convergence. The localization error per step curve in Fig. 6.73 shows less divergence on later steps, suggesting a more stable performance across episodes.

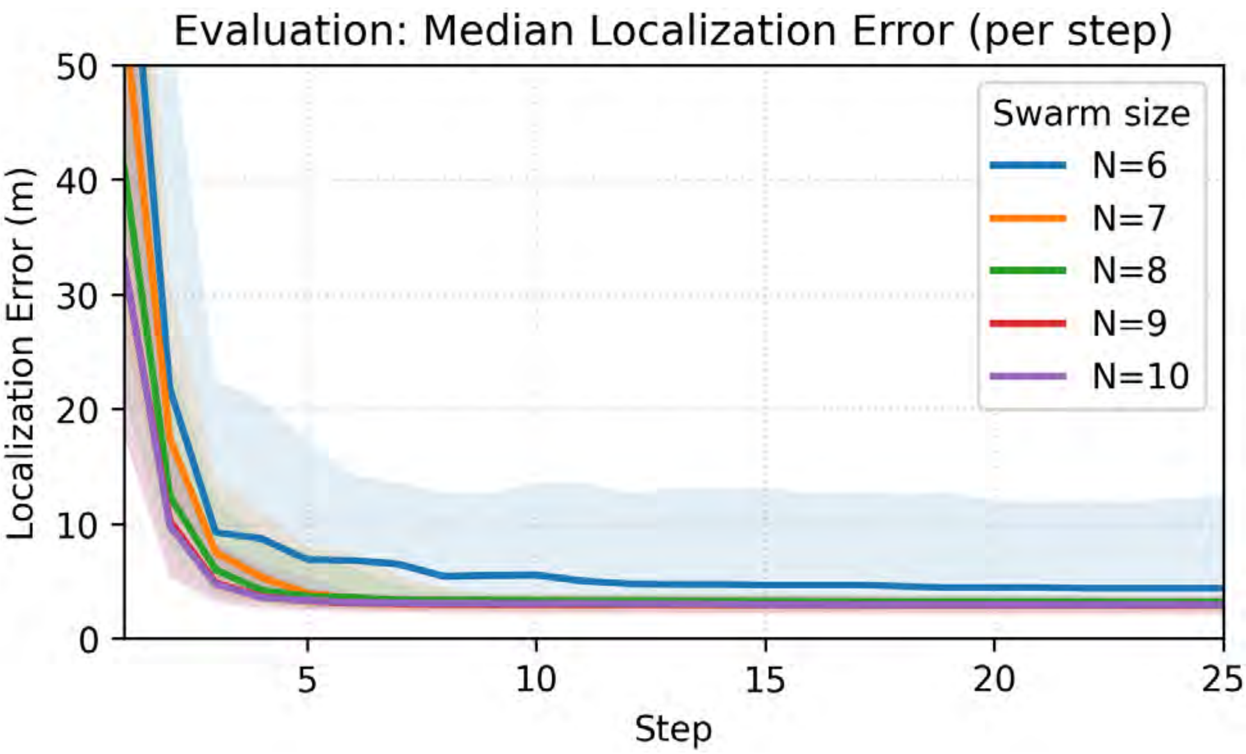

Figure 6.73: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

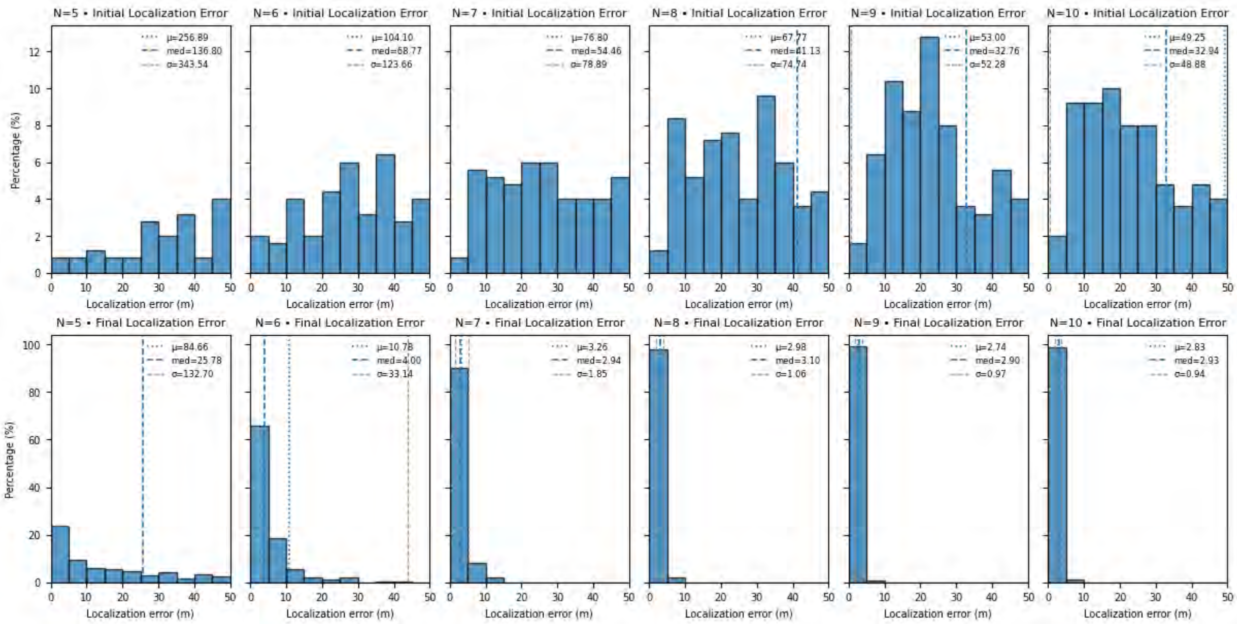

Figure 6.74: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

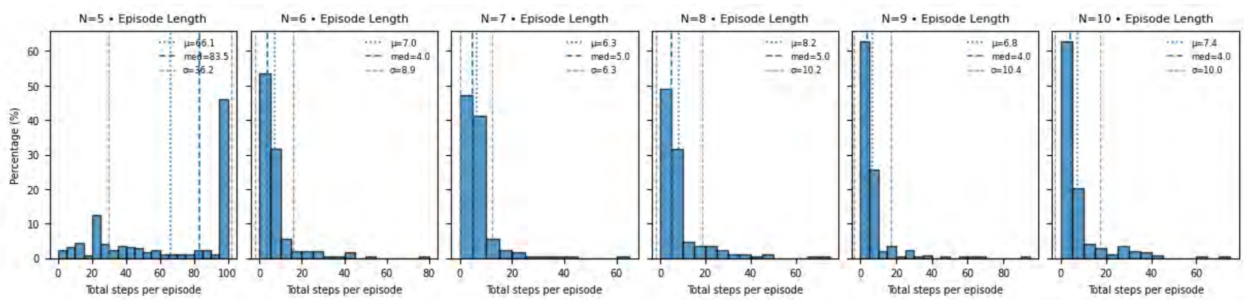

Figure 6.75: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

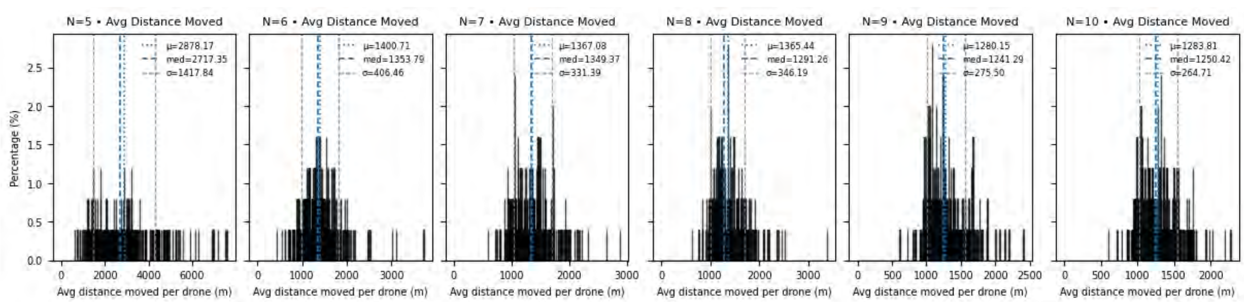

Figure 6.76: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas; system trained for 8 GHz and seeking emitter with 8 GHz signal.

6.3.6.2 Utilize Policies for 5 GHz at Other Frequencies

We next evaluate the original policy, trained at 5 GHz, at these alternate frequencies.

2.4 GHz Emitter, Directional Antennas The UAS with directional antennas trained at 5 GHz seeking a 2.4 GHz emitter shows lower median localization accuracy across episode steps and requires more steps for convergence. However, while the performance is not optimized, it appears that the policy trained at 5 GHz can be utilized at 2.4 GHz for directional antennas.

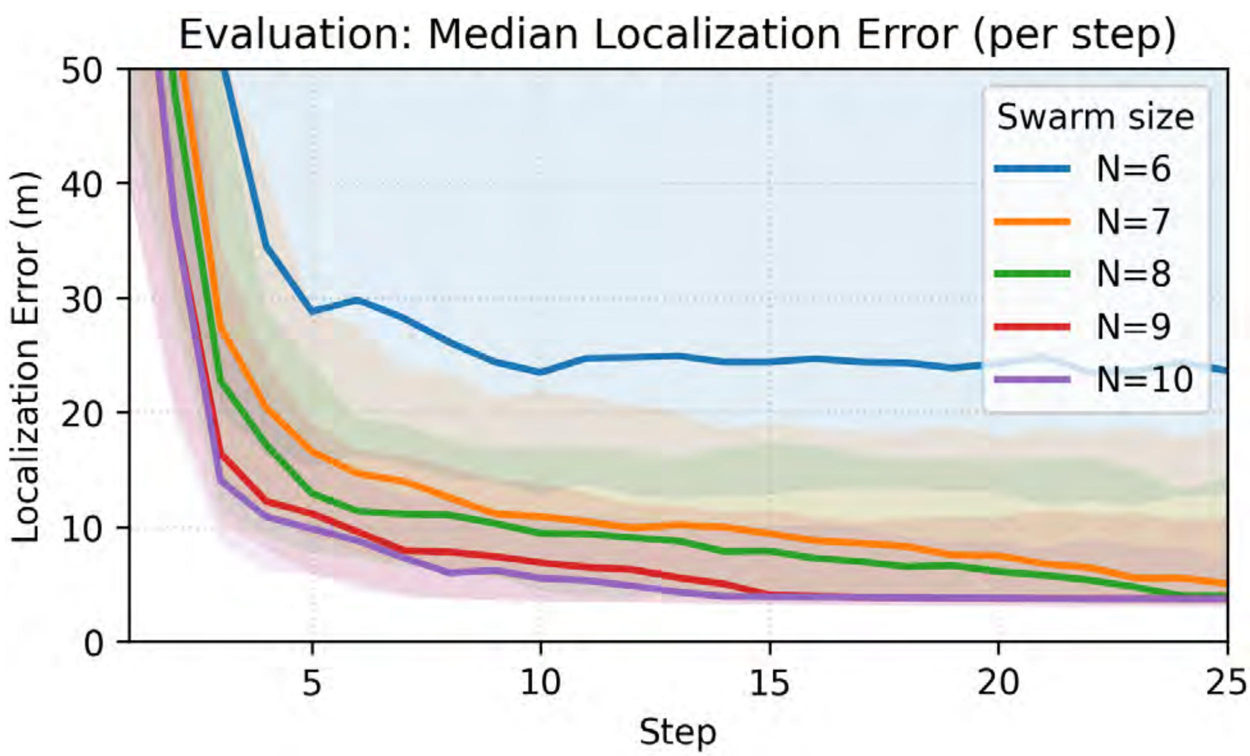

Figure 6.77: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

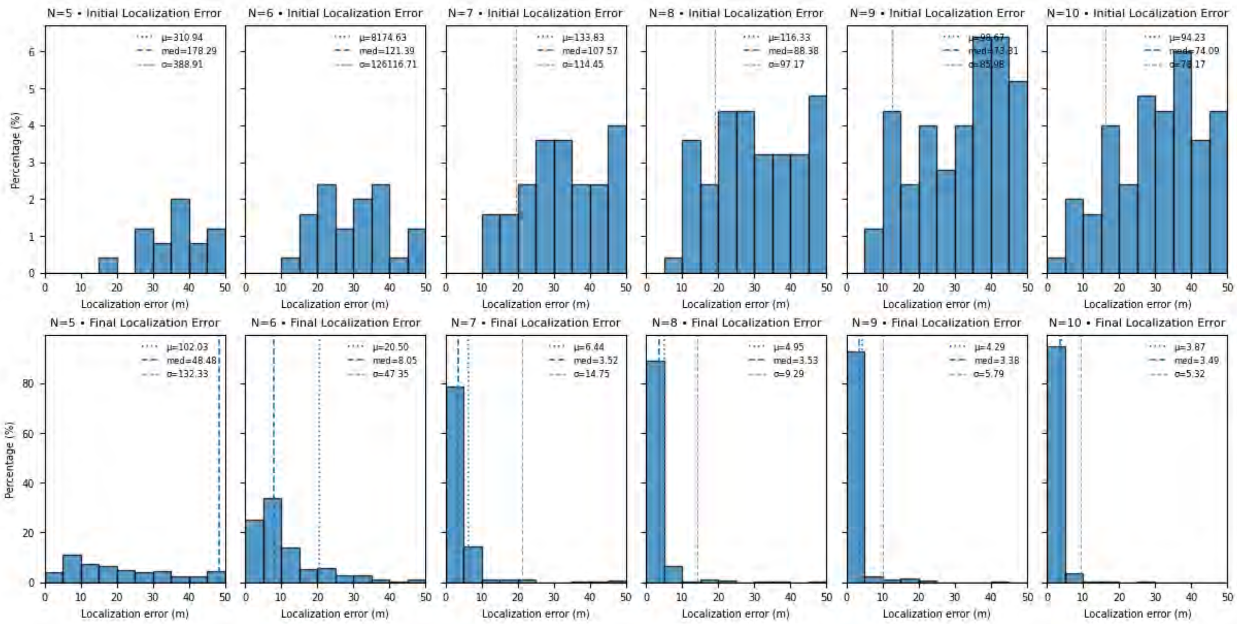

Figure 6.78: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

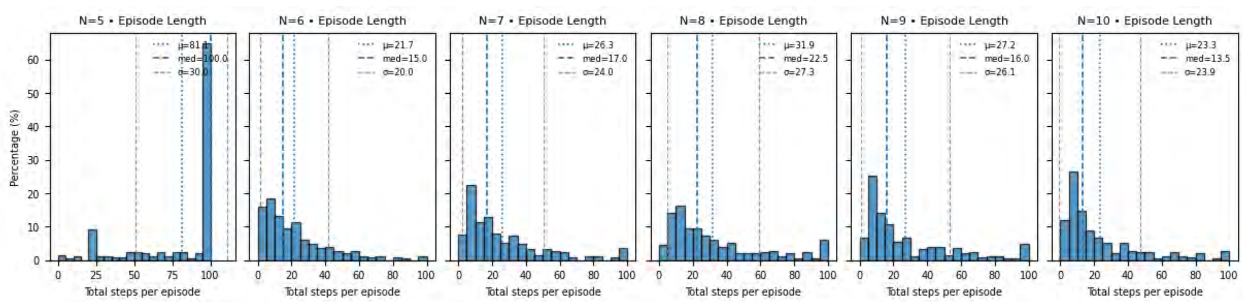

Figure 6.79: DQN final architecture episode length for varying swarm sizes and directional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

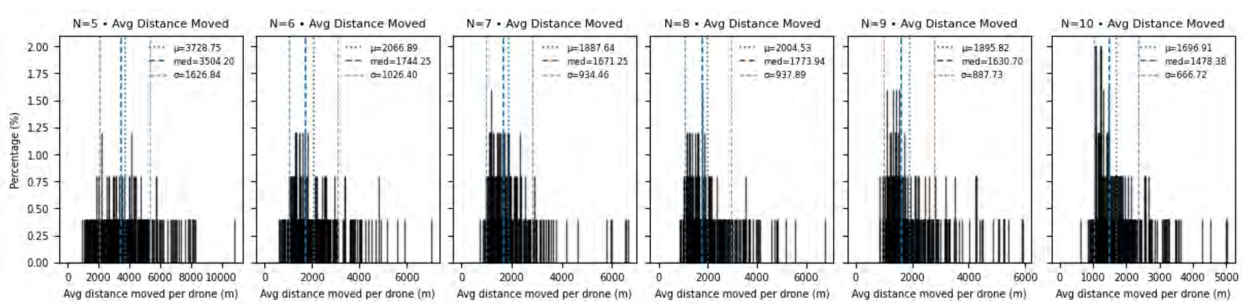

Figure 6.80: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

2.4 GHz Emitter, Omnidirectional Antennas Similar to the UAS with directional antennas, the omnidirectional UAS trained at 5 GHz seeking a 2.4 GHz emitter shows lower median localization accuracy across episode steps and requires more steps for convergence. However, while the performance is not optimized, it appears that the policy trained at 5 GHz can be utilized at 2.4 GHz for omnidirectional antennas.

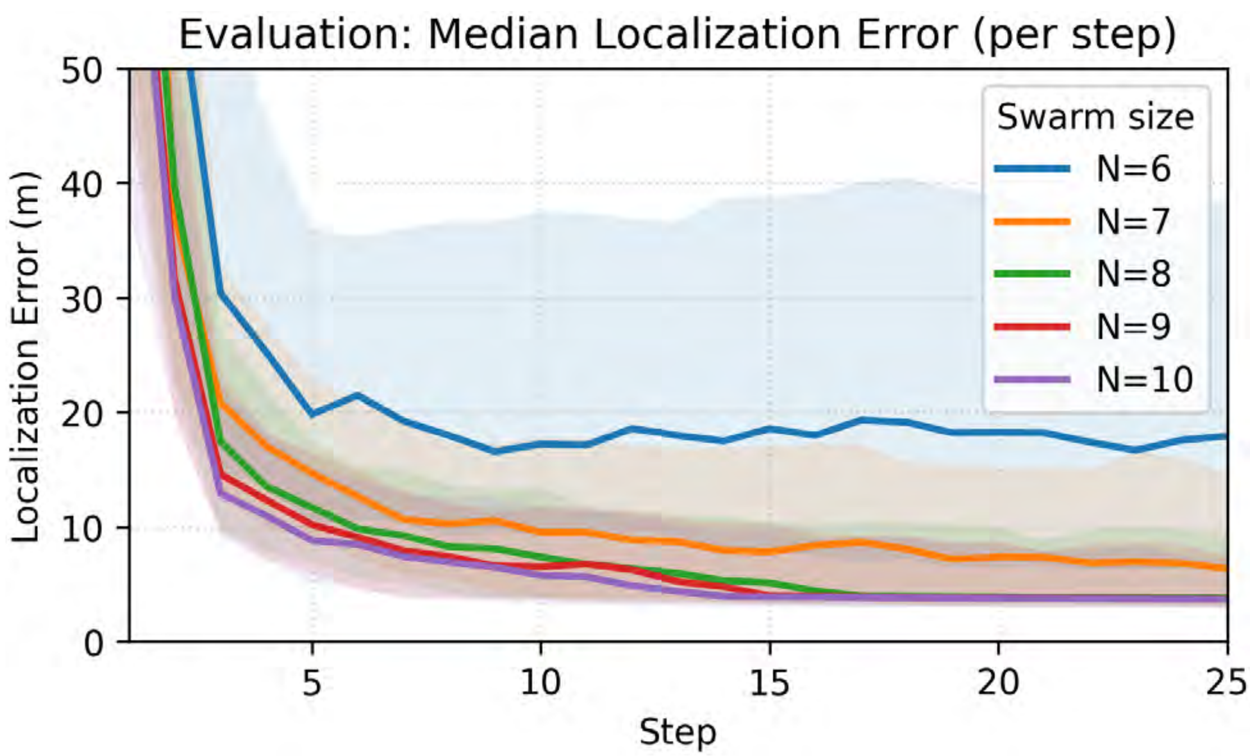

Figure 6.81: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

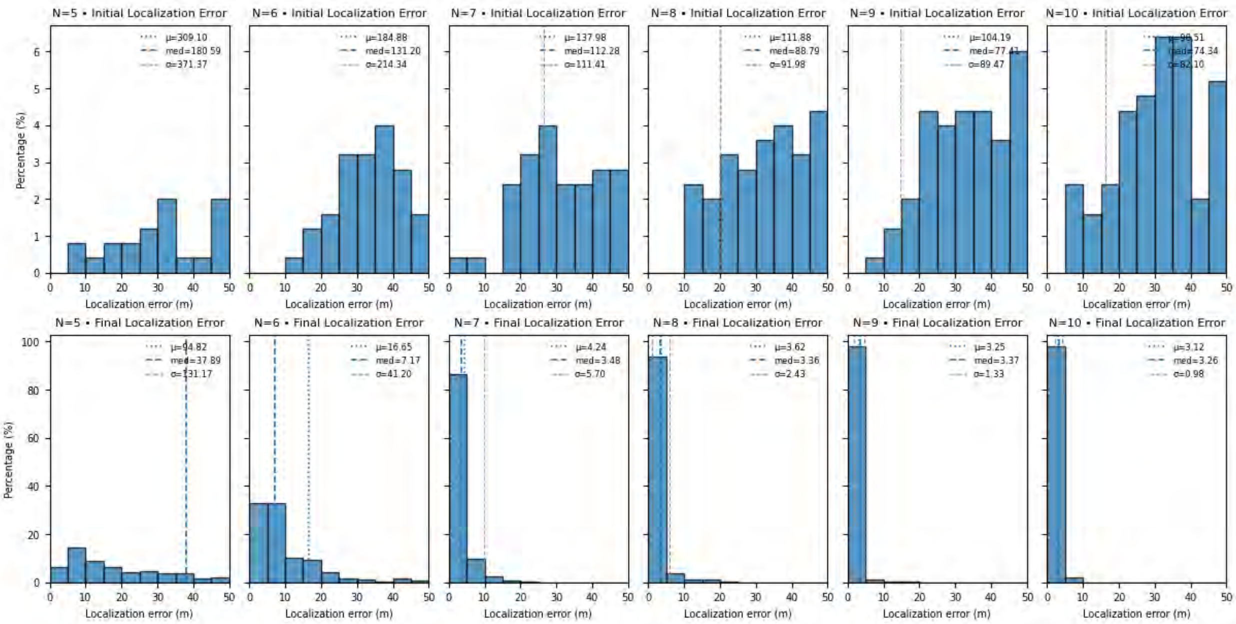

Figure 6.82: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

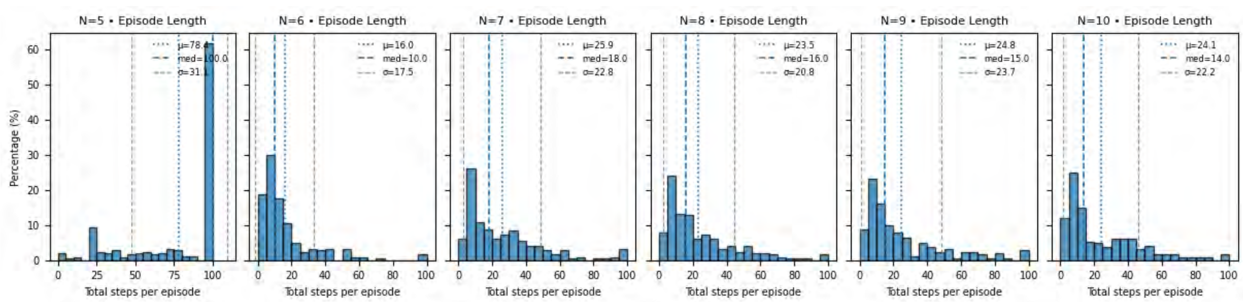

Figure 6.83: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

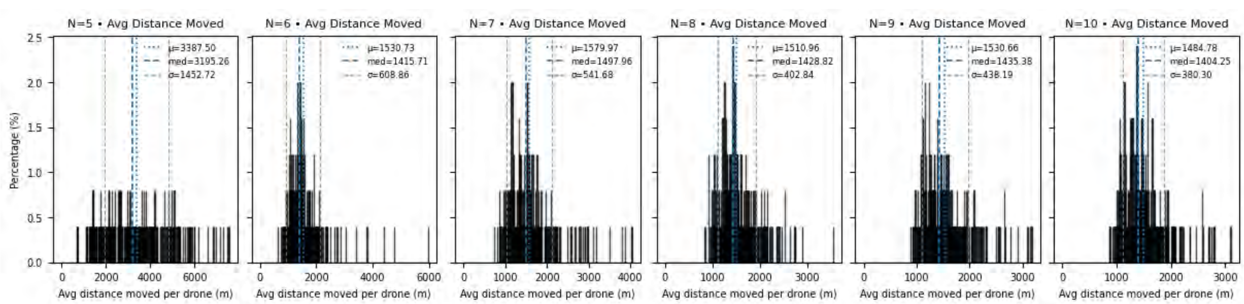

Figure 6.84: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 2.4 GHz signal.

8 GHz Emitter, Directional Antennas The UAS with directional antennas trained at 5 GHz seeking an 8 GHz emitter shows lower median localization accuracy across episode steps and requires more steps for convergence. However, while the performance is not optimized, it appears that the policy trained at 5 GHz can be utilized at 8 GHz for directional antennas.

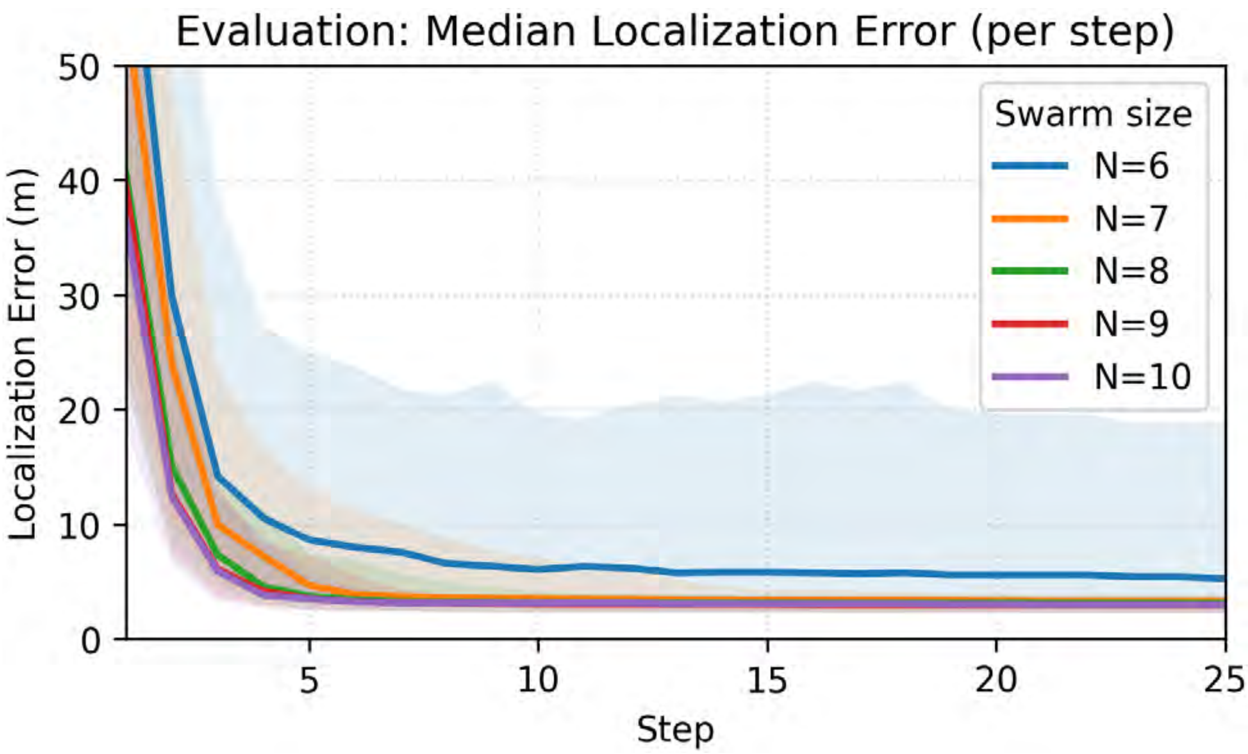

Figure 6.85: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and directional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

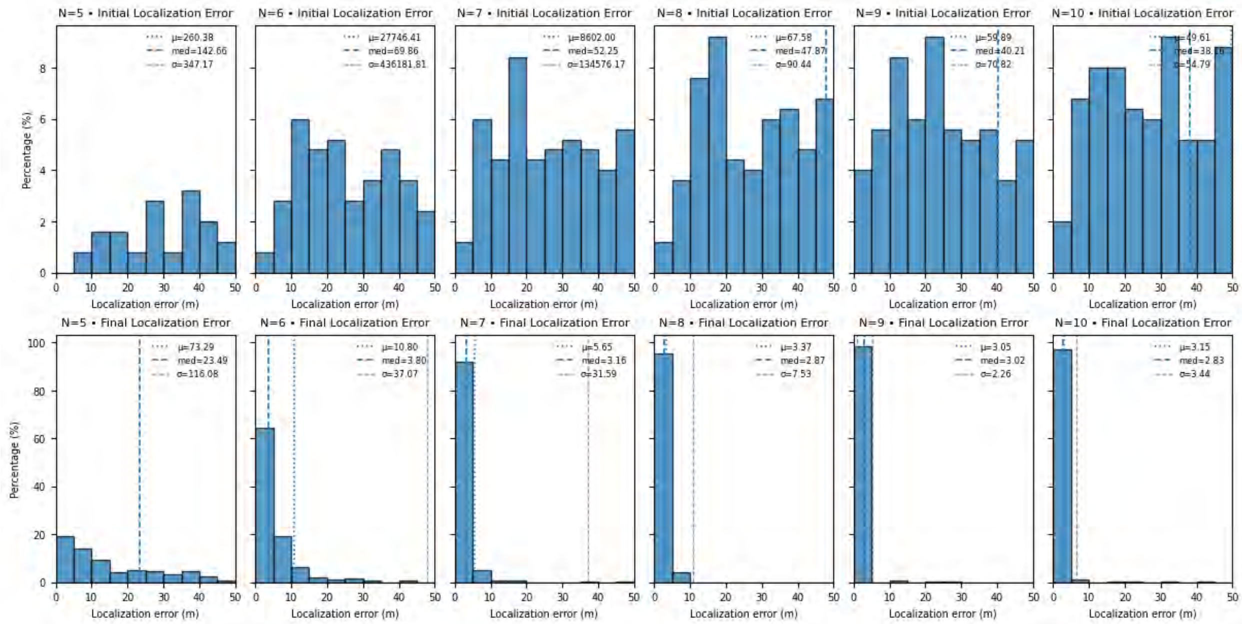

Figure 6.86: DQN final architecture comparison of initial and final localization error for varying swarm sizes and directional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

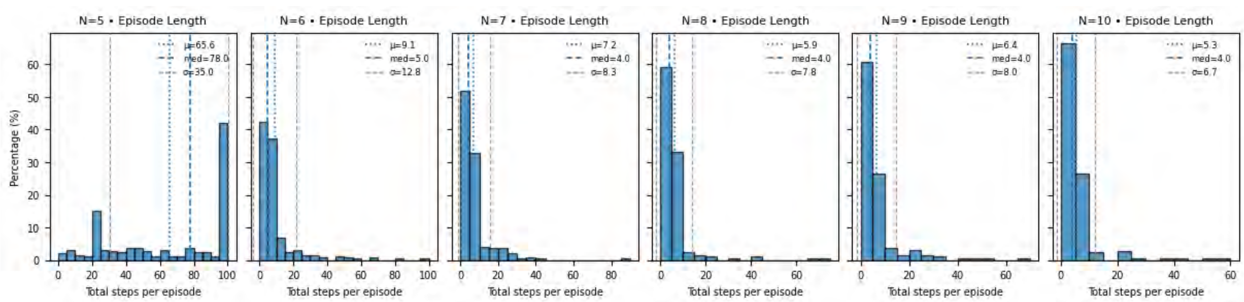

Figure 6.87: DQN final architecture episode length for varying swarm sizes and directional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

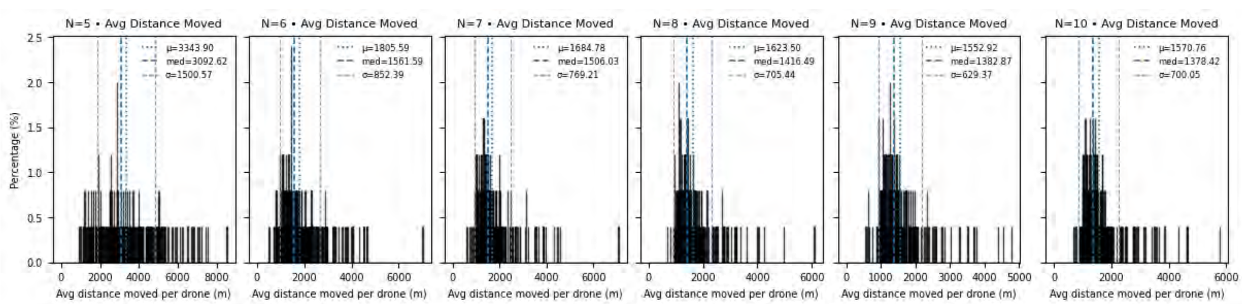

Figure 6.88: DQN final architecture average distance moved for each drone for varying swarm sizes and directional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

8 GHz Emitter, Omnidirectional Antennas The UAS with omnidirectional antennas trained at 5 GHz seeking an 8 GHz emitter shows lower median localization accuracy across episode steps and requires more steps for convergence. However, while the performance is not optimized, it appears that the policy trained at 5 GHz can be utilized at 8 GHz for directional antennas.

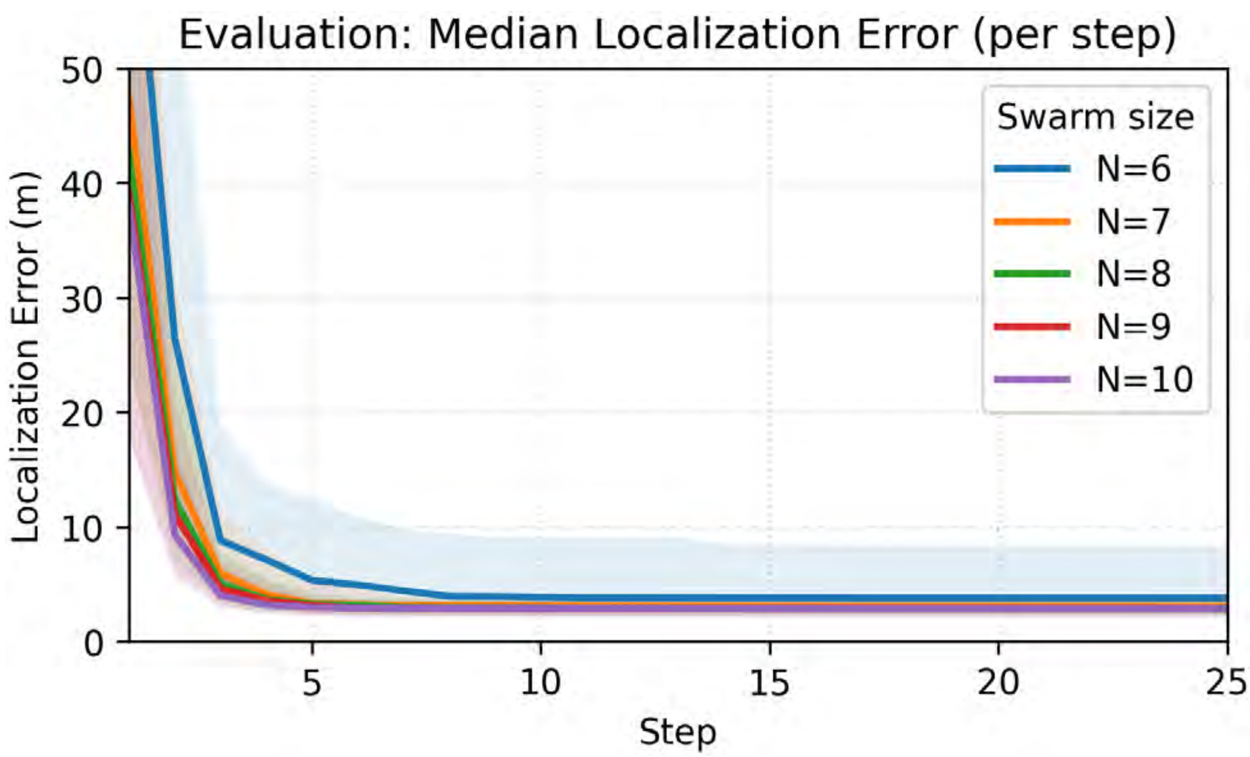

Figure 6.89: DQN geometry-aware median localization error performance for swarm sizes of 6 to 10 UAS elements with shaded standard deviation and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

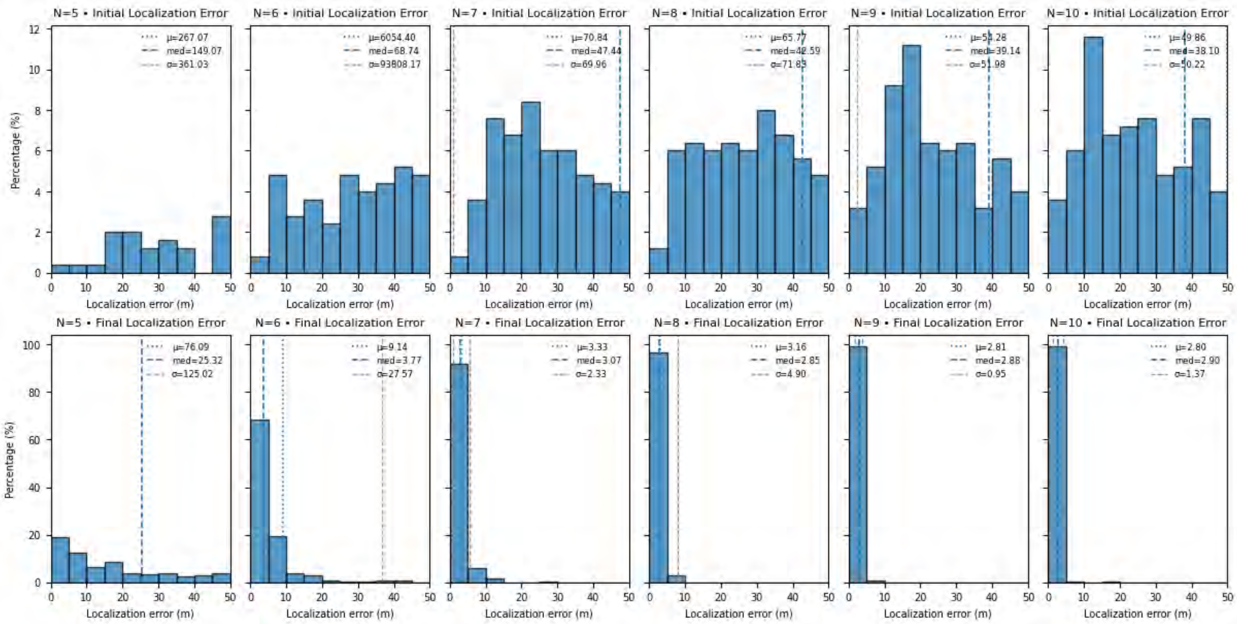

Figure 6.90: DQN final architecture comparison of initial and final localization error for varying swarm sizes and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

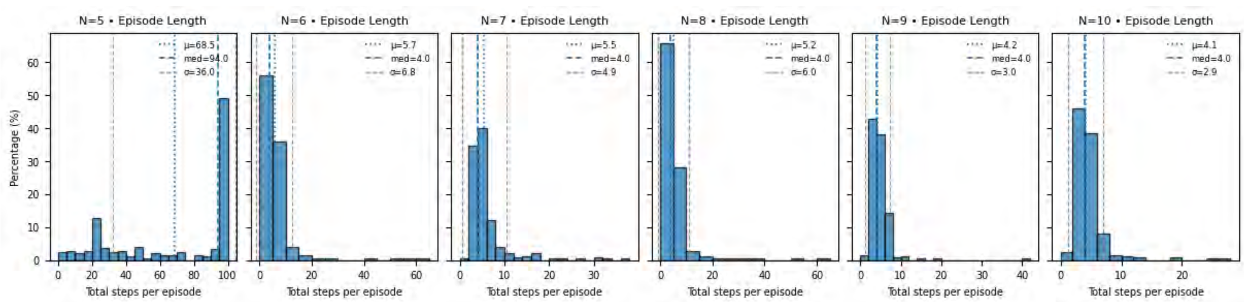

Figure 6.91: DQN final architecture episode length for varying swarm sizes and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

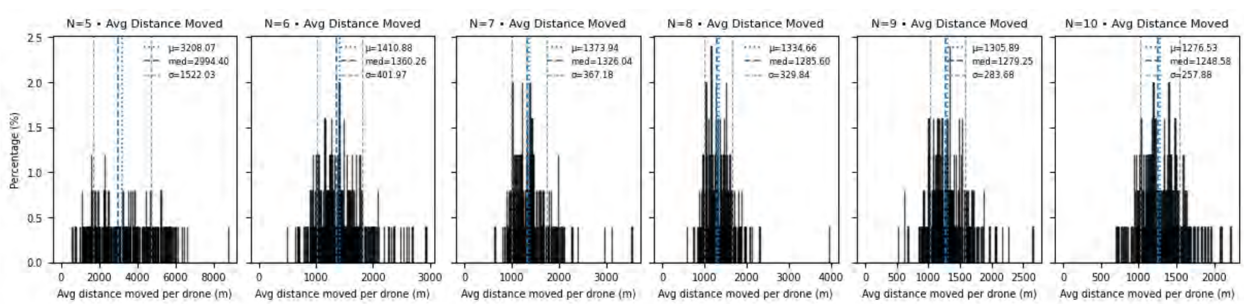

Figure 6.92: DQN final architecture average distance moved for each drone for varying swarm sizes and omnidirectional antennas; system trained for 5 GHz and seeking emitter with 8 GHz signal.

6.3.6.3 *Frequency Analysis Conclusion*

The system shows comparable performance when training at a specific frequency against training at the baseline 5 GHz frequency, meaning the policy is dependent primarily on geometry and can be applied across the frequency band. This is understandable as the LOCA equation is dependent on geometry only and our rewards are otherwise signal strength based. Other methods, like phase correlative approaches, require wavelength-driven decisions on sensor geometry. To this extent, we intentionally avoided phase-coherent payloads in this work. This observation that a policy trained at 5 GHz transfers to 2.4 GHz and 8 GHz without retraining, consistent with meter-scale geometric control rather than wavelength-scale phase observables, further shows that correlative interferometry is not well matched to the dissertation's target scenario.

6.3.7 Analysis of Final Architecture using EXata Digital-Twin

We evaluate the performance of our final architecture and its trained policies using EXata [103]. Here, we consider both omnidirectional and directional antenna performance, and quantify the performance of the system in a more robust environmental simulation. The EXata architecture utilized a system of eight UAS modeled under Rician fading in a simulated urban area where the UAS typically began its search 300 m–500 m from the emitter.

The initial starting positions of the UAS followed grid-like positions as in Sec. 6.3.3.2, where neighboring UAS were within 100 m in the x and y coordinates. These starting positions allowed the analysis to consider realistic implementations of the UAS system.

6.3.7.1 *Omnidirectional Antenna Performance*

The omnidirectional antenna performance showed nominal performance comparable to what was found in Sec. 5.6.5.2, with median localization error around 4.7 m and converging in approximately 5 steps, shown in Fig. 6.93. However, while the median performance showed positive results, the standard deviation of the results was significantly high due to the performance from the structured starting positions. This result indicates an opportunity for further refinement either in the starting position restrictions or in the controller itself to enable a wider range of UAS geometries.

We also show trajectories for the UAS in both 2D and 3D in Fig. 6.94, with starting

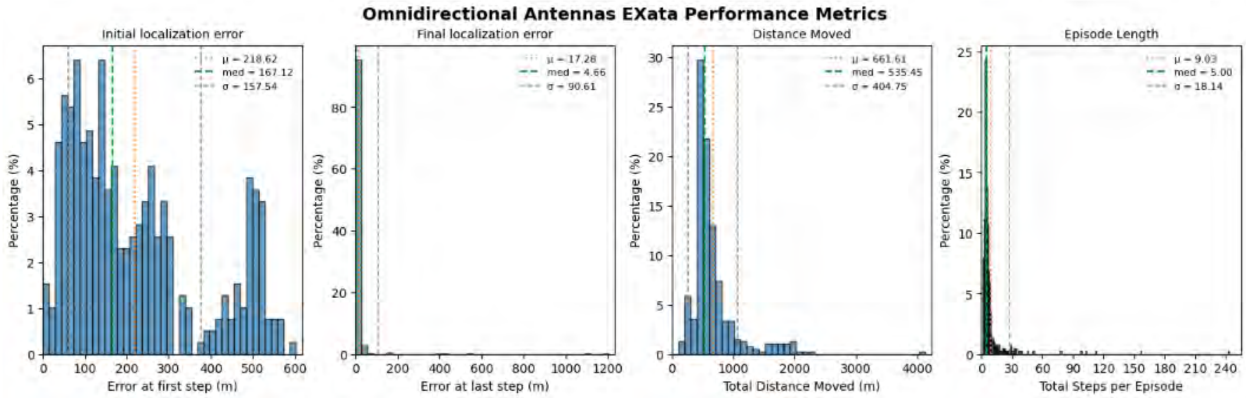

Figure 6.93: Localization performance in EXata of final architecture with omnidirectional antennas. Performance is comparable to Python simulations.

positions provided by $\circ$ and final positions provided by $\triangle$. The median performance shows the system effectively converging on the emitter in approximately two steps, however, the controller seeks to refine the SEP to a stable minimum. This result supports a looser target threshold to provide useful emitter location results in very few snapshots.

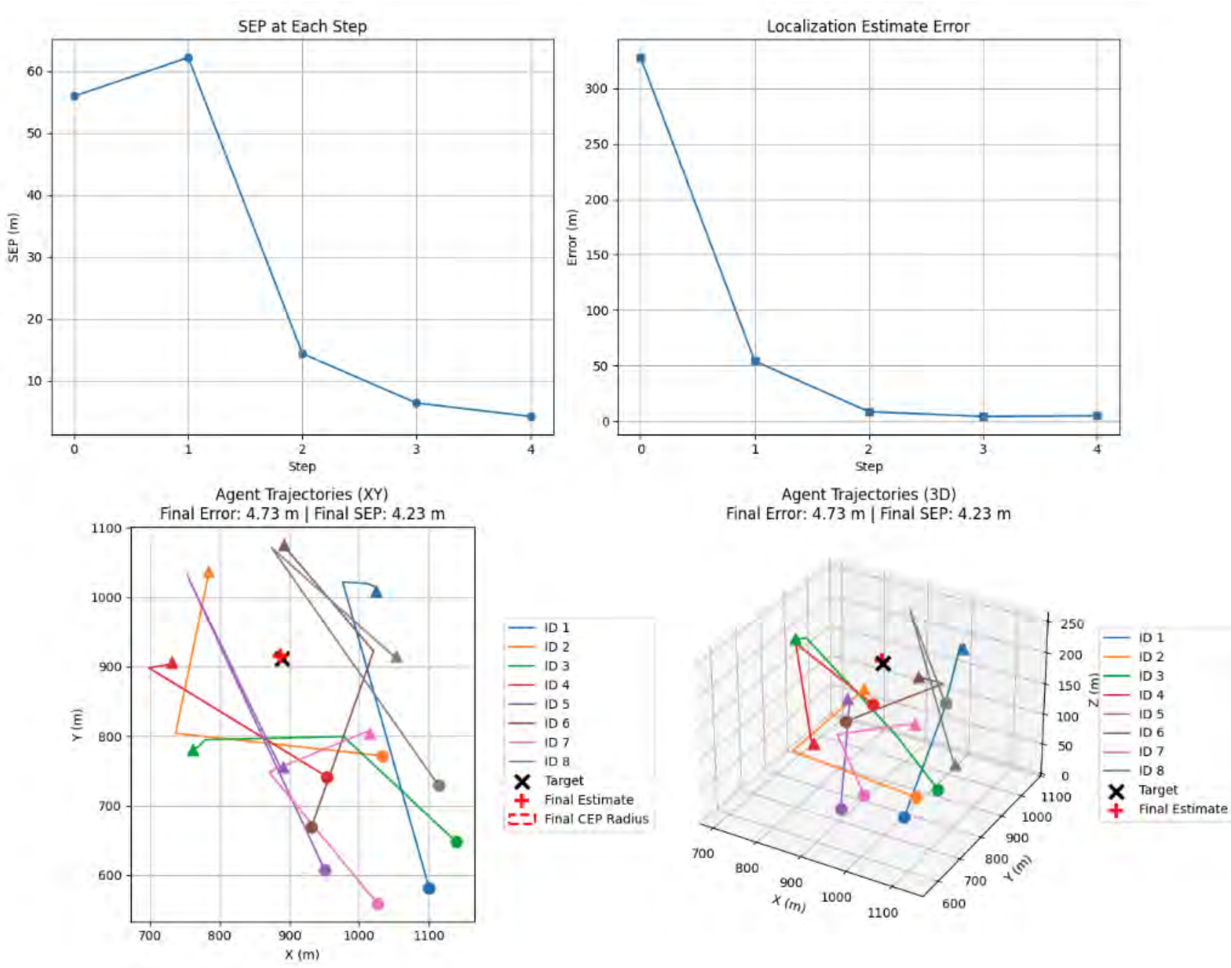

Figure 6.94: An example of the median performance of the UAS trajectory in EXata with final architecture and omnidirectional antennas, showing the SEP at each step, the localization estimate error, and a 2D and 3D depiction of the initial and final UAS positions.

6.3.7.2 *Directional Antenna Performance*

We further evaluated the architecture in EXata using directional antennas. In this evaluation, we utilized antenna patterns shown in Fig.6.95, where the horizontal pattern corresponds to the azimuth direction and the vertical pattern corresponds to the elevation direction. Because the main lobe for these patterns does not peak at 0°, but instead has nominal high gain near 180°, we provided a deterministic command to the controller to ensure the UAS heading was pointing the 180° portion of the horizontal antenna pattern towards the emitter.

The directional antenna performance showed nominal performance comparable to what was found in Sec. 6.3.1, with median localization error around 5.8 m and converging in approximately 8 steps, shown in Fig. 6.96. Again, the localization error had a significantly high standard deviation, owed to the structured starting positions. We show trajectories for the UAS in both 2D and 3D in Fig. 6.97, with starting positions provided by $\circ$ and final positions provided by $\triangle$. Similar to the omnidirectional case, the median performance shows the system effectively converging on the emitter in approximately two steps.

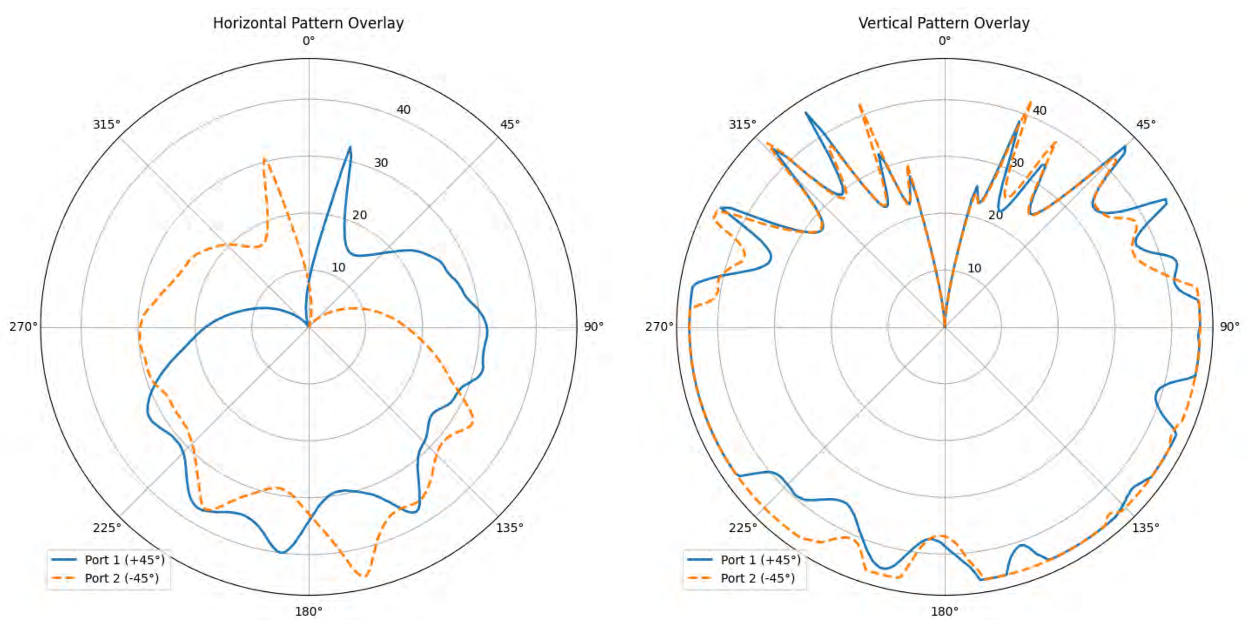

Figure 6.95: Horizontal and vertical antenna patterns used for EXata directional antenna analysis. The main lobe does not peak at 0° but instead has optimal performance near 180°.

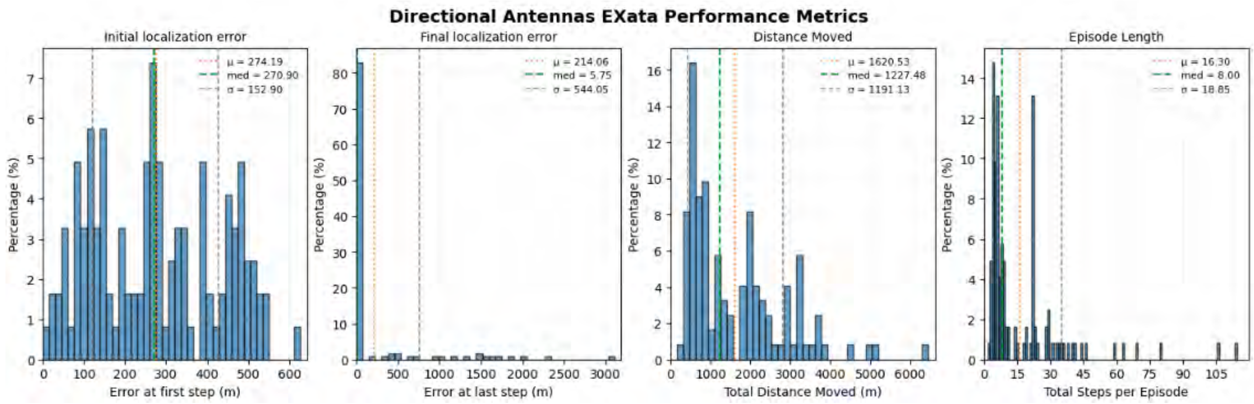

Figure 6.96: Localization performance in EXata of final architecture with directional antennas. Performance is comparable to Python simulations.

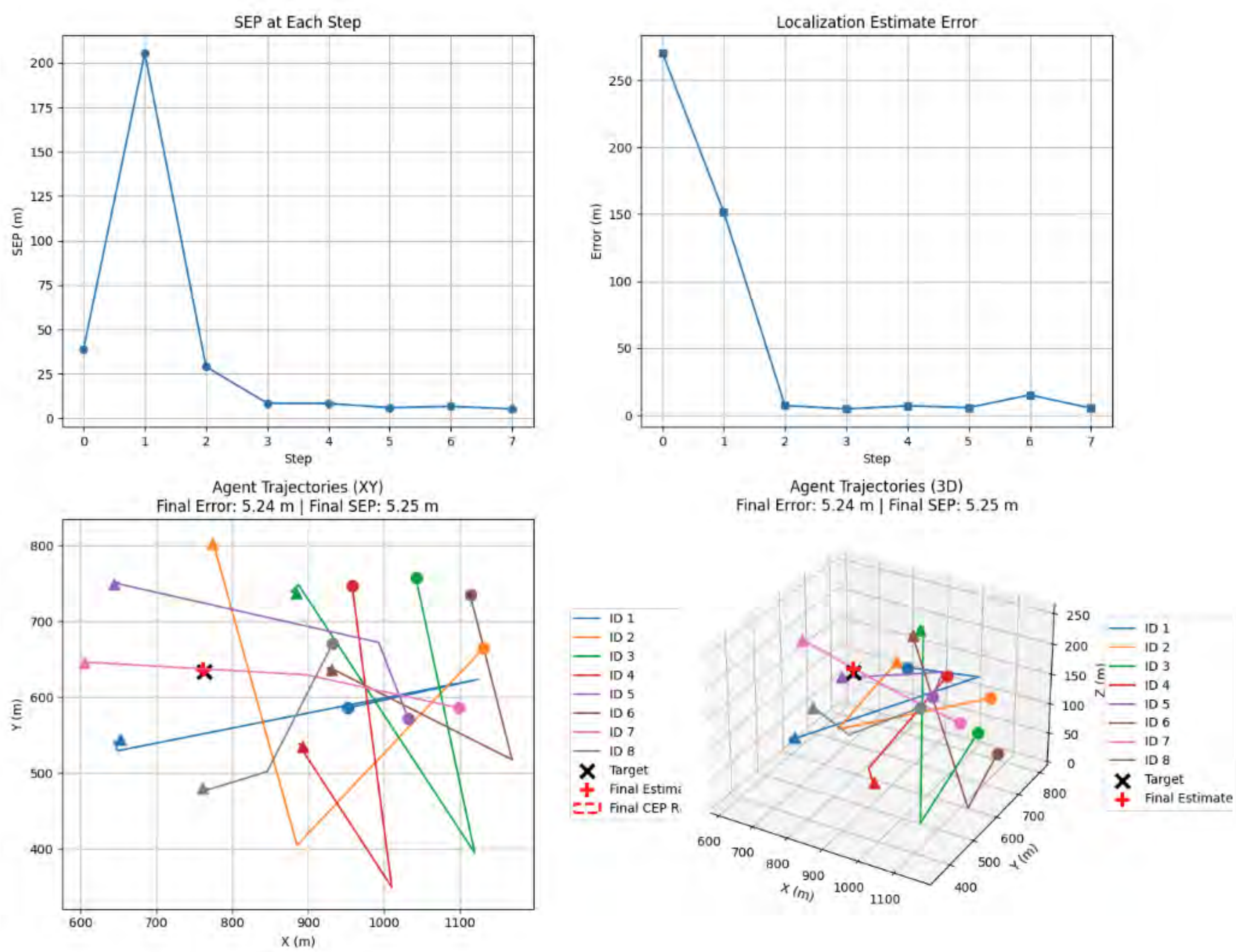

Figure 6.97: An example of the median performance of the UAS trajectory in EXata with final architecture and directional antennas, showing the SEP at each step, the localization estimate error, and a 2D and 3D depiction of the initial and final UAS positions.

Chapter 7
CONCLUSION

In this dissertation, we investigated cooperative UAS geolocation of RF emitters using time-based measurements in environments that include realistic hardware error sources and multipath. Our hypothesis was that if a cooperative UAS array can intelligently adapt its geometry, then it can efficiently reduce localization error in complex multipath environments and quickly converge on an accurate solution for an unknown-location emitter.

The central contribution of this work is the development and evaluation of geometry controllers that dynamically reposition the array to improve localization performance, and we validated our machine learning-based controllers in Monte Carlo simulations and in an EXata digital-twin environment.

7.1 Summary of Technical Findings

First, we established baseline performance and sensitivities for time-based geolocation methods under both error-free and error-induced conditions. In error-free conditions, both multilateration and LOCA improve with increasing sensor quantity, with results that improve in a roughly logarithmic manner and generally indicate that array sizes on the order of five to ten sensors can be sufficient for many mission profiles.

Second, when realistic sensor positioning and local timing errors are included, the two methods diverge significantly: multilateration remains far above the CRLB and shows limited benefit beyond modest increases in sensor count, while LOCA remains much closer to the CRLB and continues to improve with increasing sensor quantity. In these simulations, we observed diminishing improvements in emitter localization accuracy for array sizes in excess of approximately eight sensors, emphasizing that geometry and measurement quality dominate once sensor redundancy is achieved.

Third, we leveraged LOCA's subset-based solution cloud to enable estimate filtering and confidence metrics that do not require knowledge of the true emitter location. With only five sensors, a single LOCA estimate exists per snapshot; with six or more sensors, redundant subsets produce a solution cloud that can be filtered statistically with sufficiently large subset solutions per snapshot with 8 sensors to enable meaningful outlier rejection and confidence in the solution median estimates. This capability is foundational for adaptive geometry control because it provides an internal convergence signal (e.g., SEP and distribution spread) even when the true emitter position is unknown.

Fourth, we demonstrated that active geometry control provides a measurable advantage over static sensing. In comparative policy experiments across multiple RL approaches, all controllers reduced SEP within approximately 5-–10 repositioning steps, and most RL policies achieved greater than 60% error reduction by step 10 in the evaluated dense-multipath simulation setting. The results also show meaningful tradeoffs between convergence speed and motion effort: DQN and A3C were comparatively conservative in distance traveled, while MADDPG expended more distance to converge in fewer steps.

Fifth, we validated the geometry-control approach in an EXata digital twin with Rician fading and adverse weather, showing that the learned policy is not limited to the simplified Monte Carlo channel model. In EXata, the system demonstrated more stable performance, converging in fewer repositioning steps and achieving higher localization accuracy overall (e.g., mean step count 2.8 in EXata versus 9.4 in the Python simulation, with mean final SEP on the order of 4.1 m). Further, the EXata comparisons indicate that while adverse multipath increases initial error, geometry refinement rapidly overcomes these impacts once repositioning begins.

7.2 Practical Implications

The results support the conclusion that cooperative UAS geolocation performance is not determined solely by the choice of time-based algorithm, but by the combined design of (i) measurement quality bounds (sensor positioning and timing), (ii) redundancy via multi-sensor cooperation, and (iii) an adaptive geometry strategy that actively improves conditioning relative to the emitter. In practice, this suggests that system designers can trade platform count and coordination complexity against achievable accuracy, and that once a modest level of redundancy is achieved, intelligent geometry adaptation becomes the dominant discriminator for efficient convergence in dense multipath environments.

7.3 Limitations

This study assumes that sensors can consistently timestamp the same received signal feature (or signal edge) when producing TOA/TDOA measurements, and errors in feature alignment and correlation can introduce additional effective timing error that is not explicitly modeled here. Additionally, while the Monte Carlo channel model was intentionally conservative, it can over-penalize conditions where the array converges into true LoS regimes, motivating the EXata digital twin comparisons and future empirical validation.

7.4 Future work

While the geometry controller developed in this research sufficiently provides the ability to quickly find an unknown-location emitter in a dense multipath environment, future research could further enhance the operational utility of this system.

Controller Parameter Extension The controller developed in this study provides parameters for geometry shaping and fundamental sensing information for the UAS to converge on an unknown-location emitter. Other parameters can be implemented and refined for more complex use cases, such as those to improve signal correlation across measurements or to account for antenna mismatch errors. Similarly, inter-UAS collision avoidance and obstacle avoidance are assumed to be handled by other features of the autonomous UAS. In these cases, other autonomy-enabling sensors, such as the onboard LiDAR, could be used to provide continuous physical geometry feedback. Using LiDAR, when the UAS approaches physical obstructions, the controller can treat them as visual barriers and implement them as both RL reward penalties and specific deterministic control mechanisms to reduce the likelihood of collisions.

Further, when the directional antennas were implemented into the architecture, the system utilized the policies generated for the omnidirectional antennas. It is possible that including the antenna pattern into the reward function and the observation states could improve the convergence of the system instead of solely relying on the deterministic heading control of the UAS. In these cases, the system could either implement emitter bearings directly or indirectly as part of the convergence methodology, and the UAS system would use this additional environmental information to localize the emitter in fewer steps.

Multiple Emitter Use Cases This dissertation focuses on the geolocation of a single, stationary RF emitter using this cooperative UAS array. Extensions to multiple simultaneous emitters or to moving targets are outside the research scope, however, they are appropriate for future studies. In those studies, the utilization of MUSIC [21] as a signal pre-processing approach can separate the individual emitter signal information and potentially allow the UAS array to localize multiple emitters as groups of UAS subarrays.

Physical System Realization While the controller developed in this work is implemented in a quantitative simulation and in digital twin model of both the environment and the hardware, the physical implementation of the system is left as an opportunity for future development. In an extension of this work, empirical studies would be performed using real hardware to quantify end-to-end error sources and validate convergence behavior outside these simulations.

Chapter 8
APPENDIX

APPENDIX A

Table 8.1: Algorithm terms and parameters.

Term	**Description**
$s_i = (x_i, y_i, z_i)$	UAS (sensor) position
$\widehat{E} = (\widehat{x}_e, \widehat{y}_e, \widehat{z}_e)$	Emitter location estimate
Δt_{ij}	Time difference of arrival (TDOA) between sensors i and j
c	Signal propagation speed
d_i	Euclidean distance of sensor i to system origin
$\Lambda = \{\lambda_m\}_{m=1}^{M}$	Cloud of LOCA solutions (candidate emitter locations)
$SEP_{50} = q_{0.5}\left\{\left\lVert\lambda_m - \widehat{E}\right\rVert\right\}$	Spherical error probable (median) of LOCA solutions relative to $\widehat{E}$
$\sigma_{t,i}^2 = \dfrac{3R_i^2}{4\pi^2 f\, T_s\, B^2\, r_0^2\, SNR_0}$	Range-dependent timing (or measurement) variance for sensor i
f	Operating frequency
R_i	Emitter–sensor range for sensor i
T_s	Sensor integration time
B	Sensor bandwidth
r_0	Minimum sensor operational range
SNR_0	Minimum sensor signal-to-noise ratio
P	State estimation error covariance (Kalman filter)
z_k	Weighted-fused LOCA position measurement at step k
R_k	Measurement covariance associated with z_k
ω_m	Variance-aware weight for LOCA solution λ_m
$w_k,\ v_k$	Kalman filter process noise and measurement noise (at step k)

BIBLIOGRAPHY

[1] C. Yan, L. Fu, J. Zhang, and J. Wang, "A Comprehensive Survey on UAV Communication Channel Modeling," *IEEE Access*, vol. 7, pp. 107 769–107 792, 2019. 1, 17

[2] R. Chen, B. Yang, and W. Zhang, "Distributed and Collaborative Localization for Swarming UAVs," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 5062–5074, Mar. 2021. 1, 75

[3] D. Wei, L. Zhang, Q. Liu, H. Chen, and J. Huang, "UAV Swarm Cooperative Dynamic Target Search: A MAPPO-Based Discrete Optimal Control Method," *Drones*, vol. 8, no. 6, p. 214, May 2024. 1, 19, 20

[4] D. J. Torrieri, "Statistical theory of passive location systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 2, pp. 183–198, 1984. 1, 8, 11, 12, 13, 32, 41

[5] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall PTR, 1993. 1, 41, 42

[6] W. Wang, P. Bai, H. Li, and X. Liang, "Optimal Configuration and Path Planning for UAV Swarms Using a Novel Localization Approach," vol. 8, no. 6, p. 1001. [Online]. Available: https://www.mdpi.com/2076-3417/8/6/1001 1, 17, 21

[7] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, 1st ed. John Wiley & Sons, Incorporated. 2, 11, 13

[8] S. M. Sheikh, H. M. Asif, K. Raahemifar, and F. Al-Turjman, "Time difference of arrival based indoor positioning system using visible light communication," *IEEE access : practical innovations, open solutions*, vol. 9, pp. 52 113–52 124, 2021. 4, 37

[9] S. Knappe, V. Shah, P. Schwindt, L. Hollberg, J. Kitching, L.-A. Liew, and J. Moreland, "A microfabricated atomic clock," *Applied Physics Letters*, no. 85, 2004. 4, 37, 56

[10] J. Kitching, "Chip scale atomic devices," *Applied Physics Reviews*, 2018. 4, 37, 56

[11] C. Peters and M. A. Thornton, "Cooperative UAS geolocation of emitters with multi-sensor-bounded timing and localization error," in *2023 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, mar 2023, pp. 1–13. 5, 24, 39, 55, 64

[12] F. Mekelleche and H. Hafid, "Classification and comparison of range-based localization techniques in wireless sensor networks," *Journal of Communications*, vol. 12, pp. 221–227, Apr. 2017. 8

[13] SR. Leelavathy and S. Sophia, "Providing localization using triangulation method in wireless sensor networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 4, no. 6, pp. 47–50, 2014. 8

[14] K. Anup and S. Takuro, "Localization in wireless sensor networks: A survey on algorithms," *Measurement Techniques, Applications and Challenges, J. Sens. Actuator Netw*, vol. 6, 2017. 8

[15] M. K. Kumar and V. K. Prasad, "TASLT: Triangular area segmentation based localization technique for wireless sensor networks using AoA and RSSI measures – a new approach," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 2021, pp. 585–590. 8

[16] A. Norrdine, "An algebraic solution to the multilateration problem," in *Proceedings of the 15th International Conference on Indoor Positioning and Indoor Navigation, Sydney, Australia*, vol. 1315, 2012. 8

[17] M. D. Gillette and H. F. Silverman, "A linear closed-form algorithm for source localization from time-differences of arrival," *IEEE Signal Processing Letters*, vol. 15, pp. 1–4, 2008. 8

[18] S. Reddi, "An exact solution to range computation with time delay information for arbitrary array geometries," *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 485–, 1993. 8

[19] R. O. Schmidt, "A new approach to geometry of range difference location," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-8, no. 6, pp. 821–835, 1972. 9

[20] ——, "Least squares range difference location," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 234–242, 1996. 9, 25

[21] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986. 11, 178

[22] A. Paulraj, R. Roy, and T. Kailath, "Estimation Of Signal Parameters Via Rotational Invariance Techniques- Esprit," in *Nineteeth Asilomar Conference on Circuits, Systems and Computers, 1985.* IEEE, pp. 83–89. [Online]. Available: http://ieeexplore.ieee.org/document/671426/ 11

[23] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," vol. 13, no. 4, pp. 67–94. [Online]. Available: http://ieeexplore.ieee.org/document/526899/ 11

[24] K. Ho and W. Xu, "An Accurate Algebraic Solution for Moving Source Location Using TDOA and FDOA Measurements," vol. 52, no. 9, pp. 2453–2463. [Online]. Available: http://ieeexplore.ieee.org/document/1323254/ 12

[25] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., ser. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall. 13

[26] A. Zanella, "Best Practice in RSS Measurements and Ranging," vol. 18, no. 4, pp. 2662–2686. [Online]. Available: https://ieeexplore.ieee.org/document/7451182/ 13

[27] F. Gustafsson, "Particle filter theory and practice with positioning applications," vol. 25, no. 7, pp. 53–82. [Online]. Available: https://ieeexplore.ieee.org/document/5546308/ 14

[28] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, nachdr. ed., ser. A Wiley-Interscience Publication. Wiley. 14

[29] A. Amar and A. Weiss, "Direct position determination of multiple radio signals," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, pp. ii–81. [Online]. Available: https://ieeexplore.ieee.org/document/1326199/ 14

[30] W. A. Gardner, Ed., *Cyclostationarity in Communications and Signal Processing: ... Plenary Lectures at the Workshop on Cyclostationary Signals, Which Was Held August 16 - 18, 1992, at the Napa Valley Lodge in Yountville, California*. IEEE Press. 15

[31] W. Gardner and C.-K. Chen, "Signal-selective time-difference-of-arrival estimation for passive location of man-made signal sources in highly corruptive environments. I. Theory and method," vol. 40, no. 5, pp. 1168–1184. [Online]. Available: http://ieeexplore.ieee.org/document/134479/ 15

[32] C.-K. Chen and W. Gardner, "Signal-selective time-difference of arrival estimation for passive location of man-made signal sources in highly corruptive environments. II. Algorithms and performance," vol. 40, no. 5, pp. 1185–1197. [Online]. Available: http://ieeexplore.ieee.org/document/134480/ 15

[33] A. M. Reuven and A. J. Weiss, "Direct position determination of cyclostationary signals," vol. 89, no. 12, pp. 2448–2464. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0165168409001650 15

[34] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2. IEEE, pp. 775–784. [Online]. Available: http://ieeexplore.ieee.org/document/832252/ 16

[35] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," vol. 37, no. 6, pp. 1067–1080. [Online]. Available: http://ieeexplore.ieee.org/document/4343996/ 16

[36] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, J. Prieto, R. J. Durán *et al.*, "Map-Aware Models for Indoor Wireless Localization Systems: An Experimental Study," vol. 13, no. 5, pp. 2850–2862. [Online]. Available: http://ieeexplore.ieee.org/document/6786063/ 16

[37] H. Wymeersch, J. He, B. Denis, A. Clemente, and M. Juntti, "Radio Localization and Mapping With Reconfigurable Intelligent Surfaces: Challenges, Opportunities, and Research Directions," vol. 15, no. 4, pp. 52–61. [Online]. Available: https://ieeexplore.ieee.org/document/9215972/ 16

[38] A. Guerra, F. Guidi, D. Dardari, and P. M. Djurić, "Reinforcement Learning for Joint Detection and Mapping Using Dynamic UAV Networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 3, pp. 2586–2601, Jun. 2024. 17, 19, 20

[39] Y. Ding, Z. Yang, Q.-V. Pham, Y. Hu, Z. Zhang, and M. Shikh-Bahaei, "Distributed Machine Learning for UAV Swarms: Computing, Sensing, and Semantics," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7447–7473, Mar. 2024. 17, 18

[40] J. Jyoti and R. S. Batth, "Unmanned aerial vehicles (UAV) path planning approaches," in *2021 International Conference on Computing Sciences (ICCS)*, 2021, pp. 76–82. 17, 18

[41] S. Wu, "Illegal radio station localization with UAV-based Q-learning," vol. 15, no. 12, pp. 122–131. [Online]. Available: https://ieeexplore.ieee.org/document/8594722 17, 18, 19, 74, 75

[42] Y.-J. Chen, D.-K. Chang, and C. Zhang, "Autonomous Tracking Using a Swarm of UAVs: A Constrained Multi-Agent Reinforcement Learning Approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 702–13 717, Nov. 2020. 17, 18, 19, 20

[43] M. Shurrab, R. Mizouni, S. Singh, and H. Otrok, "Reinforcement learning framework for UAV-based target localization applications," *Internet of Things*, vol. 23, p. 100867, Oct. 2023. 17, 18, 82

[44] X. Chen, C. Fu, and J. Huang, "A Deep Q-Network for robotic odor/gas source localization: Modeling, measurement and comparative study," vol. 183, p. 109725. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0263224121006862 17, 18

[45] X. Cheng, F. Shu, Y. Li, Z. Zhuang, D. Wu, and J. Wang, "Optimal Measurement of Drone Swarm in RSS-Based Passive Localization With Region Constraints," *IEEE Open Journal of Vehicular Technology*, vol. 4, pp. 1–11, 2023. 17, 74, 75

[46] Y. Dong, F. Li, C. Ma, C. He, and Z. J. Wang, "UAV-Based Dynamic Object Tracking with Radio Map," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 9166–9170. 17, 19

[47] S. Aditya, A. F. Molisch, and H. M. Behairy, "A Survey on the Impact of Multipath on Wideband Time-of-Arrival Based Localization," vol. 106, no. 7, pp. 1183–1203. [Online]. Available: https://ieeexplore.ieee.org/document/8355715/ 17

[48] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. 18, 81, 83, 96, 110

[49] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Adaptive Computation and Machine Learning Series, F. Bach, Ed. Cambridge, MA, USA: MIT Press, Nov. 2018. 18, 19

[50] B. Li, J. Wang, C. Song, Z. Yang, K. Wan, and Q. Zhang, "Multi-UAV roundup strategy method based on deep reinforcement learning CEL-MADDPG algorithm," vol. 245, p. 123018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417423035200 19

[51] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, Jun. 2016, pp. 1928–1937. 19, 22, 88

[52] R. Lowe, YI. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. 20, 21, 88

[53] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and YI. WU, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 24 611–24 624. 20, 23, 88

[54] M. Narita, R. Kuroiwa, and J. C. Beck, "Reinforcement Learning-Based Heuristics to Guide Domain-Independent Dynamic Programming," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, G. Tack, Ed. Cham: Springer Nature Switzerland, 2025, pp. 137–154. 21, 88

[55] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," 2017. [Online]. Available: https://arxiv.org/abs/1611.01144 21

[56] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," 2017. [Online]. Available: https://arxiv.org/abs/1611.00712 21

[57] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2018. [Online]. Available: https://arxiv.org/abs/1506.02438 22, 23

[58] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347 22

[59] C. Peters and M. A. Thornton, "Reducing emitter localization error in urban environments with geometry adaptive UAS arrays," in *2025 IEEE International Systems Conference (SysCon)*, apr 2025, pp. 1–8. 24, 31, 39, 44, 55, 73, 74, 82, 85, 97, 99, 100, 110, 111

[60] ——, "Emitter localization using geometry aware UAVs," in *2026 IEEE Dallas Circuits and Systems Conference*, 2026. 31, 110

[61] 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. 32, 68

[62] NASA, "Calculating the CEP (Circular Error Probable)," Tech. Rep. NASA-CR-182996, 1987. [Online]. Available: https://ntrs.nasa.gov/citations/19890002104 32

[63] A. Bellés, D. Medina, P. Chauchat, S. Labsir, and J. Vilà-Valls, "Robust m-type error-state kalman filters for attitude estimation," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 840–844. 35

[64] D. Kim, J. Ha, and K. You, "Adaptive Extended Kalman Filter Based Geolocation Using TDOA/FDOA," 2011. 35, 90, 96

[65] B. Pardhasaradhi, P. Srihari, and P. Aparna, "Navigation in gps spoofed environment using m-best positioning algorithm and data association," *IEEE Access*, vol. 9, pp. 51 536–51 549, 2021. 35

[66] D. Vasisht, S. S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016, pp. 165–178. 37

[67] D. Katabi, D. Vasisht, and S. S. Kumar, "Sub-decimeter radio frequency ranging," US Patent 9,961,495, May, 2018. 37

[68] S. Prager, M. S. Haynes, and M. Moghaddam, "Wireless subnanosecond RF synchronization for distributed ultrawideband software-defined radar networks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 11, pp. 4787–4804, 2020. 37, 42

[69] L. Hollberg and J. Kitching, "Miniature frequency standard based on all-optical excitation and a micro-machined containment vessel," US Patent 6,806,784, Oct., 2004. 37

[70] Microchip. Chip scale atomic clock (CSAC). Microsemi. Accessed 1 October 2022. [Online]. Available: https://www.microsemi.com/product-directory/clocks-frequency-references/3824-chip-scale-atomic-clock-csac 37

[71] TI LIDAR pulsed time of flight reference design. Texas Instruments. Accessed 03 October 2022. [Online]. Available: https://www.ti.com/lit/pdf/tiducm1b 37, 56

[72] GPS accuracy. NOAA: National Coordination Office for Space-Based Positioning, Navigation, and Timing (PNT). Accessed 10 August 2022. [Online]. Available: https://www.gps.gov/gps-accuracy-0 37, 56

[73] L. Zhou, G. Li, Z. Zheng, and X. Yang, "TOA Estimation with Cross Correlation-Based Music Algorithm for Wireless Location," in *2014 Fourth International Conference on Communication Systems and Network Technologies*. Bhopal, India: IEEE, Apr. 2014, pp. 862–865. 40

[74] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976. 40

[75] M. Azaria and D. Hertz, "Time delay estimation by generalized cross correlation methods," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 280–285, 1984. 40

[76] J. Benesty, J. Chen, and Y. Huang, "Time-delay estimation via linear interpolation and cross correlation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 509–519, 2004. 40

[77] A. Yeredor, "Analysis of the edge-effects in frequency-domain TDOA estimation," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3521–3524. 40

[78] X. Ning, X. Sha, and X. Wu, "A TOA Estimation Method for IR-UWB Ranging Systems," in *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*. Harbin, China: IEEE, Sep. 2010, pp. 684–687. 40

[79] J. Xu, Z. Wang, Q. Liu, and L. Ren, "A novel TOA estimation method for unknown signal based on intra-pulse correlation accumulation," in *2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. Hong Kong, China: IEEE, Aug. 2016, pp. 1–4. 40

[80] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005. 41

[81] R. Kaune, J. Hörst, and W. Koch, "Accuracy analysis for TDOA localization in sensor networks," in *14th International Conference on Information Fusion*, 2011, pp. 1–8. 41, 42, 43

[82] Y. Wang and K. C. Ho, "TDOA source localization in the presence of synchronization clock bias and sensor position errors," *IEEE Transactions on Signal Processing*, vol. 61, no. 18, pp. 4532–4544, 2013. 44

[83] "Recommendation ITU-R P. 1411-12, 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," Radiocommunication Sector of International Telecommunication Union, Standard, Aug. 2023. 45, 47, 48, 50, 51, 56, 64, 78, 98, 99, 112, 153

[84] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenburger, "A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2361–2391, 2019. 47, 50

[85] "Recommendation ITU-R P. 838-3, Specific attenuation model for rain for use in prediction methods," Radiocommunication Sector of International Telecommunication Union, Standard, 2005. 49

[86] "Recommendation ITU-r P. 676-10, attenuation by atmospheric gases," Radiocommunication Sector of International Telecommunication Union, Standard, 2013. 49, 99

[87] Communications toolbox version: 8 (r2023a). The MathWorks, Inc. August 01, 2023. [Online]. Available: https://www.mathworks.com 49, 50

[88] R. K. Crane, *Electromagnetic Wave Propagation through Rain.* Wiley, 1996. 49, 57

[89] (2020, Jun.) Study on enhancement for Unmanned Aerial Vehicles (UAVs). 49

[90] D. W. Matolak and R. Sun, "Air-ground channel characterization for unmanned aircraft systems: The near-urban environment," in *MILCOM 2015 - 2015 IEEE Military Communications Conference.* Tampa, FL, USA: IEEE, Oct. 2015, pp. 1656–1660. 50

[91] J. Holis and P. Pechac, "Elevation Dependent Shadowing Model for Mobile Communications via High Altitude Platforms in Built-Up Areas," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 4, pp. 1078–1084, Apr. 2008. 50

[92] Y. Chan and K. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994. 55, 90

[93] H.-M. Huang, E. Messina, and J. Albus, "Autonomy levels for unmanned systems (ALFUS) framework, volume II: Framework models version 1.0," National Institute of Standards and Technology, Standard, Dec. 2007. 71

[94] C. Peters, M. Watts, E. C. Larson, and M. A. Thornton, "Decentralized Q-learning for emitter localization with consensus policy selection," in *2025 Asilomar Conference on Signals, Systems, and Computing*, 2025. [Online]. Available: https://cmsworkshops.com/Asilomar2025/papers/accepted_papers.php 73, 81

[95] ——, "Cooperative reinforcement learning approaches for multi-UAV localization of emitters," in *[In DRAFT for submission]: 2026 Asilomar Conference on Signals, Systems, and Computing*, 2026. 73

[96] ——, "Geometry adaptive deep Q-network for UAV-based emitter localization in cluttered RF environments," in *2026 IEEE Aerospace Conference*, 2026. 73

[97] L. Xiang, M. Zhang, and A. Klein, "Robust Dynamic Trajectory Optimization for UAV-aided Localization of Ground Target," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 7484–7489. 75

[98] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Autonomous UAV Trajectory for Localizing Ground Objects: A Reinforcement Learning Approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1312–1324, Apr. 2021. 75

[99] J. Jiang and Z. Lu, "I2Q: A Fully Decentralized Q-Learning Algorithm," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20 469–20 481, Dec. 2022. 81

[100] D. Shi, J. Tong, Y. Liu, and W. Fan, "Knowledge Reuse of Multi-Agent Reinforcement Learning in Cooperative Tasks," *Entropy*, vol. 24, no. 4, p. 470, Mar. 2022. 81, 82

[101] Z. Xu, B. Zhang, D. Li, Z. Zhang, G. Zhou, H. Chen, and G. Fan, "Consensus Learning for Cooperative Multi-Agent Reinforcement Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, pp. 11 726–11 734, Jun. 2023. 81, 82

[102] Gymnasium Documentation. https://gymnasium.farama.org/index.html. 92, 98, 112

[103] Keysight. Unlocking Network Resilience with Keysight's Network Digital Twin Ecosystem. https://www.keysight.com/us/en/assets/3122-1404/technical-overviews/Network-Digital-Twin-Ecosystem.pdf. 94, 170

[104] D. L. Young, M. Bigham, M. Bradbury, E. Larson, and M. Thornton, "SMU-DDI Cyber Autonomy Range," in *2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE Computer Society, Oct. 2022, pp. 1–5. 95, 99

[105] PettingZoo Documentation. https://pettingzoo.farama.org/index.html. 112

[106] C. Peters, M. Watts, E. C. Larson, and M. A. Thornton, "[in draft] error ellipsoid guided geometry control for cooperative uas rf emitter localization in multipath environments," in *2026 IEEE Transactions on Aerospace and Electronic Systems*, 2026.

[107] M. N. E. I. Rouabhia, A. Bellabas, M. L. Bencheikh, and M. E. M. Abdelaziz, "On the phase interferometry direction finding: Performance comparison and FPGA implementations," in *2017 Seminar on Detection Systems Architectures and Technologies (DAT)*, 2017, pp. 1–5.

[108] Z. Xiaoning, "Analysis of military application of UAV swarm technology," in *2020 3rd International Conference on Unmanned Systems (ICUS)*. Harbin, China: IEEE, Nov. 2020, pp. 1200–1204.

[109] J. Zhou, Z. Li, and Y. Deng, "An improved information volume of mass function based on plausibility transformation method," *Expert Systems with Applications*, vol. 237, p. 121663, Mar. 2024.

[110] H. Ge, G. Chen, and G. Xu, "Multi-AUV Cooperative Target Hunting Based on Improved Potential Field in a Surface-Water Environment," *Applied Sciences*, vol. 8, no. 6, p. 973, Jun. 2018.

[111] K. Ho and Y. Chan, "Geolocation of a known altitude object from TDOA and FDOA measurements," vol. 33, no. 3, pp. 770–783. [Online]. Available: http://ieeexplore.ieee.org/document/599239/

[112] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, transferred to digital print on demand ed. PTR Prentice Hall.

[113] W. Gardner, "Simplification of MUSIC and ESPRIT by exploitation of cyclostationarity," vol. 76, no. 7, pp. 845–847. [Online]. Available: http://ieeexplore.ieee.org/document/7152/