Cooperative Localization Using Received Signal Strength and Least Squares Estimation Methods

Nader Moayeri
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
moayeri@nist.gov

2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN) | 979-8-3503-2011-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/IPIN57070.2023.10332522

***Abstract*—In this paper we use two Recursive Least Squares (RLS) estimation methods to solve the cooperative localization problem based on Wi-Fi received signal strength measurements between pairs of nodes in a network that can establish communication links. A link can be between an anchor node and a non-anchor node or between a pair of non-anchor nodes. It is the latter piece of information that makes cooperation possible and this problem different from multilateration. We evaluate the performance of Newton-Raphson and Gauss-Newton RLS methods and show that cooperation can offer significant gains in performance over multilateration, which does not use cooperation.**

***Index Terms*—cooperative localization, multilateraion, least squares estimation, received signal strength, Newton-Raphson, Gauss-Newton**

I. Introduction

Generally speaking, a cooperative localization system [1]-[2] is one that estimates the locations of a set of entities – people and/or objects equipped with localization devices – based on not only the information exchanged between these entities and an infrastructure in the environment, as is the case with "ordinary" localization systems, but also based on the information exchanged between the entities themselves. Examples of the infrastructure include the constellation of GNSS satellites in the sky, the Wi-Fi access points installed in a building, and the 3D map of a building used by video odometry or LiDAR-based localization techniques. The use of cooperation in a localization system makes it possible to estimate the locations of some entities that might not be possible without cooperation and/or to achieve higher localization accuracy for all entities than would be possible with the same system without cooperation. This comes at the expense of higher computational complexity.

There are different ways of classifying cooperative localization systems [3]. A major distinction is whether the system is centralized or distributed. Centralized algorithms may be appropriate for a small set of entities, but they are not scalable when the number of entities grows. Distributed algorithms are scalable and hence more practical, but they take longer to converge to a location solution and they typically yield lower localization accuracy than comparable centralized algorithms. For every distributed localization algorithm there is a centralized counterpart, but the converse is not true. The development of centralized algorithms is important because they offer a baseline for what can be achieved with distributed algorithms.

The entities in a cooperative localization system form a network. Localization can be done based on connectivity alone [4–6], ranging among network nodes [5, 7, 8], and measured angles [1, 9, 10]. Ranging can be done based on the Received Signal Strength (RSS) of a radio frequency (RF) signal [11–13] or its Time of Flight (ToF) [14], which requires Time of Arrival (ToA) estimation.

We now turn our attention to localization based on imprecise range estimates. A performance comparison of three methods, namely trilateration, Newton-Raphson, and Gauss-Newton, for localization based on range estimates computed from Wi-Fi RSS is presented in [13], but the paper does not address cooperation. A sequential form of cooperation is proposed in [15], where any sensor node that has range estimates to at least three anchor nodes can estimate its own location and turn into an auxiliary anchor node. Recursive localization methods are used when no further anchor nodes can be added in this manner. A distributed Gauss-Newton localization method is presented in [11], where it is ensured that the global cost function is never increased as the algorithm goes through its iterations. A property of the Cayley-Menger determinant is used in [16] to formulate sensor node localization as an optimization problem with a set of quadratic equality constraints. This property puts a constraint on the distances from a sensor node to any collection of 3 (4) anchor nodes in 2D (3D) space. A weighted Recursive Least Squares (RLS) localization method based on RSS measurements is presented in [12]. It consists of using iterative multilateration, multidimensional scaling (MDS) [6, 8, 17], and maximum likelihood estimation, solved using weighted RLS techniques, one after another.

This paper focuses on centralized cooperative localization using Wi-Fi RSS in 2D and 3D spaces. We use the commonly used power law path loss model for Wi-Fi RSS [10] and RLS methods [18] – particularly Newton-Raphson and Gauss-Newton methods – to solve this problem. The rest of the paper is organized as follows. Section II presents the RLS methods used in this paper, with cooperation or without. Section III goes over the assumptions we use for system modeling and presents our simulation results. Finally, Section IV concludes the paper.

II. Recursive Least Squares Localization

Consider a network of nodes in D-dimensional space ($D = 2$ or 3) with N anchor nodes with known locations and M nodes representing the entities whose locations need to be estimated. Even though the non-anchor nodes do not have to be sensors, we refer to them as sensor nodes hereafter. For $i = 0, \ldots, M-1$, let $\mathbf{u}_i = (u_{i,0}, \ldots, u_{i,D-1})^T$ and $\hat{\mathbf{u}}_i = (\hat{u}_{i,0}, \ldots, \hat{u}_{i,D-1})^T$ denote the unknown location of the i'th sensor node and an estimate for that location, respectively. For $j = 0, \ldots, N-1$, let $\mathbf{x}_j = (x_{j,0}, \ldots, x_{j,D-1})^T$ denote the known location of the j'th anchor node. Define the super vectors $\mathbf{u} = (\mathbf{u}_0^T, \ldots, \mathbf{u}_{M-1}^T)^T$ and $\hat{\mathbf{u}} = (\hat{\mathbf{u}}_0^T, \ldots, \hat{\mathbf{u}}_{M-1}^T)^T$. The goal in the least-squares formulation of the cooperative localization problem based on range estimates (noisy or noise-free) is to find a $\hat{\mathbf{u}}$ that minimizes the cost function

$$J_1(\hat{\mathbf{u}}) = \sum_{(i,j)\in A} [R_{i,j} - \|\hat{\mathbf{u}}_i - \mathbf{x}_j\|]^2 + \sum_{(i,j)\in B} [Q_{i,j} - \|\hat{\mathbf{u}}_i - \hat{\mathbf{u}}_j\|]^2,$$

where $\|.\|$ denotes the l_2-norm,

$$\begin{aligned} A = \{&(i,j) : 0 \le i \le M-1, 0 \le j \le N-1, \\ &\text{sensor } i \text{ and anchor } j \text{ can range to each other}\}, \\ B = \{&(i,j) : 0 \le i < j \le M-1, \\ &\text{sensors } i \text{ and } j \text{ can range to each other}\}, \end{aligned}$$

and $R_{i,j}$'s and $Q_{i,j}$'s denote the range estimates between pairs of nodes in A and B, respectively. Note that not all pairs of nodes are able to range to each other because they may be out of "communication range". The question of which pairs of nodes can range to each other is addressed in the next section. In this paper, we also study the following alternative cost function:

$$J_2(\hat{\mathbf{u}}) = \sum_{(i,j)\in A} [R_{i,j}^2 - \|\hat{\mathbf{u}}_i - \mathbf{x}_j\|^2]^2 + \sum_{(i,j)\in B} [Q_{i,j}^2 - \|\hat{\mathbf{u}}_i - \hat{\mathbf{u}}_j\|^2]^2$$

Other cost functions have also been proposed [12].

The main purpose of this paper is to evaluate and compare the performance of two localization systems, one that uses cooperation and one that does not. In the latter case we end up with the multilateration problem based on range estimates from sensor nodes to anchor nodes only [19]. Specifically, for $i = 0, \ldots, M-1$, the goal in the multilateration problem is to find a $\hat{\mathbf{u}}_i$ that minimizes either

$$J_{1,i}(\hat{\mathbf{u}}_i) = \sum_{\{j:(i,j)\in A\}} [R_{i,j} - \|\hat{\mathbf{u}}_i - \mathbf{x}_j\|]^2$$

or

$$J_{2,i}(\hat{\mathbf{u}}_i) = \sum_{\{j:(i,j)\in A\}} [R_{i,j}^2 - \|\hat{\mathbf{u}}_i - \mathbf{x}_j\|^2]^2.$$

The multilateration problem has to be solved separately for each of the M sensor nodes. It is expected that a well-designed algorithm for cooperative localization would outperform a well-designed algorithm for multilateration.

In this paper we focus on Newton-Raphson and Gauss-Newton RLS methods [18] as a means of solving both the cooperative localization and multilateration problems. For example, for the cooperative localization problem each method computes a sequence of estimates for $\mathbf{u}$ until a stopping criterion is satisfied. In this paper, with $\hat{\mathbf{u}}^{(k)}$ and $\hat{\mathbf{u}}^{(k+1)}$ denoting two successive estimates of $\mathbf{u}$, an algorithm stops if $\|\hat{\mathbf{u}}^{(k+1)} - \hat{\mathbf{u}}^{(k)}\| \le \epsilon$ or the number of iterations reaches a prescribed limit K. The mathematical formulas for the recursions used by the two RLS methods can be derived by specializing the formulas given in [18] to the two cost functions defined above. For the sake of brevity, the formulas are not presented here. We just have to remember that there are two choices of RLS methods and two choices of cost functions. We have studied all four possible combinations.

Just as in any minimization problem dealing with a non-convex function, there is no guarantee of convergence to a global minimum. The situation is even more complicated with RLS methods because they attempt to find a critical point of the function to be minimized. A critical point is where the gradient of the function is zero. That point can be a local minimum, a local maximum, or a saddle point. There is also the possibility that the algorithm may diverge. These are well-known problems, for which a number of remedies are available in the optimization literature [20]. For example, convergence to a local minimum can be guaranteed with a gradient descent algorithm and an appropriate line search method [21, 22]. There are pros and cons to each of these remedies, and that's why there are so many of them out there. Our experience with the Newton-Raphson and Gauss-Newton methods was that the algorithms rarely diverge or converge to a local maximum or a saddle point when the network is well-connected, but the possibility of convergence to a local minimum is a concern.

III. Modeling, Simulations, and Performance

The first issue that needs to be addressed is whether a pair of nodes that are at a given distance d can communicate with each other. Many papers in the wireless networking literature use the so-called unit-disk graph model for connectivity. With that model, two nodes can communicate *iff* their distance is less than or equal to a given communication range. That is a deterministic model. It is not realistic because two radios that are in line of sight of each other can communicate over longer distances than two that are not. In this paper we use the well-known power law path loss model [10] given by

$$P_R = P_0 - 10\alpha \log_{10} d + X, \tag{1}$$

to decide stochastically whether two nodes can communicate. In our simulations we know the locations of all nodes, whether they are anchor nodes or sensors. Therefore, if the distance between a pair of nodes is d, we compute a possible value for P_R by generating a sample of the zero-mean Gaussian random variable X with variance σ^2 that represents lognormal

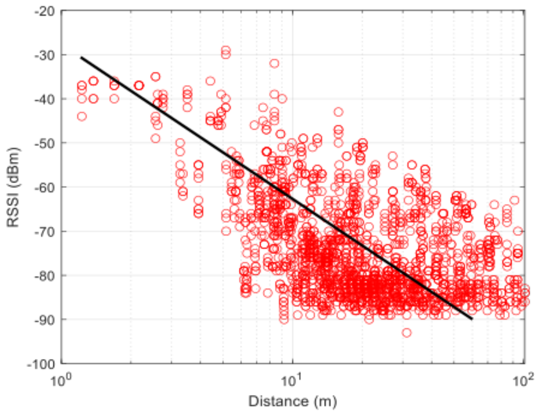

Fig. 1. A plot of Wi-Fi RSSI vs. distance in the PerfLoc data

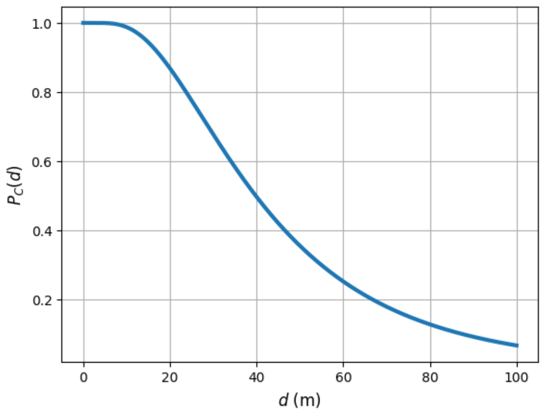

Fig. 2. Probability of link establishment for two nodes at distance d

shadow fading and then compare P_R with the known receiver sensitivity $\underline{P}$. We assume the nodes can establish a link and communicate if $P_R \geq \underline{P}$. Under these assumptions, the probability that two nodes at distance d can communicate is given by

$$P_C(d) = Q\left(\frac{\beta}{\sigma}\right) = \frac{1}{2}\text{erfc}\left(\frac{\beta}{\sigma\sqrt{2}}\right),$$

where $\beta = \underline{P} - P_0 + 10\alpha \log_{10} d$.

The parameters of the power law path loss model are computed by a least squares fit of a straight line to a point cloud of (d, P_R) points in a log-log plot. The points in the cloud are obtained via data collection in the field. For this purpose, we used the extensive set of annotated Wi-Fi RSSI data NIST collected in conjunction with the PerfLoc Prize Competition [23] for development and performance evaluation of smartphone indoor localization apps. That data set contains both line-of-sight and non-line-of-sight data. Fig. 1 shows the collected data in one of the four buildings in which PerfLoc data was collected as well as the least squares line fitted to the data.

There is one caveat. The WLAN chipset in the smartphone reports Wi-Fi RSSI for "received" packets only. Therefore, there is no data for the cases where the RSS was smaller than than the Wi-Fi receiver sensitivity. The least squares line fitted to the data would have a more negative slope than in the true path loss model. We made adjustments to the least squares fit line to account for this phenomenon. The resulting parameters for the power law path loss model, after this adjustment, turned out to be roughly $P_0 = -30$ dBm, $\alpha = 3.75$, $\sigma = 10$ dB, and $\underline{P} = -90$ dBm. Fig. 2 shows a plot of $P_C(d)$ with the selected path loss model parameters.

A pair of nodes at distance d can range to each other only if they can communicate. An estimate $\hat{d}$ for d is obtained by plugging P_R in (1), without the X term, and solving for the distance:

$$\hat{d} = 10^{(P_0 - P_R)/(10\alpha)}$$

Using (1), an alternative expression for $\hat{d}$ that relates it to the true distance d is as follows:

$$\hat{d} = d \times 10^{-X/(10\alpha)}$$

Hence, after we have generated a sample for the random variable X and decided that the given pair of nodes can communicate, we use that sample to generate $\hat{d}$.

We assume the sensor and anchor nodes reside in a square (cubic) region in the 2D (3D) localization scenarios. Specifically, let $\mathcal{C}_D = [-W/2, W/2]^D$, with $D = 2$ or 3, denote the deployment region for the nodes. The locations of the M sensor nodes are always selected randomly in an i.i.d. fashion using a uniform distribution over $\mathcal{C}_D$.

We examined two deployment strategies for the N anchor nodes. They can be placed according to a square (cubic) lattice with spacing Δ laid over $\mathcal{C}_D$ or randomly. We assume that W is an integer multiple of Δ, and we let $n = W/\Delta$ denote the number of anchor nodes along each dimension. Therefore, with the first choice, there will be $N = n^D$ anchor nodes. In addition, we examined two ways of selecting the starting point for the RLS method under consideration, with or without cooperation. One is to use a random starting point and the other is to do an exhaustive search over a fine square or cubic lattice with spacing δ laid over $\mathcal{C}_D$ to find a lattice point at which the cost function of choice for the multilateration problem is minimized and then use that as the starting point. Clearly, M such searches need to be done. An exhaustive grid search over $[-W/2, W/2]^{M \times D}$ with J_1 or J_2 would not be practical due to its prohibitively high computational complexity. There are four possible combinations depending on which choices are made vis-a-vis the two issues described above. We found that the combination of lattice deployment of anchor nodes and random starting points for RLS methods consistently outperformed the other three combinations in terms of localization accuracy. We were surprised that a random starting point turned out to be better than one obtained through a grid search. All the performance results presented hereafter were obtained using the winning combination.

We carried out simulations with these choice of parameters, $D = 2$, $W = 60$ m, $\delta = 1$ m, $M \in \{10, 15, 20, 30\}$, $n \in \{4, 5, 6, 8\}$, $\epsilon = 0.01$, and $K = 100$, to evaluate the performance of several localization methods using cost functions I and II. We use the acronyms GS = Grid Search, NR = Newton-Raphson, GN = Gauss-Newton, WOC = WithOut Cooperation, and WC = With Cooperation to name different methods. GS-I and GS-II minimize $J_{1,i}$'s and $J_{2,i}$'s, respectively. The cost for *any* method is computed by plugging the $\hat{\mathbf{u}}$ found by that method into the expression for J_1 or J_2. Likewise, we compute the cost for the "Genie Solution", which is the cost if we plug $\mathbf{u}$ in the expression for J_1 or J_2.

Table I shows the performance obtained with $M = 20$ and $n = 4$ using $L = 10,000$ randomly selected sets of locations for M sensor nodes. The anchor node density is one every $\Delta = 15$ m of distance, which corresponds to one in every 225 m^2 of area. This is the same density as what we have in our building at NIST. We make the following observations:

TABLE I
PERFORMANCE OF VARIOUS LOCALIZATION METHODS WITH $D = 2$, $W = 60$, $\delta = 1$, $M = 20$, AND $n = 4$

Localization Method	Mean Localization Error (in m)	Relative Cost Change wrt Genie Solution	Mean Number of Iterations
GS-I	11.90	-37.33%	
NR-I-WOC	12.11	-36.58%	6.51
NR-I-WC	11.27	-37.93%	55.03
GN-I-WOC	11.99	-37.17%	8.41
GN-I-WC	10.42	-46.13%	37.82
GS-II	15.13	-66.70%	
NR-II-WOC	15.14	-66.71%	5.42
NR-II-WC	13.20	-70.69%	14.34
GN-II-WOC	15.14	-66.71%	7.31
GN-II-WC	13.20	-70.70%	25.74

- Cost function II results in considerably higher mean localization error than cost function I across the board.
- RLS methods without cooperation achieve practically the same mean localization error in a few iterations as grid search methods.
- Cooperation results in lower mean localization error than not cooperating.
- GN methods achieve slightly lower mean localization errors than NR methods, particularly with cooperation.
- The relative change in cost with respect to the Genie Solution is in the same range for a given cost function. The relative change is always negative, which implies that there is no hope that any method would result in $\hat{\mathbf{u}} \approx \mathbf{u}$. However, this does not mean there does not exist a $\hat{\mathbf{u}}$ that achieves a lower cost and would even yield a lower mean localization error. It makes one wonder whether there exists a better cost function to minimize.
- It is hard to get much useful information from the results on the mean number of iterations, because each iteration of a WOC method involves inverting a $D \times D$ matrix, while each iteration of a WC method involves inverting a $(D \times M) \times (D \times M)$ matrix. In addition, each iteration of an NR method requires more computations than each iteration of a corresponding GN method.

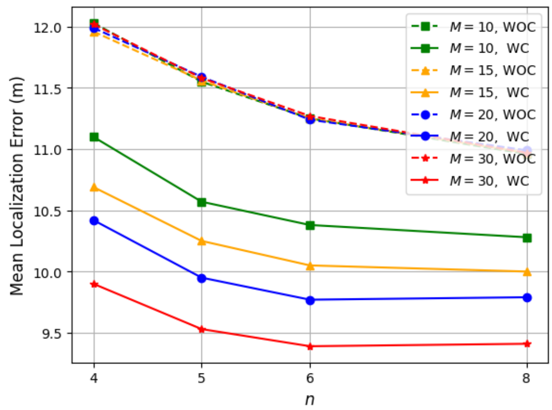

Fig. 3. Mean localization error for GN-I RLS method w/ or w/o cooperation

The trends observed in Table I were also seen with other choices of M and n. Since the lowest mean localization error is achieved with the Gauss-Newton RLS method with cost function I and using cooperation, the results presented from this point on are for that system. Of course, comparisons will be made with the GN-I-WOC method to assess the gain in performance due to cooperation.

Figure 3 plots mean localization error for systems with or without cooperation for $n \in \{4, 5, 6, 8\}$ and $M \in \{10, 15, 20, 30\}$. The curves for systems that do not use cooperation are right on top of each other. This should not come as a surprise because the performance of a system without cooperation is independent of M. Basically, we are dealing with the multilateration problem, which does not exploit sensor to sensor range information and estimates the location of each sensor independent of the other ones. Yet, the fact that these curves are for all practical purposes identical gives us confidence that our simulations are correct and maybe – maybe – they are finding the global minima of $J_{1,i}$'s. We make two other observations. First, the mean localization error of all systems, with or without cooperation, decreases with n, i.e. as the anchor node density in the building is increased. Second, the gap between the identical curves at the top of the figure and the other four curves increases with M. This is better seen in Fig. 4, which plots the cooperation gain defined as the decrease in mean localization error as one goes from a system without cooperation to one with. The cooperation gain is increasing with M, as more sensor to sensor range information becomes available, but it is decreasing with n beyond $n = 5$. As anchor node density is increased, cooperation can only help so much.

Figure 5 is a plot of the Cumulative Distribution Function (CDF) of localization error for the Gauss-Newton RLS method with cooperation and cost function I. It shows once again that the performance improves with M because a larger M leads to more cooperation.

IV. CONCLUSIONS

In this paper, we tackled the cooperative localization problem based on imprecise range estimates between network nodes using the Newton-Raphson and Gauss-Newton RLS

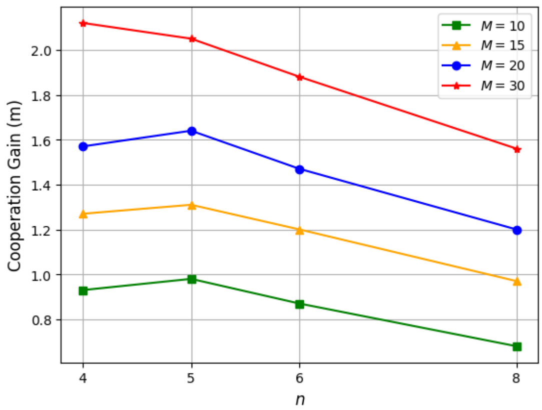

Fig. 4. Cooperation gain as a function of n and M

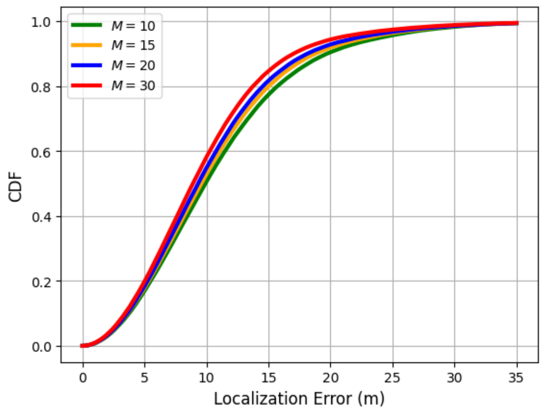

Fig. 5. Cumulative Distribution function of localization error for various values of M and $n = 4$

methods. The range estimates are between sensor nodes and anchor nodes, with known locations, as well as between pairs of sensor nodes. The latter peer-to-peer range estimates are the ones that make cooperation possible.

The setting for our simulations is a Wi-Fi network, even though the methods developed in this paper are equally applicable if the network nodes use Zigbee radios, Bluetooth Low Energy, or some other wireless technology. We chose to work with Wi-Fi because we have access to a large annotated set of Wi-Fi RSSI data at NIST. We used that data to develop a model for connectivity among network nodes – who can talk to who – as well as a relationship between imprecise, multipath-affected range estimates between pairs of connected nodes and their true distances.

It should not come as a surprise that cooperation results in a lower localization error than not using cooperation. We quantified this improvement in localization accuracy. We did this by comparing the use of RLS methods for solving the multilateration problem, which use range estimates between sensor nodes and anchor nodes only, and using those methods for solving the cooperative localization problem. We found that the improvement in localization accuracy, or decrease in mean localization error, can be as much as 25%. We expect even larger improvements when the number of sensor nodes increases, but the computational complexity of the centralized algorithms we used may become the bottle neck.

We plan to develop distributed cooperative localization methods and compare our work to existing methods that were mentioned in Section I. Comparisons to other methods and papers pose a challenge, because they use different models for range error and connectivity. We were fortunate to be able to use the invaluable PerfLoc data [23] as a basis for developing our models. We also believe that development of cooperative localization methods based on UWB ranging merits attention, and we expect the cooperation gain to be higher in that setting. We plan to look at that problem as well.

References

[1] 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, Jul. 2005.

[2] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative Localization in Wireless Networks,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[3] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[4] D. Niculescu and B. Nath, “Ad hoc positioning system (APS),” in *GLOBECOM'01. IEEE Global Telecommunications Conference*, vol. 5, Nov. 2001, pp. 2926–2931 vol.5.

[5] L. Doherty, K. Pister, and L. El Ghaoui, “Convex position estimation in wireless sensor networks,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 3, Apr. 2001, pp. 1655–1663 vol.3.

[6] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from connectivity in sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

[7] P. Biswas and Y. Ye, “Semidefinite programming for ad hoc wireless sensor network localization,” in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, Apr. 2004, pp. 46–54.

[8] X. Ji and H. Zha, “Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling,” in *IEEE INFOCOM 2004*, vol. 4, Mar. 2004, pp. 2652–2661 vol.4.

[9] D. J. Torrieri, “Statistical Theory of Passive Location Systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 2, pp. 183–198, Mar. 1984.

[10] T. S. Rappaport, *Wireless Communications: Principles and Practice, 2e.* Prentice Hall, 1996.

[11] B. Cheng, R. Hudson, F. Lorenzelli, L. Vandenberghe, and K. Yao, "Distributed Gauss-Newton method for node loclaization in wireless sensor networks," in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, 2005.*, Jun. 2005, pp. 915–919.

[12] X. Li, "Collaborative Localization With Received-Signal Strength in Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3807–3817, Nov. 2007.

[13] A. Kuntal, P. Karmakar, and S. Chakraborty, "Optimization technique based localization in IEEE 802.11 WLAN," in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, May 2014, pp. 1–5.

[14] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network Experimentation for Cooperative Localization," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 467–475, Feb. 2012.

[15] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Proceedings Ninth International Conference on Network Protocols. ICNP 2001*, Nov. 2001, pp. 35–41.

[16] M. Cao, B. D. O. Anderson, and A. S. Morse, "Sensor network localization with imprecise distances," *Systems & Control Letters*, vol. 55, no. 11, pp. 887–893, Nov. 2006.

[17] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[18] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall, 1993.

[19] K. Pahlavan, *Indoor Geolocation Science and Technology: At the Emergence of Smart World and IoT*. CRC Press, 2019.

[20] G. A. Seber and C. J. Wild, *Nonlinear Regression*. John Wiley & Sons, 1989.

[21] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Jun. 2013.

[22] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Mar. 2004.

[23] N. Moayeri, M. O. Ergin, F. Lemic, V. Handziski, and A. Wolisz, "PerfLoc (Part 1): An extensive data repository for development of smartphone indoor localization apps," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2016, pp. 1–7.