Received May 23, 2020, accepted June 16, 2020, date of publication June 22, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003984

The Sequential Fusion Estimation Algorithms Based on Gauss-Newton Method Over Multi-Agent Networked Systems

MOU WU[1,2], (Member, IEEE), LIANGJI ZHONG[2], LIANSHENG TAN[3,4], AND NAIXUE XIONG[1], (Senior Member, IEEE)

[1]College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[2]School of Computer Science and Technology, Hubei University of Science and Technology, Xianning 437100, China
[3]Discipline of ICT, School of Technology, Environments and Design, University of Tasmania, Hobart, TAS 7001, Australia
[4]Department of Computer Science, Central China Normal University, Wuhan 430079, China

Corresponding author: Liangji Zhong (yifeizlj@163.com)

This work was supported in part by the Local Science and Technology Development Fund guided by the Central Government under Grant 2019ZYYD009, in part by the Natural Science Foundation of Hubei Province under Grant 2019CFC881 and Grant 2019CFC888, and in part by the Research Development Fund of the Hubei University of Science and Technology under Grant 2019-21GP06.

ABSTRACT In multi-agent networked systems, parameter estimation problems arising in many practical applications are often required to solve Non-Linear Least Squares (NLLS) problems with the usual objective function (i.e., sum of squared residuals). The aim is to estimate a global parameter of interest across the network, such that the discrepancy between the estimation model and the real output of the system is minimized. There are challenges to face when applying the conventional Gauss-Newton method, such as non-cooperation and prosaic learning behavior. In this paper, we propose two Gauss-Newton type fusion estimation algorithms for solving overdetermined NLLS optimization problems arising frequently in multi-agent networked environment. One is the cycle-based Gauss-Newton (CGN) algorithm that is more attractive in performance due to its distributed nature than its peer: the known centralized Gauss-Newton algorithm. On the basis of CGN, we put emphasis on developing a simple but effective learning scheme leveraging an incremental technique, which is distributed on each computing agent over network. Such scheme results in the Incremental Gauss-Newton (IGN) algorithm that achieves a clear increase on convergence rate at the expense of higher computation cost than the CGN algorithm as well as the centralized one by deeper learning over the networking cycle. Both algorithms utilize Gauss-Newton iteration update in a cyclic cooperative manner, which offers the flexibility in exploiting the network topology. We provide the detailed analysis and the sufficient conditions for convergence of proposed IGN algorithm. By applying to target localization in wireless sensor networks, the numerical results confirm our convergence analysis and show that the proposed incremental scheme outperforms the centralized one in term of convergence performance.

INDEX TERMS Gauss-Newton method, incremental learning, nonlinear least squares, cyclic routing, sequential fusion.

I. INTRODUCTION

Fusion estimation [1], [2] has recently attracted a lot of attention in the field of multi-agent networked systems, especially as the agents are equipped with more and more powerful communication and computing components. An example in smart home is that the self-directing vacuum can automatically avoid obstacles by communicating with the intelligent devices deployed in a family area.

Over the past decades, the second-order learning methods such as Newton's method for solving general convex optimization problems have been largely overlooked because of high computational load, as compared with first-order methods such as gradient and subgradient methods. However, it is well known that second-order methods provide the higher accuracy for the solution because of its' learning at a deeper level. With the progress of electronic technology

The associate editor coordinating the review of this manuscript and approving it for publication was Rui-Jun Yan.

in recent years, it can be seen that the computing capability of agents in a networked system is improved significantly, for example smart phones and wireless sensor nodes. Therefore, the second-order methods have become the popular alternatives for addressing distributed optimization problems arising in networked systems, such as resource allocation [3], power dispatch in smart grids [4] and distributed tracking control [5].

Many parameter estimation problems in the fields of range-based localization [6], image restoration and alignment [7], [8], lifetime modeling of products [9], neural network [10], [11], and machine learning [12], [13] often boil down to solving the Non-Linear Least Squares (NLLS) problems, where the objective function is usually defined as the sum of squared residuals. Although first-order gradient and second-order Newton-type methods are available for solving the NLLS problmes, they ignore the special structure of the Hessian matrix in NLLS model, which is regarded as a special case for unconstrained minimization. That is, the second-order term in Hessian computation vanishes for zero residual NLLS problems and can be negligible for small residual NLLS problems [18]. On the other hand, the standard Gauss-Newton method exploits the structure to achieve a large reduction in computation cost by simply discarding the second-order term. Moreover, it is well known that the Gauss-Newton method has a locally quadratic convergence rate for zero residual problems and a linear convergence rate for small residual problems.

Unlike Newton's method that is too costly to compute the Hessian of the objective function, Gauss-Newton is a modification of Newton's method by simply discarding the second-order term in the computation of the Hessian when the initial condition is good enough, thereby resulting in the save in computational cost while maintaining fast convergence rate that can be quadratic under certain regularity conditions and linear under weaker conditions.

As the distributed algorithms with in-network processing develop, it is well known that a decentralized implementation is more efficient and robust than purely centralized or hierarchical processing schemes. Therefore, a major challenge in developing Newton type methods over networking is how to design a distributed version without the degradation of performance since the traditional Gauss-Newton method is a high-performance and centralized approach, where each agent in network sends the measured data to a fusion center for processing. A cyclic routing is typical distributed and can be found in some networked applications, such as the unmanned air vehicles using cyclic routing to monitor multiple distant targets [14] or the source location protection protocol based on cyclic routing in wireless sensor networks [15]. Moreover, such topology is easy to be deployed in various environments. For example, the communicating agents can be placed along the wall line instead of the central area in a harsh industrial scenario, thus reducing the construction difficulty. Considering the security of agents in military surveillance, the hidden agents deployed at the boundary of the monitored region are more likely to avoid being discovered and attacked.

In this paper, instead of adopting the complex matrix decomposition techniques, our aim is to develop the simple but effective fusion estimation scheme in a sequential manner, and at the same time obtaining the faster convergence performance as compared with the centralized method. To this end, we propose the intelligent sequential fusion estimation algorithms, which provide the paradigm for traditional Gauss-Newton solutions when the network is organized in a cyclic data processing structure. Firstly, we propose an embryonic Cycle-based Gauss-Newton (CGN) algorithm, which is a distributed version of centralized Gauss-Newton algorithm and provides the same performance. By simply splitting the product in descent step into two components composed of a Hessian matrix and a vector, the Gauss-Newton update is implemented only on the last agent of cyclic path. The obtained algorithm has the advantage of load balance and robustness without the degradation of estimation performance. Secondly, in order to obtain a better convergence performance than CGN, a sequential learning scheme is developed by applying the incremental strategy into CGN. The proposed intelligent algorithm is called as Incremental Gauss-Newton (IGN) that implements the Gauss-Newton update based on the same cyclic manner with CGN. In the IGN, however, each network agent obtains the descent direction by combining the updated estimate from prior agent and latest knowledge of two components of the product. The biggest benefit of such scheme is that an effect of deeper learning is achieved because the latest knowledge from neighboring agent is used immediately in the computation of descent direction at each agent for every iteration. Consequently, a faster convergence rate than the CGN algorithm can be expected. To confirm that, we first obtain the convergence conditions because the uncertainty resulted from such deep learning is introduced, under which the proposed IGN algorithm is linearly convergent at least. Then the evidence of faster convergence is provided. We also compare their performance for target localization application in an IoT network, which plays an important role as the foundation of many applications such as environmental monitoring, industrial manufacture and military field [6], [16].

In summary, the contributions of this paper are following: 1) development of a new paradigm for combining the general estimation methods and incremental computing technologies over a cyclic network; 2) detailed derivation of the convergence conditions and theoretical validation of a faster convergence for the proposed IGN algorithm; 3) numerical validation of effectiveness of the proposed algorithms by applying to target localization over wireless sensor network.

This paper is organized as follows. Section II introduces the related work. Section III describes the NLLS problems over networked system and presents the traditional centralized Gauss-Newton solution. In Section IV, we propose the motivation and the details of the fusion estimation scheme, and the convergence with sufficient conditions of IGN is analyzed. The target localization model and simulation results are provided in Section V. The whole paper is concluded and the future work is presented in Section VI.

Notation: The operator $(\cdot)^T$ denotes the transpose for matrix or vector, the operator $(\cdot)^{-1}$ denotes the inverse of a non-singular matrix. The Euclidean norm of a vector x is written as $\|x\|$, 2-norm of a matrix F is denoted by $\|F\|$. We will use indexes k and j to denote agents, and index i to denote time.

II. RELATED WORK

Traditionally, Gauss-Newton method is designed for solving the NLLS problems in data fitting [17], [18]. Recently, NLLS problems have be modeled in a wide application area, such as computer vision [8], image alignment and reconstruction [19], network-based localization [20], signal processing for direction-of-arrival estimation and frequency estimation [21], logistic regression [22] and power system state estimation [4], [23]. As a result, their solutions are highly dependent on the effective implementation of proposed optimization methods, especially in the networking environment.

Several distributed Newton type methods have been proposed recently. Gossip-based Gauss-Newton (GGN) algorithm [24] implements the Gauss-Newton step by exchanging local information with communicated neighbours via network gossiping. In [25], a distributed Gauss-Newton algorithm for state estimation of electric power systems allows each control area to calculate the state estimates of its local buses and communicate between neighboring areas. For each control area, both of the local information and the exchange of information with neighboring control areas are limited, thus restraining data flooding. The work in [26] uses the belief propagation (BP) algorithm to solve AC state estimation problem for the same electric power systems. The proposed BP algorithm has the interpretation of a distributed Gauss-Newton method with the same accuracy as the centralized Gauss-Newton. Distributed Gauss-Newton methods [27] are proposed for node localization in wireless sensor networks. The proposed method searches the descent direction of the Gauss-Newton by simply averaging local descent information exchanged by each node's in 1-hop neighborhood. A diffusion Gauss-Newton localization method with high accuracy and adaptive learning is proposed recently in our work [6]. The proposed method has an equalization effect on the unbalance noise distribution over the network in industrial environments. To avoid slow convergence in first-order methods, a distributed Newton Method [28] is proposed to solve network utility maximization (NUM) problems. The key feature of this method is that the matrix splitting techniques used in the Newton step are implemented in a decentralized manner. The descent direction and the step size computation are also obtained by using the distributed local information exchange as in the first order methods. To speedup the Newton step, a barrier-based method [29] is proposed to obtain a fast computation of inversion of the Hessian and an almost linear complexity.

The steepest descent algorithm is a first-order method that can also be applied to solve NLLS problems with good global performance and poor local convergence [18]. Some studies offer the distributed implementation for steepest descent algorithm since its descent step is simple and easy to rebuild. By considering the modes of cooperation between network nodes, there are two main types of steepest descent-based distributed methods: diffusion and incremental models. In diffusion implementations [6], [30], information is exchanged between any pair of neighboring nodes without the limits of predefined network structure. In multi-sensor networked system, the fusion center usually cannot communicate with all sensors to exchange information, while only communicating with the most accessible sensor, e.g., shortest distance geographically. The difficulty adopting diffusion model is that the results from neighborhood need be combined in an appropriate weighted form. In the incremental implementations [31]–[33], information is shared only on two neighboring nodes at a time when network nodes are connected in a cyclic manner. The size of cycle and node selection depend on the scale of network and the specific application, thus providing the flexibility in exploiting the network topology.

Some diffusion type Gauss Newton algorithms via network average consensus are proposed in [25], [27], [34], however, there is still a lack of work on developing incremental Gauss-Newton algorithms as the incremental steepest descent algorithms do, which motives the combination of Gauss-Newton learning and incremental cooperation in this paper.

III. PROBLEM STATEMENT AND THE CENTRALIZED SOLUTION

Consider a multi-agent network that consists of N agents with data processing and communication abilities, such as sensor network. Let us denote an $M \times 1$ unknown parameter vector $x = [x_1, \cdots, x_M]^T$. The objective is to estimate the vector x by solving the following unconstrained minimization problem

$$\min_x \| f(x) \|^2 \tag{1}$$

where $f(x) = [f_1(x), \cdots, f_N(x)]^T$ is a vector-valued continuous and differentiable global function throughout the network, and $f_k(x) : \Re^M \rightarrow \Re$ is the component function associated with each agent $k \in \mathcal{N} \triangleq \{1, \ldots, N\}$. Thus, $\|f(x)\|^2$ can be decoupled into a sum of N cost functions, we rewrite $\|f(x)\|^2 = \sum_{k=1}^{N} |f_k(x)|^2$.

As a consequence, the problem (1) is converted to a nonlinear least squares problem with the form

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{N} |f_k(x)|^2 \\ \text{subject to} \quad & x \in \Re^M. \end{aligned} \tag{2}$$

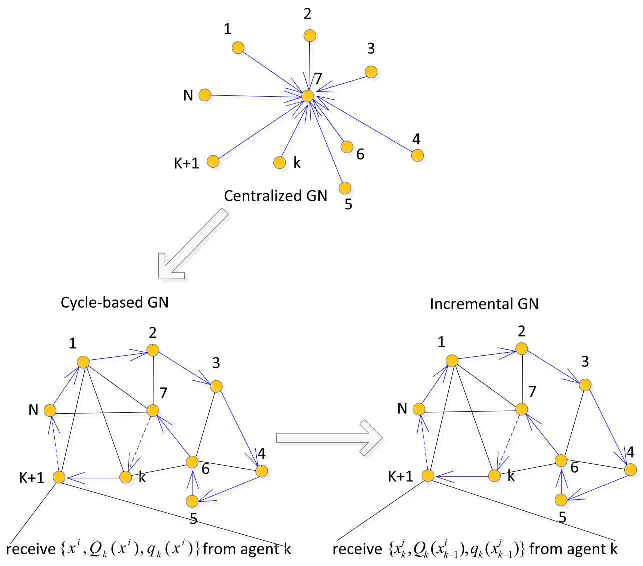
Centralized GN
Cycle-based GN
Incremental GN
1
2
3
4
5
6
7
N
K+1
k
receive $\{x^i, Q_k(x^i), q_k(x^i)\}$ from agent k
receive $\{x_k^i, Q_k(x_{k-1}^i), q_k(x_{k-1}^i)\}$ from agent k

FIGURE 1. Centralized, cycle-based and incremental Gauss-Newton cooperation schemes.

To solve the unconstrained NLLS problem (1), the most popular one is the conventional Gauss-Newton method due to its high estimation accuracy and fast convergence. Considering a centralized scheme where a fusion center is responsible for communication with all N agents as shown in Fig. 1. In other words, the fusion center collects the component $f_k(x)$ and available data from each agent k. Starting from an initial vector x^0, x is estimated iteratively by the fusion center using the Gauss-Newton method, which has the following form

$$\begin{aligned} x^{i+1} &= x^i - \alpha^i d^i \\ &= x^i - \alpha^i [F^T(x^i)F(x^i)]^{-1} F^T(x^i) f(x^i), \end{aligned} \tag{3}$$

where x^i is the estimation of x at iteration i, d^i denotes a descent direction of Gauss-Newton, α^i is the step size parameter that ensures x^{i+1} is nearer to a stationary point than x^i, and the $N \times M$ Jacobian $F(x)$ is defined with the entries $F(x)_{k,l} = \partial f_k(x)/\partial x_l$, $1 \le k \le N$, $1 \le l \le M$.

Note that N is the number of component functions distributed among N agents and M is the number of elements of the unknown parameter x. For the case of $N > M$, the problem is considered overdetermined. Conversely, if $N < M$, the problem is underdetermined and less common. In this paper, we study the overdetermined NLLS problems which occur more often since the number of agents in a network is typically more than the number of entries of the estimated vector.

Over the past few years, some adaptive step size rules based on Gauss-Newton method have been proposed in order to accelerate convergence. In this paper, we only consider that the step size is constant to simplify the later discussion, which means $\alpha = \alpha^i \in (0, 1]$ for any iteration i. However, the constraint for step size to guarantee the convergence of proposed algorithms will be derived in the section of convergence analysis.

Since the cost function is often not strictly convex, it is difficult to find the global optimum. Thus, our objective is to achieve the estimation that is sufficiently close to the unique local minimizer denoted by x^*. When the following standard assumptions are satisfied, the local convergence of Gauss-Newton method for overdetermined NLLS problems will be guaranteed and has a quadratic rate.

Assumption 1: x^* is a local minimizer of $\|f(x)\|^2$, $f(x)$ is continuously differentiable near x^*, $F(x)$ has full column rank for $N > M$.

Meanwhile, the necessary conditions of the convergence for Newton-class methods imply that at the minimizer x^*, we have

$$\nabla f(x^*) = 2F^T(x^*)f(x^*) = 0, \tag{4}$$

where $\nabla(\cdot)$ denotes the gradient operator.

IV. OUR PROPOSED SEQUENTIAL FUSION ESTIMATION SCHEME

Computing d^i in (3) results in a non-distributed algorithm and high computation and communication complexity. In this paper, we aim to establish the distributed algorithms that only use the local information to achieve the decrease of the cost function.

A. CYCLE-BASED GAUSS-NEWTON ALGORITHM

A natural idea is to split the descent direction d^i into two components of $F^T(x^i)F(x^i)$ and $F^T(x^i)f(x^i)$ because of the existence of the inverse operator. In fact, by the following definition

$$\begin{aligned} Q_N(x^i) &\triangleq F^T(x^i)F(x^i) = \sum_{k=1}^{N} F_k^T(x^i)F_k(x^i), \\ q_N(x^i) &\triangleq F^T(x^i)f(x^i) = \sum_{k=1}^{N} F_k^T(x^i)f_k(x^i), \end{aligned} \tag{5}$$

where $F_k(x^i)$ are the kth row of the matrix $F(x^i)$, the centralized Gauss-Newton method (3) can be rewritten as

$$x^{i+1} = x^i - \alpha[\sum_{k=1}^{N} F_k^T(x^i)F_k(x^i)]^{-1} \sum_{k=1}^{N} F_k^T(x^i)f_k(x^i). \tag{6}$$

Consequently, the centralized Gauss-Newton algorithm can be implemented in the following distributed way. Consider the network with N agents that are connected in a cyclical manner. Agent $k \in \mathcal{N}$ makes a summing for $Q_k(x^i)$ and $q_k(x^i)$ that are defined by

$$\begin{aligned} Q_k(x^i) &\triangleq \sum_{j=1}^{k} F_j^T(x^i)F_j(x^i) \\ q_k(x^i) &\triangleq \sum_{j=1}^{k} F_j^T(x^i)f_j(x^i). \end{aligned} \tag{7}$$

By computing $\{Q_k(x^i), q_k(x^i)\}$ on each agent k and sending them and the estimate $\{x^i\}$ to neighboring next agent, till the agent N, the following recursive relationship can be used to describe the process implemented on agent k

$$Q_k(x^i) = Q_{k-1}(x^i) + F_k^T(x^i)F_k(x^i),$$
$$q_k(x^i) = q_{k-1}(x^i) + F_k^T(x^i)f_k(x^i). \quad (8)$$

Algorithm 1: Cycle-Based Gauss-Newton Algorithm

```
1  start with an initial point $x^0$;
2  for every iteration $i \geq 1$ do
3      for agents $k = 1$ to $N$ do
4          if $k = 1$ then
5              receive $x_N^{i-1}$ from agent $N$;
6              $x^i = x_N^{i-1}$;
7              $Q_1(x^i) = F_1^T(x^i)F_1(x^i)$;
8              $q_1(x^i) = F_1^T(x^i)f_1(x^i)$;
9          else
10             receive $\{x^i, Q_{k-1}(x^i), q_{k-1}(x^i)\}$ from agent
                $k-1$;
11             $Q_k(x^i) = Q_{k-1}(x^i) + F_k^T(x^i)F_k(x^i)$;
12             $q_k(x^i) = q_{k-1}(x^i) + F_k^T(x^i)f_k(x^i)$;
13             if $k = N$ then
14                 $x_N^i = x^i - \alpha Q_N^{-1}(x^i)q_N(x^i)$;
15                 send $\{x_N^i\}$ to agent 1;
16             else
17                 send $\{x^i, Q_k(x^i), q_k(x^i)\}$ to agent $k+1$;
18             end
19         end
20     end
21 end
```

The above steps lead to a distributed Gauss-Newton algorithm that is described in Algorithm 1. Note that Algorithm 1 and (6) are equivalent on the final estimate but differ in implementation way. In the implementation of Algorithm 1, each iteration consists of N sub-communications, during which agent k starts with $\{x^i, Q_{k-1}(x^i), q_{k-1}(x^i)\}$ received from its neighbouring previous agent $k-1$ on the cycle, and does without the Gauss-Newton update. In other words, there is no need to compute the intermediate estimate of x at agent k and iteration i denoted by x_k^i, where we define $x^i \triangleq x_0^i$ and $x^{i+1} \triangleq x_N^i$. Afterwards, agent k passes x^i and the new data $\{Q_k(x^i), q_k(x^i)\}$ to agent $k+1$ until agent N is reached. The Gauss-Newton update is only implemented on agent N and the estimated result x_N^i is used for next iteration. Fig. 1 illustrates the cooperation scheme between agents in a distributed network.

Obviously, the benefit of Algorithm 1 is that a load-balanced distributed system is obtained since the mode of cyclic cooperation balances the communication loads and the links of the entire network. It is worth noting that there is very little difference in the total number of communications per iteration between Algorithm 1 and the centralized Gauss-Newton algorithm. In fact, the communication activity of Algorithm 1 occurs between neighboring agents and the total number of communications is $(N-1)(2M+M^2)+M$ scalars for one iteration, while the communication activity in the centralized Gauss-Newton algorithm occurs between the fusion center and each agent, and the total number of communications is $(N-1)(2M+M^2)$ scalars for one iteration. For a large i or M in the repeated parameter estimation applications (e.g., tracking the unknown target in a time-varying scenario), one can see the loss of Algorithm 1 on communication energy. However, it is well known that the communication consumption of (6) is magnified when the fusion center is far away from the agents, while the communicating agents of Algorithm 1 are always adjacent on distance. It's most often an advantage especially for using short-range wireless techniques such as Bluetooth communication.

Another important benefit of Algorithm 1 is that the robustness of system is improved effectively. The failure of the fusion center will cause the failure of the centralized Gauss-Newton since no other agents collect the required information for implementing Gauss-Newton step, while agent N can be replaced by the other agents that receive the information from the previous agent on the cyclic path, thereby ensuring the success of Algorithm 1. To highlight the distinctions of (6) and (8), we refer to Algorithm 1 as the cycle-based Gauss-Newton (CGN) algorithm.

B. INCREMENTAL GAUSS-NEWTON ALGORITHM

Although the CGN algorithm is distributed in nature, the global estimate x^i is an outdated information for the agents from 2 to N when they try to Gauss-Newton update. Motivated by the incremental gradient algorithms [31], [33], we propose an incremental version of Algorithm 1 that uses local computed estimate x_k^i to approximate x^i at each subiteration. The intermediate estimate x_k^i can be regarded as a result of deeper learning through the cycle, thus leading to faster and deeper learning. This is somewhat similar to the amplify-and-forward relay architecture in wireless communication, where the strength is enhanced as signal transmission hop by hop. Based on this motivation, a set of coupled N equalities implemented on N agents at an iteration can be written as

$$\begin{aligned} x_1^i &= x^i - \alpha Q_1^{-1}(x^i)q_1(x^i), \\ &\vdots \\ x_k^i &= x_{k-1}^i - \alpha Q_k^{-1}(x_{k-1}^i)q_k(x_{k-1}^i), \\ &\vdots \\ x_N^i &= x_{N-1}^i - \alpha Q_N^{-1}(x_{N-1}^i)q_N(x_{N-1}^i), \end{aligned} \quad (9)$$

where $Q_k(x_{k-1}^i) = \sum_{j=1}^{k} F_j^T(x_{j-1}^i)F_j(x_{j-1}^i)$ and $q_k(x_{k-1}^i) = \sum_{j=1}^{k} F_j^T(x_{j-1}^i)f_j(x_{j-1}^i)$.

The obtained incremental Gauss Newton (IGN) algorithm is described in Algorithm 2 which has the same communication cost with Algorithm 1, but with some important

Algorithm 2: Incremental Gauss-Newton Algorithm

```
1  start with an initial point $x^0$;
2  for every iteration $i \geq 1$ do
3      for agents $k = 1$ to $N$ do
4          if $k = 1$ then
5              receive $x_N^{i-1}$ from agent $N$;
6              $x^i = x_N^{i-1}$;
7              $Q_1(x^i) = F_1^T(x^i)F_1(x^i)$;
8              $q_1(x^i) = F_1^T(x^i)f_1(x^i)$;
9              $x_1^i = x^i - \alpha Q_1^{-1}(x^i)q_1(x^i)$;
10             send $\{x_1^i, Q_1(x^i), q_1(x^i)\}$ to agent 2;
11         else
12             receive $\{x_{k-1}^i, Q_{k-1}(x_{k-2}^i), q_{k-1}(x_{k-2}^i)\}$
                from agent $k-1$;
13             $Q_k(x_{k-1}^i) =$
                 $Q_{k-1}(x_{k-2}^i) + F_k^T(x_{k-1}^i)F_k(x_{k-1}^i)$;
14             $q_k(x_{k-1}^i) = q_{k-1}(x_{k-2}^i) + F_k^T(x_{k-1}^i)f_k(x_{k-1}^i)$;
15             $x_k^i = x_{k-1}^i - \alpha Q_k^{-1}(x_{k-1}^i)q_k(x_{k-1}^i)$;
16             if $k = N$ then
17                 send $\{x_N^i\}$ to agent 1;
18             else
19                 send $\{x_k^i, Q_k(x_{k-1}^i), q_k(x_{k-1}^i)\}$ to agent
                     $k+1$;
20             end
21         end
22     end
23 end
```

differences. As shown in Fig. 1, the big one is that each agent on the cyclic path can obtain the new estimate x_k^i that is used to substitute the old estimate x^i of the Gauss-Newton update in Algorithm 1. Intuitively, the IGN update on each agent means a descent step on the cost function. The expected better result of this incremental-type implementation is an improvement of convergence speed. Meanwhile, a negative impact on Algorithm 2 is the increase of computation cost. In Algorithm 2, each agent needs to obtain the estimate x_k^i by computing the product of matrix $Q_k^{-1}(x_{k-1}^i)$ and vector $q_k(x_{k-1}^i)$, while only agent N is required to do in Algorithm 1. In other words, the incremental strategy helps to reduce the total number of iterations while increasing the computation cost of a single iteration. The time that the system spends on floating point calculations can be decreased by faster hardware. However, the reduction to the number of iterations can only be achieved via the improvement of algorithms. We will compare the run-time for CGN and IGN by the simulation analysis.

It is important to note that the square matrix $Q_k(x_{k-1}^i)$ for $k \in \mathcal{N}$ may be not invertible during iterations. Two ways can be considered for practical usage. One is to add a small multiple ℓ_k of the identity matrix I to $Q_k(x_{k-1}^i)$, such that $Q_k(x_{k-1}^i) + \ell_k I$ will be nonsingular, which is called the Levenberg−Marquardt method [18]. The difficulty with this method is how to determine a good value for $\ell_k > 0$. The other is to compute the unique Moore−Penrose inverse, which gives a good approximate solution to the inversion of the matrix by using the singular value decomposition [35], [36]. When the following four conditions:

$$\begin{aligned}&(1) AA^+A = A,\\&(2) A^+AA^+ = A^+,\\&(3) (AA^+)^T = AA^+,\\&(4) (A^+A)^T = A^+A\end{aligned}\tag{10}$$

are satisfied for any matrix A, the Moore−Penrose inverse denoted by A^+ is an unique and least square solution of Euclidean norm $\|Ax - b\|$ for a system of linear equations $Ax = b$. For simplicity in later discussion, we assume that $Q_k(x_{k-1}^i)$ is invertible during any iteration.

Moreover, given the fact that the convergence of CGN algorithm has been confirmed in the centralized way, a challenge on evaluating the convergence of incremental strategy arises since it introduces the uncertainty of system as compared with the centralized one. In the following section, we will focus on the convergence analysis of IGN algorithm and obtain the sufficient condition for convergence from a dynamic evolution perspective.

C. CONVERGENCE ANALYSIS FOR OUR IGN ALGORITHM

In this section, we use the following assumption which was also adopted in [23], [37], [38] for analyzing the convergence of classical Gauss-Newton algorithm.

Assumption 2:

(1) $f(x)$ is bounded for all x belonging a compact set $\mathbb{X} \subset \mathbb{R}^M$ near a local minimizer x^*, and satisfies

$$\|f(x)\| \leq e_{max} \tag{11}$$

and

$$\|f(x^*)\| = e_{min} \tag{12}$$

(2) The notations $\lambda_{min}(\cdot)$ and $\lambda_{max}(\cdot)$ are denoted as the minimum and maximum eigenvalues. For all $x \in \mathbb{X}$ and $k \in \mathcal{N}$, let

$$\Sigma_{min} = \min_{x\in\mathbb{X}} \sqrt{\lambda_{min}(F^T(x)F(x))} \tag{13}$$

$$\Sigma_{max} = \max_{x\in\mathbb{X}} \sqrt{\lambda_{max}(F^T(x)F(x))}, \tag{14}$$

and

$$\sigma_{min} = \min_{x\in\mathbb{X},k\in\mathcal{N}} \sqrt{\lambda_{min}(F_k^T(x)F_k(x))} \tag{15}$$

$$\sigma_{max} = \max_{x\in\mathbb{X},k\in\mathcal{N}} \sqrt{\lambda_{max}(F_k^T(x)F_k(x))}, \tag{16}$$

where $0 < \Sigma_{min} < \Sigma_{max} < \infty$ and $0 < \sigma_{min} < \sigma_{max} < \infty$.

(3) The Jacobian $F(x)$ is Lipschitz continuous on $\mathbb{X}$ with Lipschitz constant $\omega > 0$ such that

$$\|F(x) - F(y)\| \leq \omega\|x - y\|, \tag{17}$$

for all $x, y \in \mathbb{X}$.

Assumption 2 is reasonable for a specified NLLS problem, where the value and the change speed of cost function f_x are considered to be bounded. Based on Assumption 2, we will use the corresponding results [23], [39] described in Corollary 1 for the following analysis.

Corollary 1: Let Assumption 2 hold, we have

$$\|F_k(x) - F_k(y)\| \leq \omega\|x - y\|, \tag{18}$$

for all $x, y \in \mathbb{X}$. Furthermore, we have the following results

$$\|F_k^T(x)f_k(x) - F_k^T(y)f_k(y)\| \leq \gamma_f\|x - y\| \tag{19}$$

and

$$\|F_k^T(x)F_k(x) - F_k^T(y)F_k(y)\| \leq \gamma_F\|x - y\|, \tag{20}$$

where $\gamma_f \geq \omega(e_{max} + \Sigma_{max})$ and $\gamma_F \geq 2\Sigma_{max}\omega$ are the corresponding Lipschitz constants.

1) SPATIAL-TEMPORAL RECURSION RELATION

We now proceed to show an error variance relation between the local estimate x^i of IGN algorithm and a local minimizer x^*. First, we obtain the local estimate relation between the successive two times according to the equation set (9)

$$x^{i+1} = x^i - \alpha\sum_{k=1}^{N} Q_k^{-1}(x_{k-1}^i)q_k(x_{k-1}^i). \tag{21}$$

By defining the incremental descent $d_k^i(x_{k-1}^i) \triangleq Q_k^{-1}(x_{k-1}^i)q_k(x_{k-1}^i)$ in IGN algorithm and the centralized descent $d_k^i(x^i) \triangleq Q_k^{-1}(x^i)q_k(x^i)$ in CGN algorithm, we can rewrite (21) as

$$\underbrace{x^{i+1} = x^i - \alpha d_N^i(x^i)}_{CGN} + \underbrace{\alpha(d_N^i(x^i) - \sum_{k=1}^{N} d_k^i(x_{k-1}^i))}_{discrepancy\ \ error\ between\ CGN\ and\ IGN} \tag{22}$$

The expression (22) reveals the relation on local estimates obtained by CGN and IGN algorithms. From that, we can see that a discrepancy error between them results from the descent difference by using the centralized mode and the incremental mode, respectively. Subtracting both sides from the local optimal vector x^* and using the Euclidean norm operator, we get the following recursion

$$\begin{aligned}\|x^{i+1} - x^*\| \leq \|x^i - x^* - \alpha d_N^i(x^i)\| \\ +\alpha\|d_N^i(x^i) - \sum_{k=1}^{N} d_k^i(x_{k-1}^i)\|.\end{aligned} \tag{23}$$

Based on the equalities (3) and (5), $d_N^i(x^i)$ can be written as

$$\begin{aligned}d_N^i(x^i) &= Q_N^{-1}(x^i)q_N(x^i) \\ &= [F^T(x^i)F(x^i)]^{-1}F^T(x^i)f(x^i).\end{aligned} \tag{24}$$

By denoting $F^+(\cdot)$ as the Moore−Penrose of matrix $F(\cdot)$ based on the definition (10), we have

$$d_N^i(x^i) = F^+(x^i)f(x^i). \tag{25}$$

Thus, the first term of the right side of the recursion (23) can be written as

$$\begin{aligned}x^i - x^* - \alpha d_N^i(x^i) = x^i - x^* - \alpha F^+(x^i)f(x^i) \\ +\alpha F^+(x^*)f(x^*),\end{aligned} \tag{26}$$

where $F^+(x^*)f(x^*) = 0$ according to the condition (4).

Therefore, we arrive at the expression (cf. [23] Equality (62)) that is used to analyze the convergence of traditional centralized Gauss-Newton method. Consequently, a recursion can be derived as follows

$$\|x^i - x^* - \alpha d_N^i(x^i)\| \leq T_1\|x^i - x^*\|^2 + T_2\|x^i - x^*\|, \tag{27}$$

where $T_1 \triangleq \frac{\alpha\omega}{2\Sigma_{min}}$ and $T_2 \triangleq \frac{(1-\alpha)\Sigma_{max}}{\Sigma_{min}} + \frac{\sqrt{2}\alpha\omega e_{min}}{\Sigma_{min}^2}$.

Substituting (27) in (23), a spatial-temporal recursion that shows the estimated error variance between the local estimate x^i in incremental way at each iteration and the local minimizer x^* can be obtained

$$\begin{aligned}\|x^{i+1} - x^*\| \leq T_1\|x^i - x^*\|^2 + T_2\|x^i - x^*\| \\ +\alpha\|d_N^i(x^i) - \sum_{k=1}^{N} d_k^i(x_{k-1}^i)\|.\end{aligned} \tag{28}$$

Similar to (22), the error recursion consists of two parts, $\|x^i - x^*\|$ describes the estimated error evolution as time update and $\|d_N^i(x^i) - \sum_{k=1}^{N} d_k^i(x_{k-1}^i)\|$ is the spatial descent discrepancy distributing on the incremental network which is composed of N agents.

2) BOUNDNESS OF DISCREPANCY

Based on Assumptions 1 and 2, we pursue the boundness analysis of descent discrepancy, which can be written by

$$\|d_N^i(x^i) - \sum_{k=1}^{N} d_k^i(x_{k-1}^i)\| \leq \|d_N^i(x^i)\| + \|\sum_{k=1}^{N} d_k^i(x_{k-1}^i)\| \tag{29}$$

according to the triangular inequality for norms.

For the first term of right side of (29), we have

$$\begin{aligned}\|d_N^i(x^i)\| &= \|[F^T(x^i)F(x^i)]^{-1}F^T(x^i)f(x^i)\| \\ &\leq \|[F^T(x^i)F(x^i)]^{-1}\|\|F^T(x^i)f(x^i)\| \\ &\leq \frac{\Sigma_{max}e_{max}}{\Sigma_{min}^2}.\end{aligned} \tag{30}$$

Because of $\|f(x)\|^2 = \sum_{k=1}^{N}\|f_k(x)\|^2 \leq e_{max}^2$ and $\|f(x)\|^2 \geq e_{min}^2$, $\|f_k(x)\|$ has the upper and lower bounds for all k and $x \in \mathbb{X}$. For convenience, we let $\varepsilon_{min} \leq \|f_k(x)\| \leq \varepsilon_{max}$. Thus, we have $\sigma_{min}\varepsilon_{min} \leq \|F_j^T(x)f_j(x)\| \leq \sigma_{max}\varepsilon_{max}$ for $x \in \mathbb{X}$.

For the second term of right side of (29), we have

$$\begin{aligned}\|\sum_{k=1}^{N} d_k^i(x_{k-1}^i)\| &\leq \sum_{k=1}^{N}\|d_k^i(x_{k-1}^i)\| \\ &\leq \sum_{k=1}^{N}\|[\sum_{j=1}^{k}F_j^T(x_{j-1}^i)F_j(x_{j-1}^i)]^{-1}\| \\ &\quad \times \|\sum_{j=1}^{k}F_j^T(x_{j-1}^i)f_j(x_{j-1}^i)\| \\ &\leq \sum_{k=1}^{N}\frac{\sigma_{max}\varepsilon_{max}}{\sigma_{min}^2} = \frac{N\sigma_{max}\varepsilon_{max}}{\sigma_{min}^2}, \end{aligned} \quad (31)$$

where we use

$$\frac{1}{k\sigma_{max}^2} \leq \|[\sum_{j=1}^{k}F_j^T(x_{j-1}^i)F_j(x_{j-1}^i)]^{-1}\| \leq \frac{1}{k\sigma_{min}^2} \quad (32)$$

because of $k\sigma_{min}^2 \leq \|\sum_{j=1}^{k}F_j^T(x_{j-1}^i)F_j(x_{j-1}^i)\| \leq k\sigma_{max}^2$.

Substituting (30) and (31) in (29), we obtain the boundness condition

$$\begin{aligned}&\|d_N^i(x^i) - \sum_{k=1}^{N} d_k^i(x_{k-1}^i)\| \\ &\qquad \leq \frac{\Sigma_{max}e_{max}}{\Sigma_{min}^2} + \frac{N\sigma_{max}\varepsilon_{max}}{\sigma_{min}^2} \triangleq \xi, \end{aligned} \quad (33)$$

where we introduce ξ as a positive constant.

3) CONVERGENCE WITH SUFFICIENT CONDITIONS

Giving the constant ξ that satisfies (33), the spatial-temporal error recursion (28) is rewritten as

$$\|x^{i+1} - x^*\| \leq T_1\|x^i - x^*\|^2 + T_2\|x^i - x^*\| + \alpha\xi, \quad (34)$$

which can be regarded as a nonlinear discrete dynamical system

$$y^{i+1} \leq T_1(y^i)^2 + T_2y^i + \alpha\xi, \quad (35)$$

where we define $y^i \triangleq \|x^i - x^*\|$. The dynamical system (35) is upper bounded by the following equation system

$$y^{i+1} = T_1(y^i)^2 + T_2y^i + \alpha\xi \quad (36)$$

due to $y^i > 0$. Thus, the evolution of y^i in (35) is governed by (36).

Consequently, we can resort to the steady-state equilibria theory [40] in nonlinear discrete dynamical system to obtain the steady-state equilibrium of (36), which is defined as an invariant under the law of motion described by (36). By solving the corresponding equation

$$y = T_1y^2 + T_2y + \alpha\xi, \quad (37)$$

we can easily obtain two steady-state equilibrium points

$$y_{max} = \frac{(1-T_2) + \sqrt{(1-T_2)^2 - 4T_1\alpha\xi}}{2T_1} \quad (38)$$

and

$$y_{min} = \frac{(1-T_2) - \sqrt{(1-T_2)^2 - 4T_1\alpha\xi}}{2T_1} \quad (39)$$

under the condition

$$(T_2 - 1)^2 - 4T_1\alpha\xi \geq 0. \quad (40)$$

The necessary and sufficient conditions for local stability of steady state equilibrium (Proposition 1.9 [40]) indicate that a steady state equilibrium is locally stable if $|\frac{dy^{i+1}}{dy^i}| < 1$, unstable otherwise, where $\frac{d\phi(y)}{dy}$ is the first derivative of function $\phi(y)$ with respect to y. Note that a steady state equilibrium is locally stable if the system converges to it when the initial condition is chosen from a neighborhood of this steady-state equilibrium, while a steady state equilibrium is globally stable if the system converges to it regardless of the level of the initial condition. Obviously, for y_{max} and y_{min}, we have

$$\left|\frac{dy^{i+1}}{dy^i}\Big|_{y_{max}}\right| = \left|1 + \sqrt{(T_2-1)^2 - 4T_1\alpha\xi}\right| > 1 \quad (41)$$

and

$$\left|\frac{dy^{i+1}}{dy^i}\Big|_{y_{min}}\right| = \left|1 - \sqrt{(T_2-1)^2 - 4T_1\alpha\xi}\right|. \quad (42)$$

It is easily known that y_{max} is unstable. To ensure that y_{min} is stable,

$$\left|\frac{dy^{i+1}}{dy^i}\Big|_{y_{min}}\right| < 1 \quad (43)$$

needs to be satisfied. Solving inequality (43) and combining (40) lead to the following constraint for step size

$$\max\{\frac{(T_2-1)^2 - 4}{4T_1\xi}, 0\} < \alpha < \min\{\frac{(T_2-1)^2}{4T_1\xi}, 1\}. \quad (44)$$

Based on the steady state equilibrium theory for nonlinear discrete dynamical system, we have the following theorem for convergence of proposed IGN algorithm

Theorem 1: Let Assumptions 1 and 2 hold. Under the sufficient condition (44), the estimate x^i obtained by the IGN algorithm is asymptotically upper bounded with respect to x^* as follows

$$\lim_{i\to\infty}\|x^i - x^*\| \leq y_{min}, \quad (45)$$

when the given any initial estimate x^0 is close to a local minimizer x^* within the radius y_{max} such that

$$\|x^0 - x^*\| < y_{max}, \quad (46)$$

where y_{min} and y_{max} are given (38) and (39), respectively.

Theorem 1 reveals the fact that the convergence of IGN algorithm can be guaranteed by selecting a reasonable step size depended on the specified NLLS application. One can obtain the step size by using a heuristic line search method.

The traditional Wolfe conditions [6], [18] are such an example. In each iteration, the agent finds the step size to satisfy the following conditions:

$$\|f_k(x_k^{i+1})\|^2 \leq \|f_k(x_k^i)\|^2 - b_1\alpha_k^i f_k^T(x^i)f_k(x^i)d_k^i \tag{47}$$

and

$$f_k^T(x_k^{i+1})F_k(x_k^{i+1})d_k^i \geq b_2\alpha_k^i f_k^T(x_k^i)F_k(x_k^i)d_k^i \tag{48}$$

with $0 < b_1 < b_2 < 1$. The first of the Wolfe conditions (47) guarantees that the step length α_k^i decreases the objective function $\|f_k\|^2$ at each iteration, while the second (48) tests whether the descent is sufficient. It is noted that the above line search method introduces more parameters and higher computational cost due to the heuristic searching. In practice, it is seldom considered as a necessary method.

Moreover, from (44) and (33), it can be known that the number N of agents over cyclic path has an effect on convergence of algorithm. That is, to improve the convergence performance, the practical step size α should be selected as a small value when N is large, and vice versa.

4) CONVERGENCE RATE

For further studying the convergence of IGN algorithm, we introduce the standard taxonomy [18], [41] of convergence rates.

Definition 1: Let $\{x^i\} \subset \mathbb{R}^M$ and $x^* \in \mathbb{R}^M$

(1) The sequence $\{x^i\}$ converges quadratically to x^* if x^i converges to x^* and there is $\mu > 0$ such that

$$\|x^{i+1} - x^*\| \leq \mu\|x^i - x^*\|^2. \tag{49}$$

(2) The sequence $\{x^i\}$ converges linearly to x^* if x^i converges to x^* and there is $\tau \in (0, 1)$ such that

$$\|x^{i+1} - x^*\| \leq \tau\|x^i - x^*\|. \tag{50}$$

From 34, obviously, $T_1 > 0$ is met. To guarantee $T_2 \in (0, 1)$, which is also required by $y_{max} > 0$, we have

$$0 < \Sigma_{max}\Sigma_{min} + \alpha(\sqrt{2}\omega e_{min} - \Sigma_{max}\Sigma_{min}) < \Sigma_{min}^2. \tag{51}$$

Solving (51), the sufficient condition (44) is further rewritten as follows:

$$\frac{\Sigma_{max}\Sigma_{min} - \Sigma_{min}^2}{\Sigma_{max}\Sigma_{min} - \sqrt{2}\omega e_{min}} < \alpha < \min\{\frac{\Sigma_{max}\Sigma_{min}}{\Sigma_{max}\Sigma_{min} - \sqrt{2}\omega e_{min}}, 1\} \tag{52}$$

and

$$\Sigma_{max}\Sigma_{min} > \sqrt{2}\omega e_{min}. \tag{53}$$

Therefore, $T_2 \in (0, 1)$ holds under the step size α is selected as a reasonable value based on the sufficient conditions (52) and (53). Consequently, the local estimate x^i in IGN algorithm converges at least linearly to the local minimizer x^*.

5) COMPARISON ON CONVERGENCE BEHAVIORS

In this section, we try to compare the convergence rate between IGN algorithm and CGN algorithm. From (22), the IGN update and the CGN update can be described by

$$x^{i+1} = x^i - \alpha\sum_{k=1}^{N} d_k^i(x_{k-1}^i) \tag{54}$$

and

$$x^{i+1} = x^i - \alpha d_N^i(x^i), \tag{55}$$

respectively.

Intuitively, form (54) and (55), we can obtain

$$\|x^i - x^{i+1}\| = \alpha\|\sum_{k=1}^{N} d_k^i(x_{k-1}^i)\| \tag{56}$$

for IGN, and

$$\|x^i - x^{i+1}\| = \alpha\|d_N^i(x^i)\| \tag{57}$$

for CGN. From (56) and (57), however, the conclusion of faster convergence of IGN cannot be obtained since it is difficult to determine qualitatively that $\sum_{k=1}^{N} d_k^i(x_{k-1}^i)\| > \|d_N^i(x^i)\|$. Therefore, we still consider the error evolution $\|x^i - x^*\|$ into convergence comparison instead of $\|x^i - x^{i+1}\|$.

Subtracting x^* on both sides of (54) and (55) and applying the norm operator on them, we get

$$\begin{aligned}\|x^{i+1} - x^*\| &= \|x^i - x^* - \alpha\sum_{k=1}^{N} d_k^i(x_{k-1}^i) + \alpha\sum_{k=1}^{N} d_k^i(x^*)\| \\ &\leq \|x^i - x^*\| + \alpha\|\sum_{k=1}^{N} d_k^i(x_{k-1}^i) - \sum_{k=1}^{N} d_k^i(x^*)\| \\ &\leq \|x^i - x^*\| + \alpha\sum_{k=1}^{N}\|d_k^i(x_{k-1}^i) - d_k^i(x^*)\| \end{aligned}\tag{58}$$

and

$$\begin{aligned}\|x^{i+1} - x^*\| &= \|x^i - x^* - \alpha d_N^i(x^i) + \alpha d_N^i(x^*)\| \\ &\leq \|x^i - x^*\| + \alpha\|d_N^i(x^i) - d_N^i(x^*)\|, \end{aligned}\tag{59}$$

since $q_k(x^*) = 0$ leads to $d_k^i(x^*) = Q_k^{-1}(x^*)q_k(x^*) = 0$.

Because of

$$\begin{aligned}&\|d_k^i(x_{k-1}^i) - d_k^i(x^*)\| \\ &= \|Q_k^{-1}(x_{k-1}^i)q_k(x_{k-1}^i) - Q_k^{-1}(x_{k-1}^i)q_k(x^*)\| \\ &\leq \|Q_k^{-1}(x_{k-1}^i)\|\|q_k(x_{k-1}^i) - q_k(x^*)\| \\ &\leq \frac{1}{k\sigma_{min}^2}\|\sum_{j=1}^{k} F_j^T(x_{k-1}^i)f_j(x_{k-1}^i) - \sum_{j=1}^{k} F_j^T(x^*)f_j(x^*)\| \\ &\leq \frac{1}{k\sigma_{min}^2}\sum_{j=1}^{k}\|F_j^T(x_{k-1}^i)f_j(x_{k-1}^i) - F_j^T(x^*)f_j(x^*)\| \\ &\leq \frac{1}{k\sigma_{min}^2}\sum_{j=1}^{k}\gamma_f\|x_{k-1}^i - x^*\| \end{aligned}\tag{60}$$

and

$$\begin{aligned}
&\|d_N^i(x^i) - d_N^i(x^*)\| \\
&\quad = \|Q_N^{-1}(x^i)q_N(x^i) - Q_N^{-1}(x^i)q_N(x^*)\| \\
&\quad \leq \|Q_N^{-1}(x^i)\|\|q_N(x^i) - q_N(x^*)\| \\
&\quad \leq \frac{1}{N\sigma_{min}^2}\|\sum_{j=1}^{N} F_j^T(x^i)f_j(x^i) - \sum_{j=1}^{N} F_j^T(x^*)f_j(x^*)\| \\
&\quad \leq \frac{1}{N\sigma_{min}^2}\sum_{j=1}^{N}\|F_j^T(x^i)f_j(x^i) - F_j^T(x^*)f_j(x^*)\| \\
&\quad \leq \frac{1}{N\sigma_{min}^2}\sum_{j=1}^{N}\gamma_f\|x^i - x^*\| \\
&\quad = \frac{\gamma_f}{\sigma_{min}^2}\|x^i - x^*\|, \qquad (61)
\end{aligned}$$

we rewrite (58) and (59) as

$$\|x^{i+1} - x^*\| \leq \|x^i - x^*\| + \alpha\sum_{k=1}^{N}\frac{1}{k\sigma_{min}^2}\sum_{j=1}^{k}\gamma_f\|x_{k-1}^i - x^*\| \qquad (62)$$

and

$$\|x^{i+1} - x^*\| \leq (1 + \alpha\frac{\gamma_f}{\sigma_{min}^2})\|x^i - x^*\| \qquad (63)$$

for IGN and CGN, respectively. By setting $k = N$ for the right side of (62) and comparing it with the right side of (63), we get

$$\begin{aligned}
&\|x^i - x^*\| + \alpha\sum_{k=1}^{N}\frac{1}{k\sigma_{min}^2}\sum_{j=1}^{k}\gamma_f\|x_{k-1}^i - x^*\| \\
&\quad \geq \|x^i - x^*\| + \alpha\frac{\gamma_f}{\sigma_{min}^2}\|x^i - x^*\| \\
&\quad = (1 + \alpha\frac{\gamma_f}{\sigma_{min}^2})\|x^i - x^*\|. \qquad (64)
\end{aligned}$$

The recursions (62) and (63) describe how the estimate error evolves over time for IGN and CGN, respectively. It can be known for CGN that the error $\|x^i - x^*\|$ asymptotically becomes smaller as the iteration goes on, since $1 + \alpha\frac{\gamma_f}{\sigma_{min}^2}$ is bounded. The observation leads to that the error curve becomes less steep with the increase of iteration till the system reaches the stability. A similar conclusion can be obtained for IGN algorithm.

Moreover, by comparing (62) (63) as well as (64), we see that the IGN algorithm provides a larger reduction room than the CGN algorithm in the sense of error norm, which means that the estimate x^i in IGN is more likely to be closer to the minimizer x^* at every iteration than that in CGN. Fig. 2 illustrates the evolution of estimation for IGN and CGN over discrete time. Under x^i tends to x^* in the sense of norm, i.e., $\lim_{i\to\infty}\|x^i - x^*\| = 0$, it can be believed that IGN has a faster convergence than CGN. In the following simulation section, we will confirm numerically the above conclusions.

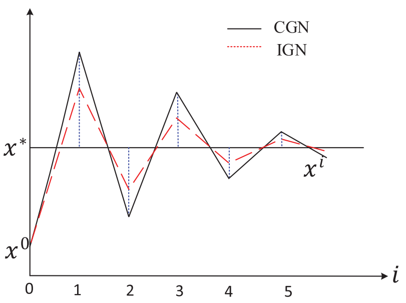

FIGURE 2. **An illustration of evolution for IGN and CGN over discrete time.**

V. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of proposed scheme by applying it to address a typical target localization NLLS problem, since location-based service has become indispensable part of our smart life. Our main objective is to verify numerically the effectiveness of proposed fusion estimation algorithms and the correctness of our analysis. All the computations were performed with MATLAB 2016a software, which is installed on a Windows 10 64-bit operating system with 8 GB of RAM and an Intel Core i7-8750H processor.

A. RANGE-BASED TARGET LOCALIZATION MODEL

Estimating the targets position or trajectory is demanded in various position-based applications. For instance, in a wireless sensor network, node localization, which is also termed as target localization when the target is equipped with sensor and communication devices, is a key technology for supporting many applications and protocols [42], [43], such as environment monitoring, object detection, location-based routing protocols and clustering algorithms. The distributed implementation over network and rapid convergence to reduce energy consumption are vital for target localization algorithms, such that more energy can be used for the main functions of the above applications.

A cyclic path through the entire network is necessary for our algorithms. It is difficult to determine such a cyclic path that covers all nodes, which is referred to as the Hamiltonian path problem in graph theory, since it is NP-complete. However, this problem can be solved by constructing an approximate Hamiltonian path in a distributed way. Several distributed methods can be found in the literatures and perform very well in practice. For instance, two distributed implementations of simulated annealing and branch-and-bound algorithms are described in [44], [45], while a considerable distributed algorithm is proposed in [46], where every node makes local decisions based on a heuristic approach and the distributed nature of the network is exploited to speed up the path construction process. Such algorithms provide the solution for establishing a cycle through the network with a high success rate. For the case of target localization, a pre-deployed infrastructure will be preferred for an easy implementation consideration. For example, in a harsh iindustrial environment or smart home, the agents located at the boundary of the monitored region are more easily deployed and hidden than elsewhere as shown in Fig. 3.

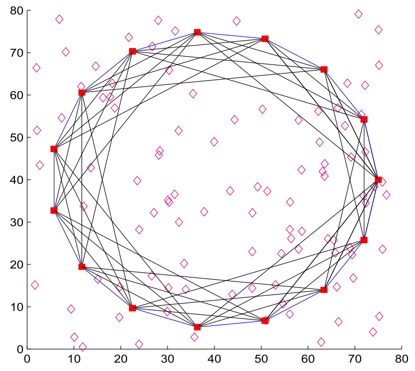

FIGURE 3. The configuration of the network with 100 random targets and 15 agents.

For the consistency of the description, we model the target localization problem by using previous same notations as much as possible. Consider an adaptive network consisting of N agents randomly deployed on a monitored area, where the unknown targets are located. Each agent $k \in \mathcal{N}$ is aware of its own location with the coordinate $c_k = [c_{k,1}, \cdots, c_{k,M}]^T$ that is not known for other agents, where $M = 2$ for two-dimensional plane or $M = 3$ for three-dimensional space, and they are connected in the cyclic form as illustrated in Fig. 1. Our goal is to estimate the actual position of the target denoted by $x = [x_1, \cdots, x_M]^T$.

The real distance R_k between the target and the kth agent is obtained by

$$R_k = \|x - c_k\|, \tag{65}$$

while the measured distance r_k between the agent k and the target can be obtained by

$$r_k = R_k(1 + noise_k), \tag{66}$$

where $noise_k$ is the Gauss white noise that follows the normal distribution $N(0, \psi_k^2)$ since the measurement is always proportional to the real distance.

Let the error between the real distance and the measured distance be denoted by

$$f_k(x) = \|R_k\|^2 - \|r_k\|^2,$$

and

$$f(x) = [f_1(x), \cdots, f_N(x)]^T,$$

thus, the objective is to minimize the cost function

$$\|f(x)\|^2 = \sum_{k=1}^{N} |f_k(x)|^2 = \sum_{k=1}^{N} |x^T x - 2c_k^T x + \|c_k\|^2 - \|r_k\|^2|^2. \tag{67}$$

Thus, the proposed CGN and IGN algorithms can be easily applied to estimate the target position by giving an initial point for the first iteration x^0 that is sufficiently good, which is not hard to achieve. Some traditional localization methods such as triangulation [47] provide good initial estimates that close to the true target location with low communication and computation cost despite the coarse-grained localization error.

B. NUMERICAL RESULTS

In this section, we provide the performance comparison based on some key metrics. In our scenarios, 100 static targets are randomly deployed in a $80m \times 80m$ two-dimensional surveillance area. In Fig. 3 that is used to describe the deployment, the small diamonds in red represent the actual position of targets, while the 15 agents represented by the red squares are deployed uniformly in a ring way. We connect geographically each agent with its d nearest neighbors, where d denotes the connectivity on any agent. Such that a cycle topology or common random topology is formed when we set $d = 2$ and $d \in \{4, 6, 8, \ldots\}$, respectively. The above scenario means that 100 independent overdetermined systems with $M = 2$ and $N = 15$ will be solved by using the proposed algorithms.

We assume that the surveillance area is inside the sensing range of all agents, so that the noisy distance measurements can be obtained based on the equality (66). Fig. 4 shows the estimated positions represented by blue diamonds for 100 targets using IGN algorithm, in which the step size $\alpha = 0.003$ and the noise variance for agent k is selected randomly from the range (0, 0.03). From Fig. 4, we see that IGN algorithm provides better accuracy for the targets in the middle of area than the ones nearby the boundary. The main reason is the unbalanced distance distribution between targets and agents, which results in the measurement error $f_k(x)$ in (67) with large deviation between different agent k due to the model (66). Two methods can be considered to mitigate the negative impact. One is to reduce the size of monitored area by using the clustering techniques. Based on the noise level, the surveillance area can be divided into several subareas with different size. At this point, our algorithms provide the scalability to match different network topology; another is

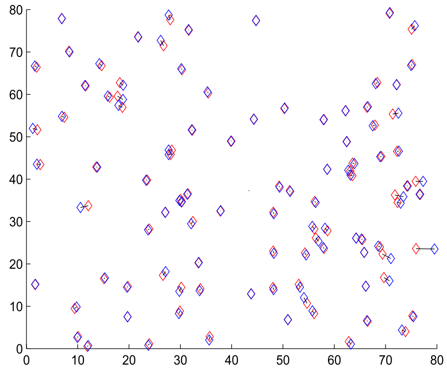

FIGURE 4. Position estimates using IGN algorithm under $\psi_k^2 \in (0, 0.03)$ and $\alpha = 0.003$.

that the static targets located in middle can become the new agents after they obtain their own accurate position.

The convergence rate is a key concern for evaluating algorithm performance. Fig. 5 and Fig. 6 show the transient error curves by choosing a random target under different step size and number of agents, respectively. In Fig. 5, a small step size leads to a slow convergence rate for both of algorithms, while the convergence performance of IGN algorithm is significantly better than CGN algorithm whether the step size with a small value $\alpha = 0.001$ or a larger value $\alpha = 0.003$. After about 500 iterations under $\alpha = 0.003$, IGN algorithm reaches the steady state, while CGN algorithm requires at least 1800 iterations.

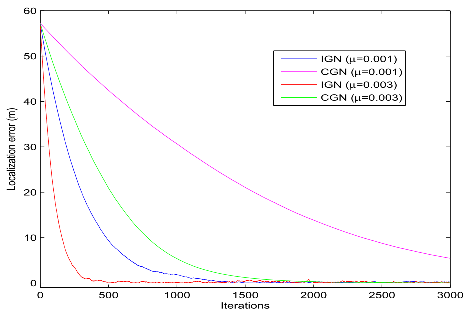

FIGURE 5. Convergence performance comparison for IGN and CGN under $N = 5$ and different step sizes.

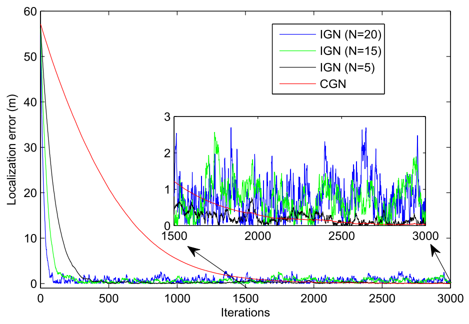

FIGURE 6. Convergence performance comparison for IGN and CGN with different number of agents under $\mu = 0.003$.

Fig. 6 shows the effect of number N of agents on convergence performance. First, the IGN algorithm converges much faster than the CGN algorithm for any given N. Second, the number N of agents have greater impact on convergence for IGN algorithm than CGN algorithm. The difference for CGN algorithm between different number of agents is not shown in the figure since it is imperceptible. Third, for a larger number $N = 20$, IGN algorithm shows big oscillations during converging. The above observations can be explained by the inherent mechanism of incremental update. In IGN algorithm, each agent on the cyclic path can obtain the new estimate x_k^i that is used to substitute the old estimate x^i of the centralized Gauss-Newton step. Intuitively, our intention is to generate a deeper descent step on the cost function, thereby obtaining an improvement of convergence speed. In this regard, a big value of N will result in the faster convergence. On the other hand, a excessive number of agents will also cause the instability of IGN algorithm during converging due to the fixed step size α that cannot guarantee the descent direction on every Gauss-Newton step with such a way that decreases the objective cost, i.e., $\|f(x^{i+1})\|^2 \leq \|f(x^i)\|^2$.

This observation can also be explained from the obtained constraint condition (44), where the upper and lower bound of step size are inversely proportional to the positive constant ξ associated with the number N of agents from the definition (33). Therefore, a conclusion is that the IGN algorithm is more sensitive to the number of agents than the CGN algorithm when the same large step size is used. As we can see from Fig. 6, the oscillations arise for IGN with $N = 20$. As a result, there is a tradeoff between the number of agents and the convergence performance for the IGN algorithm.

For the above observed tradeoff, by choosing a smaller step size or adopting the variable step size like (47) and (48) to ensure the convergence, both high convergence speed and low oscillation can be achieved by IGN algorithm when N is large. To illustrate, 20 and 30 agents that deploy uniformly at the circle are used to locate the random targets in our simulation, respectively. The results are shown in Fig. 7, where the oscillations are mitigated by using a relatively small step size $\alpha = 0.001$ while the high convergence speed and the small localization error are maintained, although the oscillation amplitude is large for IGN algorithm when $\alpha = 0.003$.

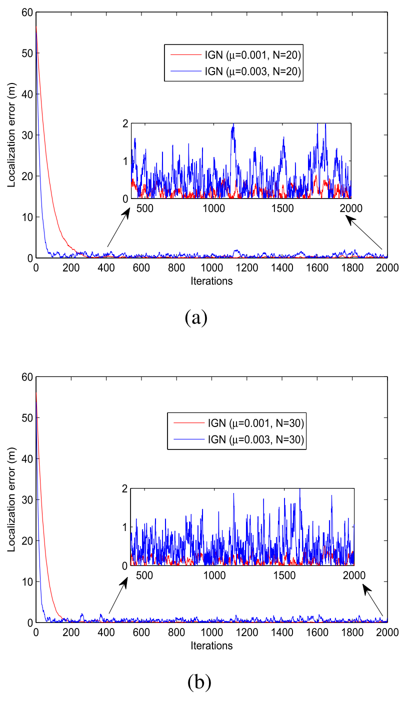

FIGURE 7. Convergence performance comparison for IGN with different step size when N = 20 and 30.

Fig. 8 illustrates the running time of IGN and CGN with different number of agents. As the number of iterations

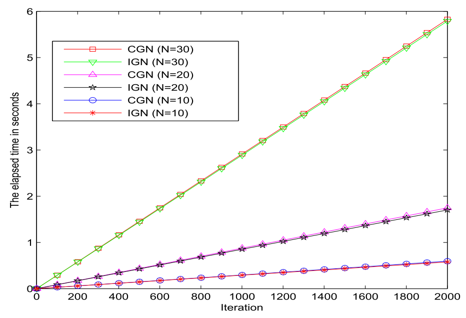

FIGURE 8. The elapsed time in seconds for IGN and CGN with different number of agents.

increases, the running time grows linearly for both IGN and CGN. For each iteration, it is known that IGN algorithm spend more time than CGN algorithm on the floating point operations. Fig. 8 shows that the running time difference between IGN and CGN is very small even if the number of iterations is large (e.g., $i = 2000$). On the other hand, from Fig. 6 and Fig. 7, about 300 iterations are required for the convergence of IGN, while over 1800 iterations are required for CGN. Thus, IGN can achieve an obvious reduction on the total convergence time, which is counted based on the chosen example of target location in our simulation.

The recently proposed diffusion Gauss-Newton (DGN) method [6], [48] is an effective cooperative scheme based on random diffusion topology to solve the same networked NLLS problems. We compare their performance by selecting different connectivity d for DGN method. DGN is a consensus-based method that can achieve the network-wide consensus estimation, and its performance highly depends on the network connectivity. From Fig. 9, we see that IGN has the similar convergence and estimation performance with DGN under high connectivity, while the communication cost of DGN is $d/2$ times ($d \geq 2$) higher than that one of IGN.

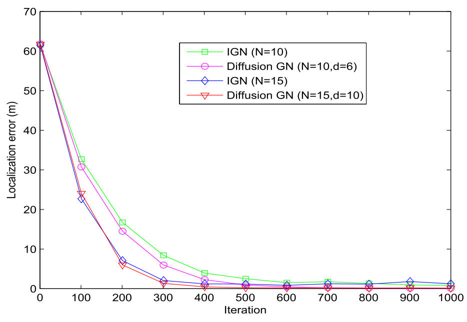

FIGURE 9. Performance comparison for IGN and diffusion GN with same number of agents.

C. COMPARISON WITH THE MODIFIED VERSIONS OF GENERAL OPTIMIZATION METHODS

To compare our methods with the general optimization methods for solving the unconstrained NLLS problem, we specify several comparable methods into our cyclic and incremental model, including the cyclic gradient, the incremental gradient and the cyclic Newton. The reason to exploit these methods is that they have very good adaptability to the NLLS problems with various objective functions. Although many excellent methods show improved performance in terms of computation and memory, such as BFGS and other Quasi-Newton methods, they require strong assumptions on both f and the initial iterate x^0, where x^0 must not only be close to x^* but also that the Hessian must be close to $\nabla^2(f(x^*))$ [18]. For example, the modified BFGS and incremental Newton methods do not guarantee the convergence through repeated experiments in this example, since the incremental technique applying to Newton type methods is unconfirmed and needs to be studied carefully.

Fig. (10) shows the results of a comparison based on the same parameter setup as Fig. (5). The following remarks are provided. (1) It should be noted that we use different step size in this experiment. The reason is that gradient type methods require a very small step size to ensure their convergence (e.g., 5×10^{-8} in Fig. (10)), since they provides the slow global convergence property. On the other hands, Newton type methods allow a large step size to provide the fast local convergence. (2) The proposed IGN and known diffusion Gauss-Newton outperform other methods in terms of convergence rate. (3) Comparing incremental gradient with cyclic/centrailzed gradient, we reach the previous conclusion that the incremental strategy has obvious effect on improving convergence performance.

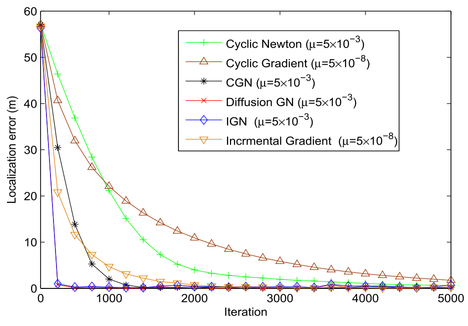

FIGURE 10. Performance comparison for proposed methods with some modified optimization methods.

Although the IGN method shows obvious improvements, its computational cost needs to be evaluated. To do this, we split the overall algorithms into four elementary operations without any accelerating computational techniques, i.e., the first partial derivative of the objective function with respect to a scalar, the inverse of an $M \times M$ matrix, the multiplication and addition of any two scalars. Table 1 lists the averaged computational cost over network per iteration, where a new notation n_k denotes the number of neighboring agents of any node k. Combining Table 1 and Fig. 10, it is observed that the first order type methods have the lowest complexity for all operations. For Newton and Gauss-Newton methods, the differences between them are mainly reflected in the number of matrix inversion, where diffusion strategy

TABLE 1. Computational complexity for each agent per iteration in different algorithms.

Operator / Algorithm	Partial derivative	Inverse	Multiplication	Addition
Cyclic Gradient	M	0	M	$(1-\frac{1}{N})M$
Incremental Gradient	M	0	M	$(1-\frac{1}{N})M$
Cyclic Newton	$2M$	$\frac{1}{N}$	$M^2+\frac{M^2}{N}+M$	$M^2+M-\frac{2}{N}M$
Diffusion GN	$M\frac{n_k}{N}$	$\frac{n_k}{N}$	$(2M^2+M)\frac{n_k}{N}$	$2M^2\frac{n_k}{N}-\frac{M^2}{N}-\frac{M}{N}$
CGN	M	$\frac{1}{N}$	$M^2+\frac{M^2}{N}+M$	$M^2+M-\frac{2}{N}M$
IGN	M	1	$2M^2+M$	$2M^2-\frac{M^2}{N}-\frac{M}{N}$

over a fully connected network leads to the same complexity with incremental technique. However, in the case, the communication cost of diffusion Gauss-Newton is N times more than our IGN method. In conclusion, with fewer iterations to reach convergence, our IGN method outperforms the various existing algorithms with some modifications in terms of total complexity.

VI. CONCLUSIONS AND FUTURE WORKS

To obtain the better learning effect, in this paper, we propose the sequential fusion estimation algorithms for Gauss-Newton method over multi-agent networked systems. As the basis for our work, we first propose a cycle-based Gauss-Newton algorithm, which has the same estimation performance and the improved robustness with comparison to the centralized Gauss-Newton algorithm. Subsequently, by utilizing the updated estimate obtained by previous agent on the cyclic path to compute the new descent direction, an incremental learning scheme is developed. The benefit of the resulted IGN algorithm is that the convergence rate is significantly improved, which is evidenced from both convergence analysis and simulation based on the target localization scenario in a wireless agent sensor network. The detailed convergence analysis for IGN will be helpful for developing the iteration-based incremental Newton-type method for optimization over a cyclic network structure. Not only for cycle-based Gauss-Newton method, our analysis provides a new perspective and insightful approach, from which the convergence behavior of Newton type method in a diffusion or consensus way can also be studied. Our methods can also be considered to solve the underdetermined NLLS problems in future works. Moreover, the network delay will result in the degradation of estimation performance because of the non-realtime data. The effect of such asynchronous cooperation on learning behavior needs to be evaluated and overcome.

REFERENCES

[1] M. Gao, S. Yang, and L. Sheng, "Distributed fault estimation for time-varying multi-agent systems with sensor faults and partially decoupled disturbances," *IEEE Access*, vol. 7, pp. 147905–147913, 2019.

[2] Z. Peng, Y. Li, and G. Hao, "The research on distributed fusion estimation based on machine learning," *IEEE Access*, vol. 8, pp. 38174–38184, 2020.

[3] Z. Deng, "Distributed algorithm design for resource allocation problems of second-order multiagent systems over weight-balanced digraphs," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 6, 2019, doi: 10.1109/TSMC.2019.2930672.

[4] X. He, D. W. C. Ho, T. Huang, J. Yu, H. Abu-Rub, and C. Li, "Second-order continuous-time algorithms for economic power dispatch in smart grids," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1482–1492, Sep. 2018.

[5] P. Duan, K. Liu, N. Huang, and Z. Duan, "Event-based distributed tracking control for second-order multiagent systems with switching networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, May 18, 2018, doi: 10.1109/TSMC.2018.2833098.

[6] M. Wu, N. Xiong, and L. Tan, "Adaptive range-based target localization using diffusion Gauss-Newton method in industrial environments," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5919–5930, Nov. 2019.

[7] D. Chwa, A. Dani, H. Kim, and W. Dixon, "Camera motion estimation for 3-D structure reconstruction of moving objects," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2012, pp. 1788–1793.

[8] X. Xiong and F. De la Torre, "Supervised descent method for solving nonlinear least squares problems in computer vision," 2014, *arXiv:1405.0601*. [Online]. Available: http://arxiv.org/abs/1405.0601

[9] Z. Lu, L. Dong, and J. Zhou, "Nonlinear least squares estimation for parameters of mixed Weibull distributions by using particle swarm optimization," *IEEE Access*, vol. 7, pp. 60545–60554, 2019.

[10] Y. Ren and D. Goldfarb, "Efficient subsampled Gauss-Newton and natural gradient methods for training neural networks," 2019, *arXiv:1906.02353*. [Online]. Available: http://arxiv.org/abs/1906.02353

[11] X. He, D. Mudigere, M. Smelyanskiy, and M. Takác, "Distributed hessian-free optimization for deep neural network," in *Proc. Workshops 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–6.

[12] A. Botev, H. Ritter, and D. Barber, "Practical Gauss-Newton optimisation for deep learning," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 557–565.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.

[14] N. Drucker, H.-M. Ho, J. Ouaknine, M. Penn, and O. Strichman, "Cyclic-routing of unmanned aerial vehicles," *J. Comput. Syst. Sci.*, vol. 103, pp. 18–45, Aug. 2019.

[15] G. Han, L. Zhou, H. Wang, W. Zhang, and S. Chan, "A source location protection protocol based on dynamic routing in WSNs for the social Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 689–697, May 2018.

[16] S. I. Chu, C. Y. Lien, W. C. Lin, Y. J. Huang, C. L. Pan, and P. Y. Chen, "A survey of localization in wireless sensor network," *Int. J. Distrib. Sensor Netw.*, vol. 2012, no. 1, pp. 385–391, 2014.

[17] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA, USA: SIAM, 1996.

[18] C. T. Kelley, *Iterative Methods for Optimization*. Philadelphia, PA, USA: SIAM, 1999.

[19] F. De la Torre and M. Hoai Nguyen, "Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[20] É. L. Souza, E. F. Nakamura, and R. W. Pazzi, "Target tracking for sensor networks: A survey," *ACM Comput. Surv.*, vol. 49, no. 2, p. 30, Jun. 2016.

[21] J. R. Jensen, M. G. Christensen, and S. H. Jensen, "Nonlinear least squares methods for joint DOA and pitch estimation," *IEEE Trans. Audio Speech, Lang. Process.*, vol. 21, no. 5, pp. 923–933, May 2013.

[22] Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin, "Distributed Newton methods for regularized logistic regression," in *Advances in Knowledge Discovery and Data Mining*. Cham, Switzerland: Springer, 2015, pp. 690–703.

[23] X. Li and A. Scaglione, "Convergence and applications of a gossip-based Gauss-Newton algorithm," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5231–5246, Nov. 2013.
[24] X. Li and A. Scaglione, "Robust decentralized state estimation and tracking for power systems via network gossiping," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1184–1194, Jul. 2013.
[25] A. Minot, Y. M. Lu, and N. Li, "A distributed Gauss-Newton method for power system state estimation," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3804–3815, Sep. 2016.
[26] M. Cosovic and D. Vukobratovic, "Distributed Gauss-Newton method for AC state estimation: A belief propagation approach," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Nov. 2016, pp. 643–649.
[27] B. Bejar, P. Belanovic, and S. Zazo, "Distributed Gauss-Newton method for localization in ad-hoc networks," in *Proc. Conf. Rec. Forty 4th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2010, pp. 1452–1454.
[28] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization—I: Algorithm," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2162–2175, Sep. 2013.
[29] S. Wang, Z.-H. Zhou, M. Ge, and C. Wang, "Resource allocation for heterogeneous cognitive radio networks with imperfect spectrum sensing," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 464–475, Mar. 2013.
[30] C. Li, P. Shen, Y. Liu, and Z. Zhang, "Diffusion information theoretic learning for distributed estimation over network," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4011–4024, Aug. 2013.
[31] F. S. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1465–1480, Apr. 2011.
[32] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.
[33] M. Wu and L. Tan, "An adaptive distributed parameter estimation approach in incremental cooperative wireless sensor networks," *AEU-Int. J. Electron. Commun.*, vol. 79, pp. 307–316, Sep. 2017.
[34] G. C. Calafiore, L. Carlone, and M. Wei, "A distributed Gauss-Newton approach for range-based localization of multi agent formations," in *Proc. IEEE Int. Symp. Comput.-Aided Control Syst. Design*, Sep. 2010, pp. 1152–1157.
[35] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ., 2009.
[36] S. Campbell and C. Meyer, *Generalized Inverses of Linear Transformations*. Philadelphia, PA, USA: SIAM, 2009.
[37] T. Yamamoto, "Historical developments in convergence analysis for Newton's and Newton-like methods," *J. Comput. Appl. Math.*, vol. 124, nos. 1–2, pp. 1–23, Dec. 2000.
[38] A. Galántai, "The theory of Newton's method," *J. Comput. Appl. Math.*, vol. 124, nos. 1–2, pp. 25–44, 2000.
[39] C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[40] O. Galor, *Discrete Dynamical Systems*. Cham, Switzerland: Springer, 2007.
[41] J. Nocedal and S. J. Wright, *Numerical Optimization*. Cham, Switzerland: Springer, 2006.
[42] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: A survey," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2419–2436, 2013.
[43] W. Xu, F. Quitin, M. Leng, W. P. Tay, and S. G. Razul, "Distributed localization of a RF target in NLOS environments," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1317–1330, Jul. 2015.
[44] J. R. A. Allwright and D. B. Carpenter, "A distributed implementation of simulated annealing for the travelling salesman problem," *Parallel Comput.*, vol. 10, no. 3, pp. 335–338, May 1989.
[45] S. Tschoke, R. Lubling, and B. Monien, "Solving the traveling salesman problem with a distributed branch-and-bound algorithm on a 1024 processor network," in *Proc. 9th Int. Parallel Process. Symp.*, Apr. 1995, pp. 182–189.
[46] A. Gasparri, B. Krishnamachari, and G. S. Sukhatme, "A framework for multi-robot node coverage in sensor networks," *Ann. Math. Artif. Intell.*, vol. 52, nos. 2–4, pp. 281–305, Apr. 2008.
[47] K. K. Saab and S. S. Saab, "Application of an optimal stochastic Newton-raphson technique to triangulation-based localization systems," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2016, pp. 981–986.
[48] N. Xiong, M. Wu, V. C. M. Leung, and L. T. Yang, "The effective cooperative diffusion strategies with adaptation ability by learning across adaptive network-wide systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Aug. 13, 2019, 10.1109/TSMC.2019.2931060.

MOU WU (Member, IEEE) received the Ph.D. degree in radio physics from Central China Normal University, in June 2015. Since July 2015, he has been with the School of Computer Science and Technology, Hubei University of Science and Technology. In 2018, he joined the College of Intelligence and Computing, Tianjin University, as a Postdoctoral Researcher. His research interests include wireless sensor networks, distributed algorithms for performance optimization over networks, and computer communication.

LIANGJI ZHONG received the B.S. degree from the Hubei University of Technology, in 2003, and the M.S. degree from Wuhan University, in 2009. He is currently an Associate Professor with the School of Computer Science and Technology, Hubei University of Science and Technology, China. His research interests include the Internet of Things, smart city and home, and design in single-chip microcomputer.

LIANSHENG TAN received the Ph.D. degree from Loughborough University, U.K., in 1999. He was a Research Fellow with the Research School of Information Sciences and Engineering, The Australian National University, Australia, from 2006 to 2009, and a Postdoctoral Research Fellow with the School of Information Technology and Engineering, University of Ottawa, Canada, in 2001. He has also held a number of visiting research positions at Loughborough University, the University of Tsukuba, the City University of Hong Kong, and The University of Melbourne. He is currently a Professor with the Department of Computer Science, Central China Normal University. His research interests include modeling, congestion control analysis, and performance evaluation of computer communication networks, resource allocation and management of wireless and wireline networks, and routing and transmission control protocols.

NAIXUE XIONG (Senior Member, IEEE) received the Ph.D. degree in sensor system engineering from Wuhan University and the Ph.D. degree from the Japan Advanced Institute of Science and Technology. He was with Georgia State University, Wentworth Technology Institution, and Colorado Technical University for about ten years. He is currently an Associate Professor with the Department of Mathematics and Computer Science, Northeastern State University, OK, USA. He has published over 200 international journal articles and over 100 international conference papers. His research interests include cloud computing, security and dependability, and parallel and distributed computing, networks, and optimization theory. He is serving as the Editor-in-Chief, an Associate Editor, or an Editor Member for over ten international journals.