Exploration With Massive Sensor Swarms

Stephan Schlupkothen, Guido Dartmann, and Gerd Ascheid
Chair for Integrated Signal Processing Systems, RWTH Aachen University, Germany
{schlupkothen,dartmann,ascheid}@ice.rwth-aachen.de

***Abstract*—In this paper algorithms for wireless sensor networks (WSN) localization and tracking are presented. They are optimized for the exploration of inaccessible environments with unknown shape and dynamics. The algorithms are based on semidefinite programming (SDP) and unscented Kalman filtering (UKF). The SDP approach is derived from a recent solution for WSN localization with anchor node uncertainty, but has been formulated iteratively and allows, therefore, the utilization of massive sensors swarms, which has not been investigates so far.**

***Index Terms*—Localization, senors, tracking, low power, exploration**

I. Introduction

Wireless Sensor Networks (WSN) have recently attracted a lot of researchers because of their flexibility and applicability in various fields. A crucial property of these sensors is their size, from which constraints on battery consumption, communication range and provided measurements are deduced. WSNs are for example very important in the field of exploration of inaccessible environments such as groundwater and other underground systems. Because of the sensor size and underground application a direct measurement via GPS or other satellite demanding techniques is often impossible. Therefore, the WSN has to have the ability to localize itself by limited information, such as range measurements. The localization and tracking problem results hence in a high-dimensional complex optimization problem and is in general non-convex.

Related Work: The problem of WSN localization has already been studied for many years but the focus lay mainly on theoretical question under which conditions a unique solution exists, e.g. [1], [2]. In [3]–[5] the first convex formulations using SDPs were proposed. Later other also analyzed the solvability with numerical optimization algorithms/solvers, [6]. None of these investigated the special problem of huge moving sensor networks.

Contribution: The application scenario of our paper is the exploration of inaccessible environments, which requires a huge amount of moving sensors. Hence, this paper addresses the problem of localization and tracking of sensors, that are traveling through a pipe or tunnel like system. This further complicates the estimation since nodes with known absolute positions exist only at the in- and outflow. For this problem, we evaluated the most promising already existing SDP localization algorithm, [7], and extended it to an iterative formulation which accounts for the exponentially increasing complexity and resource demands when using huge sensors networks, and therefore also exponentially growing constraint sets. In addition, we propose a new unscented Kalman filter (UKF) tracking algorithm for this problem.

II. The Environment Model

The model of our evaluation scenario is closely related to the underground water system. For simplicity, we assume to have a system with one inflow and only one outflow, see Figure 1a. The sensors are inserted constantly into the system to enable at one point a ”steady-state” where sensors are distributed all over the system. Measurements are performed every period. Simultaneously, also two sensors are released into the system. To evaluate the performance of the algorithms, we assume a model which is easier to represent in terms of fluid dynamics, simple to reproduce and more challenging concerning the tracking performance. It is based on a so called sinusoidal path. This means that the center of the path has a sinusoidal shape with a certain amplitude, frequency, phase offset and thickness d_{max}, see Figure 1b. For the sake of simplicity, the fluid is assumed to have a constant velocity profile.

III. Problem Formulation

The mathematical problem of sensor localization with range measurements only is well known and can be written as follows, [3], [5], [7]. The problem is in general non-convex and $\mathcal{NP}$-hard, [6]. We assume to have k anchor nodes whose absolute position, $\boldsymbol{a}_1, \boldsymbol{a}_2 \ldots, \boldsymbol{a}_k \in \mathbb{R}^2$, with $\boldsymbol{a}_i^T = \left[x_i, y_i\right]$, is known. Furthermore we want to localize n moving sensors with the coordinates $\boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+2}, \ldots, \boldsymbol{x}_{k+n} \in \mathbb{R}^2$, with $\boldsymbol{x}_i^T = \left[x_i, y_i\right]$. Then, the Euclidean distance between the sensor i and j will be denoted as r_{ij} and can be written as:

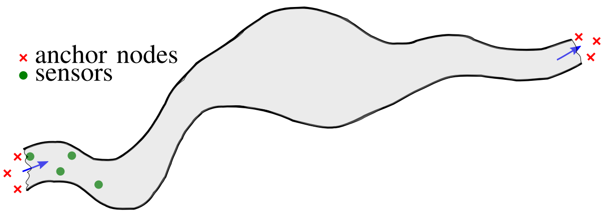
$$r_{ij} = r_{ji} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2 ; \quad i, j = k+1, k+2, \ldots, n; (i,j) \in \mathcal{E} \qquad (1)$$

and similar between anchor node i and sensor node j: $r_{ij} = r_{ji} = \left\|\boldsymbol{a}_i - \boldsymbol{x}_j\right\|_2$, where $m = n + k$ denotes the total amount of sensors and anchor nodes. $\mathcal{E}$ is the set of all anchor to sensor and sensor to sensor distance measurements, which may also be referred to as the set of edges when viewing this as a graph-theoretic problem, i.e. $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{V}$ denotes the set of vertices. The measurements in presence of noise are: $\tilde{r}_{ij} = r_{ij} + n_{ij}$. Furthermore, n_{ij} is assumed to be a Gaussian noise process. The general optimization problem can then be written as

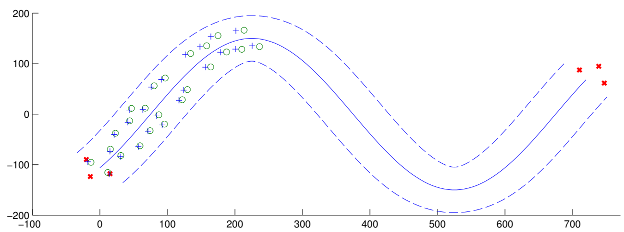
$$\underset{\{\boldsymbol{x}_i\}}{\arg\min} \sum_{(i,j)\in\mathcal{E}} \left[\tilde{r}_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right]^2 \qquad (2)$$

IV. The Semidefinite Programming (SDP) Approach

The general SDP relaxation of Problem (2) is already well known and has been derived in several manners, see [3], [4], [8]. Moreover, they are known to produce accurate results



(a) An underground water model with fixed anchor nodes and entering sensors.



(b) Sensors are continuously inserted in the example path which has been used for the performance evaluation. The red crosses mark the positions of anchor nodes, the green circles the actual position and the blue the current estimates. In this example the pipe diameter is $d_{max} = 45$ and the sensors and anchor nodes sensing radius is $R = 80$.

Fig. 1. The assumed underground system and its model.

in sensor localization problems. For our purposes, we have chosen one of the most recent and advanced formulations of the SDP relaxation that also considers the presence of noisy measurements. Here, we only want to refer the reader to SDP formulation (27) in [7], which we have chosen for a performance evaluation which is comparable to many other formulations that have been published so far.

V. The Iterative Semidefinite Programming (ISDP) Approach

Our iterative SDP algorithm is based on formulation (25) in [7], which considers the case of anchor node uncertainty, i.e. when the absolute positions of the anchor nodes are disturbed by (white) Gaussian noise. We have introduced two further constraints, that preserve the convexity of the problem and reflect the disc graph property: $R^2 \leq \gamma_{ij}$ for $(i,j) \notin \mathcal{E}$ and $r_{ij}^2 \leq \gamma_{ij} \leq R^2$ for $(i,j) \in \mathcal{E}$. This means a connection between two sensors or anchor nodes exists only if their distance is less than the sensors sensing radius R and vice versa. Furthermore, the noise variances have been generalized to parameters, $1/\alpha_{ij}$ and $1/\beta_i$, that can be freely selected and allow further optimization. This is reasonable since in the case of an iterative approach the anchor node noise can not be assumed to be Gaussian anymore. The modified objective is: $\sum_{i=k+1}^{m} \sum_{j=1}^{k} \hat{\alpha}_{ij} \left[\gamma_{ij} - 2\tilde{r}_{ij} r_{ij}\right] + \sum_{i=1}^{k} \beta_i \left[y_{ii} - 2\tilde{\boldsymbol{a}}_i^T \boldsymbol{x}_i\right]$, with:

$$\hat{\alpha}_{ij} := \begin{cases} \delta_{ij}\alpha_{ij} & \text{if } i > k, j > k \\ \delta_{ij}\alpha_{ij}/2 & \text{otherwise} \end{cases} \tag{3}$$

This modified SDP will be denoted as MSDP. In Algorithm (1), the inner loop over variable l represents the iterative behavior of the ISDP. In each iteration l we select a set of nodes $\mathcal{V}_{\text{est}}$ which will be estimated and the corresponding set of nodes

Algorithm 1: The Iterative SDP (ISDP) Algorithm

for $(k = k_{start};\ k \leq k_{end};\ k++)$ **do**
  Initialize set of already determined nodes: $\mathcal{V}_{\text{det}} := \mathcal{V}_{\text{anchor nodes}}$.
  Initialize set of nodes to be estimated in this iteration: $\mathcal{V}_{\text{est}} := \{\}$.
  Copy the set of all vertices from the current graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$, $\bar{\mathcal{V}}_k = \mathcal{V}_k \backslash \mathcal{V}_{\text{det}}$.
  for $(l = 0;\ l \leq \infty;\ l++)$ **do**
    $\mathcal{V}_{\text{est}} := \{$ nodes in $\bar{\mathcal{V}}_k$ which are connected to at least q nodes in $\mathcal{V}_{\text{det}}\}$.
    $\bar{\mathcal{V}}_{\text{det}} := \{$ nodes in $\mathcal{V}_{\text{det}}$ that are connected to nodes in $\mathcal{V}_{\text{est}}\}$.
    Set $\alpha_{ij} = f_\alpha(l)$ and $\beta_i = f_\beta(l)$ and run MSDP with anchor nodes $\bar{\mathcal{V}}_{\text{det}}$ and sensors $\mathcal{V}_{\text{est}}$, save estimated coordinates for $\mathcal{V}_{\text{est}}$.
    $\mathcal{V}_{\text{det}} := \mathcal{V}_{\text{det}} \cup \mathcal{V}_{\text{est}}$.
    $\bar{\mathcal{V}}_k := \bar{\mathcal{V}}_k \backslash \mathcal{V}_{\text{det}}$.
    if $\left|\bar{\mathcal{V}}_k\right| = 0$ **then** break;
  end
end

from $\mathcal{V}_{\text{det}}$ that will help to translate the relative measurements into absolute positions. Based on these sets we will run the above described MSDP in every iteration l with the parameters α_{ij}, β_i, q and $\mathcal{E}_k$: $\hat{\boldsymbol{x}}_{\mathcal{V}_{\text{est}}} = \text{MSDP}_l(\mathcal{V}_{\text{est}}, \mathcal{V}_{\text{det}}, \alpha_{ij}, \beta_i, q, \mathcal{E}_k)$. This procedure is continued until all sensor coordinates have been estimated, i.e. $\left|\bar{\mathcal{V}}_k\right| = 0$ and repeated for every time instant k. Parameter $q \geq 0$ influences how many iterations are necessary to determine all sensor coordinates. For $q = 0$, only a single iteration will be run which is equivalent to the non-iterative SDP formulation. For the highest amount of iterations, it is advised to set q to the maximum value of edges between each node in $\mathcal{V}_{\text{est}}$ and $\mathcal{V}_{\text{det}}$. A reasonable guideline for the choice of q may be the minimum amount of distance measurements q_{min} that is needed in the noise free case for triangulation, i.e. in the two-dimensional case $q_{\text{min}} = 3$. Since the size of the SDP is determined by the size of $\mathcal{V}_{\text{est}}$ and $\bar{\mathcal{V}}_{\text{det}}$, parameter q also gives a trade-off between complexity, accuracy and needed iterations. The function $f_\alpha(l)$ and $f_\beta(l)$ give a further degree of freedom, and allow an iteration dependent weighting between anchor nodes and sensors. Since in this approach the reliability of the estimate decreases with every iteration due to numerical inaccuracy, noise and other factors, we have chosen an declining function: $f_\beta(l) = 1/l$ and $f_\alpha(l) = 1$. This is equivalent to the assumption that the variance of the noise process that disturbs all distance measurements increases in each iteration.

VI. Unscented Kalman Filter Tracking (UKF) Approach

Since the sensors are moving over time, a tracking approach is reasonable, despite the fact that due to increased amount of sensors the complexity rises exponentially with the size of the state vector. Furthermore, this tracking approach reduces - by velocity estimation and prediction - flip ambiguities, i.e. non-unique solutions, which occur for example in these SDP-based algorithms. Regarding highly non-linear systems, the UKF is a promising compromise between performance, accuracy and complexity, [9]. An extended Kalman filter (EKF) is known to lack precision in this case, [10], and has, therefore, not been chosen. Because of the assumed fluid dynamics, see Section II, we have selected the following movement model, where the absolute value of the velocity v is estimated separately. It is assumed to be constant and estimated independent of the velocity angle $\phi = \arctan(\dot{y}/\dot{x})$, [11]. The state vector $\boldsymbol{x}$ is given as: $\boldsymbol{x}_k^T = \begin{bmatrix} x & y & v & \phi & \omega \end{bmatrix}$. Where x and y are the Cartesian coordinates of the sensors, v is the absolute value of the velocity, i.e. $v = \sqrt{\dot{x}^2 + \dot{y}^2}$, and ω is the turning rate. The state evolution and measurements are given by

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{4}$$

$$= \begin{bmatrix} x + \frac{2}{\omega} v \sin(\frac{\omega T}{2}) \cos(\phi + \frac{\omega T}{2}) \\ y + \frac{2}{\omega} v \sin(\frac{\omega T}{2}) \sin(\phi + \frac{\omega T}{2}) \\ v \\ \phi + \omega T \\ \omega \end{bmatrix} + \boldsymbol{w}_k \tag{5}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{6}$$

$$= \begin{bmatrix} \sqrt{(x_{1,k} - x_{2,k})^2 + (y_{1,k} - y_{2,k})^2} \\ \vdots \\ \sqrt{(x_{i,k} - x_{j,k})^2 + (y_{i,k} - y_{j,k})^2} \\ \vdots \end{bmatrix} + \boldsymbol{v}_k \tag{7}$$

where $\boldsymbol{w}_k$ models the process noise and is assumed to be sampled from a Gaussian noise process with zero mean and covariance $\boldsymbol{Q} \in \mathbb{R}^{5n \times 5n}$. The measurement noise is denoted as $\boldsymbol{v}_k$ and is also sampled from a Gaussian noise process with zero mean and covariance $\boldsymbol{R} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$. The measurement function $h(\boldsymbol{x})$ provides the Euclidean distances between different sensors or between sensors and anchor nodes, i.e. for all $(i,j) \in \mathcal{E}$, and stacks them into a single vector. Since the dimensionality of the problem changes over time - sensors may enter and/or exit at any measurement instance or simply a connection between two sensors has been lost - also the size of the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ changes. In Algorithm (2), n_k and $|\mathcal{E}_k|$ refers to the number of sensors and distance measurements at time instant k. At this point in time k, the current graph is called $\mathcal{G}_k = (\mathcal{E}_k, \mathcal{V}_k)$. The functions `ukfPredict()` and `ukfUpdate()` perform the UKF time update and measurement updates respectively, [9].

Algorithm 2: The Unscented Kalman Filter Algorithm

Initialize mean and covariance estimates $\boldsymbol{M}_{k_{\text{start}}} \in \mathbb{R}^{5n_{k_{\text{start}}}}$ and $\boldsymbol{P}_{k_{\text{start}}} \in \mathbb{R}^{|\mathcal{E}_{k_{\text{start}}}| \times |\mathcal{E}_{k_{\text{start}}}|}$.
for $(k = k_{start};\ k \leq k_{end};\ k++)$ **do**
    $\boldsymbol{M}_k := \boldsymbol{M}_{k-1}, \boldsymbol{P}_k := \boldsymbol{P}_{k-1}$
    if $\mathcal{G}_{k-1} \neq \mathcal{G}_k$ **then**
        Determine set of removed and added sensors $\mathcal{V}^- := \{v_1^-, v_2^-, \ldots, v_l^-\}$ $\mathcal{V}^+ := \{v_1^+, v_2^+, \ldots, v_p^+\}$.
        for $(i = 1;\ i \leq |\mathcal{V}^-|;\ i++)$ **do**
            Remove line in $\boldsymbol{M}_k$ corresponding to v_i^-.
            Remove columns and rows in $\boldsymbol{P}_k$ corresponding to v_i^-.
        end
        for $(i = 1;\ i \leq |\mathcal{V}^+|;\ i++)$ **do**
            Insert and initialize line in $\boldsymbol{M}_k$ corresponding to v_i^+ with values from $\boldsymbol{M}_0$.
            Insert and initialize columns and rows in $\boldsymbol{P}_k$ corresponding to v_i^- with values from $\boldsymbol{P}_0$.
        end
    end
    $[\boldsymbol{M}_k, \boldsymbol{P}_k] :=$ `ukfPredict`$(\boldsymbol{M}_k, \boldsymbol{P}_k, \boldsymbol{Q}, f(\boldsymbol{x}))$;
    $[\boldsymbol{M}_k, \boldsymbol{P}_k] :=$ `ukfUpdate`$(\boldsymbol{M}_k, \boldsymbol{P}_k, \boldsymbol{R}, h(\boldsymbol{x}))$;
end
return $\boldsymbol{M}_k$.

VII. Evaluation Criteria

For detecting and exploring the sensor environment, we are mainly interested in the trajectory estimate accuracy. Therefore, we propose the following criteria, where $\hat{x}$ denotes the estimate of x and x^* its true value:

a) The root mean square error (RMSE) is first computed for every sensor i over all time instances $k = k_{\text{start}}, k_{\text{start}} + 1, \ldots, k_{\text{end}}$, $K = k_{\text{end}} - k_{\text{start}}$, and then averaged over all sensors n: $\text{RMSE} = \frac{1}{n}\sum_{i=1}^{n} \text{RMSE}_i$, where

$$\text{RMSE}_i = \sqrt{\frac{1}{K}\sum_{k=k_{\text{start}}}^{k_{\text{end}}} (\hat{x}_{i,k} - x_{i,k}^*)^2 + (\hat{y}_{i,k} - y_{i,k}^*)^2} \tag{8}$$

b) After the first sensors have reached the outflow, a more accurate estimate is possible since more sensors are directly connected to anchor nodes and, thus, the amount of needed ISDP iterations is reduced. Hence, we also examine the RMSE after the first two sensors have reached the outflow, RMSE_{out}.

c) In addition, we consider the distance between the estimated position and the true sensor trajectory. This is comparable to a time invariant RMSE and is independent of the current sensor position: $\text{RMSE}_{\text{traj}} = \frac{1}{n}\sum_{i=1}^{n} \text{RMSE}_{\text{traj}}^i$, with $l \in [k_{\text{start}}, k_{\text{end}}]$:

$$\text{RMSE}_{\text{traj}}^i = \sqrt{\frac{1}{K}\sum_{k=k_{\text{start}}}^{k_{\text{end}}} \min_l \left\| \begin{pmatrix} \hat{x}_{i,k} \\ \hat{y}_{i,k} \end{pmatrix} - \begin{pmatrix} x_i^*(l) \\ y_i^*(l) \end{pmatrix} \right\|_2^2} \tag{9}$$

Where $x_i^*(l), y_i^*(l)$ denotes the sensors real, time continuous trajectory.

VIII. Evaluation

We provide evaluation results for the sinusoidal path, see Figure 1b, with a radial extend of $d = 35$, amplitude $A = 150$, frequency $f = 1/600$, phase offset $\Delta\phi = -\pi/4$, maximal velocity $v_{max} = 30$ and variable sensing radius R for 1.25 periods. The simulation starts after the insertion of the first sensor pair and has been terminated after all sensors have reached the outflow and, thus, have been collected. In total

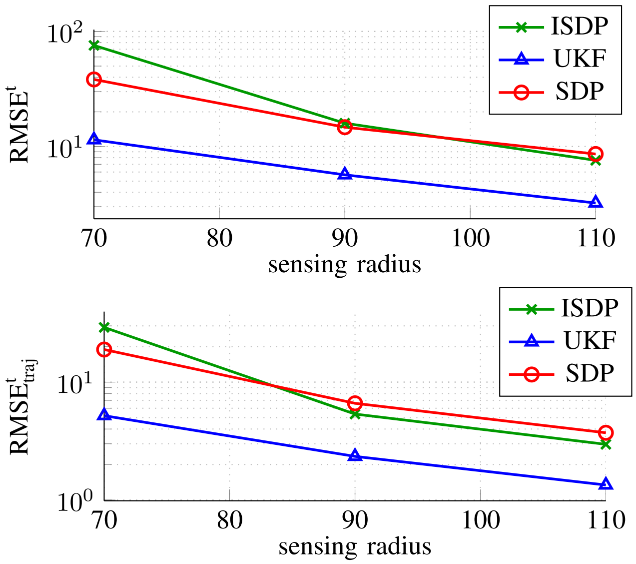


Fig. 2. Evaluation results for $\sigma^2 = 0.1$

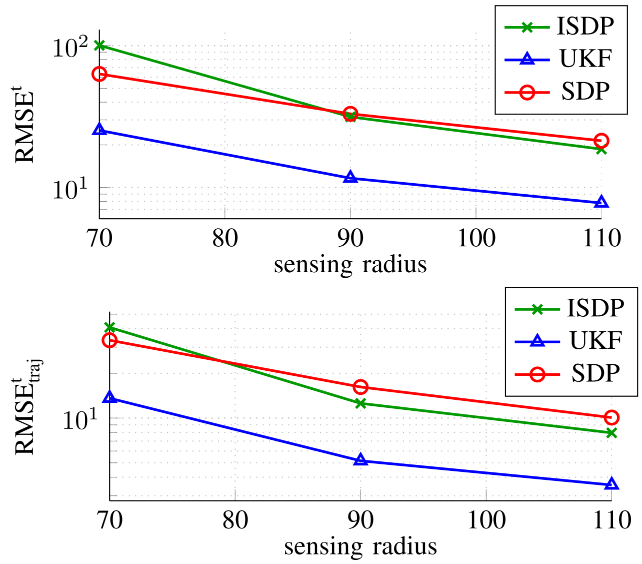


Fig. 3. Evaluation results for $\sigma^2 = 1$

200 sensors have been simulated. For the ISDP algorithm, we chose $q = q_{max}$. In the figure legends, the following abbreviations are used:

- ISDP refers to Algorithm 1
- UKF corresponds to Algorithm 2
- SDP refers to equation (27) in [7], see also Section IV

Figure 2 and 3 show that throughout the simulations the UKF algorithm outperforms all SDP-based ones. This is due to the fact that the UKF considers the verlocity information, in constrast to the SDP-based algorithms. Moreover, especially the ISDP method suffers from error propagation because of its iterative structure. However, for large sensing radii - i.e. when the connectivity is high and less ISDP iterations are executed - the ISDP algorithm performs better than the SDP. In our scenario, the computational complexity and resource requirements of the ISDP algorithms is significant lower compared to the SDP algorithm, see Table I. This is not only because less sensors are used in each iteration, but also because the amount of constraints scales exponentially with this. Especially linear matrix inequalities (LMI) are complex and memory-intensive. In addition, we have seen that adding more nodes from set $\mathcal{V}_{\text{det}}$ to $\bar{\mathcal{V}}_{\text{det}}$ increases the accuracy, at cost of complexity. The computational time for the SDP and ISDP method was, in our simulations, higher by a factor of 1.76 and 2.66, respectively, than for the UKF.

TABLE I
COMPLEXITY COMPARISON

	SDP	ISDP for sensing radii R				
		70	75	80	85	90
avg. inner iterations	1	23.9	22.2	21.9	21.3	21.9
avg. nodes/inner iteration	48.6	5.7	6.1	6.5	6.7	7.1

IX. CONCLUSION

This paper investigates the WSN localization and tracking problem. It provides two algorithms that are especially suitable for the application of many sensors. The UKF method, using a movement model to increase the estimation accuracy and the ISDP method was proposed. The various parameters of the ISDP enable the adaption to several application cases and offer a trade-off between complexity and accuracy. Further investigations on this seems to be interesting.

REFERENCES

[1] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. D. O. Anderson, and P. Belhumeur, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, 2006.

[2] B. Jackson and T. Jordan, "Graph theoretic techniques in the analysis of uniquely localizable sensor networks," 2008.

[3] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005, pp. 405–414.

[4] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, 2006.

[5] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.

[6] G. Calafiore, L. Carlone, and M. Wei, "Position estimation from relative distance measurements in multi-agents formations," in *18th Mediterranean Conference on Control Automation (MED), 2010.*, 2010, pp. 148–153.

[7] K. W. K. Lui, W.-K. Ma, H. So, and F. K. W. Chan, "Semi-definite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 752–763, 2009.

[8] Z. Wang, S. Zheng, S. Boyd, and Y. Ye, "Further relaxations of the sdp approach to sensor network localization," Tech. Rep., 2006.

[9] E. Wan and R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC), 2000.*, 2000, pp. 153–158.

[10] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[11] X. Li and V. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, 2003.