

Tight Two-Dimensional Outer-Approximations of Feasible Sets in Wireless Sensor Networks

Siamak Yousefi, *Student Member, IEEE*, Henk Wymeersch, *Member, IEEE*, Xiao-Wen Chang, and Benoit Champagne, *Senior Member, IEEE*

Abstract—Finding a tight ellipsoid that contains the intersection of a finite number of ellipsoids is of interest in positioning applications for wireless sensor networks (WSNs). To this end, we propose a novel geometrical method in 2-dimensional (2-D) space. Specifically, we first find a tight polygon, which contains the desired region and then obtain the tightest ellipse containing the polygon by solving a convex optimization problem. For demonstrating the usefulness of this method, we employ it in a distributed algorithm for elliptical outer-approximation of feasible sets in co-operative WSNs. Through simulations, we show that the proposed method gives a tighter bounding ellipse than conventional methods, while having similar computational cost.

Index Terms—Computational geometry, localization, non-line-of-sight, optimization.

I. INTRODUCTION

LOCALIZATION of sensor nodes in a wireless sensor network (WSN) is of great interest in many public safety and commercial applications [1]. In particular, cooperative localization has received special attention since it can improve localization accuracy and coverage [2]. In contrast to non-cooperative WSN, in which only measurements between the sensors being localized and anchors with known positions are performed, cooperative WSNs also use sensor-to-sensor measurements.

In non-line-of-sight (NLOS) situations, in which the range measurements become positively biased, the unknown location of each sensor is restricted to the intersection of multiple balls (or discs in 2-dimensional (2-D) space), with centres corresponding to the locations of neighbouring nodes, i.e., anchors and sensors, and with radii equal to the biased range measurements. The intersection of these balls is a convex feasible set, which can serve as a rough approximation of the uncertainty in the true sensor's position. However, since this feasible set cannot be generally described by a few parameters, outer-approximating it by a simple shape, e.g., a ball or an ellipsoid, is needed.

Manuscript received October 31, 2015; accepted January 11, 2016. Date of publication January 14, 2016; date of current version March 8, 2016. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the European Research Council (ERC) under Grant 258418 (COOPNET). The associate editor coordinating the review of this paper and approving it for publication was D. Qiao.

S. Yousefi and B. Champagne are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: siamak.yousefi@mail.mcgill.ca; benoit.champagne@mcgill.ca).

H. Wymeersch is with the Division of Communication Systems, Information Theory, and Antennas, Chalmers University of Technology, Gothenburg SE-412 96, Sweden (e-mail: henkw@chalmers.se).

X.-W. Chang is with the School of Computer Science, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: chang@cs.mcgill.ca).

Digital Object Identifier 10.1109/LCOMM.2016.2518186

In cooperative WSNs, finding outer-approximations of these feasible sets is not straightforward as the centres of the balls corresponding to the locations of the neighbouring sensors are unknown. To address this issue, a distributed iterative algorithm was proposed in [3], where a ball is used for an outer-approximation of a feasible set. The algorithm has been improved in [4] by using ellipsoids instead of balls, on the basis that an ellipsoid can generally capture a complex convex set more tightly due to its additional degrees of freedom. The core operation in [4] is the outer-approximation of multiple ellipsoids by a tight ellipsoid.

Finding the tightest ellipsoidal outer-approximation of the intersection of multiple ellipsoids is NP-hard [5], and to the best of our knowledge, there is no algorithm to find the optimal solution. However, there are several sub-optimal solutions, including the ones considering two ellipsoids [6]–[9], as well as standard convex optimization methods for a larger number of ellipsoids [5, p.44], [10, p.414]. In [4], the method from [10] has been employed, where first, the largest volume ellipsoid contained in the intersection of multiple ellipsoids is determined by solving a convex optimization problem. Then by expanding this ellipsoid with the dimension of the space, an ellipsoid that covers the intersection region can be found.

The sub-optimal methods from [5], [10] are based on convex relaxations and may not generally offer a tight enough ellipsoid. As the localization problems can often be considered in a 2-D space (i.e., unknown latitude and longitude), there is a special interest in developing geometrical methods in 2-D that can find tighter ellipses. To this end, we propose a novel method, in which we first efficiently determine a tight polygon containing the intersection of ellipses, and then solve a convex optimization problem to obtain the tightest ellipse covering the vertices of the polygon. We employ the proposed method in the distributed algorithms considered in [4] for outer-approximations of 2-D feasible sets in cooperative WSNs and show that it offers significant improvements in tightness with similar computational cost, compared to the case that the method from [10] is employed.

Notation: The vector 2-norm is denoted by $\|\cdot\|$. The set of $\nu \times \nu$ symmetric positive-definite matrices is denoted by \mathbb{S}_{++}^{ν} .

II. SYSTEM MODEL AND BACKGROUND

A. System Model

We consider a 2-D WSN with N sensor nodes at unknown locations denoted by $\mathbf{x}_i \in \mathbb{R}^2$ for $i \in \{1, \dots, N\}$, and M anchors with known locations $\mathbf{a}_i \in \mathbb{R}^2$, for $i \in \{N + 1, \dots, N + M\}$.

$1, \dots, N + M$). Two nodes are regarded as neighbours if they are within communication range, i.e., they are within the given distance R_{\max} of each other. For each sensor node j we define two sets \mathcal{A}_j and \mathcal{S}_j which include the indices of all the neighbouring anchors and sensors, respectively. The range measurements of the j -th sensor are modelled as

$$r_{ij} = \|\mathbf{a}_i - \mathbf{x}_j\| + b_{ij} + n_{ij}, \quad i \in \mathcal{A}_j \quad (1)$$

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| + b_{ij} + n_{ij}, \quad i \in \mathcal{S}_j \quad (2)$$

where n_{ij} are measurement noises, and $b_{ij} > 0$ represent the biases due to the NLOS, while for LOS measurements $b_{ij} = 0$. The noise terms are often assumed to be independent identically distributed Gaussian random variables with zero-mean and variance σ_n^2 , while the bias terms b_{ij} have been modelled as exponential [11], or uniformly distributed random variables [12]. We assume that (1) and (2) correspond to the NLOS measurements only, which can be identified from LOS ones using NLOS identification techniques, as done in [11], [12].¹ Furthermore, to make our algorithm more robust, no knowledge is assumed about the distribution of n_{ij} and b_{ij} . In many applications, the bias dominates over the measurement noise [13], so that $b_{ij} + n_{ij} \geq 0$. Hence it follows that each sensor \mathbf{x}_j is restricted to be inside the intersection area of multiple discs,² defined as

$$\mathcal{D}_{ij}^{\mathcal{A}} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x} - \mathbf{a}_i\| \leq r_{ij}\}, \quad i \in \mathcal{A}_j \quad (3)$$

$$\mathcal{D}_{ij}^{\mathcal{S}} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x} - \mathbf{x}_i\| \leq r_{ij}\}, \quad i \in \mathcal{S}_j \quad (4)$$

Therefore, $\mathbf{x}_j \in \mathcal{D}_j$ where

$$\mathcal{D}_j = \left(\bigcap_{i \in \mathcal{A}_j} \mathcal{D}_{ij}^{\mathcal{A}} \right) \cap \left(\bigcap_{i \in \mathcal{S}_j} \mathcal{D}_{ij}^{\mathcal{S}} \right). \quad (5)$$

Our objective is to determine an outer-approximation of the convex feasible set \mathcal{D}_j for every sensor \mathbf{x}_j through a distributed approach.³ Note that each $\mathcal{D}_{ij}^{\mathcal{A}}$ is available to sensor j , while each $\mathcal{D}_{ij}^{\mathcal{S}}$ is not *a-priori* available since \mathbf{x}_i is unknown. Therefore, the solution is not straightforward.

B. Definition of Ellipsoids

An ellipsoid ξ_i in ν -dimensional space \mathbb{R}^ν can be defined in many different ways [10], including:

(i) The image of the unit ball under an affine transformation:

$$\xi_i = \{\mathbf{x} = \mathbf{P}_i \mathbf{y} + \mathbf{x}_{c,i} : \|\mathbf{y}\| \leq 1, \mathbf{y} \in \mathbb{R}^\nu\}, \quad (6)$$

where $\mathbf{x}_{c,i}$ is the centre of the ellipsoid, and without loss of generality $\mathbf{P}_i \in \mathbb{S}_{++}^\nu$.

(ii) A quadratic form:

¹The LOS measurements can later be used in conventional localization algorithms (e.g., nonlinear least-squares or SPAWN [2]) by taking advantage of the bounds obtained in this work.

²If the condition $b_{ij} + n_{ij} \geq 0$ can not be guaranteed (e.g., due to large σ_n), a constant can be added to each r_{ij} in the right hand side of (3) and (4) [14] to ensure that the position of each sensor is restricted to the intersection of discs with neighbouring nodes as centres.

³We assume that for every sensor j , there is at least one neighbouring node with pairwise measurement r_{ij} such that \mathcal{D}_j is not empty.

$$\xi_i = \{\mathbf{x} \in \mathbb{R}^\nu : \|\mathbf{B}_i \mathbf{x} + \mathbf{d}_i\| \leq 1\}, \quad (7)$$

where $\mathbf{B}_i \in \mathbb{S}_{++}^\nu$ and $\mathbf{d}_i \in \mathbb{R}^\nu$ is a translation vector. When $\mathbf{B}_i = \mathbf{P}_i^{-1}$ and $\mathbf{d}_i = \mathbf{P}_i^{-1} \mathbf{x}_{c,i}$, the two ellipsoids in (6) and (7) are identical.

III. OUTER-APPROXIMATION OF FEASIBLE SETS

In this section, we first describe the proposed method for outer-approximation of the intersection of ellipses, and then apply it to the distributed bounding algorithm given in [4].

A. Tight Outer-Approximation of the Intersection of Ellipses

We show how it is possible to efficiently find a tight polygon, represented by \tilde{m} vertices $\mathbf{w}^{(l)}$ for $l = 1, \dots, \tilde{m}$, to cover the intersection of ellipses. The ellipses are denoted with ξ_i for $i = 1, \dots, p$, and their intersection with \mathcal{E} , i.e., $\mathcal{E} = \bigcap_{i=1}^p \xi_i$. The smallest area ellipse that contains these vertices (and hence contains \mathcal{E}) is found by solving the following convex optimization problem [10], [15]:

$$\begin{aligned} \min_{\mathbf{B}, \mathbf{d}} \quad & \log \det(\mathbf{B}^{-1}) \\ \text{s.t.} \quad & \mathbf{B} \in \mathbb{S}_{++}^2, \|\mathbf{B} \mathbf{w}^{(l)} + \mathbf{d}\| \leq 1, l = 1, \dots, \tilde{m}, \end{aligned} \quad (8)$$

where $\det(\mathbf{B}^{-1})$ is proportional to the area of the ellipse. Since each inequality in (8) can be written as a linear matrix inequality, this optimization problem can be formulated as a standard semi-definite programming (SDP) problem. For the ellipse to tightly bound \mathcal{E} , the polygon which bounds \mathcal{E} has to be tight. Hence, the problem reverts to the determination of a polygon that covers \mathcal{E} tightly. We propose below a method with three steps to achieve this:

Step 1 (generating discrete points): We first generate a number of discrete points on the boundary of \mathcal{E} . One way to do so is to generate a fixed number of points on the boundary of each ξ_i forming \mathcal{E} and then reject those that do not lie on \mathcal{E} . Harnessing the fact that an ellipse is an image of the unit disc under an affine transformation, we first generate m points $\mathbf{y}^{(l)}$ for $l = \{1, \dots, m\}$, uniformly on a unit circle and then map these points onto the desired ellipse ξ_i as defined in (6), through the transformation $\mathbf{z}_i^{(l)} = \mathbf{P}_i \mathbf{y}^{(l)} + \mathbf{x}_{c,i}$. After rejecting the points among those $m \times p$ points that are not on the boundary of \mathcal{E} , we denote the remaining points by $\tilde{\mathbf{z}}^{(l)}$ for $l = \{1, \dots, \tilde{m}\}$ and the associated ellipse index for each point by $i^{(l)}$. The remaining points are shown with white color in Fig. 1.

Step 2 (generating half planes): Utilizing the form (7), the tangent lines to the i -th ellipse at the points $\tilde{\mathbf{z}}^{(l)}$ can be obtained, and hence the half planes are formed

$$(\mathbf{B}_{i^{(l)}} \tilde{\mathbf{z}}^{(l)} + \mathbf{d}_{i^{(l)}})^T (\mathbf{B}_{i^{(l)}} \mathbf{x} + \mathbf{d}_{i^{(l)}}) \leq 1, \quad l = 1 \dots, \tilde{m}. \quad (9)$$

Step 3 (determining the vertices of a polygon): For a sufficient number of points $\tilde{\mathbf{z}}^{(l)}$, the intersection of these half planes forms a closed polygon covering \mathcal{E} . One way to find this polygon is to obtain the intersection point of every pair of tangent lines by solving a linear system of two equations, and to verify if this point is inside the intersection region of all the remaining half-planes, i.e., if it satisfies all the remaining $\tilde{m} - 2$ affine

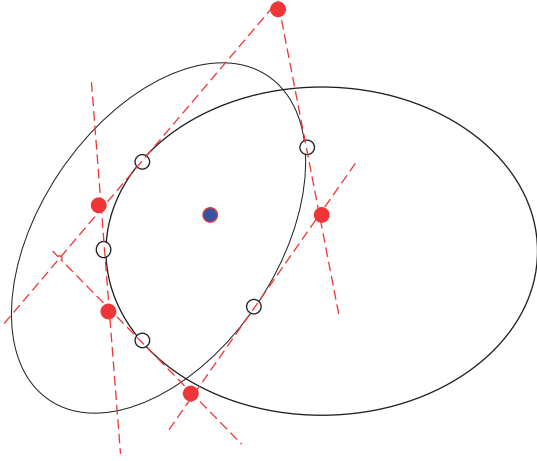


Fig. 1. The diagram of the intersecting ellipses and the half-planes forming a closed convex polygon. The white, dark blue, and red points correspond to $\tilde{z}^{(l)}$, z_{mean} , and $w^{(l)}$, respectively.

inequalities. Note that there is a total of $\tilde{m}(\tilde{m} - 1)/2$ of such linear systems. The complexity of the above procedure is $\mathcal{O}(\tilde{m}^3)$ flops. Hence, this procedure might be very time consuming when m is large. Herein, we make use of the fact that $\nu = 2$ to develop a more efficient approach.

- *Step 3a:* Given the points $\tilde{z}^{(l)}$ for $l = \{1, \dots, \tilde{m}\}$, we first compute the average $z_{\text{mean}} = \frac{1}{\tilde{m}} \sum_{l=1}^{\tilde{m}} \tilde{z}^{(l)} \in \mathcal{E}$, which is shown with dark blue color in Fig. 1. The vectors $\mathbf{v}^{(l)} = \tilde{z}^{(l)} - z_{\text{mean}}$ connecting z_{mean} to the points $\tilde{z}^{(l)}$ are sorted according to the angles $\alpha^{(l)} \in [0, 2\pi)$, measured with respect to the horizontal axis. This sorting imposes an order to the points $\tilde{z}^{(l)}$.
- *Step 3b:* For any two sequential points $\tilde{z}^{(l)}$ in the ordering, we determine the intersection point of the corresponding two tangent lines. The obtained intersection points, $w^{(l)}$ for $l = \{1, \dots, \tilde{m}\}$, which are shown with red color in Fig. 1, form the vertices of the polygon and are used as an input to (8).

In terms of complexity, in *Step 3*, the proposed technique requires solving \tilde{m} linear systems of two equations to find the polygon, hence the computational cost is $\mathcal{O}(\tilde{m})$.

We note that some degenerate cases can occur in this method such that a closed and convex polygon that covers the intersection of ellipses cannot be formed. This problem can be avoided if m is large enough. In practice, a proper value of m can be chosen by preliminary experiments.

B. Distributed Outer-Approximation for Positioning

In the first iteration of the distributed bounding algorithm, each sensor with index j , finds a tight ellipse that contains the intersection of multiple discs corresponding to the neighbouring anchors, with each disc \mathcal{D}_{ij}^A described in (3). While in [4], the method from [10, p.414] has been employed, we obtain an ellipse using the method described in Section III-A, by solving (8) and finding the parameters \mathbf{B}_j and \mathbf{d}_j .⁴ Then, each sensor exchanges the information of its bounding ellipse with its neighbouring sensors through the communication link.

⁴If a sensor has no neighbouring anchors, then an arbitrary large ellipse that contains the area under study is employed to represent its uncertainty.

In the second and next iterations, each sensor j uses the information of the neighbouring sensors with index $i \in \mathcal{S}_j$ as well to reduce the area of the ellipse obtained so far. Since \mathcal{D}_{ij}^S is not *a-priori* available, the bounding ellipse of node i , which has been obtained so far as

$$\{\mathbf{x} : \|\mathbf{B}_i \mathbf{x} + \mathbf{d}_i\| \leq 1\}, \quad i \in \mathcal{S}_j \quad (10)$$

is expanded by r_{ij} along its semi-axes, and thus is guaranteed to contain \mathbf{x}_j . The semi-axes of the i -th ellipse in (10) are the eigenvalues of $\mathbf{P}_i = \mathbf{B}_i^{-1}$. Let $\mathbf{P}_i = \mathbf{V}_i \mathbf{\Gamma}_i \mathbf{V}_i^T$ be the eigen-decomposition of \mathbf{P}_i where $\mathbf{\Gamma}_i = \text{diag}(\lambda_{1,i}, \lambda_{2,i})$. In order to expand the ellipse by r_{ij} , we replace \mathbf{P}_i by $\tilde{\mathbf{P}}_{ij} = \mathbf{V}_i \tilde{\mathbf{\Gamma}}_{ij} \mathbf{V}_i^T$ where $\tilde{\mathbf{\Gamma}}_{ij} = \text{diag}(\lambda_{1,i}, \lambda_{2,i}) + r_{ij} \mathbf{I}_2$. Then for every $i \in \mathcal{S}_j$ the expanded ellipse (calculated at node j) is

$$\{\mathbf{x} : \|\tilde{\mathbf{B}}_{ij} \mathbf{x} + \tilde{\mathbf{d}}_i\| \leq 1\}, \quad i \in \mathcal{S}_j \quad (11)$$

where $\tilde{\mathbf{B}}_{ij} = \tilde{\mathbf{P}}_{ij}^{-1}$ and $\tilde{\mathbf{d}}_i = \tilde{\mathbf{P}}_{ij}^{-1} \mathbf{P}_i \mathbf{d}_i$.

Then, the next step for every sensor is to find a tight ellipse that contains the intersection of multiple discs corresponding to the neighbouring anchors, and multiple expanded ellipses corresponding to the neighbouring sensors. Therefore, the proposed method in Section III-A is employed again so that each sensor updates its current bounding ellipse. Then each sensor exchanges its updated ellipse parameters with the neighbouring sensors. This procedure continues iteratively until convergence or when a predefined number of iterations K , which is chosen by the user according to the time and accuracy constraints, is reached. The distributed iterative bounding algorithm is summarized in Algorithm 1.

Algorithm 1. Distributed Outer-approximating Algorithm

```

1: for  $k = 1$  until convergence (or predefined  $K$ ) do
2:   for  $j = 1, \dots, N$  in parallel do
3:     if  $k = 1$  then
4:       for all  $i \in \mathcal{A}_j$  do
5:         Generate  $m$  points  $z_i^{(l)}$  on the discs in (3).
6:       end for
7:       Reject the points outside  $\mathcal{E}_j$ , i.e., the intersection of
         discs in (3).
8:     else
9:       for each  $i \in \mathcal{S}_j$  do
10:        Expand the  $i$ -th ellipse in (10) to obtain (11).
11:        Generate  $m$  points  $z_i^{(l)}$  on the ellipses in (11).
12:      end for
13:      Reject the points outside  $\mathcal{E}_j$ , i.e., the intersection of
        discs in (3) and ellipses in (11).
14:    end if
15:    Find the half planes tangent to  $\mathcal{E}_j$  at  $\tilde{z}^{(l)}$ , i.e., (9).
16:    Calculate  $z_{\text{mean}}$ ,  $\mathbf{v}^{(l)}$ , and  $\alpha^{(l)}$  for  $l = \{1, \dots, \tilde{m}\}$ .
17:    Sort the vectors  $\mathbf{v}^{(l)}$  according to the angles  $\alpha^{(l)}$ .
18:    Find the intersection point of the tangent lines of every
      two neighbouring points to obtain  $w^{(l)}$ .
19:    Find  $\mathbf{B}_j$  and  $\mathbf{d}_j$  by solving (8).
20:    Exchange the updated  $\mathbf{B}_j$  and  $\mathbf{d}_j$  with neighbours.
21:  end for
22: end for

```

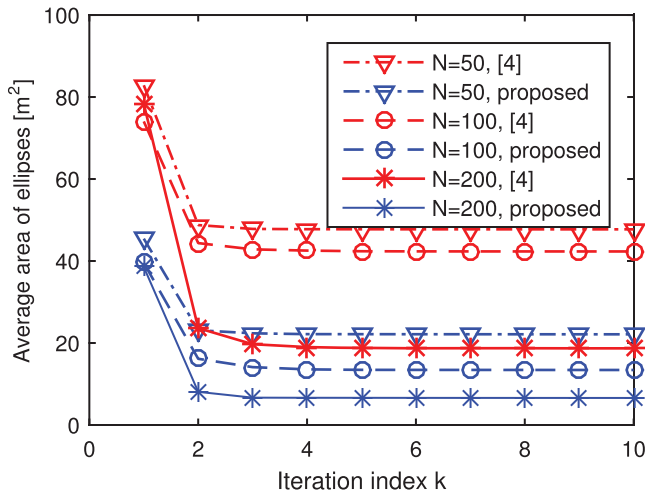


Fig. 2. Comparison of the average area of bounding ellipses as a function of the iteration index k for different N .

TABLE I
COMPARISON OF COMPUTATION TIMES PER SENSOR

Methods	$N = 50$	$N = 100$	$N = 200$
Technique from [4]	7.2 s	12.2 s	23.4 s
Proposed method	8.1 s	14.5 s	26.7 s

IV. NUMERICAL PERFORMANCE EVALUATION

We consider three scenarios where 10 anchors are located on a $100 \text{ m} \times 100 \text{ m}$ 2-D area while 50, 100, and 200 sensors, respectively, are distributed uniformly within this area. The communication range is set to $R_{\max} = 50 \text{ m}$ and the measurement between each pair of neighbouring nodes is obtained by adding to the true range an exponentially distributed positive error with mean equal to 5 m, and zero-mean Gaussian noise with $\sigma_n = 0.5 \text{ m}$. For solving the optimization problems, we use the CVX toolbox [16] in Matlab. We set $m = 256$ since with smaller m , sometimes degenerate cases could occur. The performance is evaluated in terms of the average area of the ellipses in each iteration, quantified by $\det(\mathbf{B}_j^{-1})$. As a benchmark, we use the method presented in [4].

In Fig. 2, we show the average area of the covering ellipses versus the iteration number for different numbers of sensors N . The results show that the distributed algorithm converges rapidly for both outer-approximation methods, although our proposed method converges to outer-approximating ellipses with almost half the area.

In Table I, we compare the computation time of each algorithm for the three scenarios after convergence, i.e., the CPU time required such that the difference between average areas in two consecutive iterations is less than 0.01 m^2 . Since the results are obtained by processing the information centrally on a CPU, we divide the computation time by the number of sensors N to have a better insight of the computation time in a distributed WSN. The results show that the proposed method has similar computation time compared to the one in [4]. Therefore, in terms of the trade-off between accuracy and computational cost, the proposed method is clearly preferred.

V. CONCLUSION

In this letter, we developed a method for tight outer-approximation of the intersection of multiple ellipses in 2-D space. This method was used as part of a distributed algorithm in cooperative WSNs for outer-approximation of the feasible sets containing the positions of the sensors. Through simulations, it was shown that the proposed method results in a tighter approximation of the feasible sets compared to existing techniques, while having a similar computational cost.

REFERENCES

- [1] A. Sayed, A. Tarighat, and N. Khajehpour, "Network-based wireless location: Challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 24–40, Jul. 2005.
- [2] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.
- [3] M. Gholami, H. Wymeersch, E. Strom, and M. Rydstrom, "Wireless network positioning as a convex feasibility problem," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, pp. 1–15, 2011.
- [4] M. Gholami, H. Wymeersch, S. Gezici, and E. Strom, "Distributed bounding of feasible sets in cooperative wireless network positioning," *IEEE Commun. Lett.*, vol. 17, no. 8, pp. 1596–1599, Aug. 2013.
- [5] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA, USA: SIAM, 1994, vol. 15.
- [6] F. L. Chernousko, "Ellipsoidal bounds for sets of attainability and uncertainty in control problems," *Opt. Control Appl. Methods*, vol. 3, no. 2, pp. 187–202, 1982.
- [7] W. Kahan, "Circumscribing an ellipsoid about the intersection of two ellipsoids," *Can. Math. Bull.*, vol. 11, no. 3, pp. 437–441, 1968.
- [8] D. Maksarov and J. Norton, "State bounding with ellipsoidal set description of the uncertainty," *Int. J. Control*, vol. 65, no. 5, pp. 847–866, 1996.
- [9] L. Ros, A. Sabater, and F. Thomas, "An ellipsoidal calculus based on propagation and fusion," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 32, no. 4, pp. 430–442, Aug. 2002.
- [10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [11] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.
- [12] S. Venkatesh and R. Buehrer, "Non-line-of-sight identification in ultra-wideband systems based on received signal statistics," *IET Microw. Antennas Propag.*, vol. 1, no. 6, pp. 1120–1130, Dec. 2007.
- [13] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, Aug. 2009.
- [14] S. Yousefi, X.-W. Chang, and B. Champagne, "Mobile localization in non-line-of-sight using constrained square-root unscented Kalman filter," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2071–2083, May 2015.
- [15] N. Z. Shor and O. Berezovski, "New algorithms for constructing optimal circumscribed and inscribed ellipsoids," *Optim. Methods Softw.*, vol. 1, no. 4, pp. 283–299, 1992.
- [16] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming, version 2.1," Sep. 2014.