Signal Processing 174 (2020) 107574

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Robust MIMO radar target localization based on lagrange programming neural network

Zhanglei Shi, Hao Wang, Chi Shing Leung, Hing Cheung So*, Member EURASIP

Department of Electrical Engineering, City University of Hong Kong, China

ARTICLE INFO

Article history:
Received 30 August 2019
Revised 29 January 2020
Accepted 9 March 2020
Available online 30 April 2020

Keywords:
Multiple-input multiple-output (MIMO) radar
Target localization
Lagrange programming neural network (LPNN)
Locally competitive algorithm (LCA)
Outlier,

ABSTRACT

In a multiple-input multiple-output (MIMO) radar system, there are a number of transmitters and receivers. We can use a set of range measurements from MIMO system to locate a target. Each range measurement is the sum of the transmitter-to-target distance and target-to-receiver distance, which corresponds to elliptic localization. This paper addresses the MIMO radar target localization problem with possibly outlier measurements. We formulate the problem via non-smooth constrained optimization with an ℓ_1-norm objective function, which is non-differentiable, and the Lagrange programming neural network (LPNN) is adopted as the solver. As the LPNN framework cannot handle non-differentiable objective functions, we utilize two techniques, namely, approximation of the ℓ_1-norm and locally competitive algorithm, to develop two LPNN based algorithms. Moreover, the stability of the LPNN-based algorithms is studied. Simulation results demonstrate that the proposed algorithms outperform two state-of-the-art algorithms.

1. Introduction

Conventional radar systems process a single waveform at a time. On the other hand, a multiple-input multiple-output (MIMO) radar system [1–5] uses multiple antennas to transmit and receive a number of waveforms and process the received signals jointly. In doing so, the MIMO technology is superior to the conventional systems in target detection and localization. There are two MIMO radar configurations. One is colocated configuration in which the antennas are placed close together [4]. A colocated MIMO radar system utilizes the waveform diversity for performance improvement. Another setting is widely distributed configuration in which the antennas are widely separated [3]. A widely distributed MIMO radar system employs the spatial diversity to improve its localization accuracy. This paper considers the target localization problem using the widely distributed MIMO radar.

For widely distributed MIMO radar systems, there are two approaches to locate a target. They are direct and indirect methods. In the direct approach, including the maximum likelihood (ML) estimators [1,6], we calculate the target position directly using the measurements collected by the antennas. These methods are based on two-dimensional search, which requires enormous computational power. While in the indirect approach [1,7–22], we first estimate the range measurements. Each range measurement is the sum of the transmitter-to-target distance and target-to-receiver distance. By solving a set of elliptic equations of the range measurements, we can estimate the target location. In this case, the ML methods can also be employed to locate the target iteratively with an initial position estimate [1,7]. On the other hand, the elliptic equations can be transformed to a set of linear equations. To this end, the commonly used technique is linearization. Afterwards, we can use the least squares (LS) [2,8–10] or weighted least squares (WLS) [11,13,17–20] to estimate the target location. To name a few, in [2], the elliptic equations are linearized into two groups of linear equations first. Then the target location is estimated based on LS. In [11], a quadratically constrained quadratic program (QCQP) problem is obtained after linearization of the range measurements. By solving the QCQP with the WLS concept, unbiased estimation of target position can be achieved.

Moreover, positioning methods can be broadly divided into two groups, including iterative and closed-form approaches. Due to the highly nonconvex and nonlinear nature of positioning problem, iterative approaches require initialization being close to the true target location [1,7,15,16]. In comparison, the closed-form positioning approaches are computationally preferable due to their low complexity and global convergence [11–13,21,22]. . It is worth noting that most algorithms are designed to locate a stationary target. Recently, new advancements [23–32] are made in locating a moving target where both position and velocity of the target need to be estimated.

* Corresponding author.
E-mail address: hcso@ee.cityu.edu.hk (H.C. So).

https://doi.org/10.1016/j.sigpro.2020.107574
0165-1684/

In designing the localization approach, the noise is mostly assumed to be Gaussian distributed. However, in practical situations, there may exist non-line-of-sight (NLOS) propagation and signal interference [33–37]. Hence, noise, as well as, range measurements, may contain outliers, which degrade the performance of LS based algorithms. Another alternative is the maximum correntropy criterion (MCC) [38]. But the complexity of MCC-based formulation is high since it involves convex optimization in every iteration step.

In this paper, we develop two robust target localization algorithms for distributed MIMO radar systems. Our solutions are based on the Lagrange programming neural network (LPNN) framework [15,39–43]. This framework is a universal solver for general differentiable constrained optimization problems with equality constraints, in which the objective function and the constraints are twice differentiable.

We formulate the MIMO localization problem as a constrained optimization problem, namely, minimizing an ℓ_1-norm objective function subject to a set of equality and inequality constraints. Note that the ℓ_1-norm based objective function aims at achieving robustness against outliers. Our initial formulation for the constrained optimization problem contains some inequality constraints which cannot be handled by the LPNN framework. Nevertheless, in the latter stage, we show that those inequality constraints can be removed.

Another important issue is that the LPNN cannot deal with non-differentiable objective functions and constraints. To resolve this issue, we use two strategies. In the first proposed algorithm, we introduce a differentiable approximation function to replace the ℓ_1-norm term in the objective function. In the second algorithm, the internal state concept from the locally competitive algorithm (LCA) [44] is utilized to avoid the non-differentiable issue of the ℓ_1-norm. After the modification, the LPNN approach can be employed for target localization. In addition, we apply the augmented term concept to stabilize the neural dynamics at equilibrium states.

The rest of this paper is organized as follows. The background of MIMO radar target localization, LPNN and LCA are described in Section 2. In Section 3, two target localization algorithms are developed. The local stability of the two algorithms is proved in Section 4. Numerical results for algorithm evaluation and comparison are provided in Section 5. Finally, conclusions are drawn in Section 6.

2. Background

2.1. Notation

We use a lower-case or upper-case letter to represent a scalar, while vectors and matrices are denoted by bold lower-case and upper-case letters, respectively. $\boldsymbol{I}$ denotes an identity matrix with appropriate dimensions. The transpose operator is represented by $(\cdot)^{\mathrm{T}}$. The $\mathrm{diag}(c_1, \ldots, c_n)$ denotes a diagonal matrix whose diagonal components are $c_1, \ldots c_n$. In addition, $\mathbf{1}_{m\times n}$ and $\mathbf{0}_{m\times n}$ represent the $m \times n$ matrix of 1 and $m \times n$ zero matrix, respectively. Other mathematical symbols are defined in their first appearance.

2.2. MIMO Radar localization

A MIMO radar localization system [15,37] consists of M transmitters and N receivers, which is illustrated in Fig. 1 with $M = N = 3$. To locate the target, each transmitter sends out a distinct electromagnetic wave. All these electromagnetic waves are reflected by the target. Afterwards, they are collected by the receivers. That is, each receiver receives and processes M signals jointly.

Let $\boldsymbol{p} = [x, y]^{\mathrm{T}}$ be the unknown position of the target. Let $\boldsymbol{t}_i = [x_i^t, y_i^t]^{\mathrm{T}}$ and $\boldsymbol{r}_j = [x_j^r, y_j^r]^{\mathrm{T}}$ be ith transmitter and jth receiver, respectively, where $i = 1, \ldots, M$ and $j = 1, \ldots, N$. Let d_i^t be the distance from the transmitter $\boldsymbol{t}_i$ to target, and d_j^r be the distance from the target to receiver $\boldsymbol{r}_j$. They are given by

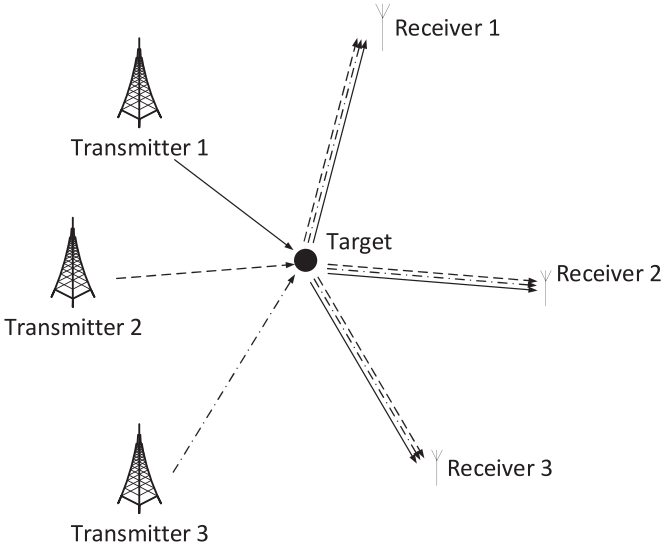

Fig. 1. MIMO radar localization system.

$$d_i^t = \|\boldsymbol{p} - \boldsymbol{t}_i\|_2 = \sqrt{(x_i^t - x)^2 + (y_i^t - y)^2}, \tag{1}$$

$$d_j^r = \|\boldsymbol{p} - \boldsymbol{r}_j\|_2 = \sqrt{(x_j^r - x)^2 + (y_j^r - y)^2}. \tag{2}$$

It should be noticed that we do not know these range values. Instead, we are only able to estimate the overall propagation time $\tau_{i,j}$ from the ith transmitter to the jth receiver. The total distance can be computed from $\tau_{i,j}$, given by

$$d_{i,j} = d_i^t + d_j^r = c\tau_{i,j}, \tag{3}$$

for $i = 1, \ldots, M,\ j = 1, \ldots, N$, where c is the speed of light.

In the practical scenarios, the measured propagation time $\tau_{i,j}$ usually contains noise. Hence we can model the distance as

$$\hat{d}_{i,j} = d_{i,j} + \epsilon_{i,j}, \tag{4}$$

where $\epsilon_{i,j}$ denotes the noise. The aim of target localization is to estimate the position of target $\boldsymbol{p}$ from $\{\boldsymbol{t}_i\}$'s, $\{\boldsymbol{r}_j\}$'s and $\{\hat{d}_{i,j}\}$'s. Conventionally, it is assumed that the noise components $\epsilon_{i,j}$'s follow zero-mean Gaussian distribution. According to the LS [10,15], the problem is formulated as

$$\min_{\boldsymbol{p}} \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\hat{d}_{i,j} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2 - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2\right)^2. \tag{5}$$

Traditional gradient based approaches may not be suitable to minimize the cost function, which is clearly observed in (5) as the gradient vector of the objective function includes factors $(1/\|\boldsymbol{p} - \boldsymbol{t}_i\|_2)$ and $(1/\|\boldsymbol{p} - \boldsymbol{r}_j\|_2)$. Based on (1) and (2), the problem can be rewritten as

$$\min_{\boldsymbol{p}, d_i^t, d_j^r} \quad \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\hat{d}_{i,j} - d_i^t - d_j^r\right)^2, \tag{6a}$$

$$\text{s.t.} \quad d_i^t = \|\boldsymbol{p} - \boldsymbol{t}_i\|_2, \quad i = 1, \ldots, M, \tag{6b}$$

$$d_j^r = \|\boldsymbol{p} - \boldsymbol{r}_j\|_2, \quad j = 1, \ldots, N. \tag{6c}$$

Denote

$$\boldsymbol{d}^t = [d_1^t, \ldots, d_M^t, d_1^t, \ldots, d_M^t, \ldots, d_1^t, \ldots, d_M^t]^{\mathrm{T}},$$

$$\boldsymbol{d}^r = [d_1^r, \ldots, d_1^r, d_2^r, \ldots, d_2^r, \ldots, d_N^r, \ldots, d_N^r]^{\mathrm{T}},$$
$$\hat{\boldsymbol{d}} = [\hat{d}_{1,1}, \ldots, \hat{d}_{M,1}, \hat{d}_{1,2}, \ldots, \hat{d}_{M,2}, \ldots, \hat{d}_{1,N}, \ldots, \hat{d}_{M,N}]^{\mathrm{T}},$$

where they are all $MN \times 1$ vectors. Then (6) can be rewritten as a constrained optimization problem, given by

$$\min_{\boldsymbol{p}, d_i^t, d_j^r} \quad \left\| \hat{\boldsymbol{d}} - \boldsymbol{d}^t - \boldsymbol{d}^r \right\|_2^2, \tag{7a}$$

$$\text{s.t.} \quad d_i^{t\,2} = \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2, \quad i = 1, \ldots, M, \tag{7b}$$

$$d_j^{r\,2} = \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2, \quad j = 1, \ldots, N, \tag{7c}$$

$$d_i^t \geq 0, \quad i = 1, \ldots, M \tag{7d}$$

$$d_i^t \geq 0, \quad j = 1, \ldots, N. \tag{7e}$$

In many real situations, the measured distances may contain outliers. For example, there may have some obstacles in the environment, resulting in NLOS propagation in transmission paths between the transmitters and target and/or target and receivers. In (7), the objective function is an ℓ_2-norm term indicating that it cannot eliminate the influence caused by the outlier measurements.

2.3. Lagrange programming neural network

The LPNN, introduced in [39], is an analog neural network approach. It has the capability to solve the general nonlinear constrained optimization problem, given by

$$\min_{\boldsymbol{z}} \quad f(\boldsymbol{z}) \tag{8a}$$

$$\text{s.t.} \quad \boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{0}_{1\times n}, \tag{8b}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $\boldsymbol{z} = [z_1, \ldots, z_n]^{\mathrm{T}}$ is the collection of the decision variables, and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^m$ $(m < n)$ represents m equality constraints.

In the LPNN framework, we first construct the Lagrangian of (8), given by

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\zeta}) = f(\boldsymbol{z}) + \boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{z}), \tag{9}$$

where $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_m]^{\mathrm{T}}$ is the collection of Lagrange multipliers.

In this LPNN, there are n variable neurons and m Lagrangian neurons. The n variable neurons are used to hold the values of the n decision variables. The m Lagrangian neurons are used to hold the values of the m Lagrange multipliers. The dynamics of the neurons can be defined as

$$\frac{1}{\epsilon}\frac{d\boldsymbol{z}}{dt} = -\frac{\partial \mathcal{L}(\boldsymbol{z}, \boldsymbol{\zeta})}{\partial \boldsymbol{z}}, \tag{10a}$$

$$\frac{1}{\epsilon}\frac{d\boldsymbol{\zeta}}{dt} = \frac{\partial \mathcal{L}(\boldsymbol{z}, \boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}, \tag{10b}$$

where ϵ is the characteristic time constant depending on the impedance of the neural circuit. For simplicity, we set ϵ equal to one. The dynamic in (10a) aims at seeking for a solution with the minimum objective value. In addition, the dynamics in (10b) aims at seeking for a solution that satisfies the constraints. The state of the network will reach a stable state under some mild conditions [15,39,43]. After the dynamics settle down at an equilibrium point, the output of the neurons is the solution of the optimization problem. Obviously, both f and $\boldsymbol{g}$ should be differentiable. It should be noticed that if they are not differentiable, we cannot define or implement the dynamics in (10).

2.4. Locally competitive algorithm

The LCA concept [44] is also an analog neural network model that is used to solve the following unconstrained optimization problem:

$$L_{\mathrm{lca}} = \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{z}\|_2^2 + \tau\|\boldsymbol{z}\|_1, \tag{11}$$

where $\boldsymbol{z} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}(m < n)$. To construct the dynamics of LCA, we need to calculate the gradient of L_{lca} with respect to $\boldsymbol{z}$. However, $\tau\|\boldsymbol{z}\|_1$ is non-differentiable, and the gradient of (11) is

$$\partial_{\boldsymbol{z}} L_{\mathrm{lca}} = -\boldsymbol{\Phi}(\boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{z}) + \tau\,\partial\|\boldsymbol{z}\|_1, \tag{12}$$

where $\partial\|\boldsymbol{z}\|_1$ denotes the sub-differential of $\|\boldsymbol{z}\|_1$. According to the definition of sub-differential, at the non-differentiable point, the sub-differential is equal to a set.[1] To handle this issue, the LCA concept introduces an internal state vector $\boldsymbol{u} = [u_1, \ldots, u_n]^{\mathrm{T}}$ and defines a relationship between $\boldsymbol{u}$ and $\boldsymbol{z}$ as

$$z_i = T_\tau(u_i) = \begin{cases} 0, & |u_i| \leq \tau, \\ u_i - \tau\,\mathrm{sign}(u_i), & |u_i| > \tau, \end{cases} \tag{13}$$

where $i = 1, \ldots, n$. In the LCA, $\boldsymbol{z}$ and $\boldsymbol{u}$ are known as the output state variable vector and the internal state variable vector, respectively, and τ denotes the threshold of the function. Furthermore, from (13), we can deduce that

$$\boldsymbol{u} - \boldsymbol{z} \in \tau\,\partial\|\boldsymbol{z}\|_1. \tag{14}$$

Hence, LCA defines its dynamics on $\boldsymbol{u}$ rather than $\boldsymbol{z}$ as

$$\frac{d\boldsymbol{u}}{dt} = -\partial_{\boldsymbol{z}} L_{\mathrm{lca}} = -\boldsymbol{u} + \boldsymbol{z} + \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{z}). \tag{15}$$

It is worth noting that if the dynamics of $\boldsymbol{z}$ are used directly, we need to calculate $\partial\|\boldsymbol{z}\|_1$ which is equal to a set. Therefore, the LCA uses $d\boldsymbol{u}/dt$ to define the dynamics rather than $d\boldsymbol{z}/dt$.

3. Development of proposed algorithms

3.1. Problem formulation

Most algorithms for target localization assume that the noise in (7) is Gaussian distributed. Thus the positioning problem is solved by minimizing the ℓ_2-norm of the range measurement errors. However, in real situations, the range measurements may contain outliers due to signal interference or NLOS propagation. It is well known that, compared with the ℓ_2-norm, the ℓ_1-norm is less sensitive to outliers. Hence in our proposed formulation, the MIMO radar localization problem is modified as:

$$\min_{\boldsymbol{p}, d_i^t, d_j^r} \quad \left\| \hat{\boldsymbol{d}} - \boldsymbol{d}^t - \boldsymbol{d}^r \right\|_1, \tag{16a}$$

$$\text{s.t.} \quad d_i^{t\,2} = \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2, \quad i = 1, \ldots, M, \tag{16b}$$

$$d_j^{r\,2} = \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2, \quad j = 1, \ldots, N, \tag{16c}$$

$$d_i^t \geq 0, \quad i = 1, \ldots, M, \tag{16d}$$

$$d_i^t \geq 0, \quad j = 1, \ldots, N. \tag{16e}$$

It is clear that in (16) the formulation includes $M + N$ equality constraints and $M + N$ inequality constraints.

To use the LPNN framework, we need to address two issues in our formulation. First, the LPNN framework cannot handle problems with inequality constraints, but the formulation

[1] For the absolute function $|z|$, the sub-differential $\partial|z|$ at $z = 0$ is equal to $[-1, 1]$.

in (16d) and (16e) includes inequality constraints. Secondly, the LPNN approach requires that the objective function and constraints should be twice differentiable. Obviously, due to the ℓ_1-norm term, the objective function in (16a) does not satisfy this requirement.

From (3) and (4), we know that

$$\hat{d}_{i,j} = d_i^t + d_j^r + \epsilon_{i,j}. \tag{17}$$

For small noise level (the magnitude of $\epsilon_{i,j}$ is small) and NLOS noise ($\epsilon_{i,j}$ is positive), the measurement value $\hat{d}_{i,j}$ is greater than d_i^t and d_i^t. Hence, it is reasonable to assume that $\hat{d}_{i,j} > d_i^{t*}$ and $\hat{d}_{i,j} > d_j^{r*}$, for i and j, where d_i^{t*}'s and d_j^{t*}'s are the optimal values. Based on the above assumption, we derive Theorem 1, which indicates that the inequality constraints in (16d) and (16e) can be removed.

Theorem 1. *From the signal model (4), we can assume that $\hat{d}_{i,j} > d_i^{t*}$ and $\hat{d}_{i,j} > d_j^{r*}$, where d_i^{t*}'s and d_j^{t*}'s are the optimal values. The optimization problem in (16) is then equivalent to*

$$\min_{\boldsymbol{p}, d_i^t, d_j^r} \left\| \hat{\boldsymbol{d}} - \boldsymbol{d}^t - \boldsymbol{d}^r \right\|_1, \tag{18a}$$

$$\text{s.t. } d_i^{t2} = \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2, \quad i = 1, \ldots, M, \tag{18b}$$

$$d_j^{r2} = \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2, \quad j = 1, \ldots, N. \tag{18c}$$

Proof. Suppose that $[\boldsymbol{p}^{*\mathrm{T}}, d_1^{t*}, \ldots, d_M^{t*}, d_1^{r*}, \ldots, d_N^{r*}]^{\mathrm{T}}$ is the optimal solution of (18). According to (18b) and (18c), we have that $[\boldsymbol{p}^{*\mathrm{T}}, |d_1^{t*}|, \ldots, |d_M^{t*}|, |d_1^{r*}|, \ldots, |d_N^{r*}|]^{\mathrm{T}}$ is a feasible solution.

Since $\hat{d}_{i,j}$'s are range measurements, we have $\hat{d}_{i,j} \geq 0$. From the model defined in (3), we have

$$\hat{d}_{i,j} = d_i^t + d_j^r + \epsilon_{i,j}. \tag{19}$$

Hence it can be assumed that $\hat{d}_{i,j} > d_i^{t*}$ and $\hat{d}_{i,j} > d_j^{r*}$, since $\hat{d}_{i,j} \geq 0$, $d_i^t > 0$, and $d_j^r > 0$. Then we have the following inequality:

$$\left| \hat{d}_{i,j} - d_i^{t*} - d_j^{r*} \right| \geq \left| \hat{d}_{i,j} - |d_i^{t*}| - |d_j^{r*}| \right|, \tag{20}$$

for $\forall i \in \{1, \ldots, M\}$ and $\forall j \in \{1, \ldots, N\}$.

To validate (20), we check the sign of $\left| \hat{d}_{i,j} - d_i^{t*} - d_j^{r*} \right|^2 - \left| \hat{d}_{i,j} - |d_i^{t*}| - |d_j^{r*}| \right|^2$ under the assumption $\hat{d}_{i,j} > d_i^{t*}$ and $\hat{d}_{i,j} > d_j^{r*}$. After rearrangement, let

$$\begin{aligned} R &= \left| \hat{d}_{i,j} - d_i^{t*} - d_j^{r*} \right|^2 - \left| \hat{d}_{i,j} - |d_i^{t*}| - |d_j^{r*}| \right|^2 \\ &= 2\hat{d}_{i,j}(|d_i^{t*}| + |d_j^{r*}| - d_i^{t*} - d_j^{r*}) + 2(d_i^{t*} d_j^{r*} - |d_i^{t*}||d_j^{r*}|). \end{aligned} \tag{21}$$

There are four cases.

- Case (a) $d_i^{t*} \geq 0$, $d_j^{r*} \geq 0$:
  In this case, it is obvious that
  $$R = 2\hat{d}_{i,j}(d_i^{t*} + d_j^{r*} - d_i^{t*} - d_j^{r*}) + 2(d_i^{t*} d_j^{r*} - d_i^{t*} d_j^{r*}) = 0. \tag{22}$$
- Case (b) $d_i^{t*} > 0$, $d_j^{r*} < 0$:
  $$\begin{aligned} R &= 2\hat{d}_{i,j}(d_i^{t*} - d_j^{r*} - d_i^{t*} - d_j^{r*}) + 2(d_i^{t*} d_j^{r*} + d_i^{t*} d_j^{r*}) \\ &= -4d_j^{r*}(\hat{d}_{i,j} - d_i^{t*}). \end{aligned} \tag{23}$$
  According to our assumption, $\hat{d}_{i,j} - d_i^{t*} > 0$. Together with $-4d_j^{r*} > 0$, we have $R > 0$.
- Case (c) $d_i^{t*} < 0$, $d_j^{r*} > 0$:
  $$\begin{aligned} R &= 2\hat{d}_{i,j}(-d_i^{t*} + d_j^{r*} - d_i^{t*} - d_j^{r*}) + 2(d_i^{t*} d_j^{r*} + d_i^{t*} d_j^{r*}) \\ &= -4d_i^{t*}(\hat{d}_{i,j} - d_j^{r*}). \end{aligned} \tag{24}$$
  Similar to Case (b), $\hat{d}_{i,j} - d_j^{r*} > 0$ and $-4d_i^{t*} > 0$. Hence, $R > 0$.
- Case (d) $d_i^{t*} < 0$, $d_j^{r*} < 0$:
  $$\begin{aligned} R &= 2\hat{d}_{i,j}(-d_i^{t*} - d_j^{r*} - d_i^{t*} - d_j^{r*}) \\ &= -4\hat{d}_{i,j}(d_j^{t*} + d_i^{r*}). \end{aligned} \tag{25}$$
  Since $d_j^{t*} + d_i^{r*} < 0$ and $\hat{d}_{i,j} \geq 0$, we have $R = -4\hat{d}_{i,j}(d_j^{t*} + d_i^{r*}) > 0$.

To sum up, from (22)–(25), we conclude that (20) holds, and obtain

$$\sum_{i=1}^{M}\sum_{j=1}^{N} \left| \hat{d}_{i,j} - d_i^{t*} - d_j^{r*} \right| \geq \sum_{i=1}^{M}\sum_{j=1}^{N} \left| \hat{d}_{i,j} - |d_i^{t*}| - |d_j^{r*}| \right|. \tag{26}$$

Inequality (26) implies that the optimal objective value is greater than or equal to the objective value achieved by the feasible solution $[\boldsymbol{p}^{*\mathrm{T}}, |d_1^{t*}|, \ldots, |d_M^{t*}|, |d_1^{r*}|, \ldots, |d_N^{r*}|]^{\mathrm{T}}$. Since $[\boldsymbol{p}^{*\mathrm{T}}, d_1^{t*}, \ldots, d_M^{t*}, d_1^{r*}, \ldots, d_N^{r*}]^{\mathrm{T}}$ is the optimal solution, the equality in (26) must hold. That means, $d_i^{t*} = |d_i^{t*}|$ for $\forall i \in \{1, \ldots, M\}$, and $d_j^{r*} = |d_j^{r*}|$ for $\forall j \in \{1, \ldots, N\}$. In other words, $d_i^{t*} \geq 0$ for $\forall i \in \{1, \ldots, M\}$, and $d_j^{r*} \geq 0$ for $\forall j \in \{1, \ldots, N\}$. Therefore, we can remove the inequality constraints in (16). The proof is completed. □

Due to the non-differentiability of the ℓ_1-norm term in (18), the gradient of the Lagrangian is not well-defined. To calculate the gradient of ℓ_1-norm at the non-differentiable points, we utilize two strategies to resolve the issue. Accordingly, they are developed in the following subsection.

3.2. LPNN for robust MIMO radar localization

Algorithm 1: ℓ_1-norm LPNN

In the first algorithm, we consider using a differentiable ℓ_1-norm approximation function [45], given by

$$g(x) = \frac{1}{\gamma} \ln(\cosh(\gamma x)), \tag{27}$$

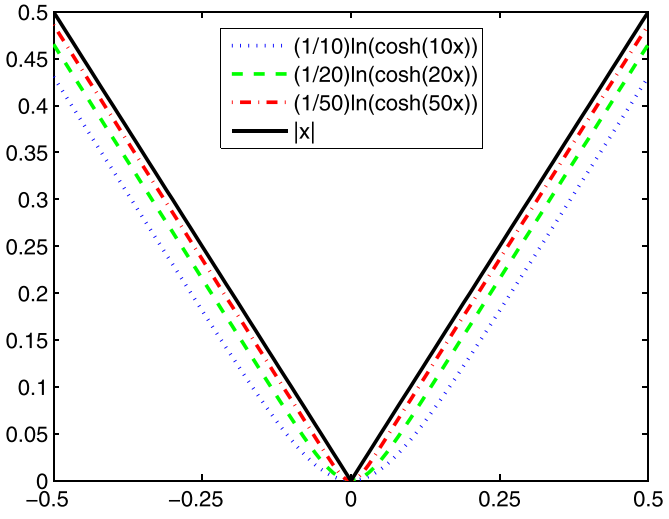

Fig. 2. $(1/\gamma)\ln(\cosh(\gamma x))$.

where $\gamma > 1$ is a constant. Fig. 2 shows the shapes of $g(x)$ with several γ's. It is observed that $|x| \approx \frac{1}{\gamma}\ln(\cosh(\gamma x))$ for a sufficiently large γ.

To define the neural dynamics, we only need to implement the gradient of $g(x)$. It is worth noting that the gradient of $g(x)$ with respect to x is equal to the hyperbolic tangent function $\tanh(\gamma x)$ which is a frequently used activation function in neural networks. With this approximate function, (18) can be recast as

$$\min_{\boldsymbol{x}} \quad \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{\gamma}\ln(\cosh(\gamma(\hat{d}_{i,j}-d_i^t-d_j^r))), \tag{28a}$$

$$\text{s.t.} \quad {d_i^t}^2 = \|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2, \quad i=1,\ldots,M, \tag{28b}$$

$${d_j^r}^2 = \|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2, \quad j=1,\ldots,N, \tag{28c}$$

where $\boldsymbol{x} = [\boldsymbol{p}^{\mathrm{T}}, d_1^t,\ldots,d_M^t, d_1^r,\ldots,d_N^r]^{\mathrm{T}}$.

To improve the convexity and stability [39], we include an augmented term, given by

$$\frac{C}{2}\left(\sum_{i=1}^{M}\left({d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2\right)^2+\sum_{j=1}^{N}\left({d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2\right)^2\right) \tag{29}$$

into (28a), where $C > 0$ is a constant. For a sufficiently large C [39], the augmented term can improve the convexity of (28a) and accelerate the convergence of the dynamics. In practical situations, C can be selected empirically and the appropriateness can be reflected by the stability of the dynamics. Very large/small C would lead to instability of the dynamics, while a proper C would lead to stable and convergent dynamics.

In the LPNN framework, the Lagrangian neurons are used to guide the state into the feasible region. With the aid of the augmented term, this process can be accelerated. Because the impact of the augmented term is to introduce a penalty into the cost function, it will penalize the state of network for any unfulfillment of constraints. Normally, the state of the network is initialized with a random point. The constraints may be significantly violated. With a large C, the neural state is forced to approach the feasible region rapidly. For any point close enough to the feasible region, the augmented term will have little influence on the network state.

When the network reaches a stable state, the augmented term equals 0. Hence, the augmented term will not influence the optimal solution. It should also be noticed that without the augmented term, the resultant LPNN dynamics may not be stable and convergent. According to [39], with sufficient large C, around a local minimum, the Hessian of the objective function is positive definite, i.e., the local minimum is stable. For the details of the augmented term, readers are referred to [39].

With the augmented term (29), the Lagrangian of (28) is

$$\begin{aligned}\mathcal{L}(\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\lambda}) &= \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{\gamma}\ln(\cosh(\gamma(\hat{d}_{i,j}-d_i^t-d_j^r)))\\ &+\sum_{i=1}^{M}\beta_i\left({d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2\right)+\sum_{j=1}^{N}\lambda_j\left({d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2\right)\\ &+\frac{C}{2}\left(\sum_{i=1}^{M}\left({d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2\right)^2+\sum_{j=1}^{N}\left({d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2\right)^2\right)\end{aligned} \tag{30}$$

where $\boldsymbol{x}$ is the decision variable vector, $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_M]^{\mathrm{T}}$ and $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_N]^{\mathrm{T}}$ are Lagrangian variable vectors.

Then the LPNN can be utilized to handle the corresponding analog circuit. According to the LPNN concepts given in (10), the dynamics of our formulation become:

$$\begin{aligned}\frac{dd_i^t}{dt} &= -\frac{\partial\mathcal{L}(\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial d_i^t}\\ &= \sum_{j=1}^{N}\tanh(\gamma(\hat{d}_{i,j}-d_i^t-d_j^r))-2\beta_i d_i^t-2Cd_i^t\left({d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2\right),\end{aligned} \tag{31}$$

$$\begin{aligned}\frac{dd_j^r}{dt} &= -\frac{\partial\mathcal{L}(\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial d_j^r}\\ &= \sum_{i=1}^{M}\tanh(\gamma(\hat{d}_{i,j}-d_i^t-d_j^r))\\ &\quad -2\lambda_j d_j^r-2Cd_j^r\left({d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2\right),\end{aligned} \tag{32}$$

$$\begin{aligned}\frac{d\boldsymbol{p}}{dt} &= -\frac{\partial\mathcal{L}(\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial\boldsymbol{p}}\\ &= 2\sum_{i=1}^{M}\beta_i(\boldsymbol{p}-\boldsymbol{t}_i)+2\sum_{j=1}^{N}\lambda_j(\boldsymbol{p}-\boldsymbol{r}_j)\\ &\quad +2C\left(\sum_{i=1}^{M}(\boldsymbol{p}-\boldsymbol{t}_i)\left({d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2\right)\right.\\ &\quad \left.+\sum_{j=1}^{N}(\boldsymbol{p}-\boldsymbol{r}_j)\left({d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2\right)\right),\end{aligned} \tag{33}$$

$$\frac{d\beta_i}{dt} = \frac{\partial\mathcal{L}(\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial\beta_i} = {d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2, \tag{34}$$

$$\frac{d\lambda_j}{dt}, = \frac{\partial\mathcal{L}(\boldsymbol{x},\boldsymbol{\beta},\boldsymbol{\lambda})}{\partial\lambda_j} = {d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2., \tag{35}$$

where $i=1,\ldots,M$ and $j=1,\ldots,N$. Eqs. (31)–(33) are used for seeking the minimum objective value, while (34)–(35) constrain the equilibrium point into the feasible region. In this formulation, there are $M+N+2$ variable neurons to hold the $M+N+2$ decision variables and $M+N$ Lagrangian neurons for the $M+N$ Lagrangian variables.

Algorithm 2: ℓ_1-norm LPNN-LCA

In this algorithm, we utilize the LCA technique to solve the subdifferentiable problem in (18). We introduce a dummy variable $\boldsymbol{z}$ and rewrite (18) as

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{z}\|_1, \tag{36a}$$

$$\text{s.t.} \quad \boldsymbol{z} = \hat{\boldsymbol{d}}-\boldsymbol{d}^t-\boldsymbol{d}^r, \tag{36b}$$

$${d_i^t}^2 = \|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2, \quad i=1,\ldots,M, \tag{36c}$$

$${d_j^r}^2 = \|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2, \quad j=1,\ldots,N, \tag{36d}$$

where $\boldsymbol{x} = [\boldsymbol{p}^{\mathrm{T}}, \boldsymbol{z}^{\mathrm{T}}, d_1^t,\ldots,d_M^t, d_1^r,\ldots,d_N^r]^{\mathrm{T}}$ and $\boldsymbol{z} = [z_1,\ldots,z_{MN}]^{\mathrm{T}}$.

For this formulation, we also introduce an augmented term into the objective function to make the system more stable. Thus, (36) can be modified as:

$$\begin{aligned}\min_{\boldsymbol{x}} \quad & \|\boldsymbol{z}\|_1+\frac{C}{2}\|\boldsymbol{z}-\hat{\boldsymbol{d}}+\boldsymbol{d}^t+\boldsymbol{d}^r\|_2^2\\ &+\frac{C}{2}\sum_{i=1}^{M}\left({d_i^t}^2-\|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2\right)^2\\ &+\frac{C}{2}\sum_{j=1}^{N}\left({d_j^r}^2-\|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2\right)^2,\end{aligned} \tag{37a}$$

$$\text{s.t.} \quad \boldsymbol{z} = \hat{\boldsymbol{d}} - \boldsymbol{d}^t - \boldsymbol{d}^r, \tag{37b}$$

$$d_i^{t\,2} = \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2, \quad i = 1, \ldots, M, \tag{37c}$$

$$d_j^{r\,2} = \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2, \quad j = 1, \ldots, N, \tag{37d}$$

where $C > 0$ is a constant.

The Lagrangian of (37) is given by

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \\
&= \|\boldsymbol{z}\|_1 + \boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{z} - \hat{\boldsymbol{d}} + \boldsymbol{d}^t + \boldsymbol{d}^r) \\
&\quad + \sum_{i=1}^{M} \beta_i \left(d_i^{t\,2} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2\right) + \sum_{j=1}^{N} \lambda_j \left(d_j^{r\,2} - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2\right) \\
&\quad + \frac{C}{2}\left(\sum_{i=1}^{M}\left(d_i^{t\,2} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2\right)^2 + \sum_{j=1}^{N}\left(d_j^{r\,2} - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2\right)^2\right) \\
&\quad + \frac{C}{2}\|\boldsymbol{z} - \hat{\boldsymbol{d}} + \boldsymbol{d}^t + \boldsymbol{d}^r\|_2^2
\end{aligned} \tag{38}$$

According to the LCA, we introduce internal variables $\boldsymbol{u} = [u_1, \ldots, u_{MN}]^{\mathrm{T}}$ for $\boldsymbol{z}$. The relationship between $\boldsymbol{u}$ and $\boldsymbol{z}$ is given by

$$z_l = T_\tau(u_l) = \begin{cases} 0, & |u_l| \le \tau, \\ u_l - \tau \operatorname{sign}(u_l), & |u_l| > \tau, \end{cases} \tag{39}$$

where $l = 1, \ldots, MN$. For simplicity, we set τ to 1 in our formulation. Note that according to the definition of sub-differential, we have

$$\boldsymbol{u} - \boldsymbol{z} \in \tau \partial \|\boldsymbol{z}\|_1. \tag{40}$$

Combining the dynamics of the LPNN in (10) with the concept of LCA in (40), we deduce that:

$$\begin{aligned}
\frac{d\boldsymbol{u}}{dt} &= -\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \\
&= -\boldsymbol{u} + \boldsymbol{z} - \boldsymbol{\alpha} - C\left(\boldsymbol{z} - \hat{\boldsymbol{d}} + \boldsymbol{d}^t + \boldsymbol{d}^r\right),
\end{aligned} \tag{41}$$

$$\begin{aligned}
\frac{dd_i^t}{dt} &= -\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial d_i^t} \\
&= -\sum_{j=1}^{N} \alpha_{i+(j-1)M} - 2\beta_i d_i^t - C\sum_{j=1}^{N}\left(z_{i+(j-1)M} - \hat{d}_{i,j} + d_i^t + d_j^r\right) \\
&\quad -2C d_i^t\left(d_i^{t\,2} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2\right),
\end{aligned} \tag{42}$$

$$\begin{aligned}
\frac{dd_j^r}{dt} &= -\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial d_j^r} \\
&= -\sum_{i=1}^{M} \alpha_{i+(j-1)M} - 2\lambda_j d_j^r - C\sum_{i=1}^{M}\left(z_{i+(j-1)M} - \hat{d}_{i,j} + d_i^t + d_j^r\right) \\
&\quad -2C d_j^r\left(d_j^{r\,2} - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2\right),
\end{aligned} \tag{43}$$

$$\begin{aligned}
\frac{d\boldsymbol{p}}{dt} &= -\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial \boldsymbol{p}} \\
&= \sum_{i=1}^{M} 2\beta_i(\boldsymbol{p} - \boldsymbol{t}_i) + \sum_{j=1}^{N} 2\lambda_j\left(\boldsymbol{p} - \boldsymbol{r}_j\right) \\
&\quad +2C\sum_{i=1}^{M}(\boldsymbol{p} - \boldsymbol{t}_i)\left(d_i^{t\,2} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2\right) \\
&\quad +2C\sum_{j=1}^{N}\left(\boldsymbol{p} - \boldsymbol{r}_j\right)\left(d_j^{r\,2} - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2\right),
\end{aligned} \tag{44}$$

$$\begin{aligned}
\frac{d\boldsymbol{\alpha}}{dt} &= \frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial \boldsymbol{\alpha}} \\
&= \boldsymbol{z} - \hat{\boldsymbol{d}} + \boldsymbol{d}^t + \boldsymbol{d}^r,
\end{aligned} \tag{45}$$

$$\begin{aligned}
\frac{d\beta_i}{dt} &= \frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial \beta_i} \\
&= d_i^{t\,2} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2,
\end{aligned} \tag{46}$$

$$\begin{aligned}
\frac{d\lambda_j}{dt} &= \frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda})}{\partial \lambda_j} \\
&= d_j^{r\,2} - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2.
\end{aligned} \tag{47}$$

In this formulation, there are $MN + M + N + 2$ variable neurons to hold the $MM + M + N + 2$ decision variables and $MN + M + N$ Lagrangian neurons for the $MN + M + N$ Lagrangian variables.

Two typical examples of the dynamics of the proposed algorithms are given in Figs. 3 and 4. The setting is described in Section 5.B. It is seen that both networks can settle down in around 50 characteristic times. According to the analysis in Section 4, our proposed algorithms are locally stable. Hence, starting from a random state, the dynamics fluctuate during the transient state and finally reach a stable state.

4. Local stability of proposed algorithms

In the analog network approach, one important aspect is the stability of the proposed algorithms. Since the constraints are not convex, we cannot investigate the global stability. Instead, we show the local stability. Local stability means that a local minimum point of the constrained optimization problem should be a stable point of the dynamics. Note that if a local minimum is not a stable point, the network will not converge to this minimum point. According to Theorem 1 in [39], local stability is guaranteed under two sufficient conditions. They are

- The convexity of the Lagrangian, i.e., the Hessian matrix of the Lagrangian (with respect to the decision variables) at a local minimum point should be positive definite.
- The gradient vectors of the constraints with respect to the decision variable vector at the local minimum point should be linearly independent.

For our proposed algorithms, the first condition has been achieved by introducing the augmented terms. That means, with the use of a sufficiently large C, the Hessian matrix is positive definite under mild conditions [15,39,41–43]. Then we only need to show that the gradient vectors of the constraints at the local minimum are linear independent.

4.1. ℓ_1-norm LPNN algorithm

For the ℓ_1-norm LPNN algorithm, let $\boldsymbol{x}$ be the collection of all decision variables, i.e., $\boldsymbol{x} = \left[\boldsymbol{p}^{\mathrm{T}}, d_1^t, \ldots, d_M^t, d_1^r, \ldots, d_N^r\right]^{\mathrm{T}}$. Let $\{\boldsymbol{x}^*, \boldsymbol{\beta}^*, \boldsymbol{\lambda}^*\}$ be a local minimum of the constrained optimization problem. The second condition is that the gradient vectors of the constraints with respect to $\boldsymbol{x}$ should be linearly independent at $\boldsymbol{x}^*$. In our case, there are $M + N$ constraints as stated in (28b) and (28c). Denote h_i as the ith constraint. The constraints can be rearranged as

$$h_i(\boldsymbol{x}) = d_i^{t\,2} - \|\boldsymbol{p} - \boldsymbol{t}_i\|_2^2, \tag{48}$$

$$h_{M+j}(\boldsymbol{x}) = d_j^{r\,2} - \|\boldsymbol{p} - \boldsymbol{r}_j\|_2^2, \tag{49}$$

where $i = 1, \ldots, M$ and $j = 1, \ldots, N$. The gradient vectors of these constraints at $\boldsymbol{x}^*$ are given by

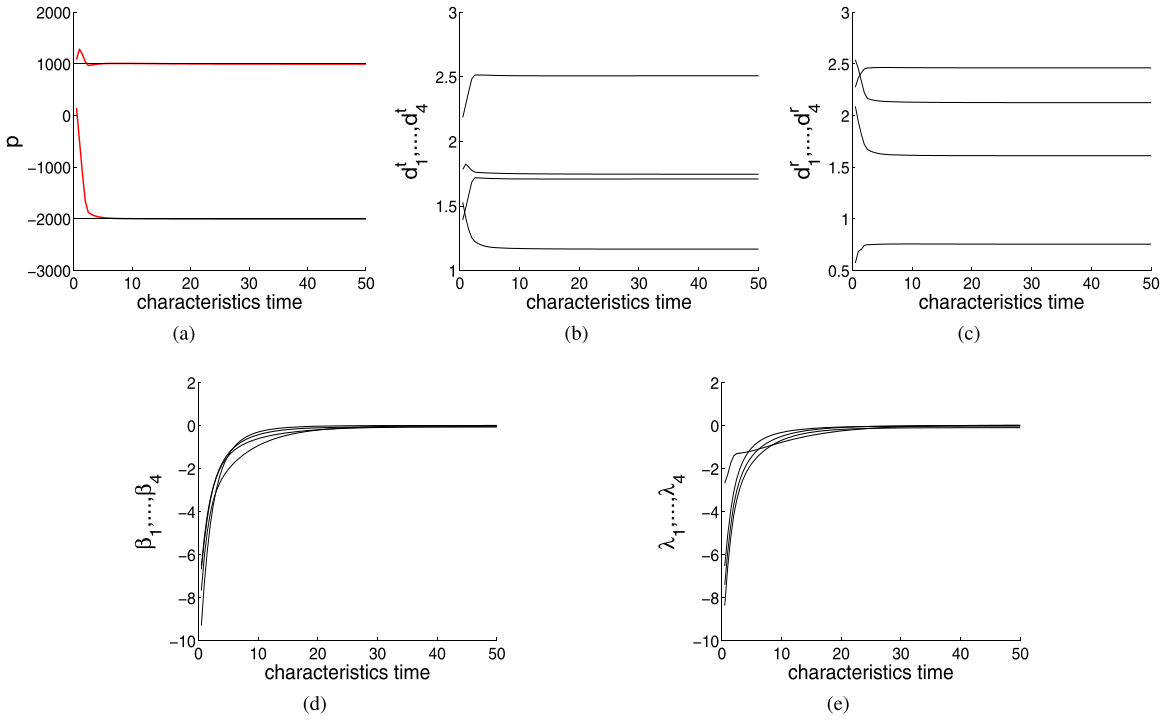

Fig. 3. Dynamics of estimated parameters of ℓ_1-norm LPNN approach when the variance of Gaussian noise is 100 (m^2). The true target position is $[-2000, 1000]^T$. (a) $\boldsymbol{p}$; (b) $d_1^t,\ldots,d_4^t$; (c) $d_1^r,\ldots,d_4^r$; (d) $\beta_1,\ldots,\beta_4$; (e) $\lambda_1,\ldots,\lambda_4$.

$$\left.\frac{\partial h_i(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = 2\Big[(\boldsymbol{t}_i-\boldsymbol{p}^*)^{\mathrm{T}}, \mathbf{0}_{1\times(i-1)}, d_i^{t*}, \mathbf{0}_{1\times(M-i)}, \mathbf{0}_{1\times N}\Big]^{\mathrm{T}}, \quad i=1,\ldots,M, \tag{50}$$

$$\left.\frac{\partial h_{M+j}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = 2\Big[(\boldsymbol{r}_j-\boldsymbol{p}^*)^{\mathrm{T}}, \mathbf{0}_{1\times M}, \mathbf{0}_{1\times(j-1)}, d_j^{r*}, \mathbf{0}_{1\times(N-j)}\Big]^{\mathrm{T}}, \quad j=1,\ldots,N. \tag{51}$$

From (50) and (51), we can obtain the following matrix

$$\left[\tfrac{\partial h_1(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\ldots,\tfrac{\partial h_M(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\tfrac{\partial h_{M+1}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\ldots,\tfrac{\partial h_{M+N}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}}\right] = \left[\begin{array}{c|c} 2[\boldsymbol{t}_1-\boldsymbol{p}^*,\ldots,\boldsymbol{t}_M-\boldsymbol{p}^*] & 2[\boldsymbol{r}_1-\boldsymbol{p}^*,\ldots,\boldsymbol{r}_N-\boldsymbol{p}^*] \\ \hline 2\,\mathrm{diag}(d_1^{t*},\ldots,d_M^{t*}) & \mathbf{0}_{M\times N} \\ \hline \mathbf{0}_{N\times M} & 2\,\mathrm{diag}(d_1^{r*},\ldots,d_N^{r*}) \end{array}\right]. \tag{52}$$

In general, the transmitter and receiver positions are different from the target position. That is, all $\{d_i^{t*}\}$ and $\{d_j^{r*}\}$ are not equal to 0. Thus, it is clear that the gradient vectors of equalities (48) and (49) are linear independent at $\boldsymbol{x}^*$. Hence the gradient vectors of the constraints of ℓ_1-norm LPNN algorithm satisfy the linear independence condition. The dynamics of the ℓ_1-norm LPNN algorithm around a minimum point are locally stable.

4.2. ℓ_1-norm LPNN-LCA algorithm

For the ℓ_1-norm LPNN-LCA algorithm, the proof is similar to that of the ℓ_1-norm LPNN algorithm. Let $\boldsymbol{x} = \left[\boldsymbol{p}^{\mathrm{T}}, \boldsymbol{u}^{\mathrm{T}}, d_1^t,\ldots,d_M^t, d_1^r,\ldots,d_N^r\right]^{\mathrm{T}}$ be the collection of decision variables. Let $\{\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\lambda}^*\}$ be a local minimum of the problem. We need to show that, at the minimum point $\boldsymbol{x}^*$, the gradient vectors of constraints with respect to $\boldsymbol{x}$ are linearly independent. In the second algorithm, we have $MN+M+N$ constraints, namely,

$$h_{i+(j-1)M}(\boldsymbol{x}) = z_{i+(j-1)M} - \hat{d}_{i,j} + d_i^t + d_j^r, \tag{53}$$

$$h_{MN+i}(\boldsymbol{x}) = d_i^{t\,2} - \|\boldsymbol{p}-\boldsymbol{t}_i\|_2^2, \tag{54}$$

$$h_{MN+M+j}(\boldsymbol{x}) = d_j^{r2} - \|\boldsymbol{p}-\boldsymbol{r}_j\|_2^2, \tag{55}$$

where $i=1,\ldots,M$ and $j=1,\ldots,N$. We assume that $z_i \neq 0$, i.e., all data points are influenced by noise. Hence according to (13), we have $\frac{\partial z_l}{\partial u_l}=1$ for $\forall l=1,\ldots,MN$. Therefore, we obtain $\frac{\partial h_l}{\partial u_l} = \frac{\partial h_l}{\partial z_l}\frac{\partial z_l}{\partial u_l}=1$. Then the gradient vectors of the constraints (53) are given by

$$\left.\frac{\partial h_{i+(j-1)M}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = \Big[\mathbf{0}_{1\times 2}, \mathbf{0}_{1\times(j-1)M}, \mathbf{0}_{1\times(i-1)}, 1, \mathbf{0}_{1\times(M-i)}, \mathbf{0}_{1\times M(N-j)}, \mathbf{0}_{1\times(i-1)}, 1, \mathbf{0}_{1\times(M-i)}, \mathbf{0}_{1\times(j-1)}, 1, \mathbf{0}_{1\times(N-j)}\Big]^{\mathrm{T}} \text{ for } i=1,\ldots,M \text{ and } j=1,\ldots,N. \tag{56}$$

In addition, the gradient vectors of the constraints (54) and (55) are given by:

$$\left.\frac{\partial h_{MN+i}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = 2\Big[(\boldsymbol{t}_i-\boldsymbol{p}^*)^{\mathrm{T}}, \mathbf{0}_{1\times MN}, \mathbf{0}_{1\times(i-1)}, d_i^{t*}, \mathbf{0}_{1\times(M-i)}, \mathbf{0}_{1\times N}\Big]^{\mathrm{T}} \text{ for } i=1,\ldots,M. \tag{57}$$

$$\left.\frac{\partial h_{MN+M+j}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = 2\Big[(\boldsymbol{r}_j-\boldsymbol{p}^*)^{\mathrm{T}}, \mathbf{0}_{1\times MN}, \mathbf{0}_{1\times M}, \mathbf{0}_{1\times(j-1)}, d_j^{t*}, \mathbf{0}_{1\times(N-j)}\Big]^{\mathrm{T}} \text{ for } j=1,\ldots,N. \tag{58}$$

From (56), (57), and (58), we obtain (59).

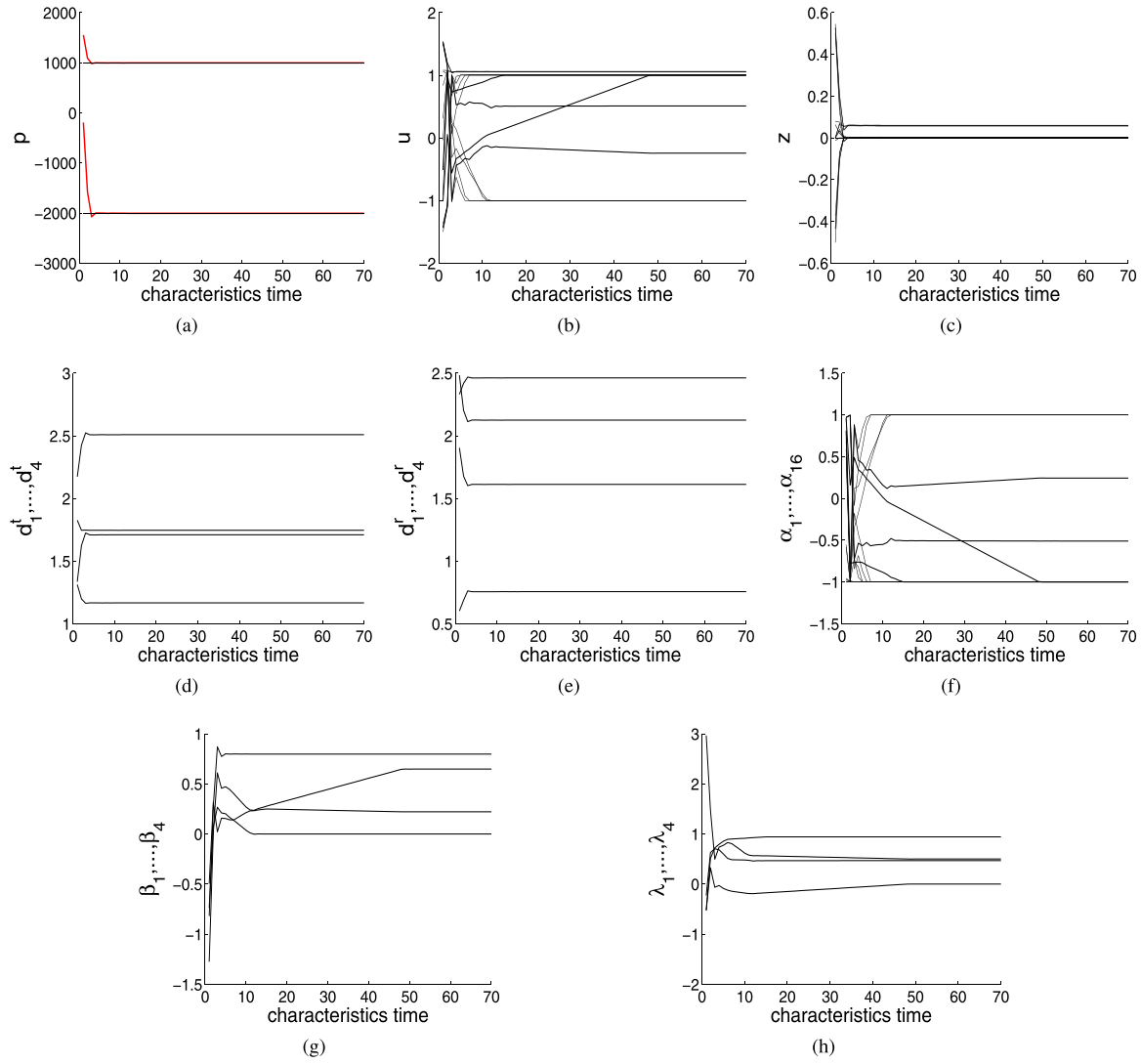

Fig. 4. Dynamics of estimated parameters of ℓ_1-norm LPNN-LCA approach when the variance of Gaussian noise is 100 (m^2). The true target position is at $[-2000, 1000]^T$ (m). (a) $\boldsymbol{p}$; (b) $\boldsymbol{u}$; (c) $\boldsymbol{z}$; (d) $d_1^t,\ldots,d_4^t$; (e) $d_1^r,\ldots,d_4^r$; (f) $\alpha_1,\ldots,\alpha_{16}$; (g) $\beta_1,\ldots,\beta_4$; (h) $\lambda_1,\ldots,\lambda_4$.

$$\boldsymbol{A} \triangleq \left[\frac{\partial h_1(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\ldots,\frac{\partial h_{MN}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\frac{\partial h_{MN+1}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\ldots,\frac{\partial h_{MN+M}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\frac{\partial h_{MN+M+1}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}},\ldots,\frac{\partial h_{MN+M+N}(\boldsymbol{x}^*)}{\partial \boldsymbol{x}}\right]. \tag{59}$$

It can be rewritten as

$$\boldsymbol{A} \triangleq \left[\begin{array}{c|c|c} \boldsymbol{0}_{2\times MN} & 2[\boldsymbol{t}_1-\boldsymbol{p}^*,\ldots,\boldsymbol{t}_M-\boldsymbol{p}^*] & 2[\boldsymbol{r}_1-\boldsymbol{p}^*,\ldots,\boldsymbol{r}_N-\boldsymbol{p}^*] \\ \hline \boldsymbol{I}_{MN} & \boldsymbol{0}_{MN\times M} & \boldsymbol{0}_{MN\times M} \\ \hline \boldsymbol{I}_M \quad,\ldots,\quad \boldsymbol{I}_M & 2\,\mathrm{diag}(d_1^{t*},\ldots,d_M^{t*}) & \boldsymbol{0}_{M\times N} \\ \hline \begin{bmatrix}\boldsymbol{1}_{1\times M}\\ \boldsymbol{0}_{(N-1)\times M}\end{bmatrix},\ldots,\begin{bmatrix}\boldsymbol{0}_{(N-1)\times M}\\ \boldsymbol{1}_{1\times M}\end{bmatrix} & \boldsymbol{0}_{N\times M} & 2\,\mathrm{diag}(d_1^{r*},\ldots,d_N^{r*}) \end{array}\right]. \tag{60}$$

It can be seen that the first MN columns in (60) are linear independent. Moreover, all $\{d_i^{t*}\}$ and $\{d_j^{r*}\}$ are not equal to 0. Hence the last $M+N$ columns are also linear independent.

The proof is based on contradiction. Assume that all the $MN+M+N$ columns are linear dependent. That is, there are $\varphi_1,\ldots,\varphi_{MN+M+N}$ which are not all zero, such that

$$\boldsymbol{A}\boldsymbol{\varphi} = \boldsymbol{0}_{(2+MN+M+N)\times 1}, \tag{61}$$

where $\boldsymbol{\varphi} = [\varphi_1,\ldots,\varphi_{MN+M+N}]^{\mathrm{T}}$. After rearrangement, the above linear system can be written as follows. For the first two rows in (61), we have

$$\sum_{i=1}^{M} 2\varphi_{MN+i}(\boldsymbol{t}_i-\boldsymbol{p}^*) + \sum_{j=1}^{N} 2\varphi_{MN+M+j}(\boldsymbol{r}_j-\boldsymbol{p}^*) = \boldsymbol{0}_{2\times 1}. \tag{62}$$

From the third row to the $(MN+2)$ rows in (61), we have

$$\boldsymbol{I}_{MN}[\varphi_1,\ldots,\varphi_{MN}]^{\mathrm{T}} = \boldsymbol{0}_{MN\times 1}. \tag{63}$$

From the $(MN+3)$th row to the $(MN+M+2)$th rows in (61),

$$2\varphi_{MN+i}d_i^{t*} + \sum_{j=1}^{N}\varphi_{(j-1)M+i} = 0, \quad \text{for } i = 1,\ldots,M. \tag{64}$$

From the $(MN+M+3)$th row to the $(MN+M+N)$th rows in (61),

$$2\varphi_{MN+M+j}d_j^{r*} + \sum_{i=1}^{M}\varphi_{(j-1)M+i} + = 0, \quad \text{for } j = 1,\ldots,N. \tag{65}$$

From (63), $\varphi_1,\ldots,\varphi_{MN}$ are equal to zero. Substituting $\varphi_{(j-1)M+i} = 0$, $j = 1,\ldots,N$ into (64), we obtain $2\varphi_{MN+i}d_i^{t*} = 0$ for $i = 1,\ldots,M$. Since $d_i^{t*} \neq 0$, we can deduce that $\varphi_{MN+i} = 0$ for $i = 1,\ldots,M$. In a similar way, $\varphi_{MN+M+j} = 0$ for $j = 1,\ldots,N$ can be obtained from (65). Therefore, $\boldsymbol{\varphi}$ is a zero vector. That leads to a contradiction. Hence, all the $MN+M+N$ columns of $\boldsymbol{A}$ cannot be linear dependent. We conclude that the gradient vectors in (60) are linear independent and the dynamics of ℓ_1-norm LPNN-LCA algorithm around a minimum point are stable.

5. Simulation

This section compares our algorithms with the target localization algorithms in [15] and [37]. The former is also based on the LPNN framework, but it assumes that the noise follows Gaussian distribution and uses the ℓ_2-norm in its objective function. The latter one is a robust target localization algorithm for MIMO radar system. It introduces the MCC [38] into the conventional ML method to deal with outliers, and applies half-quadratic optimization technique to handle the corresponding nonconvex nonlinear function. We refer them to as ℓ_2-norm LPNN algorithm and MCC algorithm, respectively.

The MCC method is a numerical method. Its complexity at each iteration is $\mathcal{O}((M+N)^{4.5}\log\frac{1}{\epsilon}+6MN)$ [37], where $\epsilon = 2.22^{-8}$. The LPNN framework is an analog approach. It may not be appropriate to compare the complexity between the MCC method and the three LPNN approaches. However, in the LPNN framework, the circuit complexity is mainly determined by the complexity for computing the derivatives. The complexity of the ℓ_2-norm LPNN algorithm is $\mathcal{O}(12M+12N)$. The complexities of the ℓ_1-norm LPNN and the ℓ_1-norm LPNN-LCA are $\mathcal{O}(11M+11N+MN)$ and $\mathcal{O}(12M+12N+MN)$, respectively. Thus, for each iteration, the LPNN based approaches have lower complexity than the MCC.

In our MIMO radar localization system, there are 4 transmitters and 4 receivers, i.e., $M = N = 4$. They are at $\boldsymbol{t}_1 = [-5000, 6000]^{\mathrm{T}}$(m), $\boldsymbol{t}_2 = [0, -7500]^{\mathrm{T}}$(m), $\boldsymbol{t}_3 = [10500, 0]^{\mathrm{T}}$(m), $\boldsymbol{t}_4 = [6000, 4000]^{\mathrm{T}}$(m), $\boldsymbol{r}_1 = [-10000, -6000]^{\mathrm{T}}$(m), $\boldsymbol{r}_2 = [-9000, 5000]^{\mathrm{T}}$(m), $\boldsymbol{r}_3 = [0, 4200]^{\mathrm{T}}$(m), and $\boldsymbol{r}_4 = [6400, -8000]^{\mathrm{T}}$(m). The true position of the target is $\boldsymbol{p} = [-2000, 1000]^{\mathrm{T}}$(m). The geometry of transmitters, receivers and target is shown in Fig. 5.

For the proposed algorithms, we set $C = 20$. For ℓ_1-norm LPNN method, we set $\gamma = 50$ in (27). The initial values of variables $\boldsymbol{p}$, $\boldsymbol{z}$, $d_i^t (i = 1,\ldots,M)$, $d_j^r (j = 1,\ldots,N)$, $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_{MN}]$, $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_M]$, and $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_N]$ are randomly generated.

5.1. Experiment 1: target localization in Gaussian noise

In this subsection, we study the root mean squared error (RMSE) performance of the proposed algorithms under Gaussian noise environment without introducing any outliers. The standard deviation of the Gaussian noise varies from 1 (m) to 10^2 (m). For each noise level, we repeat the experiment 100 times. We measure the RMSE between the true target position and the estimated position at each noise level. The results are shown in Fig. 6.

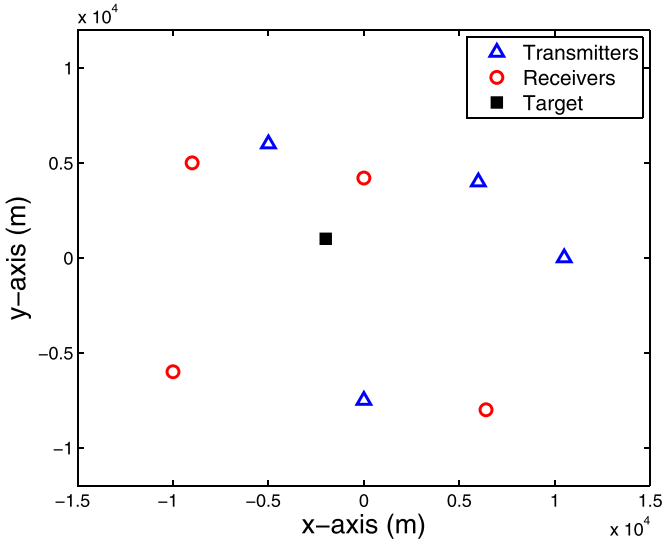

Fig. 5. Configuration of $M = N = 4$.

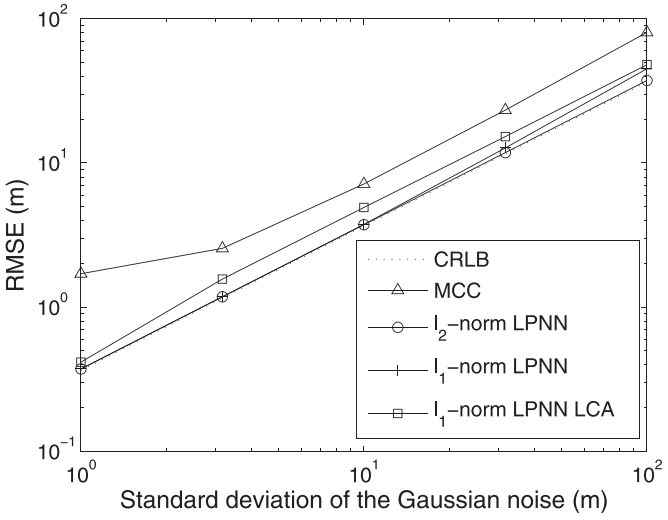

Fig. 6. RMSE of different algorithms in Gaussian noise environment.

In Fig. 6, we also plot the Cramér-Rao lower bound (CRLB), which denotes the minimum variance bound. It can be seen that the performance of our two robust algorithms is close to the CRLB in Gaussian noise environment. They are better than the robust MCC algorithm and are slightly inferior to the ℓ_2-norm LPNN algorithm, which is based on the Gaussian noise model.

5.2. Experiment 2: Target localization in Gaussian noise with NLOS outliers

In this subsection, we fix the variance of Gaussian noise to 100 (m^2), and introduce outliers into the range measurements. We assume that there exists NLOS propagation between a transmitter and the target or between the target and a receiver. Thus all measurements associated with this transmitter or receiver include NLOS outliers [33,34]. NLOS outliers are positive and are generated by the exponential distribution in the following experiments. When NLOS happens, the noisy range measurements are always larger than the true values. The standard deviation of the exponential distribution (outlier level) is varied from 10^1 (m) to 10^4 (m).

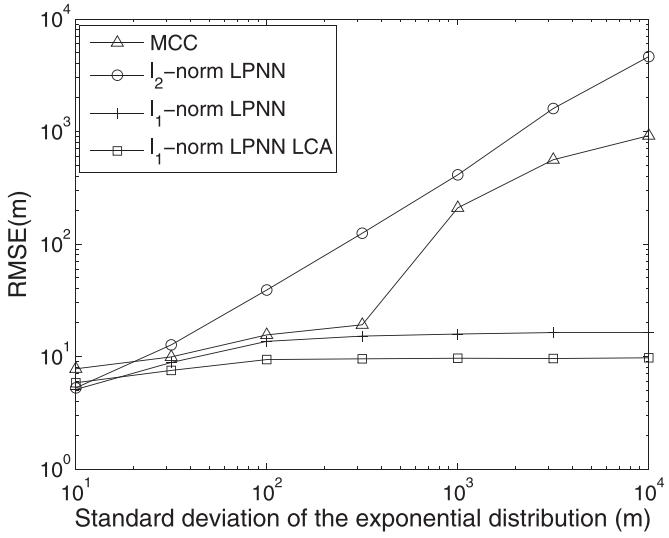

Fig. 7. RMSE of different algorithms with $M = N = 4$. In this experiment, one transmitter or one receiver corresponds to NLOS propagation.

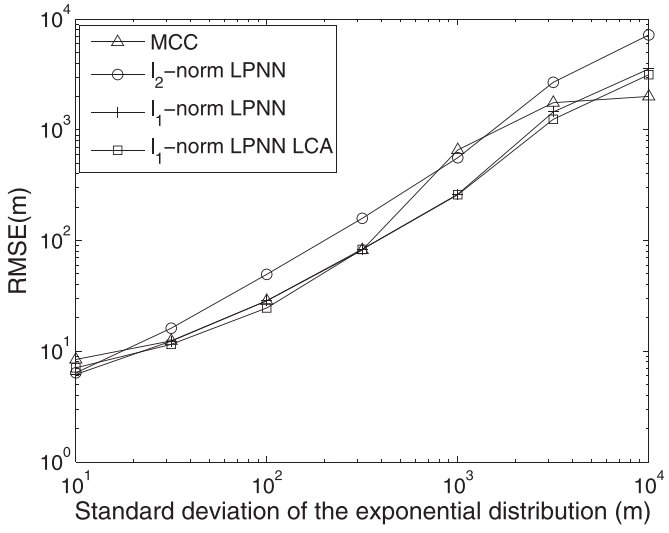

Fig. 8. RMSE of different algorithms with $M = N = 4$. In this experiment, one transmitter and one receiver correspond to NLOS propagation.

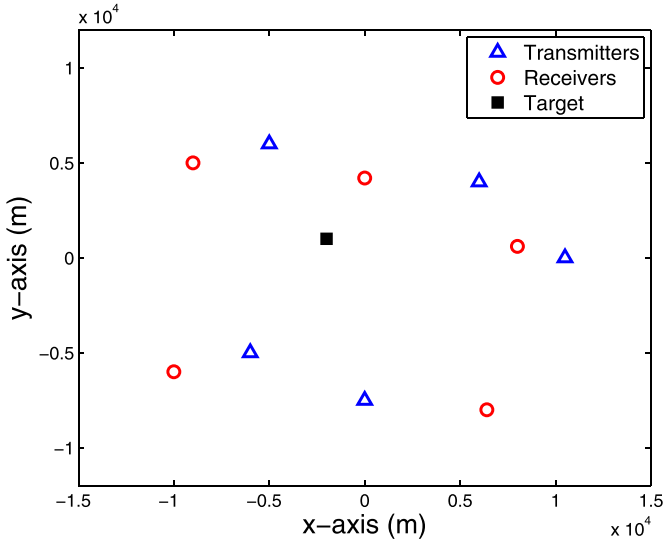

Fig. 9. Configuration of $M = N = 5$.

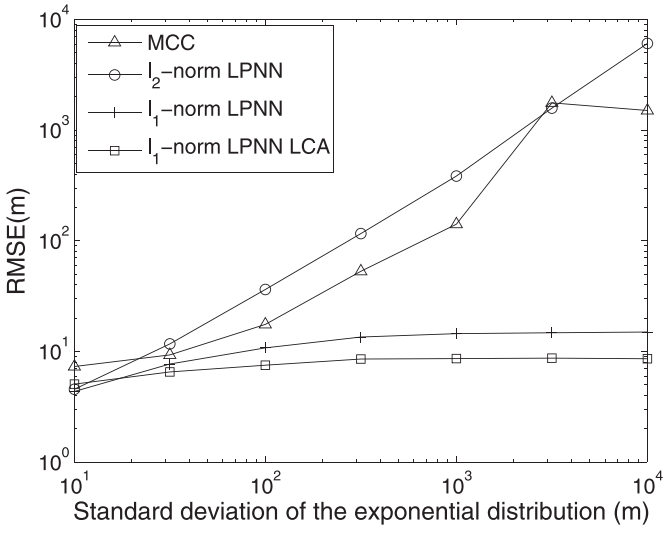

Fig. 10. RMSE of different algorithms with $M = N = 5$. In this experiment, one transmitter and one receiver correspond to NLOS propagation.

Case 1:

We randomly choose one of the transmitters or receivers and add NLOS outliers into its relevant measured values. In other words, in the 4×4 range measurement matrix, there is one column or row influenced by NLOS outliers, i.e., **4 range measurements** are highly distorted. We repeat the experiment 100 times at each outlier level. We measure the RMSE between the true target position and the estimated position and the results are shown in Fig. 7. It can be seen that the error curves of our algorithms are lower than those of the two existing algorithms. Also, as the outlier level approaches $10^{3.5}$ (m), the MCC algorithm breaks down but our algorithms do not.

Case 2:

In this case, we demonstrate that if there are too many outliers, all algorithms cannot give satisfactory results. The settings are similar to those of Case 1, except that we randomly choose one transmitter and one receiver, and then add NLOS outliers into their relevant measurements. Thus, there are **7 elements** influenced by outliers in the 4×4 measurement matrix. The RMSE results are shown in Fig. 8. It can be seen that due to the high proportion of outliers, all the algorithms cannot give satisfactory results.

Case 3:

In this case, we further add one transmitter at $\boldsymbol{t}_5 = [-6000, -5000]^{\mathrm{T}}$(m) and one receiver at $\boldsymbol{r}_5 = [8000, 600]^{\mathrm{T}}$(m) into the system given by Fig. 5. The new geometry of transmitters and receivers is depicted in Fig. 9. We randomly choose one transmitter and one receiver, and then add NLOS outliers into their relevant measurements. There are **9 elements** influenced by outliers in the 5×5 measurement matrix. The RMSE results are shown in Fig. 10. We can see that the ℓ_2-norm LPNN and the MCC are sensitive to the outliers. However, the performance of our proposed algorithms is insensitive to different outlier levels.

In summary, our proposed algorithms can give better localization results under NLOS measurements. The localization accuracy of the ℓ_1-norm LPNN-LCA algorithm is higher than that of the ℓ_1-norm LPNN. However, there are $M + N + 2$ variable neurons and $M + N$ Lagrangian neurons in the ℓ_1-norm LPNN algorithm, while there are $MN + M + N + 2$ variable neurons and $MN + M + N$ Lagrangian neurons in the ℓ_1-norm LPNN-LCA algorithm.

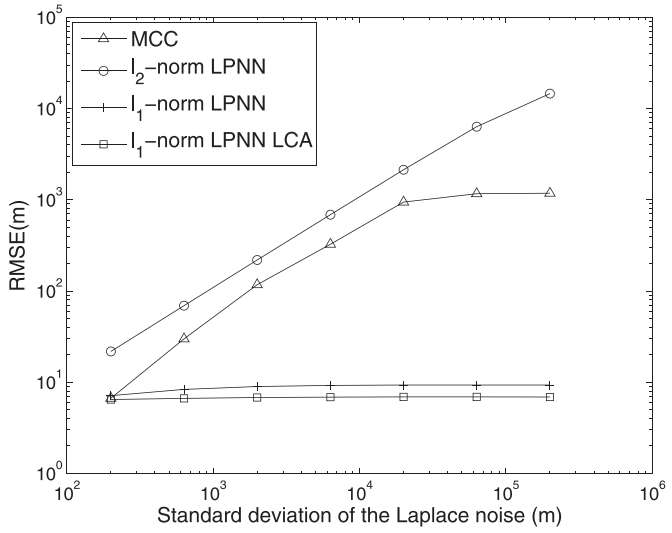

Fig. 11. RMSE of different algorithms with $M = N = 4$. In this experiment, 5 range measurements $\hat{d}_{1,2}$, $\hat{d}_{1,4}$, $\hat{d}_{2,3}$, $\hat{d}_{3,4}$, and $\hat{d}_{4,1}$ are influenced by SINR outliers.

5.3. Experiment 3: Target localization in Gaussian noise with SINR outliers

In the third experiment, we evaluate the performance of our proposed algorithms under low signal-to-interference-noise ratio (SINR) environment. This experiment is implemented based on the MIMO radar system given by Fig. 5. First, we set the variance of Gaussian noise to 100 (m^2) and introduce outliers, which are generated to model the low SINR environment. SINR outliers can be either positive or negative values and are generated by the Laplace distribution. When SINR happens, the noisy range measurements can be larger or smaller than the true values.

Assume that **5 measurement values** $\hat{d}_{1,2}$, $\hat{d}_{1,4}$, $\hat{d}_{2,3}$, $\hat{d}_{3,4}$, and $\hat{d}_{4,1}$ are influenced by SINR outliers. The standard deviation of the Laplace distribution ranges from 2×10^2 (m) to 2×10^5 (m). At each outlier level, we repeat the experiment 100 times. The RMSE results are shown in Fig. 11. It can be seen that under the low SINR environments, the performance of our proposed algorithms is also better that of the two existing algorithms. In addition, the performance of the ℓ_1-norm LPNN-LCA is the best.

Here we use the same radar configuration used in Section 5.B Case 2. However, the proposed algorithms fail to locate the target under NLOS situation shown in Fig. 8. On the contrary, they have higher localization accuracy under SINR situation shown in Fig. 9. One reason is that the number of outliers is reduced from 7 to 5 in the 4×4 measurement matrix. In addition, when NLOS happens to one transmitter or receiver, all measurements related to the corresponding transmitter or receiver are distorted. The information of the target position provided by the corresponding transmitter or receiver is totally broken. Unlike NLOS situation, SINR only influences part of the measurements provided by the corresponding transmitter or receiver. In other words, position information of the target provided by the corresponding transmitter or receiver is preserved. Hence, our approaches have higher localization accuracy under SINR outlier situation with the configuration of $M = N = 4$ in our experiments.

5.4. Checking of Theorem 1

In Theorem 1, there are two assumptions. The first assumption is that $d_i^{t*} \geq 0$ and $d_j^{r*} \geq 0$ for all i and j. The second is that $\hat{d_{i,j}} \geq d_i^{t*}$ and $\hat{d_{i,j}} \geq d_j^{r*}$ for all i and j. In this subsection, we report the number of cases that satisfy the two assumptions stated in Theorem 1 in our simulation. For each experiment, we consider several noise levels, and for each noise level we run the simulation 100 times. Table 1 summarizes the numbers of cases that satisfy the two assumptions.

Table 1
Numbers of cases that fulfil the assumptions in Theorem 1.

Experiment	Noise	ℓ_1-norm LPNN	ℓ_1-norm LPNN-LCA
Section 5.A	Gaussian	500/500	500/500
Section 5.B Case 1	NLOS	700/700	700/700
Section 5.B Case 1	NLOS	692/700	693/700
Section 5.B Case 3	NLOS	700/700	700/700
Section 5.C	SINR	394/700	394/700

Table 2
Numbers of cases that fulfill the assumptions for Section 5.B Case 2 .

Method\Noise level	10^1	$10^{1.5}$	10^2	$10^{2.5}$	10^3	$10^{3.5}$	10^4
ℓ_1-norm LPNN	100	100	100	100	100	99	93
ℓ_1-norm LPNN-LCA	100	100	100	100	100	99	94

For Experiment 1 (Gaussian noise in Section 5.A), we consider five noise levels, and hence there are 500 runs for each algorithm. The numbers of cases that satisfy the assumptions are reported in the first row of Table 1. From this row, for the Gaussian noise situation in Section 5.A, there is no violation case for the assumptions in our simulation.

For Experiment 2 (NLOS with $M = N = 4$ and one outlier in Section 5.B Case 1) we consider seven noise levels and hence there are 700 runs for each algorithm. The numbers of cases that satisfy the assumptions are reported in the second row of Table 1. From this row, for Section 5.B Case 1, there is no violation case for the assumptions in our simulation.

For Experiment 2 (NLOS with $M = N = 4$ and two outliers in Section 5.B Case 2) we consider seven noise levels, and hence there are 700 runs for each algorithm. The numbers of cases that satisfy the two assumptions are reported in the third row of Table 1. From this row, for Section 5.B Case 2, there are a few cases that do not fulfil the assumptions. Table 2 shows the numbers of fulfilled cases for each noise level. From the table, those cases that do not meet the assumptions appear in high noise levels.

For Experiment 2 (NLOS with $M = N = 5$ and two outliers in Section 5.B Case 3), we consider seven noise levels, and hence there are 700 runs for each algorithm. The numbers of cases that satisfy the assumptions are reported in the fourth row of Table 1. From this row, for Section 5.B Case 3, there is no violation case.

For Experiment 3 (SINR noise with $M = N = 4$ and five outliers in Section 5.C Case 2) we consider seven noise levels, and hence there are 700 runs for each algorithm. The numbers of cases that satisfy the two assumptions are reported in the fifth row of Table 1. From this row, for Section 5.C, there are around 42% cases that do not fulfil the assumptions. Table 3 shows the numbers of fulfilled cases for each noise level. From the table, those cases that do not meet the assumptions appear in high noise levels. Although the numbers of cases that do not fulfil is a bit high for high noise levels, the errors in the estimated positions are still very small (compared to the other two algorithms), as shown in Fig. 11.

According to Tables 1–3, our assumptions tend to hold for Gaussian noise and NLOS outliers. In particular, even NLOS propagation happens to two transmitter/receiver in the radar configuration of $M = N = 4$, i.e., Case 2 of Section 5.B, our assumptions still hold for most cases. Unfortunately, when SINR outliers are with large noise levels, it is less likely that our assumptions hold. This is because the SINR noise is of Laplace distribution. When the noise level is high, it may be with a large negative value. However, the errors in the estimated positions are still very small (compared to the other two algorithms), as shown in Fig. 11.

Table 3
Numbers of cases that fulfill the assumptions for Section 5.C.

Method\Noise level	2×10^2	$2 \times 10^{2.5}$	2×10^3	$2 \times 10^{3.5}$	2×10^4	$2 \times 10^{4.5}$	2×10^5
ℓ_1-norm LPNN	100	100	99	67	21	6	1
ℓ_1-norm LPNN-LCA	100	100	99	67	21	6	1

6. Conclusion

In this paper, we proposed two algorithms for robust target localization problem in a distributed MIMO radar system. To alleviate the influence of outliers, we formulated the target localization as a constrained optimization problem with an ℓ_1-norm objective. The two proposed algorithms are based on the concept of LPNN. Since the objective function is non-differentiable, two strategies are utilized for solving this problem. In the first method, we use a differentiable function to approximate the ℓ_1-norm in the objective function. While in the second method, we utilize the LCA concept to solve the non-differentiable problem of ℓ_1-norm. In addition, the augmented term concept is introduced to make the network more stable for both approaches. We showed that the dynamics of the proposed algorithms are locally stable. Through numerical experiments, we investigated the robustness of the proposed techniques under the impact of outliers. Simulation results showed that the proposed methods can effectively reduce the influence of outliers and are superior to two state-of-the-art MIMO radar target localization algorithms. The ℓ_1-norm LPNN-LCA algorithm is superior to the ℓ_1-norm LPNN algorithm in terms of estimation performance but more neurons, implying higher complexities, are needed.

One of the future directions is to consider the Huber-norm [36] which can address both the noise and outliers. When we use the Huber-norm to formulate a robust model, it should be noted that this norm is also only first-order differentiable. Hence, the LPNN framework cannot be directly utilized to solve the Huber-norm based formulation. After approximating the Huber-norm by the pseudo-Huber loss [46], the resultant model can be tackled by LPNN. It is worth pointing out that prior knowledge of the noise and outliers as well as a threshold parameter are needed. However, these key factors for robust localization are difficult to determine in real situations.

Declaration of Competing Interest

The authors declare that they do not have any financial or non-financial conflict of interests

CRediT authorship contribution statement

Zhanglei Shi: Writing - original draft, Visualization, Methodology, Software. **Hao Wang:** Investigation, Validation. **Chi Shing Leung:** Conceptualization, Methodology, Supervision, Project administration. **Hing Cheung So:** Writing - review & editing.

Acknowledgments

The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 9610431).

References

[1] H. Godrich, A.M. Haimovich, R.S. Blum, Target localization accuracy gain in MIMO radar-based systems, IEEE Trans. Inf. Theory 56 (6) (2010) 2783–2803.
[2] M. Dianat, M.R. Taban, J. Dianat, V. Sedighi, Target localization using least squares estimation for MIMO radars with widely separated antennas, IEEE Trans. Aerosp. Electron. Syst. 49 (4) (2013) 2730–2741.
[3] A.M. Haimovich, R.S. Blum, L.J. Cimini, MIMO radar with widely separated antennas, IEEE Signal Process. Mag. 25 (1) (2007) 116–129.
[4] J. Li, P. Stoica, MIMO radar with colocated antennas, IEEE Signal Process. Mag. 24 (5) (2007) 106–114.
[5] R. Amiri, H. Zamani, F. Behnia, F. Marvasti, Sparsity-aware target localization using TDOA/AOA measurements in distributed MIMO radars, Ict Express 2 (1) (2016) 23–27.
[6] O. Bar-Shalom, A.J. Weiss, Direct positioning of stationary targets using MIMO radar, Signal Process. 91 (10) (2011) 2345–2358.
[7] L. Rui, K. Ho, Elliptic localization: performance study and optimum receiver placement, IEEE Trans. Signal Process. 62 (18) (2014) 4673–4688.
[8] Y. Huang, J. Benesty, G.W. Elko, R.M. Mersereati, Real-time passive source localization: a practical linear-correction least-squares approach, IEEE Trans. Speech Audio Process. 9 (8) (2001) 943–956.
[9] K.W. Cheung, H.C. So, W.-K. Ma, Y.T. Chan, A constrained least squares approach to mobile positioning: algorithms and optimality, EURASIP J. Adv. Signal Process. 2006 (1) (2006) 1–23.
[10] H.C. So, Source localization: algorithms and analysis, in: Handbook of Position Location: Theory, Practice, and Advances, 2011, pp. 25–66.
[11] M. Einemo, H.C. So, Weighted least squares algorithm for target localization in distributed MIMO radar, Signal Process. 115 (2015) 144–150.
[12] C.-H. Park, J.-H. Chang, Closed-form localization for distributed MIMO radar systems using time delay measurements, IEEE Trans. Wirel. Commun. 15 (2) (2015) 1480–1490.
[13] R. Amiri, F. Behnia, H. Zamani, Asymptotically efficient target localization from bistatic range measurements in distributed MIMO radars, IEEE Signal Process. Lett. 24 (3) (2017) 299–303.
[14] A. Noroozi, A.H. Oveis, M.A. Sebt, Iterative target localization in distributed MIMO radar from bistatic range measurements, IEEE Signal Process. Lett. 24 (11) (2017) 1709–1713.
[15] J. Liang, C.S. Leung, H.C. So, Lagrange programming neural network approach for target localization in distributed MIMO radar, IEEE Trans. Signal Process. 64 (6) (2016) 1574–1585.
[16] J. Liang, Y. Chen, H.C. So, Y. Jing, Circular/hyperbolic/elliptic localization via euclidean norm elimination, Signal Process. 148 (2018) 102–113.
[17] R. Amiri, F. Behnia, An efficient weighted least squares estimator for elliptic localization in distributed MIMO radars, IEEE Signal Process. Lett. 24 (6) (2017) 902–906.
[18] R. Amiri, F. Behnia, M.A.M. Sadr, Exact solution for elliptic localization in distributed MIMO radar systems, IEEE Trans. Veh. Technol. 67 (2) (2017) 1075–1086.
[19] R. Amiri, F. Behnia, A. Noroozi, Efficient algebraic solution for elliptic target localisation and antenna position refinement in multiple-input–multiple-output radars, IET Radar, Sonar Navig. 13 (11) (2019) 2046–2054.
[20] R. Amiri, S.A.R. Kazemi, F. Behnia, A. Noroozi, Efficient elliptic localization in the presence of antenna position uncertainties and clock parameter imperfections, IEEE Trans. Veh. Technol. 68 (10) (2019) 9797–9805.
[21] R. Amiri, F. Behnia, H. Zamani, Efficient 3-D positioning using time-delay and AOA measurements in MIMO radar systems, IEEE Commun. Lett. 21 (12) (2017) 2614–2617.
[22] S.A.R. Kazemi, R. Amiri, F. Behnia, Efficient convex solution for 3-D localization in MIMO radars using delay and angle measurements, IEEE Commun. Lett. (2019).
[23] R. Amiri, F. Behnia, M.A.M. Sadr, Efficient positioning in MIMO radars with widely separated antennas, IEEE Commun. Lett. 21 (7) (2017) 1569–1572.
[24] Y. Du, P. Wei, An explicit solution for target localization in noncoherent distributed MIMO radar systems, IEEE Signal Process. Lett. 21 (9) (2014) 1093–1097.
[25] R. Amiri, F. Behnia, M.A.M. Sadr, Positioning in MIMO radars based on constrained least squares estimation, IEEE Commun. Lett. 21 (10) (2017) 2222–2225.
[26] H. Yang, J. Chun, An improved algebraic solution for moving target localization in noncoherent MIMO radar systems, IEEE Trans. Signal Process. 64 (1) (2015) 258–270.
[27] A. Noroozi, M.A. Sebt, A.H. Oveis, Efficient weighted least squares estimator for moving target localization in distributed MIMO radar with location uncertainties, IEEE Syst. J. (2019).
[28] R. Amiri, F. Behnia, H. Zamani, Closed-Form positioning in MIMO radars with antenna location uncertainties, IET Radar Sonar Navig. (2019).
[29] H. Song, G. Wen, L. Zhu, D. Li, A novel TSWLS method for moving target localization in distributed MIMO radar systems, IEEE Commun. Lett. (2019).
[30] H. Song, G. Wen, L. Zhu, An approximately efficient estimator for moving target localization in distributed MIMO radar systems in presence of sensor location errors, IEEE Sens. J. (2019).
[31] R. Amiri, F. Behnia, A. Noroozi, Efficient joint moving target and antenna localization in distributed MIMO radars, IEEE Trans. Wirel. Commun. 18 (9) (2019) 4425–4435.

[32] F. Zhang, Y. Sun, J. Zou, D. Zhang, Q. Wan, Closed-form localization method for moving target in passive multistatic radar network, IEEE Sens. J. (2019).
[33] J. Nouvel, M. Lesturgie, Study of NLOS detection over urban area at Ka band through multipath exploitation, in: IEEE International Radar Conference, Lille, France, 2014, pp. 1–5.
[34] P. Setlur, T. Negishi, N. Devroye, D. Erricolo, Multipath exploitation in non-LOS urban synthetic aperture radar, IEEE J. Sel. Top. Signal Process. 8 (1) (2014) 137–152.
[35] S.S. Al-Samahi, Y. Zhang, K. Ho, Elliptic and hyperbolic localizations using minimum measurement solutions, Signal Process. 167 (2020) 107273.
[36] R. Zhang, F. Höflinger, L. Reindl, TDOA-based localization using interacting multiple model estimator and ultrasonic transmitter/receiver, IEEE Trans. Instrum. Meas. 62 (8) (2013) 2205–2214.
[37] J. Liang, D. Wang, L. Su, B. Chen, H.C. Chen, So, Robust MIMO radar target localization via nonconvex optimization, Signal Process. 122 (2016) 33–38.
[38] W. Liu, P.P. Pokharel, J.C. Príncipe, Correntropy: properties and applications in non-Gaussian signal processing, IEEE Trans. Signal Process. 55 (11) (2007) 5286–5298.
[39] S. Zhang, A. Constantinides, Lagrange programming neural networks, IEEE Trans. Circuits Syst. II 39 (7) (1992) 441–452.
[40] M. Nagamatu, T. Yanaru, On the stability of Lagrange programming neural networks for satisfiability problems of prepositional calculus, Neurocomputing 13 (2) (1996) 119–133.
[41] X. Zhu, S.-W. Zhang, A.G. Constantinides, Lagrange neural networks for linear programming, J. Parallel Distrib. Comput. 14 (3) (1992) 354–360.
[42] V. Sharma, R. Jha, R. Naresh, An augmented Lagrange programming optimization neural network for short term hydroelectric generation scheduling, Eng. Optim. 37 (2005) 479–497.
[43] J. Liang, H.C. So, C.S. Leung, J. Li, A. Farina, Waveform design with unit modulus and spectral shape constraints via Lagrange programming neural network, IEEE J. Sel. Top. Signal Process. 9 (8) (2015) 1377–1386.
[44] C.J. Rozell, D.H. Johnson, R.G. Baraniuk, B.A. Olshausen, Sparse coding via thresholding and local competition in neural circuits, Neural Comput. 20 (10) (2008) 2526–2563.
[45] C.-S. Leung, J. Sum, A.G. Constantinides, Recurrent networks for compressive sampling, Neurocomputing 129 (2014) 298–305.
[46] J.T. Barron, A general and adaptive robust loss function, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4331–4339.