“Sensing Vulnerabilities:” Novel Applications of Sensor Network Localization in Healthcare and Disaster Management

Hannah Ashai
Department of Electrical Engineering
Stanford University
Stanford, CA
hyashai@stanford.edu

Amod Hegde
Department of Management Science and Engineering
Stanford University
Stanford, CA
amod96@stanford.edu

***Abstract*—Sensor network localization (SNL) is a complex problem involving the estimation of sensor positions based on distance measurements. This project investigates the practical applications of SNL in healthcare and disaster management domains, focusing on evaluating the performance of following localization algorithms: LS, SOCP, SDP, SDP with Projection, and ADMM. We also consider these algorithms under different noise levels and radio ranges. Our simulations consider anchor-sensor configurations in adult care, low-power healthcare, and disaster management scenarios. We conclude that anchor placement significantly impacts localization accuracy and suggest an SDP formulation for most application scenarios.**

I. Introduction

A sensor network is a collection of sensors with unknown locations and anchors with known locations. In such networks, each sensor can detect others within radio range and measure the distances to nearby sensors. The objective of this problem is to determine the precise locations of all sensors based on these known distance measurements. These wireless sensor networks are increasingly used for disaster management and home monitoring, capturing data that can localise individuals in physical space. Typically, these networks consist of numerous densely deployed sensor nodes that need to gather local data and communicate with other nodes.

Mathematically, the problem can be described as follows. Let there be n_x distinct sensor points in R^d, with their locations to be determined, and n_a fixed points known as anchor points, denoted as $a_1, a_2, \cdots, a_{n_a}$. The Euclidean distance d_{ij} between the i^{th} and j^{th} sensor points is known if $(i,j) \in N_x$, and the distance $\hat{d}_{ik}$ between the i^{th} and k^{th} anchor point is known if $(i,k) \in N_a$. $N_x = \{(i,j) : ||x_i - x_j||_2 = d_{ij} \leq r_d\}$ and $N_a = \{(i,k) : ||x_i - a_k||_2 = \hat{d}_{ik} \leq r_d\}$, where r_d is a fixed parameter representing the radio range.

The sensor network localization problem aims to find a vector $x_i \in R^d$ for each sensor $i = 1, 2, \cdots, n$, such that for all $||x_i - a_k|| = \hat{d}_{ik} \quad \forall (i,k) \in N_a$.

The sensor network localization problem poses significant computational difficulties and has attracted extensive research efforts to develop efficient algorithms for accurate sensor localization. In the following sections, we investigate several algorithms: including Least Squares (LS); Second Order Cone Programming (SOCP); Semidefinite Programming (SDP); A refined version of SDP where we pass the SDP solution into LS (SDP Refined); SDP with Projection (Projection); and Alternating Direction Method of Multipliers (ADMM). We evaluate their performance in terms of the root mean square distance (RMSD) from the true placement of the sensors under different noise levels and radio ranges.

To investigate the effect of anchor placement, we conduct simulations considering three different scenarios. These scenarios provide insights into various applications and their corresponding anchor-sensor configurations.

1) Adult Care Application:
   In this scenario, we simulate the placement of anchors along the perimeter of the area of interest, while sensors are randomly distributed within the boundary. This configuration mimics the adult care scenario, where anchors represent fixed reference points (such as walls or furniture) and sensors represent individuals being monitored. We analyze the performance of localization algorithms in this setup (see figure 1) and assess the accuracy of sensor positioning.

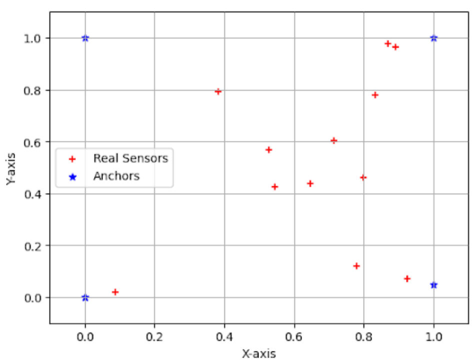

Figure 1. Visualization of a sample grid for the Adult Care scenario.

2) Low Power Healthcare Application:
   In this scenario, we explore two sub-scenarios. First, we place anchors in the interior of the area of interest, while sensors are positioned beyond the convex hull formed by the anchors. This setup mimics a low power healthcare application where sensors are spread throughout the body, and anchors act as central reference points outside the body. Second, we consider a scenario where only sensors are present without any anchors. This configuration represents a minimalistic approach to localization, where sensors rely solely on their spatial relationships with each other. We evaluate the performance of localization algorithms in both cases (see Figure 2) to assess the feasibility and accuracy of sensor positioning.
3) Disaster Management/Wildlife tracking Scenario:
   In this scenario, we use a quasirandom space-filling set of anchors (using the Halton Sequence) throughout the area of interest, mimicking the unpredictable and dynamic nature of disaster management situations (see Figure 3). The sensors are distributed randomly. We analyze the impact of this anchor-sensor configuration on localization accuracy and assess the robustness of the algorithms under varying anchor densities and placements.

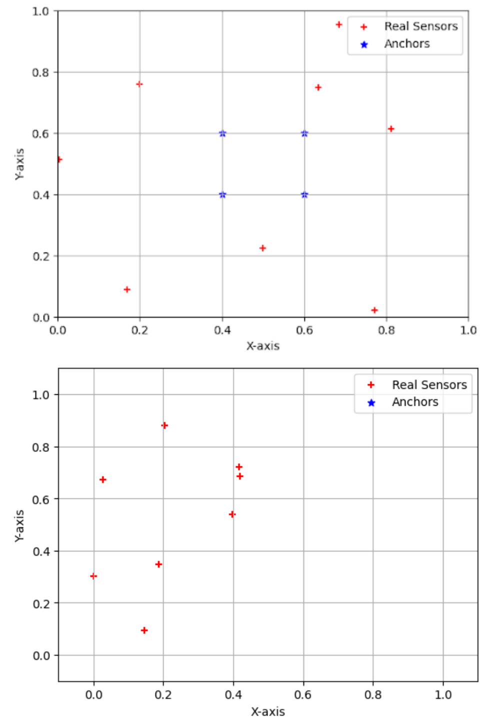

Figure 2. Visualization of sample grid for the Low Power Healthcare Scenario.

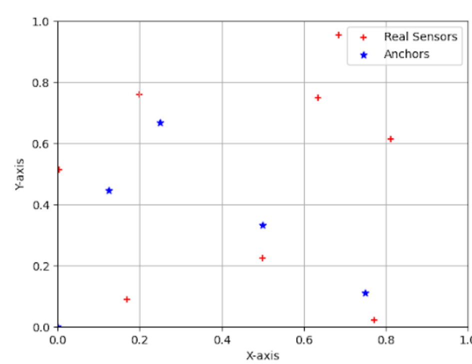

Figure 3. Visualization of a sample grid for the Disaster Management/Wildlife tracking scenario.

These simulation studies provide valuable information about the impact of anchor placement. We also study the influence of radius and noise on localization accuracy in different application scenarios. The findings contribute to the understanding of sensor network localization and can guide the selection and deployment of localization algorithms in practical contexts.

II. Problem Formulation

The sensor network localization problem is: given the locations $a_k \in R^d$ of all n_a anchors, distance measurements $d_{ij} \forall (i,j) \in N_x$ and $\hat{d}_{kj} \forall (k,j) \in N_a$, find the coordinates $x_i \in R^d$ of all the n_x sensors, such that:

$$||x_i - x_j||_2^2 = d_{ij}^2 \quad \forall (i,j) \in N_x, \quad i < j$$
$$||x_i - a_k||_2^2 = \hat{d}_{ik}^2 \quad \forall (i,k) \in N_a$$

This is an NP-hard problem to solve. In the subsequent subsections, we look at different relaxations and formulations to solve the SNL problem.

A. Least Squares Method (LS)

At first glance, it would be relatively simple to model this as an unconstrained nonlinear problem, which we can solve using least squares regression. In short, we are aiming to do the following:

$$\min_{x_i} \sum_{(i,j) \in N_x} (||x_i - x_j||_2^2 - d_{ij}^2)^2 + \sum_{(i,k) \in N_a} (||x_i - a_k||_2^2 - \hat{d}_{kj}^2)^2$$

Unfortunately, the objective is non-convex, meaning it can have multiple local minima. This characteristic makes the optimization process sensitive to the initial conditions and prone to getting trapped in local minimas, resulting in overall poor performance.

To address these challenges and make the problem more tractable, a relaxation method may be necessary. One approach is to convert the strict equality constraints in our model to inequality constraints. This allows us to formulate the problem as a Second Order Cone Program (SOCP), which has a more structured and solvable form.

B. Second Order Cone Program (SOCP)

In the SOCP formulation, the objective is to minimize a convex function subject to a set of linear and second-order cone constraints. The relaxation we have introduced from strict equality to inequality constraints has turned our problem to a Second Order Cone Program of the following structure:

$$\min_{x_i} 0 \cdot x_i$$

such that:

$$||x_i - x_j||_2^2 \le d_{ij}^2 \quad \forall (i,j) \in N_x, \quad i < j$$
$$||x_i - a_k||_2^2 \le \hat{d}_{ik}^2 \quad \forall (i,k) \in N_a$$

The second-order cone method offers efficient polynomial complexity for solving Euclidean metric problems. However, its effectiveness is contingent on the placement of anchor nodes. When anchor nodes are positioned on the outer boundary of the network, the method yields accurate results, with estimated positions falling within the convex hull of the anchor nodes. In contrast, if anchor nodes are placed within the interior of the network, the estimated positions of unknown nodes tend to converge towards the interior, leading to highly inaccurate results, as shown in Figure 10. Therefore, careful consideration of anchor node placement is crucial to ensure accurate position estimation using the second-order cone method. [1] Therefore, alternative approaches need to be explored to address the limitations and improve the performance of the localization problem.

C. Semi-Definite Program (SDP)

Observe that our SNL problem can be also be framed as follows:

$$(\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z = d_{ij}^2 \quad \forall (i,j) \in N_x$$
$$(a_k; -e_j)(a_k; -e_j)^T \bullet Z = d_{ij}^2 \quad \forall (k,j) \in N_a$$
$$Z = \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \tag{1}$$
$$Y = X^T X \tag{2}$$

We note a few notations for clarity. Matrix $X = [x_1, x_2, \ldots, x_{n_x}] \in R^{d \times N_x}$ is matrix with sensor coordinates. $P \bullet Q$ refers to the action of Trace($P^T Q$).

$e_j \in R^{n_x}$ is the vector with all zeros except -1^{th} the j^{th} position and $\mathbf{0} \in R^d$ is a vectors of zeros. The vector $(v; w)$ is a vector where v is stacked on top of w.

If we relax (2) to $Y \succcurlyeq X^T X$, then it is equivalent to setting $Z \succcurlyeq 0$ as shown in [4]. This gives us our Semi-Definite Program:

$$\min \mathbf{0} \bullet Z$$

$$\begin{aligned} Z_{1:d,1:d} &= I \\ (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z &= d_{ij}^2 \quad \forall (i,j) \in N_x \\ (a_k; -e_j)(a_k; -e_j)^T \bullet Z &= d_{ij}^2 \quad \forall (k,j) \in N_a \\ Z &\succcurlyeq 0 \end{aligned}$$

Theorem 1:

If the SNL problem is *uniquely d-realizable*, then the solution matrix to the SDP

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$$

where $\bar{Y} = \bar{X^T}\bar{X}$.

The definition of *uniquely d-realizable* and the proof of the theorem can be found in [3] which we skip for the sake of brevity. Essentially, the theorem says that, if the SNL solution has a unique solution, then SDP retrieves this exact solution, with $\bar{Y} = \bar{X^T}\bar{X}$.

D. SDP for SNL with noisy data

In practical problems, there is often noise in the distance information. To deal with possible noise, the SDP relaxation approach (II-C) can be modified to minimize the L1 norm of the errors as follows:

$$\min_{Z,\delta,\delta',\hat{\delta},\hat{\delta}'} \sum_{(i,j)\in N_x} (\delta_{ij} + \delta'_{ij}) + \sum_{(k,j)\in N_a} (\hat{\delta}_{kj} + \hat{\delta}'_{kj})$$

s.t. $Z_{1:d,1:d} = I$
$(\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z + \delta_{ij} - \delta'_{ij} = d_{ij}^2$
$(a_k; -e_j)(a_k; -e_j)^T \bullet Z + \hat{\delta}_{kj} - \hat{\delta}'_{kj} = d_{ij}^2$
$Z \succcurlyeq 0$

The SDP solution

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$$

may often not be rank d so that $\bar{X} \in R^{d\times n_x}$ cannot be the best possible localization of the sensors. Next, we talk about a way to further refine the solution from SDP.

E. SDP Refined

In the SDP refined approach, we leverage the solution obtained from the SDP algorithm as the initial solution for a subsequent least squares optimization step. This strategy aims to enhance the accuracy and convergence of the optimization process. By utilizing the SDP solution as a starting point, we can refine and improve the estimated positions of the unknown nodes by incorporating additional constraints and information from the least squares method.

Moreover, this refinement step allows us to take advantage of suboptimal solutions obtained from the SDP algorithm, even if they have been computed for a shorter duration. By building upon these initial solutions, we can further enhance the overall performance and quality of the localization results.

As the SDP problem grows in size, the associated matrix cone dimension and the number of unknown variables increase quadratically. This quadratic growth poses a challenge for solving large-scale SDP problems, with its runtime complexity being arbitrarily polynomial. [2]

Therefore, in the interest of improving the runtime, we discuss a faster first-order method to solve the SDP problem.

F. Steepest Descent Projecting method for SDP

The Steepest Descent Projection method recasts our SDP problem (II-C) to a following least squares problem:

$$\begin{aligned} min_Z \quad & f(Z) := \frac{1}{2}||\mathcal{A}Z - b||^2 \\ s.t. \quad & Z \succcurlyeq 0 \end{aligned}$$

where $\mathcal{A}Z = \begin{pmatrix} \mathcal{A}_1 \bullet Z \\ \vdots \\ \mathcal{A}_m \bullet Z \end{pmatrix}$. Each of the $\mathcal{A}_i$'s are calculated such that $\mathcal{A}_i \bullet Z$ maps to the one of the $m = N_x + N_a + d^2$ linear constraints in our original SDP problem.

We first initialiZe our solution Z with a random PSD matrix Z_0 The gradient $\nabla_Z f(Z) = \mathcal{A}^T(\mathcal{A}Z - b)$. In each iteration, after the gradient step, the

Algorithm 1 Steepest Descent Projection

$\mathbf{Z}^1 \leftarrow \mathbf{Z}_0$.
for $k \in \{1, \dots n\}$ and $|f(Z^k) - f(Z^{k-1})| > \epsilon$ **do**
  $\mathbf{\hat{Z}^{k+1}} \leftarrow \mathbf{Z^k} - \eta \nabla_{\mathbf{Z}} f(\mathbf{Z^k})$.
  $\mathbf{\hat{Z}^{k+1}} = \mathbf{V \Lambda V^T}$
  $\mathbf{Z^{k+1}} = \mathbf{V} \max\{\mathbf{0}, \mathbf{\Lambda}\}\mathbf{V^T}$
end for
Return $\mathbf{Z^{n+1}}$

resulting matrix $\hat{Z}^{k+1}$ may not be a PSD as required. Therefore, we project $\hat{Z}^{k+1}$ onto the cone of PSD matrices by computing its eigenvalue decomposition and retaining only its positive eigenvalues.

But, in practice, computing the eigenvalue decomposition in each iteration may prove to be expensive. We can reduce this overhead by just working with the best p rank approximation of Z_{k+1}. This boils down to, as per Eckart–Young–Mirsky theorem, computing the first $p < (d + n_x)$ largest eigenvalues and eigenvectors of $\hat{Z}^k$ and setting $Z^{k+1} = V \max\{0, \Lambda\} V^T \approx V_p \max\{0, \Lambda_p\} V_p^T$, where Λ_p is the diagonal matrix with the largest p eigenvalues and V_p contains the corresponding p eigenvectors.

In the Figure 4, we can see the convergence of this method for a scenario with 8 sensors and 4 anchors. The solution converges pretty quickly and manages to retrieve the right answer.

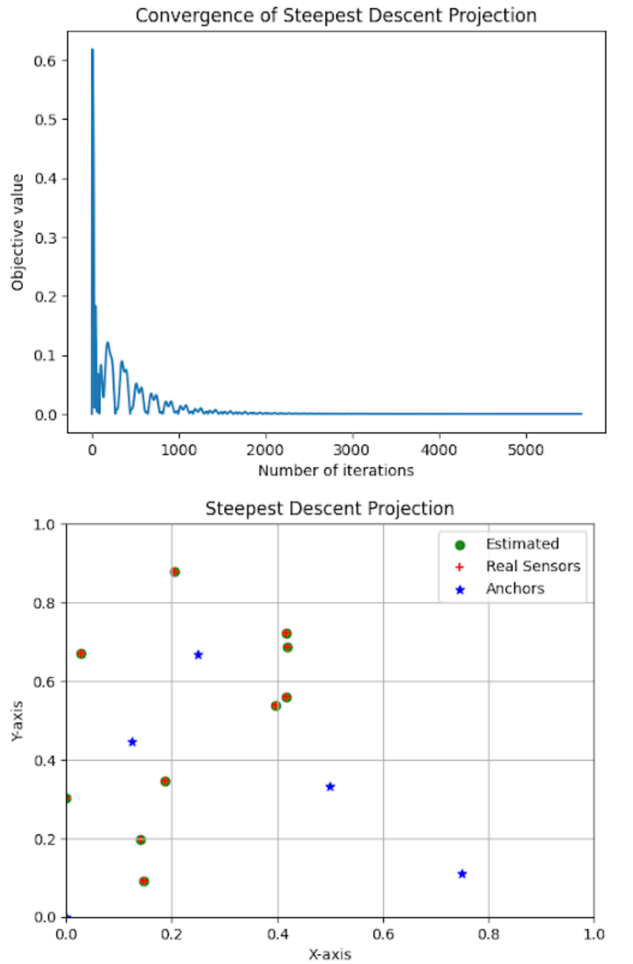

Figure 4. Convergence of the Steepest Descent Projection Method for a scenario with 8 sensors and 4 anchors

G. Alternating Direction Method of Multipliers (ADMM)

Another speed-up may be using ADMM approach. The Alternating Direction Method of Multipliers is an optimization algorithm used to solve optimization problems with separable objectives and convex constraints. The algorithm splits the original problem into smaller subproblems, each involving a subset of the variables and constraints. By iteratively updating the variables while enforcing consistency between them, ADMM can converge to an optimal solution.

In our case, we can reformulate the nonlinear least squares model as follows:

$$\begin{aligned} \min \sum_{(i,j)\in N_x} & [(x_i - x_j)^T (y_i - y_j) - d_{ij}^2] \\ & + \sum_{(k,j)\in N_a} [(a_k - x_j)^T (a_k - y_j) - d_{kj}^2] \\ & s.t. \quad x_j - y_j = 0, \forall j \in \{1, 2, \dots n_x\} \end{aligned} \tag{3}$$

For fixed y, the objective function is a quadratic function of x; and for fixed x, the objective function is a quadratic function of y. The above optimization problem is solved using the augmented Lagrangian method.

The augmented Lagrangian for 3 would be

$$L(x,y,z) = \sum_{(i,j)\in N_x} [(x_i - x_j)^T(y_i - y_j) - d_{ij}^2]$$
$$+ \sum_{(k,j)\in N_a} [(a_k - x_j)^T(a_k - y_j) - d_{kj}^2]$$
$$- \sum_i z_i(x_i - y_i) + \frac{\beta}{2}\sum_i ||x_i - y_i||_2^2$$

We minimize $L(x,y,z)$ iteratively as follows:

$$x_{k+1} = \arg\min_x L(x, y_k, z_k) \tag{4}$$

$$y_{k+1} = \arg\min_y L(x_k, y, z_k) \tag{5}$$

$$z_{k+1} = z_k - \beta(x_{k+1} - y_{k+1}) \tag{6}$$

We start with some random x_0, y_0, z_0 and repeat until convergence of the objective value. In the end, we use $(x+y)/2$ as the localized coordinates. Since, steps 4 and 5 are simple quadratic minimization problems, the ADMM approach to solve the SNL problem is simpler and faster to solve.

Therefore, ADMM would be a good choice for large scale scenarios where solving a pair of quadratic minimization probelms iteratively is more tractable than solving a complex optimization algorithm like SDP.

III. Results

A. Performance in Application Scenarios

This section presents an analysis of the performance of different optimization algorithms in three application scenarios: random sensor-anchor placement for disaster management, home healthcare monitoring with anchors on the perimeter, and low power healthcare sensing with sensors placed beyond the convex hull. The evaluation is based on the Root Mean Square Deviation (RMSD) error (7) observed in 25 iterations for each algorithm. We have considered 8 sensors, 4 anchors, 0.5 radio range and no noise.

$$RMSD = \frac{1}{n_x}\left(\sum_{i=1}^{n_x} ||\hat{x}_i - x_i||_2^2\right)^{1/2} \tag{7}$$

In the random sensor-anchor placement scenario, SDP, SOCP, and ADMM emerged as effective optimization approaches. Both SDP and SOCP exhibited good performance, indicating their suitability for accurately localizing sensors with anchors placed using the quasi-random Halton Sequence. However, ADMM also showed promise but with significantly higher variance in the obtained solutions.

For home healthcare monitoring with anchors on the perimeter, SDP demonstrated superior performance. SOCP also proved to be effective in localizing sensors and showed similar variance to SDP. This is advantageous since SOCP is generally more efficient and quicker to run than SDP, making it a viable option for this scenario. SOCP's ability to handle convex hull localization contributed to its success.

In the low power healthcare sensing scenario where sensors are placed beyond the convex hull, SOCP's performance significantly degraded. It performed worse on average than even LS, which is generally considered less effective. ADMM, on the other hand, displayed relatively good average performance but suffered from extremely high variance, making it less suitable for this scenario. For the scenario with no anchors, SOCP and SDP perform poorly! ADMM is the clear winner.

Overall, SDP consistently provided reliable results across all cases, making it the most reliable and effective optimization approach. Its robust performance can be attributed to its ability to handle various scenarios and its reliable convergence properties. On the other hand, LS consistently performed the worst due to its susceptibility to being trapped in local minima and its inability to handle non-convex problems.

The analysis of the box plot results indicates that SDP is generally the most reliable and effective optimization approach for the considered scenarios, while SOCP proves to be a suitable alternative in certain cases. Understanding the strengths and weaknesses of these optimization algorithms can guide their selection in specific application scenarios, leading to improved sensor localization and overall system performance.

B. Effect of Varying Radio Range and Noise Factor

In order to address the challenges posed by varying noise and radio range, our study narrows its focus specifically on disaster management, where these factors are particularly prominent, highlighting their impact on communication and proposing

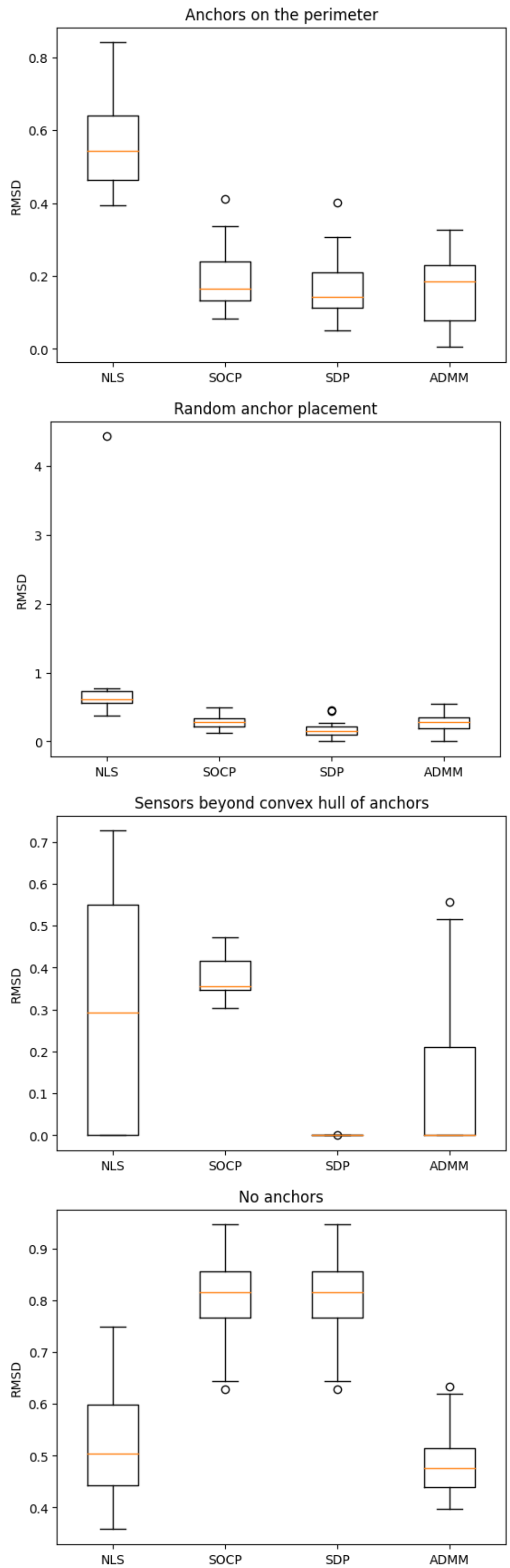

Figure 5. Box plots of RMSD for the algorithms with 8 sensors, 4 anchors, 0.5 radio range and no noise.

strategies to mitigate their effects. We simulated this for the other application domains, and the results remained consistent with what we found for this particular case. However, we include the results from only disaster management, as we would like to avoid repitition of our analysis.

We introduce noise in our distance measurements as follows:

$$d_{ij} = d_{ij}(1 + \epsilon_{ij} * \sigma) \ \forall(i,j) \in N_x, i < j$$
$$\hat{d}_{kj} = \hat{d}_{kj}(1 + \epsilon_{kj} * \sigma) \ \forall(k,j) \in N_a$$

where $\epsilon_{ij}, \epsilon_{kj} \sim N(0,1)$ and σ is the noise factor.

These figures illustrate the root mean square deviation (RMSD) values plotted against the radio range for three different algorithms: SDP, SDP refined, and ADMM. The experiments were conducted with varying levels of noise, including none, low (noise fatcor = 0.1), and high (noise factor = 0.25). To ensure accuracy, the results were averaged over five different seed values. The RMSD values provide insights into the performance of each algorithm under different noise conditions and radio ranges.

From Figure 6, a few key observations can be made. Firstly, as the noise level increases, the root mean square deviation also increases. This indicates that higher levels of noise negatively impact the accuracy of the localization algorithms, which is very logical. Secondly, as the radio range increases, the RMSD decreases. This is an expected result, as a larger radio range means more distance measurements between sensors and anchors. Lastly, it is interesting to note that the SDP algorithm performs better than its refined version, contrary to expectations. This suggests that the least squares 'refinement' of the SDP solution basically worsens the solution by moving it to a local minima.

Additionally, an interesting observation is that the ADMM algorithm outperforms both forms of SDP algorithms for higher noise levels. This implies that ADMM is more robust and effective in handling noisy sensor network localization scenarios, where the accuracy of distance measurements may be compromised. The superior performance of ADMM in such conditions highlights its potential for practical applications in real-world environments with significant noise interference.

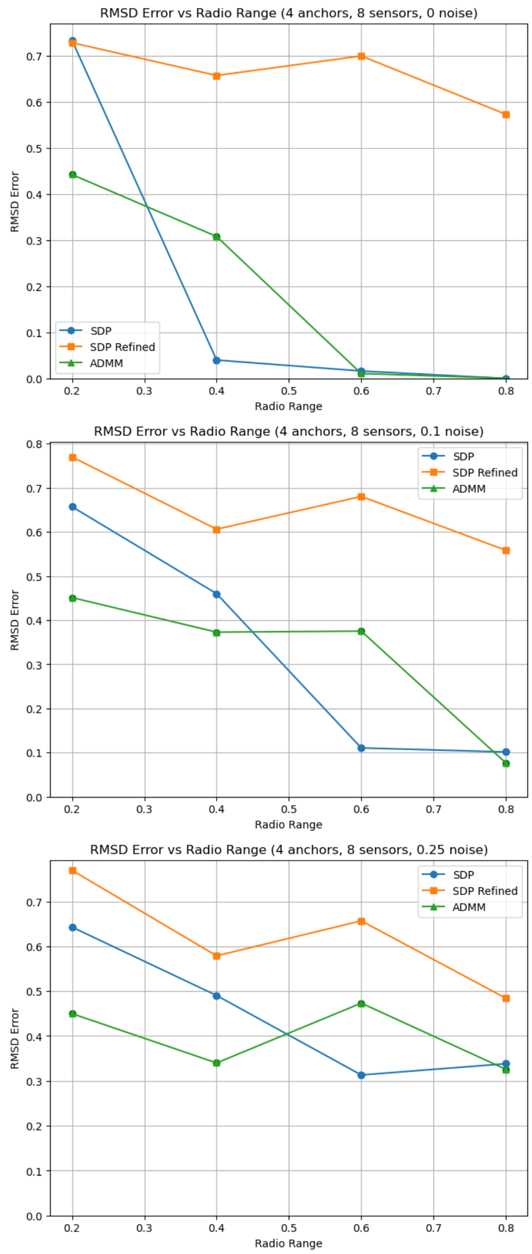

Figure 6. These figure show the RMSD vs Radio Range for Different Noise Levels

C. Effect of number of anchors

Next, we understand the effect of the number of anchors with respect to the performance of the algorithms. Towards this, we fix the number of sensors to 15, and vary the number of anchors and track the error for all the algorithms, resulting in Figure 7.

For all algorithms except Least Squares (LS), the RMSD error reduces drastically with increasing number of anchors. Therefore, the performance gain from deploying additional anchors is significant and hence highly recommended. With Least Squares, the RMSD tends to increase with increasing number of anchors. This is because, the more complex the problem, the higher is its sensitivity to the initial conditions and the likelihood of getting trapped in local minimas.

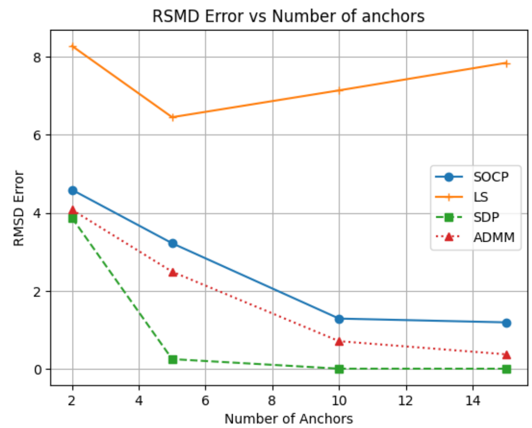

Figure 7. This figure shows the improvement in the performance of algorithms with increasing number of anchors when placed randomly

D. Timing Analysis

We conducted a runtime analysis for varying numbers of sensors. We focused on the general case with low noise level (noise standard deviation equal to 0.1) and high radio range. (See Figure 8). In this case, we consider the disaster management (anchors placed via Halton sequence) and adult care scenario (anchors placed on perimeter) only.

Firstly, we observed that the runtime increases as the number of sensors increases, ranging from 5 sensors to 50 sensors. This is expected, as a larger number of sensors leads to a more complex problem, requiring more computation time.

Secondly, when comparing the three algorithms, we found that the ADMM algorithm consistently exhibited the highest runtimes across all cases, by a significant margin. This is odd, as we expected ADMM to perform more quickly than SDP. However, due to the hyperparameter tuning required for quicker results, our current implementation of ADMM might not have been fully optimized, and further investigation is needed to determine its potential performance advantages over SDP. Without

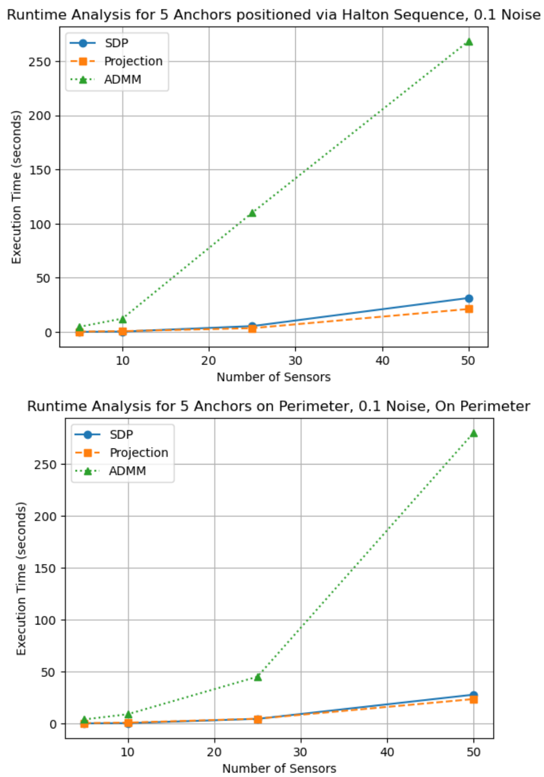

Figure 8. The runtimes for 5 anchors vs number of sensors. The top image shows the runtime for halton sequence anchor placement, and the bottom image shows the runtime for perimeter placement.

this tuning to each use case, ADMM may not be suitable for low-power and less efficient computers typically found in home monitoring systems, requiring simpler algorithms better optimized for resource-constrained environments.

Furthermore, we observed that the SDP with Projections algorithm generally outperformed the other algorithms. This is reasonable, considering that it is a relaxation of the SDP approach. However, the performance gap between SDP and SDP with projection is not substantial for few sensors, especially when the anchors are placed on the convex hull.

Conclusion

This project examined the effectiveness of algorithms such as Least Squares (LS), Second Order Cone Programming (SOCP), Semidefinite Programming (SDP), SDP Refined, SDP with Projection, and Alternating Direction Method of Multipliers (ADMM) in solving the localization problem. We consider three different application scenarios, including adult care, low power healthcare, and disaster management, to evaluate the impact of anchor placement. We end up finding that SDP tends to be the most effective algorithm in localizing the sensors correctly; however, it suffers from long runtimes that may make methods like SOCP (for the adult care scenario in particular) and SDP with projection more appropriate choices if resources are constrained.

Moving forward, there are several potential next steps that we could consider to further enhance the research on sensor network localization in wireless sensor networks:

1) Real-world validation: While simulations provide valuable insights, conducting real-world experiments would add credibility to the findings. Collaborating with healthcare facilities or disaster management agencies to deploy sensor networks in practical scenarios could validate the effectiveness of the proposed algorithms.
2) Scalability and resource constraints: Investigating the scalability of the algorithms is crucial, as real-world sensor networks often involve a large number of nodes. Analyzing the proposed algorithms handle resource constraints, such as limited power or computational capabilities, would be beneficial for practical implementation, especially for dense networks of sensors.
3) Robustness and fault tolerance: Exploring the resilience of the localization algorithms to various types of errors, uncertainties, and network failures would contribute to their reliability. Assessing the algorithms' ability to adapt to changing environmental conditions, node failures, or dynamic network topologies would be valuable. Different noise models could also be investigated.

References

[1] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in Proceedings of the third international symposium on Information processing in sensor networks, ACM Press, 2004, pp. 46–5
[2] Wang, S. Zheng, S. Boyd, and Y. Ye, "Further relaxations of the SDP approach to sensor network localization," Tech. Rep., 2006.

[3] A. So. A Semidefinite Programming Approach to the graph realization problem: Theory, Applications and Extensions Phd Thesis, Stanford University, 2007.

[4] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. Linear Matrix Inequalities in System and Control Theory. SIAM., 1994

APPENDIX

The code repository for this project is included here as a github link: https://github.com/hya00/CME307-Final-Project

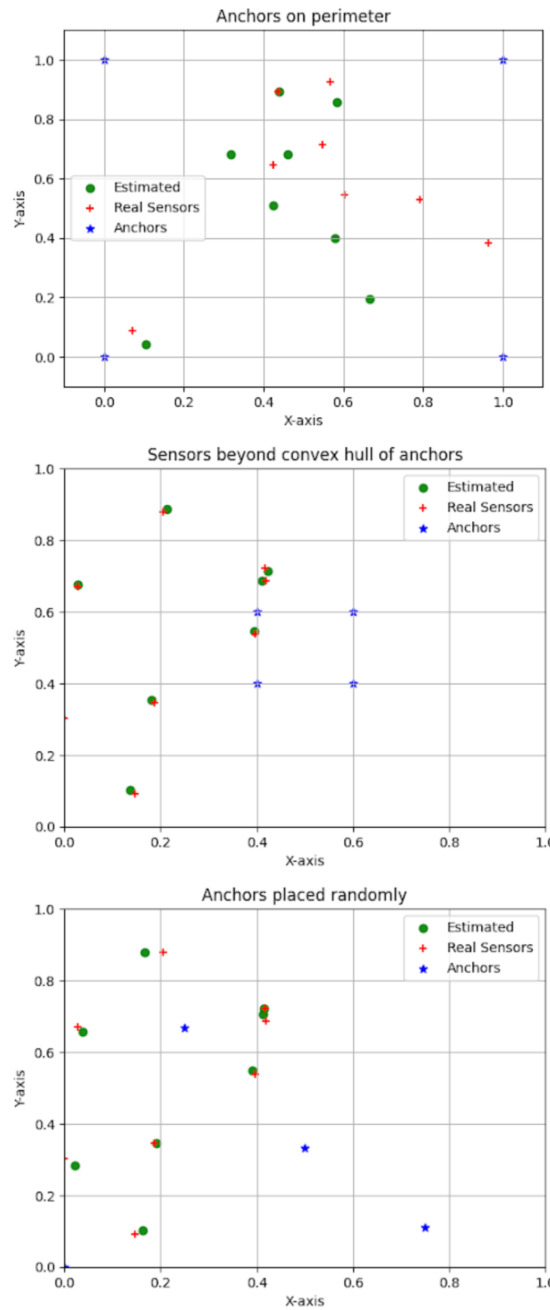

Figure 9. Sample Grid Visualization of localization by Least Squares for each application scenraio

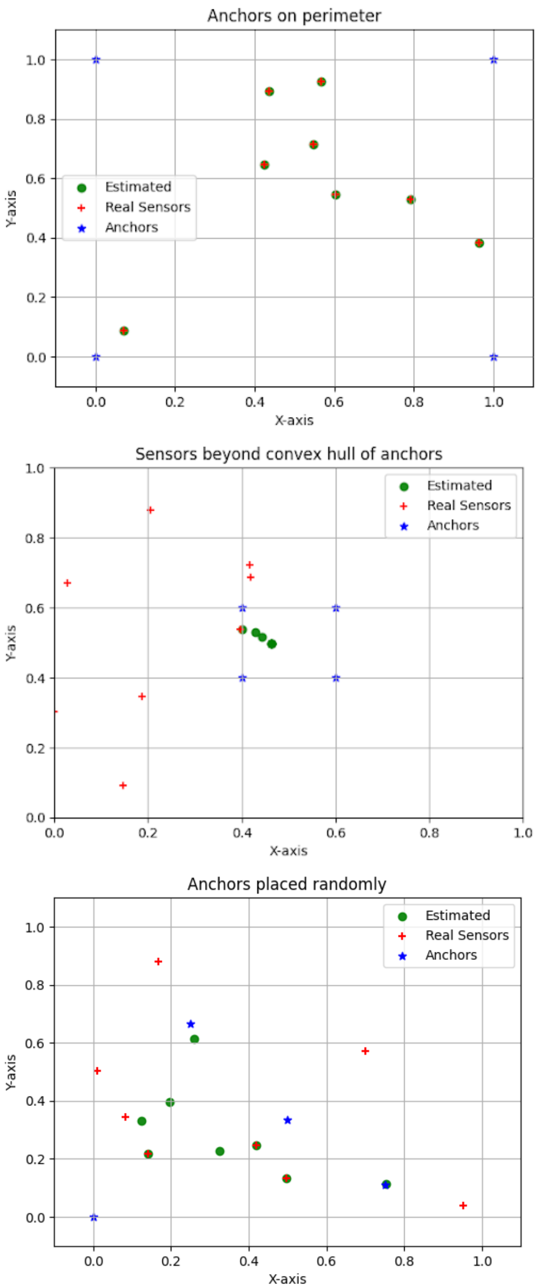

Figure 10. Sample Grid Visualization of localization by SOCP for each application scenraio

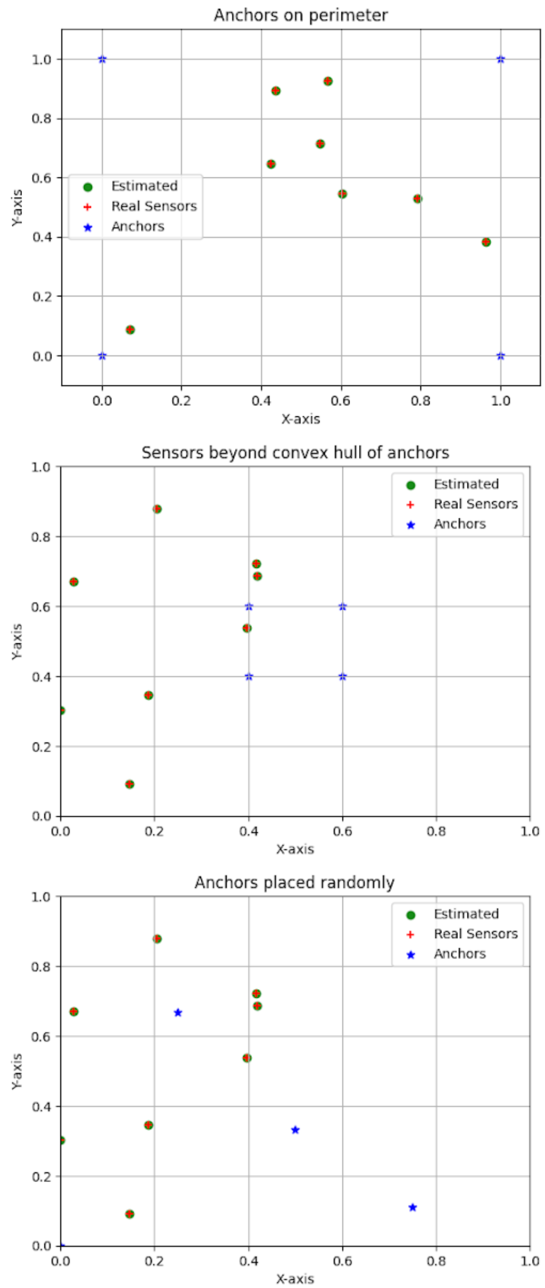

Figure 11. Sample Grid Visualization of localization by SDP for each application scenraio

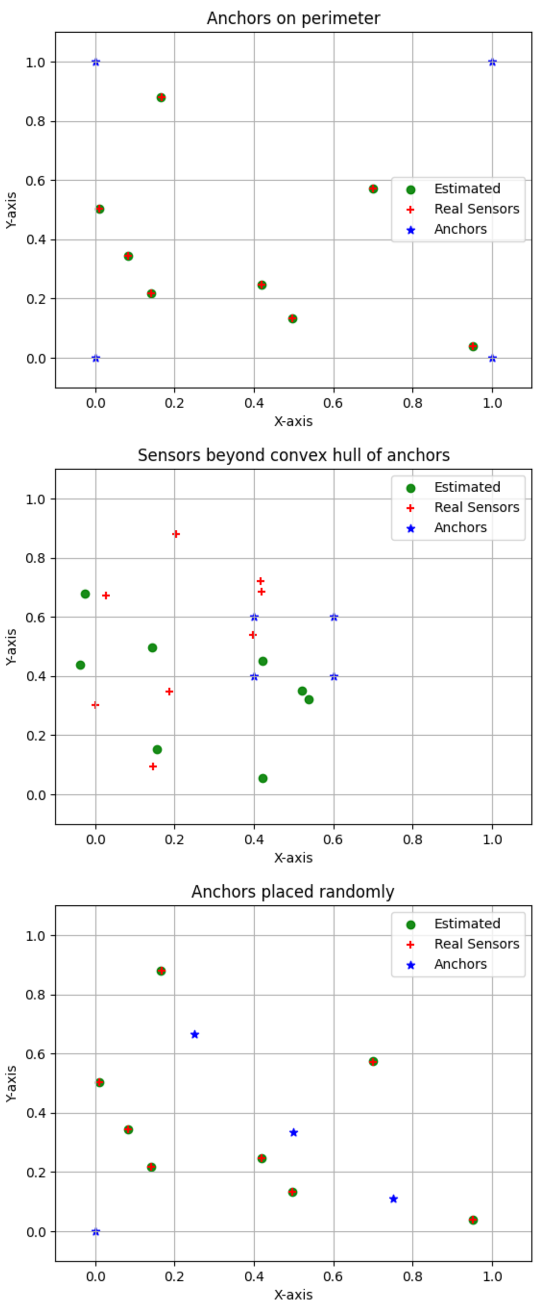

Figure 12. Sample Grid Visualization of localization by ADMM for each application scenraio