Real-time Trajectory Optimization for Collaborative Self-Localization in Random Aircraft Formations

1st Jonathon S. Gipson
Autonomy and Navigation Technology Center
U.S. Air Force Institute of Technology
Wright-Patterson AFB, OH
jonathon.gipson@afit.edu

2nd Christine M. Schubert Kabban
Professor, Department of Mathematics
U.S. Air Force Institute of Technology
Wright-Patterson AFB, OH
christine.schubert@afit.edu

3rd Robert C. Leishman
Director, Autonomy and Navigation Technology Center
U.S. Air Force Institute of Technology
Wright-Patterson AFB, OH
robert.leishman@afit.edu

4th Juan D. Jurado
Director, Department of Education
U.S. Air Force Test Pilot School
Edwards AFB, CA
juan.jurado@us.af.mil

I. Abstract

In network localization scenarios, the geometry between the anchor nodes and the node of interest determines the minimum variance of the relative position error, commonly known as the Cramér-Rao Lower Bound. With modern computing resources, methods for real-time trajectory optimization are possible. The primary contribution of this paper is a distributed method for real-time self-localization trajectory optimization in a wireless sensor network of random formation aircraft. To perform this task, this paper introduces Self-Aligning Swarm, a distributed heuristic algorithm which generates a real-time trajectory for optimized self-localization based on local minimization of the trace of the Cramér-Rao Lower Bound. Using a 10K-trial Monte Carlo simulation, the proposed algorithm is shown to improve 2D position MSE of the optimized aircraft, reduce MSE variance, and demonstrate the MSE distribution approaches normality.

***Index Terms*—distributed collaborative localization, real-time, Cramér-Rao, cofactor matrix, pseudolites, design matrix, Fisher information, error ellipse, heuristic, particle swarm.**

II. Introduction

Persistent self-localization requires a resilient navigation solution, so it is critically important that we understand how to take advantage of performance benefits associated with collaborative localization and navigation [10]. Self-localization is the ability to accurately determine one's position which is dependent on minimizing absolute and relative positioning errors. Early localization trajectory optimization methods were limited to offline estimation for a known reference trajectory [4], [5]. In [5], Helferty employed minimization of the $trace(CRLB_{\hat{\theta}})$ for trajectory optimization for offboard localization of a single target. Several recent techniques have been presented to optimize self-localization with a static network topology [3, 6, 7, 1, 8]. The relevance of localization optimization is underscored by recent developments in practical methods for maximizing localization performance under network communication resource constraints [9]. This paper contributes a real-time heuristic method of trajectory generation for a single aircraft (referred to as the optimized node) to minimize relative positioning error in a wireless sensor network composed of a simulated formation of maneuvering aircraft.

With the proliferation of wireless sensor networks, there is an opportunity to collaboratively augment other aircraft in a degraded navigation environment. An anchor node can be defined as any node whose absolute location can be estimated accurately (e.g. GPS, highly reliable alternative navigation, etc.). In this scenario, simulated aircraft (anchor nodes) flying in a random dynamic formation are able to share their true position with nearby nodes over a wireless network. A single aircraft (optimized node) obtains noisy range measurements to each formation aircraft and combines this information with aircraft positions reported over a distributed wireless sensor network. Real-time trajectory guidance is generated for the optimized node based on real-time minimization of relative positioning error. Since relative position error is based on the geometry between the optimized node and nearby formation nodes, this paper introduces Self-Aligning Swarm (SAS), a real-time trajectory generation algorithm for self-localization.

In the following section, we review relevant background material which examines trade-offs associated with local and global cost minimization. We discuss the importance of algorithm scalability and conservation of resources given diminishing returns associated with capturing a global minimum which ultimately drove SAS towards local minimization. Next, we provide a derivation of the 2-D Cramér-Rao Lower Bound for a known variance which clarifies the importance of diverse angular coverage for reduction in relative positioning error. An introduction to the pseudolite cofactor matrix and comparison of SAS cost functions is performed and an anti-collision parameter is introduced. Next, we introduce the kinematics model for the simulated formation nodes and walk through an example of trajectory generation for a SAS-optimized node. Once the details of the simulation are explained, we provide the numerical results for a 10K trial Monte Carlo simulation. The results show the benefits of SAS optimization over a baseline formation trajectory. The paper concludes with our overall assessment of SAS, potential applications for the algorithm, and recommendations for future work.

III. Self-Aligning Swarm (SAS) Algorithm for Self-localization

A. Background

In a denied environment, vehicles equipped with all-source navigation capabilities (e.g. magnetic, visual, etc.) can maintain assured PNT solutions and serve as anchors for other vehicles in a surveillance network. Vehicles not equipped with these features are subject to integration drift and can benefit from relative navigation updates. In a distributed surveillance network, vehicles can perform relative measurement updates using radio ranging. In a time-synchronized network, localization accuracy for vehicles equipped only with radio ranging is primarily dependent on multilateration geometry to anchors. The primary contribution of this paper is a real-time control scheme to optimize self-localization in a distributed network.

SAS was originally conceived as a distributed heuristic global optimization algorithm for self-localization. The initial goal was to optimize the real-time trajectory for single node self-localization across a complex, dynamic cost surface. At each time step, a CPU-intensive global search of the cost surface was performed. The complexity of this computation has motivated previous approximation methods such as the alpha shape for efficient estimation of localization performance [8]. A vector from the optimized node's current position to the particle exhibiting the lowest cost was used to generate a guidance vector. This method required knowledge of the entire cost surface (which is likely unavailable) to guarantee a global minimum. The computation required to perform this search grows exponentially with search area. In a dynamically maneuvering formation of anchor nodes, the location of the global cost minimum can vary significantly, especially in the case of multiple competing minima. For example, there could be two or more minima fluctuating with approximately the same cost value in opposite directions from the optimized node. It would be counterproductive for a physical aircraft with momentum to expend resources to abruptly change direction to chase an ephemeral global minimum. There are diminishing returns associated with chasing a global minimum for self-localization.

For these reasons, a distributed heuristic local minimization algorithm was conceived with a small search area centered around the optimized node. To improve particle swarm efficiency, the size of the search area is no larger than the velocity and/or maneuvering capabilities of the optimized node. This means the search is constrained to positions the optimized node can actually achieve. This permits the designer to spawn a small, fixed number of particles for each heuristic search. By constraining the search area to the capabilities of the optimized platform, the local method does not guarantee a global minimum, nor does it guarantee the most direct trajectory to the minimum. However, the local method has a small, scalable computing burden and provides a stable, predictable trajectory to a dynamic self-localization optimum for reduction in relative position error and does not waste resources attempting to locate an unachievable solution.

B. 2-D CRLB for Range Measurements in AWGN

A node of interest receives range measurements corrupted with zero-mean independent bivariate Additive White Gaussian Noise (AWGN) from other nearby nodes in 2-D space. In the example, nearby nodes have unique positions separated by true distance D_n and each range observation R_n is corrupted by zero-mean AWGN with magnitude σ. Ranging error V_n is known and clock synchronization is assumed.

$$\underline{R}_n = \underline{D}_n + \underline{V}_n, \text{ where } \underline{V}_n \hookrightarrow \mathcal{N}(\underline{\mu}, \underline{\underline{S}}) \tag{1}$$

where $\underline{R}_n, \underline{D}_n$, and $\underline{V}_n$ are vectors of dimension N × 1 with $n = \{1, 2, ..., N\}$ nearby nodes. $\underline{\underline{S}} = \underline{\sigma}^2 \underline{\underline{I}}$ with $\sigma = 5$m with $\mu = [\underline{0}]$. At each time step, i, a node receives a range measurement R_n from each nearby node

$$\text{with nodes positioned at } \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, ... \begin{bmatrix} x_N \\ y_N \end{bmatrix}.$$

Next we derive the Cramer-Rao Lower Bound (CRLB) for the self-localization position estimate $\hat{\Theta}$, where $\hat{\Theta}_i = \hat{x}_0$ and $\hat{\Theta}_j = \hat{y}_0$ using range measurements corrupted by zero-mean independent AWGN. The Fisher Information Matrix (FIM), $\underline{\underline{I}}(\Theta)$, describes the observable information from the range measurements. The inverse of the FIM is the CRLB which provides a best-case estimate of the lower bound of the variance of the observable information (Eq. 2).

$$\underline{\underline{I}}(\hat{\Theta})^{-1} = CRLB_{\hat{\Theta}} \tag{2}$$

To obtain the FIM, we examine the expected value of the derivative of the pdf for each range measurement R_n with respect to our self-localization position estimate $\hat{\Theta}$ to understand how much information is available from R_n (Eq, 3).

$$\underline{\underline{I}}(\hat{\Theta}) = E\{\big[\frac{d}{d\hat{\Theta}} p(R|\hat{\Theta})\big]^2\}$$

$$= \int_{-\infty}^{\infty} \left(\frac{\frac{d}{d\hat{\Theta}}\big[p(R|\hat{\Theta})\big]}{p(R|\hat{\Theta})} \right)^2 p(R|\hat{\Theta}) dR \tag{3}$$

The pdf is

$$p(R|\hat{\Theta}) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(R-\hat{\Theta})^2}{2\sigma^2}}. \tag{4}$$

The partial derivative of Eq. 4 with respect to $\hat{\Theta}$ divided by the pdf from Eq. 4 is

$$\frac{\frac{d}{d\hat{\Theta}}p(R|\hat{\Theta})}{p(R|\hat{\Theta})} = \frac{\frac{d}{d\hat{\Theta}}\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(R-\hat{\Theta})^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(R-\hat{\Theta})^2}{2\sigma^2}}} = \frac{R-\hat{\Theta}}{\sigma^2}$$

when combined with Eq. 3 results in Fisher information shown in Eq. 5.

$$I(\hat{\Theta}) = -\int_{-\infty}^{\infty}\left(\frac{R-\hat{\Theta}}{\sigma^2}\right)^2\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(R-\hat{\Theta})^2}{2\sigma^2}}dR$$
$$= -\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}R^2e^{\frac{-R^2}{2}}dR = \frac{1}{\sigma^2} \qquad (5)$$

To obtain the Fisher Information matrix, we begin by taking the partial derivatives of each distance, d_n with respect to the node's location, x_0 and y_0.

$$I_{ij} = \frac{1}{\sigma^2}\sum_{n=0}^{N}\frac{\mathrm{d}d_n}{d\hat{\Theta}_i}\frac{\mathrm{d}d_n}{d\hat{\Theta}_j} \text{ where } \hat{\Theta}_i = x_0, \hat{\Theta}_j = y_0, \qquad (6)$$

and where $d_n = \sqrt{(x_0-x_n)^2+(y_0-y_n)^2)}$.

$$\frac{\mathrm{d}d_n}{dx_0} = \frac{d}{d\hat{\Theta}_i}\sqrt{(x_0-x_n)^2+(y_0-y_n^2)}$$
$$= \frac{1}{2}\left[(x_0-x_n)^2+(y_0-y_n^2)\right]^{(-\frac{1}{2})}[2(x_0-x_n)]$$
$$= \frac{x_0-x_n}{d_n} \qquad (7)$$

$$\frac{\mathrm{d}d_n}{dy_0} = \frac{d}{d\hat{\Theta}_j}\sqrt{(x_0-x_n)^2+(y_0-y_n^2)}$$
$$= \frac{1}{2}\left[(x_0-x_n)^2+(y_0-y_n^2)\right]^{(-\frac{1}{2})}[2(y_0-y_n)]$$
$$= \frac{y_0-y_n}{d_n} \qquad (8)$$

We can avoid taking the log-likelihood by making use of basic trigonometry. Figure 1 shows that the partial derivatives of the distance between nodes are equivalent to the following trigonometric identities:

$$\cos(\phi_n) = \frac{x_0-x_n}{d_n}$$

$$\sin(\phi_n) = \frac{y_0-y_n}{d_n}$$

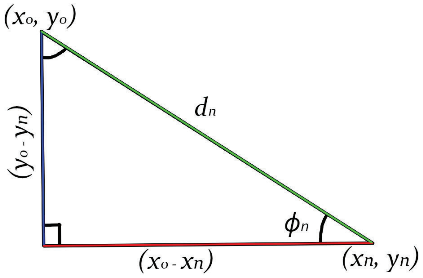

Fig. 1. Visualization of Partial Derivatives of d_n

If we manipulate Eq. 6, we can show it is proportional to the expected value of the partial derivatives of the distance measurements:

$$I_{ij} = \frac{N}{\sigma^2}\frac{1}{N}\sum_{n=0}^{N}\frac{\mathrm{d}d_n}{d\hat{\Theta}_i}\frac{\mathrm{d}d_n}{d\hat{\Theta}_j}. \qquad (9)$$
$$= \frac{N}{\sigma^2}E\left[\frac{\mathrm{d}d_n}{d\hat{\Theta}_i}\frac{\mathrm{d}d_n}{d\hat{\Theta}_j}\right] \qquad (10)$$

The resulting Fisher Information matrix is

$$\underline{\underline{I}} = \frac{N}{\sigma^2}\begin{bmatrix} E[cos^2(\phi_n)] & E[cos(\phi_n)sin(\phi_n)] \\ E[sin(\phi_n)cos(\phi_n)] & E[sin^2(\phi_n)]\end{bmatrix} \qquad (11)$$

accordingly, the Cramér-Rao Lower Bound is

$$Cov(\hat{\theta}) \geq \frac{\sigma^2}{N}\begin{bmatrix} E[cos^2(\phi_n)] & E[cos(\phi_n)sin(\phi_n)] \\ E[sin(\phi_n)cos(\phi_n)] & E[sin^2(\phi_n)]\end{bmatrix}^{-1} \qquad (12)$$

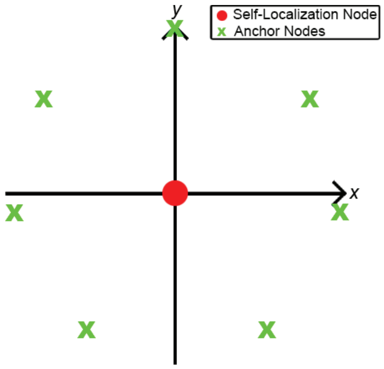

Fig. 2. Optimal 2-D Self-Localization Geometry for $N = 7$ Anchor Nodes

The diagonal terms of the $CRLB_{\hat{\theta}}$ matrix define the dimensions of a 2-D covariance ellipse for relative positioning error. For example, minimum relative self-localization error for $N = 7$ anchor nodes is achieved when anchor nodes are evenly spaced and centered on the the self-localization node (See Figure 2). In this example, the diagonal terms of the FIM are

$$E[cos^2(\phi_n)] = E[sin^2(\phi_n)] \approx 0.5$$

and the off-diagonal terms are

$$E[cos(\phi_n)sin(\phi_n)] = E[sin(\phi_n)cos(\phi_n)] \approx 0.$$

The resulting CRLB is

$$COV(\hat{\theta}) \geq \frac{\sigma^2}{7}\begin{bmatrix}0.5 & 0 \\ 0 & 0.5\end{bmatrix}^{-1} = \frac{2\sigma^2}{7}\begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix}.$$

Assuming the anchor nodes remain static, the diagonal terms of the FIM will decrease if the Self-Localization node moves from its current position. This results in a larger value for the $trace(CRLB_{\hat{\theta}})$. This forms the motivation for minimization of the $trace(CRLB_{\hat{\theta}})$ for generating an optimized self-localization trajectory.

C. Pseudolite Cofactor Matrix and SAS Cost Function

The definition of $CRLB_{\hat{\theta}}$ in Eq. 12 elucidates the importance of diverse angular coverage for reduction in relative positioning error variance. For a 2-D pseudolite-based self-localization problem, the design matrix, P for N nodes (Eq. 13), can be expressed in a similar manner as Eq. 7 and 8.

$$\underline{P} = \begin{bmatrix} \frac{x_1 - x_0}{R_1} & \frac{y_1 - y_0}{R_1} \\ \frac{x_2 - x_0}{R_2} & \frac{y_2 - y_0}{R_2} \\ \dots & \dots \\ \frac{x_N - x_0}{R_N} & \frac{y_N - y_0}{R_N} \end{bmatrix} \text{ where } R \text{ is defined by Eq. 1.} \quad (13)$$

The pseudolite cofactor matrix $\underline{\underline{\Sigma}}_c$ for this example can be computed using a least squares method shown in Eq. 14.

$$\underline{\underline{\Sigma}}_c = (\underline{P}^T \underline{P})^{-1} = \frac{\sigma^2}{N} \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \quad (14)$$

Minimization of the $trace(\underline{\underline{\Sigma}}_c)$ (Eq. 15) is equivalent to minimizing the $trace(CRLB_{\hat{\theta}})$.

$$trace(\underline{\underline{\Sigma}}_c) = \frac{\sigma^2}{N}(\sigma_x^2 + \sigma_y^2) \quad (15)$$

It is assumed that the initial starting position $[x_0, y_0]^T$ of the optimized node is known and is used to initialize an iterative nonlinear self-localization position estimator which relies on MATLAB's $fitlnm$ nonlinear model fit function to perform a self-localization estimate. For all $i > 1$, the t_{i-1} position estimate is used to initialize the self-localization estimator. The cost function employed in the SAS algorithm is composed of trajectory optimization with an anti-collision function (Eq. 16).

$$SAS_{Cost} = trace(\underline{\underline{\Sigma}}_c) + A^{\sum_1^N R_n^{-1}} \quad (16)$$

where A is a desired anti-collision avoidance factor, N is the total number of nearby formation nodes, and R_n is a noisy range measurement from node n.

The avoidance term in the cost function is approximately 1 until the optimized node is in proximity of a formation node. A determines the gradient of this anti-collision function. In this simulation, A was chosen to mitigate incursions inside 100 meters of other nodes. SAS minimizes Eq. 16 at each time step to determine an optimal trajectory for the optimized node.

D. Simulation

The general shape of a surface defined by the $trace(CRLB_{\hat{\theta}})$ for a formation of networked nodes is convex, with the global minimum typically existing somewhere near the center of the formation network. Since self-localization performance is based primarily on angular coverage from anchor nodes, there can be significant benefits to trajectory optimization with suboptimal formation topology.

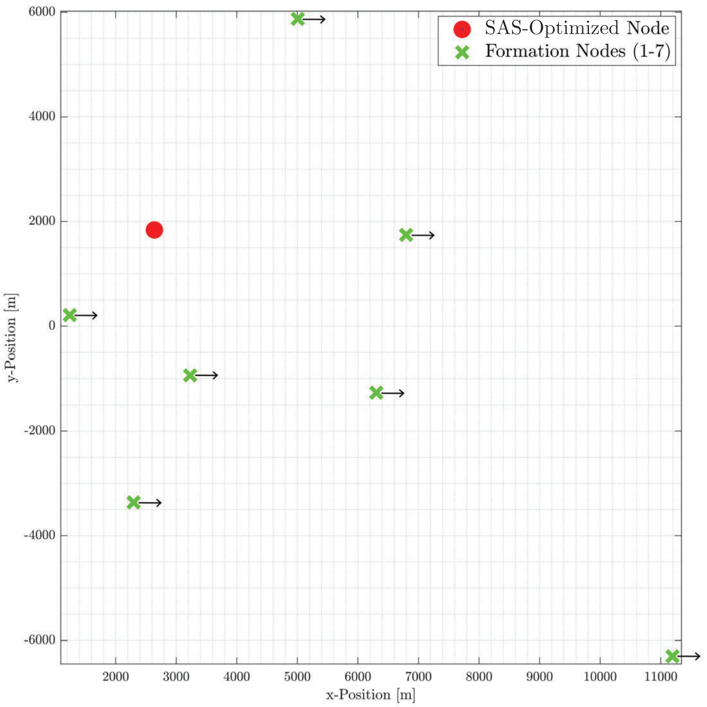

Fig. 3. SAS Initialization with Random Node Locations and Initial Trajectory, t_0 = 0.0 sec

In the scenario illustrated in Figure 3, there is a single optimized node and $N = 7$ anchor nodes in a random, dynamic formation. Figure 3 shows the results of this initialization with a single optimized node and N formation nodes visible on a 2-D plane at t_0 = 0.0 sec. N nodes provide their absolute positions to a optimized node over a simulated digital wireless network. The single optimized node is able to collect noisy range measurements to each formation node with $\sigma_{range} = 5$ m (detailed explanation presented in Section III-C). $10,000$ Monte Carlo trials were performed lasting a duration of $T_{trial} = 150$ sec each. At the beginning of each run, all of the nodes are initialized at random uniform 2-D locations on a square 400 km^2 grid.

To simulate a randomly maneuvering formation of fixed-wing aircraft travelling in the same direction, the formation nodes are initialized with a stable 2-D velocity V_0 = 100 m/s initially in the $+X$ direction with a Gaussian random angular perturbation defined by Eq. 17.

$$\alpha_i = 0.8\alpha_{i-1} + 0.2\phi \quad (17)$$

where time step size t_i is 1 second, α is the current 2-D velocity angle and ϕ is a normally distributed random angle between [0, 2π] radians generated for each formation node every time step t_i. The effect of Eq. 17 is visible in Figure 4.

In the same manner as the formation nodes, the optimized node is initialized at a random 2-D location on the 400 km^2 grid. Noisy range measurements are performed between the optimized node and each formation node at each time step. MATLAB's 'fit nonlinear model' function $fitnlm$ is used to

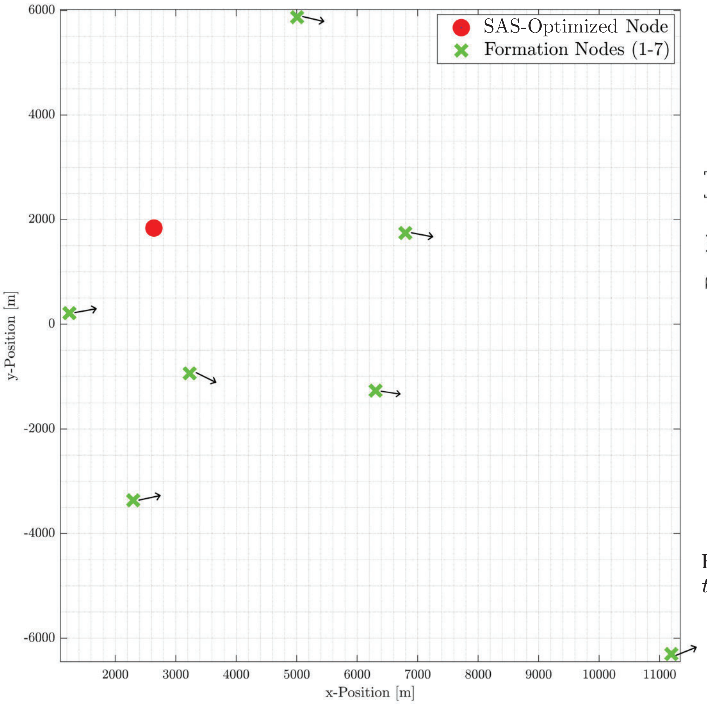

Fig. 4. SAS Formation Node Trajectories with Angular Perturbations, t = 1.0 sec

obtain an iterative least-squares position estimate for the optimized node's absolute 2-D position using the noise-corrupted range measurements and true absolute positions of the nearby nodes. Detailed explanations of the range measurements and absolute position estimate methods are presented in Section III-C.

The optimized node is infinitely maneuverable with a maximum velocity of 250 m/s. The trajectory of the optimized node is based on a particle swarm search of size 10 particles inside a local circular area of radius 250 m centered on the location of the optimized node. The optimized node trajectory is established by following the local gradient of the cost function at each time step. Once a local minimum is captured inside the search area, the optimized node follows the surface gradient to the minimum and maneuvers as necessary to maintain the local minimum. The optimized node will continue to follow the surface gradient towards a local minimum, resulting in a real-time optimized self-localization trajectory.

Figure 5 shows the contour of the particle swarm cost function given by Eq. 16 for the configuration shown in Figure 3 from the perspective of the optimized node. The peaks resulting from the anti-collision term visible in Figure 5 indicate where the formation nodes are located. These are the same nodes visible in Figure 3. Figure 6 shows the contour surface of $trace(CRLB_{\hat{\theta}})$ without the anti-collision term. It is apparent that the locations of the minima of the $trace(CRLB_{\hat{\theta}})$ are unaffected by the anti-collision term which ensures that minimization of the cost function is equivalent to minimization of the $trace(\Sigma_c)$ and $trace(CRLB_{\hat{\theta}})$, from Equations 12 and 14.

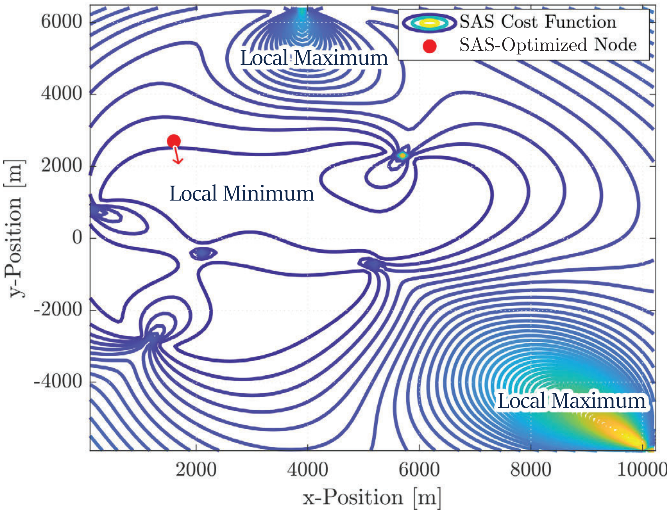

Fig. 5. Initial Node Trajectory Towards Local Minimum of Cost Function, t_0 = 0.0 sec

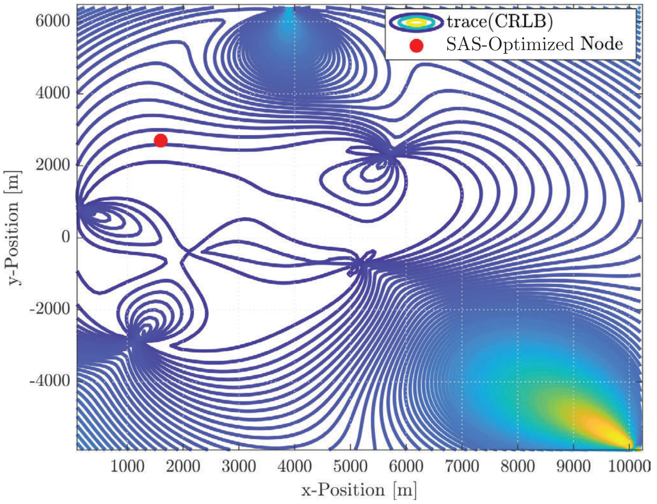

Fig. 6. Contour of $trace(CRLB_{\hat{\theta}})$ for Formation Node Network, t_0 = 0.0 sec

The optimized node's current trajectory follows the gradient of the cost surface in Figure 5 towards the minimum value of the surface and is identified by a vector arrow. Section III-B presents a derivation for the lower bound of the 2-D error variance. A proof for the convergence of the bounds of total relative localization error to the minimization of the $trace(CRLB)$ is presented in [2].

As the simulation progresses, the heuristic search is repeated at each time step to establish a new local minimum for self-localization trajectory optimization. Once the optimized node arrives at the local minimum, its velocity will stabilize with the surrounding formation and the optimized node will maneuver to maintain optimal self-localization geometry within the local search area. At each time step, position residuals are recorded

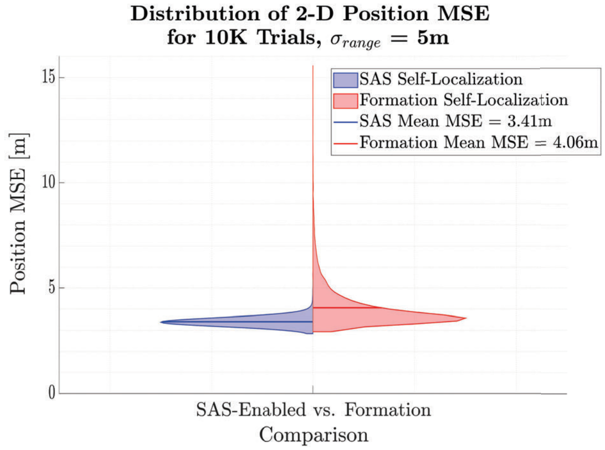

Fig. 7. Distribution of Mean Position MSE for 10K Trials, T_{trial} = 150 sec, σ_{range} = 5 m

for the optimized node and position MSE is recorded for the duration of the run. The results for a 10K-trial Monte Carlo simulation are discussed in Section IV.

IV. Numerical Results

TABLE I
Mean Position MSE for 10K Trials, T_{trial} = 150 sec, σ_{range} = 5 m

Type/Behavior	$\hat{\mu}_{MSE}$	$\hat{\sigma}_{MSE}$
Formation Nodes	**4.06 m**	**0.981 m**
SAS-Optimized Node	**3.41 m**	**0.184 m**

The estimated mean position MSE for the SAS-optimized node and each of the 7 non-optimized formation nodes are shown for 10K trials in Table I. To develop a baseline for comparison, localization estimates for the formation nodes were performed using the same noisy ranging technique described in Section III-C. The formation nodes perform a 'default' behavior by generally maintaining the same relative formation position for the duration of the simulation. This comparison quantifies the difference between mean position MSE observed from maintaining formation and from following the SAS-optimized trajectory. Figure 8 shows the mean position MSE distributions for each of the 7 non-SAS formation nodes have nearly the same distribution for 10,000 trials. A t-test for significant differences in mean variances indicates the variance of position MSE for the SAS-optimized node is smaller than that of formation nodes with a p-value < 0.0001. The reduction in variance is shown by a violin-plot comparison of the position MSE distributions for 10K Monte Carlo trials shown in Figure 7.

The overall reduction in mean position MSE and variance can be explained with an example. If a node is randomly spawned at an unfavorable formation position, it is expected to have a large CRLB (and position MSE) compared to a node spawned in a more favorable position within the formation. In both situations, a SAS-optimized node will follow a trajectory to minimize the $trace(CRLB_{\hat{\theta}})$ which results in lower position MSE. In contrast, a formation node simply flies along with the formation with kinematics governed by Eq. 17 and does not optimize its position MSE. This difference in behavior results in the reduced mean and variance for the SAS-optimized mean position MSE distribution.

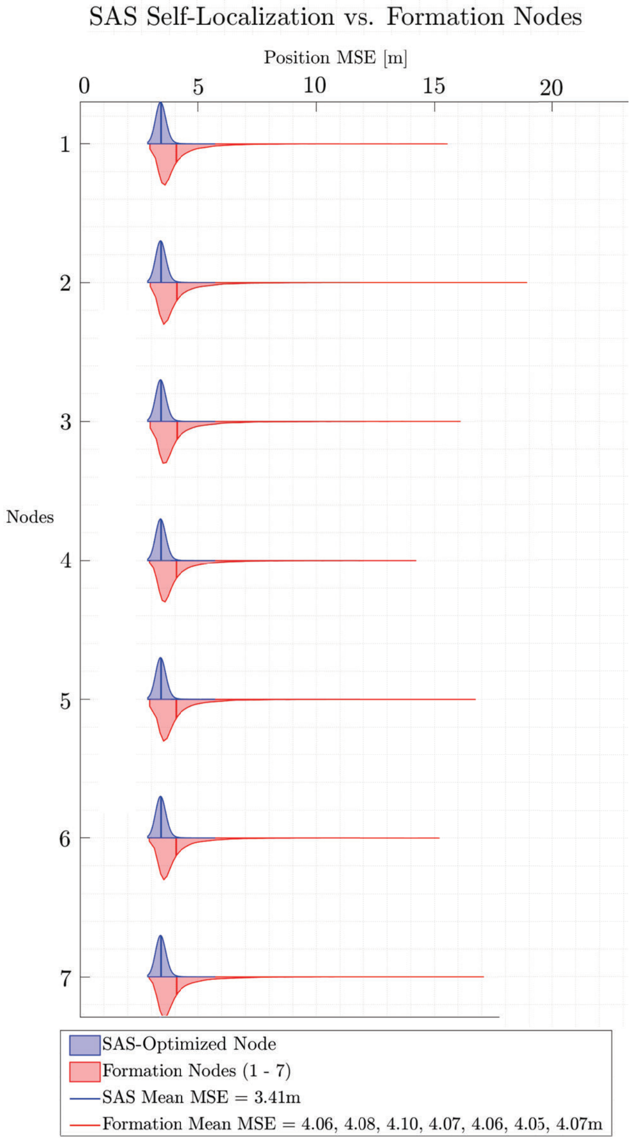

Fig. 8. Comparison of Mean Position MSE Distributions for 10K Trials, T_{trial} = 150 sec, σ_{range} = 5 m

A comparison of Normal-Gaussian distribution models fit to the SAS-optimized and Formation Node 1 distributions is visible in Figure 9. The mean position MSE for the SAS-optimized node approaches a Normal-Gaussian distribution which is a property not shared by the formation trajectory nodes. Normally-distributed position error is desirable for subsequent use with Bayesian estimators which often make this simplifying assumption regardless of the actual error distribution.

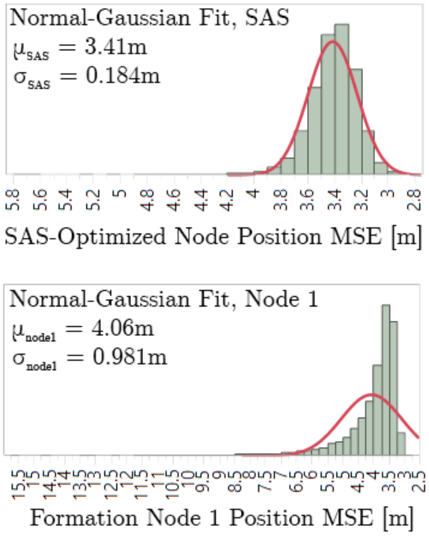

Fig. 9. Comparison of Normal-Gaussian Models Fit to SAS-Optimized and Formation Node 1 Mean Position MSE Distributions for 10K trials, T_{trial} = 150 sec, σ_{range} = 5 m

V. Conclusion

This paper presents a novel real-time heuristic method of trajectory optimization for a single SAS-optimized aircraft to minimize self-localization error in proximity to a formation of randomly maneuvering aircraft. SAS is a compact, scalable heuristic method to optimize collaborative self-localization performance for a single aircraft in a random formation. SAS is shown to be a useful method for dynamic position error minimization. A 10K-trial Monte Carlo simulation demonstrated SAS-optimized flight path trajectory exhibits reduced mean position MSE and reduced variance for position MSE. The simulation also shows the distribution of position MSE for the SAS-optimized aircraft approaches normality whereas any other random formation aircraft exhibited a heavily-skewed error distribution. A potential practical application for SAS is implementation on a potentially GPS-denied aircraft whose goal is to minimize Inertial Navigation System drift in a collaborative network composed of all-source contributors. Recommended future work includes investigation of multiple cooperative anchor nodes for optimized localization performance of off-board aircraft on a mission-dictated trajectory in a dynamic environment.

References

[1] Marko Angjelichinoski et al. "Cramér – Rao Lower Bounds of RSS-Based Localization With Anchor Position Uncertainty". In: *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2807–2834. DOI: 10.1109/TIT.2015.2409270.

[2] Joshua N. Ash and Randolph L. Moses. "On the relative and absolute positioning errors in self-localization systems". In: *IEEE Transactions on Signal Processing* 56.11 (2008), pp. 5668–5679. ISSN: 1053587X. DOI: 10.1109/TSP.2008.927072.

[3] Joshua N Ash et al. "Locating the nodes: cooperative localization in wireless sensor networks". In: *IEEE Signal Processing Magazine* 22.July (2005), pp. 54–69. DOI: 10.1109/MSP.2005.1458287.

[4] D. Cappel. "Optimal Observer Maneuver for Bearings-Only Tracking". In: 34.3 (1998).

[5] James P Helferty, David R Mudgett, and John E Dzielski. "Trajectory Optimization for Minimum Range Error in Bearings-only Source Localization ". In: (1993).

[6] Tao Jia et al. "Collaborative Position Location". In: *IEEE Transactions on Wireless Communications* 9.1 (2010), pp. 374–383. DOI: 10.1109/TWC.2010.01.090869.

[7] Wei Li, Zhen Yang, and Haifeng Hu. "A new variant of sum-product algorithm for sensor self-localization in wireless networks". In: *2013 IEEE/CIC International Conference on Communications in China, ICCC 2013* Iccc (2013), pp. 628–632. DOI: 10.1109/ICCChina.2013.6671189.

[8] Richard K Martin. "Based Positioning Performance". In: *IEEE Signal Processing Letters* 18.12 (2011), pp. 741–744. DOI: 10.1109/LSP.2011.2173193.

[9] Ruixin Niu, Aditya Vempaty, and Pramod K. Varshney. "Received-Signal-Strength-Based Localization in Wireless Sensor Networks". In: *Proceedings of the IEEE* 106.7 (2018), pp. 1166–1182. ISSN: 00189219. DOI: 10.1109/JPROC.2018.2828858.

[10] Moe Z. Win et al. "Network localization and navigation via cooperation". In: *IEEE Communications Magazine* 49.5 (2011), pp. 56–62. ISSN: 01636804. DOI: 10.1109/MCOM.2011.5762798.