ALGEBRAIC CONNECTIVITY ESTIMATION BASED ON DECENTRALIZED INVERSE POWER ITERATION

Yue Wei, Hao Fang, Jie Chen, and Bin Xin

ABSTRACT

In this work, we propose a new scheme to estimate the algebraic connectivity of the graph describing the network topology of a multi-agent system. We consider network topologies modeled by undirected graphs. The main idea is to propose a new decentralized conjugate gradient algorithm and a decentralized compound inverse power iteration scheme. The matrix inversion computation in this scheme is replaced by solving the non-homogeneous linear equations relying on the proposed decentralized conjugate gradient algorithm. With this scheme, we can achieve a fast convergence rate in estimating the algebraic connectivity by setting the parameter $\mu$ properly. Simulation results demonstrate the effectiveness of the proposed scheme.

Key Words: Graph Laplacian, inverse power iteration, decentralized estimation.

I. INTRODUCTION

Communication between the agents in multi-agent systems has attracted much attention due to the importance of information exchange. It relies on the graph constructs, especially the distance-based weighted graphs. The information spread in the network usually attenuate with the distance [1–3]. In this context, multi-hop communication means the same to network connectivity, which describes the property of a graph to spread message in the network.

Algebraic graph theory [4,5] can offer great tools to analyse a graph. For instance, for the Laplacian matrix associated to an undirected graph, the spectrum can display some topological properties like average degree, algebraic connectivity, connectivity measures and spectral gap [6]. In multi-agent systems, such information is very important to indicate the dynamics of the whole multi-agent system [7]. The second smallest eigenvalue of the Laplacian matrix, which is called the algebraic connectivity, is a significant measurement of the convergence rate of consensus algorithms [8]. It's also used in more general cases, like leader-follower control of multi-agent systems with local interactions which is described by the Laplacian matrix [9].

Decentralized control of multi-agent systems (MAS) has many applications, such as exploration of unknown environments and connectivity preservation [10,11]. Estimating the algebraic connectivity of the MAS topology in a decentralized manner becomes the key issue to achieve the flocking property, e.g., connectivity preservation, see [12–14]. A lot of estimation algorithms of the Laplacian eigenvalue are based on the centralized methods, which rely on the knowledge of all the elements of the Laplacian matrix. Some papers about decentralizing these estimation algorithms have been noticed [15–18]. Paper [15] presented a decentralized orthogonal iteration algorithm for computing eigenvectors of a symmetric matrix, but the initialization step of this algorithm is not decentralized. Reference [16] proposed a decentralized estimation scheme based on the power iteration algorithm combined with the PI average consensus estimators [16,19]. However, the convergence rate of the power iteration method depends on the ratio of the sub-dominant eigenvalue to the dominant eigenvalue of the Laplacian matrix. The authors wish to thank the Beijing Advanced Innovation Center for Intelligent Robots and Systems (Beijing Institute of Technology) for its many kindly help.

© 2016 Chinese Automatic Control Society and John Wiley & Sons Australia, Ltd
combined with the relaxed Jacobi method, but the convergence rate of the relaxed Jacobi iteration method depends on the selection of the relaxation factor, which is not easy to be determined. Reference [18] introduced a novel interaction rule to convert the decentralized eigenvalue estimation problem into a standard signal processing problem. Since this algorithm must estimate the whole spectrum of the Laplacian, it may cost unnecessary time and energy in the algebraic connectivity estimation problem.

With respect to the state of the art in estimation of the algebraic connectivity, the goal of this paper is to present a novel decentralized algebraic connectivity estimator based on compound inverse power iteration method with fast convergence rate. The contributions of this paper are listed as follows:

1. We propose a new decentralized conjugate gradient algorithm in which the problem of using the global information is handled by a distributed PI average consensus estimator, so that the non-homogeneous linear equations can be solved in a totally decentralized way.

2. A new scheme of decentralized estimation of \( \lambda_2 \) as well as its corresponding eigenvector \( \theta_2 \) based on the decentralized inverse power iteration algorithm is built.

The remainder of the paper is organized as follows. In Section II, the problem is formally formulated and preliminaries are provided. An algorithm for fully decentralized algebraic connectivity estimation based on inverse power iteration is presented in Section III. Non-trivial comparative simulation results are provided in Section IV. Concluding remarks as well as directions for future work are stated in Section V.

II. PROBLEM FORMULATION AND PRELIMINARIES

Consider a group of \( N \) mobile agents moving in the two-dimensional Euclidean plane with second-order dynamics, which is described by

\[
\begin{align*}
\dot{q}_i &= u_i, \\
\dot{u}_i &= q_i - p_i, \\
& i = 1, 2, \cdots, N
\end{align*}
\]

where \( q_i \in \mathbb{R}^2 \) is the position vector of agent \( i \), \( p_i \in \mathbb{R}^2 \) is the velocity vector of agent \( i \), \( u_i \in \mathbb{R}^2 \) is the control input of agent \( i \). \( q_{ij} = q_i - q_j \) is the position vector for agent \( i \) and agent \( j \). Let \( q = [q_1^T, q_2^T, \cdots, q_N^T] \) denotes the stack position vector of the multi-agent systems. Each agent is considered to have the same limited communication radius \( R \). The state dependent, switched system (1) induces the undirected dynamic graph

\[
\mathcal{G} = \{\mathcal{V}, \mathcal{A}, \mathcal{E}\}
\]

where \( \mathcal{V} = \{1, 2, \cdots, N\} \) corresponds to the \( N \) agents, \( N \) is assumed known in this paper. If not, the information about \( N \) can be obtained straightforwardly by using the algorithm proposed by [20]. \( \mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N} \) is the weighted adjacency matrix. \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \) is the set of communication links among agents. \( a_{ij} \) is defined as

\[
a_{ij} (\|q_j\|_2) = \begin{cases} 
1, & \|q_j\|_2 \in [0, \tau R) \\
\frac{1}{2} \left[ 1 + \cos \left( \pi \frac{\|q_j\|_2}{R - \tau} \right) \right], & \|q_j\|_2 \in [\tau R, R) \\
0, & \text{otherwise}
\end{cases}
\]

where \( \|q_j\|_2 \) is the Euclidean norm of the vector \( q_j \), and we assume \( a_{ij} = 0 \). \( \tau \) is the parameter showing how the communication quality changes with the distance between agent \( i \) and \( j \). \( 0 < \tau < 1 \). \( \mathcal{N}_i = \{ j \in \mathcal{V} \mid (i, j) \in \mathcal{E} \} \) is the neighbour set of an agent \( i \) with symmetry property \( i \in \mathcal{N}_j \iff j \in \mathcal{N}_i \). Define the weighted degree matrix \( \mathcal{D} = \text{diag}(d_i) \) with the degree \( d_i = \sum_{j \in \mathcal{N}_i} a_{ij} \) as the diagonal elements. The Laplacian matrix of \( \mathcal{G} \) is given by

\[
\mathcal{L} = \mathcal{D} - \mathcal{A}
\]

Let \( 0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N \) be the eigenvalues of \( \mathcal{L} \) in the non-decreasing order with the corresponding unit eigenvectors \( \{v_1, v_2, \cdots, v_N\} \). The graph Laplacian exhibits the following remarkable properties:

**Lemma 1.** [4]. Given an undirected graph \( \mathcal{G} \)

(i) \( \mathcal{L}(\mathcal{G}) \) is always symmetric and positive semi-definite and satisfies

\[
x^T \mathcal{L} x = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} a_{ij} (x_i - x_j)^2
\]

where \( x = (x_1, x_2, \cdots, x_N)^T \in \mathbb{R}^N \);

(ii) \( \lambda_1 = 0, \lambda_2 > 0 \) if and only if \( \mathcal{G} \) is connected;

(iii) \( v_i^T \mathcal{L} v_j = 0, \forall 1 \leq i, j \leq N, i \neq j \). Especially, \( v_1 = 1_N \), where \( 1_N \) is an \( N \)-dimensional column vector with all its elements being one;

(iv) The spectral radius is \( \rho(\mathcal{L}) = \lambda_N \leq N \).
III. DECENTRALIZED ESTIMATION OF ALGEBRAIC CONNECTIVITY $\lambda_2$

In order to overcome the drawback of the power iteration algorithm, we devise a decentralized inverse power iteration scheme, which can estimate the algebraic connectivity much faster than the standard power iteration scheme. The operation of the decentralized inverse power iteration can be converted into solving a related set of non-homogeneous linear equations. For each set of non-homogeneous linear equations, the conjugate gradient (CG) algorithm [21] would be employed to solve the non-homogeneous linear equations. A decentralized summation estimation algorithm inspired by [16] is further applied in this paper to make the CG algorithm fully decentralized.

3.1 Decentralized conjugate gradient algorithm

It is well known that CG algorithm has been widely used in solving linear equations, but normally the key procedure in computation of CG algorithm, for example, the inner product calculation, has to rely on the knowledge of all the elements in the correlation matrix, which makes the traditional CG algorithm centralized. In order to cope with this problem, a decentralized CG algorithm is proposed based on the decentralized summation computation algorithm which inspired by [16], the distributed PI average consensus algorithm is applied in this paper to decentralize the summation operation.

Algorithm 1 (Decentralized summation estimation).

1. For each agent $i$, the PI average consensus estimator has the following form [16]

$$
\begin{align*}
\dot{z}_i' &= \gamma (a'_i - z_i') - K_p \sum_{j \in N_i} [z_i' - z_j'] - K_f \sum_{j \in N_i} [W_j - W_i'] \\
\dot{W}_i' &= -K_f \sum_{j \in N_i} [z_i' - z_j']
\end{align*}
$$

where $a'_i$ is the initial scalar measured by agent $i$, $z_i'$ is the average estimate, $\gamma$ is the rate that new information replaces old information, and $K_f$, $K_p$ are estimator gains.

2. The estimation for the sum of the initial scalars of each agent $\sum_{i=1}^{N} a_i'$ can be obtained by

$$Z' = z' \ast N$$

where $N$ is the number of the agents in the MAS network, and the $Z'$ is the estimation result.

Based on above decentralized summation computation algorithm, the traditional CG algorithm [21] can be decentralized as follows.

Algorithm 2 (Decentralized conjugate gradient).

1. For non-homogeneous linear equations $Ax = b$, where $A$ is a $N$-dimension positive square matrix, $b$ is a known $N$-dimension vector, $x$ is the solution vector of these equations. Set $w^{(0)}$ associated to the random initial vector $x^{(0)}$ and the equations by

$$w^{(0)} = -r^{(0)} = b - Ax^{(0)}$$

for the initial iteration step $k = 0$, where $w^{(k)}$ is the direction-revising vector and $r^{(k)}$ is the residual vector in the CG iteration step $k$.

2. At the iteration step $k$, $x^{(k+1)}$ can be updated as

$$x_i^{(k+1)} = x_i^{(k)} + a_i^{(k)} w_i^{(k)}, \quad i = 1, \ldots, N$$

where

$$a_i^{(k)} = -\frac{\langle r_i^{(k)}, w_i^{(k)} \rangle}{\langle w_i^{(k)}, A w_i^{(k)} \rangle}$$

and $\langle r^{(k)}, w^{(k)} \rangle = \sum_{i=1}^{N} r_i^{(k)} w_i^{(k)}, \langle w^{(k)}, A w^{(k)} \rangle = \sum_{i=1}^{N} \sum_{m=1}^{N} w_i^{(k)} A_{im} w_m^{(k)}$. Since the computations of $\langle r^{(k)}, w^{(k)} \rangle$ and $\langle w^{(k)}, A w^{(k)} \rangle$ have to rely on all the elements of $r^{(k)}$, $w^{(k)}$ and the matrix $A$, Algorithm 1 is applied to make these computations decentralized. After that, $r_i^{(k+1)}$, $\beta^{(k+1)}$, $w_i^{(k+1)}$ can be updated for the manipulation of the next iteration step as follows.

$$
\begin{align*}
\beta_i^{(k+1)} &= \frac{\langle w_i^{(k)}, A w_i^{(k)} \rangle - \langle w_i^{(k+1)}, A w_i^{(k+1)} \rangle}{\beta_i^{(k)}} \\
\gamma_i^{(k+1)} &= -r_i^{(k)} + \beta_i^{(k+1)} w_i^{(k)}
\end{align*}
$$

© 2016 Chinese Automatic Control Society and John Wiley & Sons Australia, Ltd
where the computation of $\langle \alpha(k+1), Aw(k) \rangle = \sum_{i=1}^{N} (r_i^{(k+1)})$.

3. When the algorithm step 2 is finished, set $k = k + 1$.

Remark 1. In Algorithm 2, $\langle \cdot \rangle$ denotes the inner product of two vectors; $A_{i,m}$ is the $(i, m)$th component of matrix $A$; $w_m^{(k)} \left( x_i^{(k)}, x_j^{(k)}, r_i^{(k)}, w_j^{(k)} \right)$ is the $m$th $(i, \cdot)$th component of vector $w^{(k)} \left( x^{(k)}, x^{(k)}, r^{(k)}, w^{(k)} \right)$. Please note that the computation of $\alpha^{(k)}$ and $\beta^{(k)}$ has been decentralized by using Algorithm 1.

Theorem 1. By the calculation of the Algorithm 2, the solution of the non-homogeneous linear equations $Ax = b$ can be achieved, and the resulting vector $x^{(k+1)}$ will converge to the solution of $Ax = b$ in a fully decentralized way.

Proof. According to the analysis of [19], the consensus estimators made by Algorithm 1 can run much faster than the step 2 in the Algorithm 2 if the gains $\gamma, Kp$ and $Ki$ are chosen large enough, so we can guarantee the estimation results of the summations in Algorithm 2 will converge. Therefore, after a transient period, the decentralized CG algorithm 2 will agree with the centralized traditional CG algorithm and will converge. Therefore, after a transient period, the decentralized CG algorithm 2 will agree with the centralized traditional CG method introduced in [21]. Derived from the analysis of [21], the centralized traditional CG algorithm will converge and obtain the solution of $Ax = b$. Therefore, when the Algorithm 2 is accomplished, the resulting vector $x^{(k+1)}$ will converge to the solution of $Ax = b$ in a fully decentralized way.

3.2 Decentralized inverse power iteration scheme

Since the operation of the decentralized inverse power iteration can be converted into solving a related set of non-homogeneous linear equations, the following deflation of $L$ is applied

$$L = L + \frac{N + \delta}{N} 11^T$$

(12)

where $\delta \in \mathbb{R}, \delta > 0$, the set of eigenvalues of $\hat{L}$ is $\{ \lambda_2, \cdots, \lambda_N, N + \delta \}$ with the associated set of eigenvectors $\begin{bmatrix} v_2 \cdots v_m \end{bmatrix} / \sqrt{N}$.

Define the matrix $(\hat{L} - \mu I)^{-1}$ with $\mu \in \mathbb{R}$, it follows that the set of the eigenvectors is the same as that of $\hat{L}$ with the corresponding set of the eigenvalues $\{ (N + \delta - \mu)^{-1}, (\lambda_2 - \mu)^{-1}, \cdots, (\lambda_N - \mu)^{-1} \}$

(13)

The inverse power iteration procedure would converge to the dominant eigen-pair of $(\hat{L} - \mu I)^{-1}$, which is $\{ (\lambda_2 - \mu)^{-1}, v_2 \}$, with the following iteration

$$\left( \hat{L} - \mu I \right)^{-1} \hat{v}^{(k)} = \hat{v}^{(k+1)}$$

(14)

where $k + 1$ is the step index of the inverse power iteration.

The convergence rate of $\hat{\lambda}_2$ approaching to $\lambda_2$ is related with the convergence factor $\gamma = \left| \frac{\lambda_2 - \mu}{\lambda_2 - \mu} \right|$, where the closer $\gamma$ is to 0, the higher the convergence rate is. The symmetric and positive definiteness of $(\hat{L} - \mu I)^{-1}$ can be ensured when $0 \leq \mu < \lambda_2$. Hence the convergence rate of the inverse power iteration would be much higher than that of the standard power iteration when the parameter $\mu$ is thoughtfully adjusted [21]. In this paper, $\mu$ can be deliberately chosen according to the properties of $\lambda_2$ proposed in [22].

In the inverse power iteration algorithm, computation of matrix inversion is not only necessary, but also very time-consuming. Moreover, since matrix inversion computation usually has to rely on all the elements of the matrix, for example all the elements of the Laplacian $L$ in this case, which unfortunately is inaccessible for each agent. Therefore, direct methods of matrix inversion computation cannot be employed in multi-agent systems. So instead of any direct method, an indirect iteration method for computing matrix inversion is applied.

At the inverse power iteration step $k + 1$, assume that $\hat{v}_2^{(k)}$ has been figured out from the preceding step, (14) can be transformed into a set of non-homogeneous linear equations as the following update law

$$\left( \hat{L} - \mu I \right) \hat{v}_2^{(k+1)} = \hat{v}_2^{(k)}$$

(15)

To realize this update law in a decentralized way, the proposed decentralized conjugate gradient algorithm is used to figure out the $(k + 1)th$ iteration vector $\hat{v}_2^{(k+1)}$ of $E \hat{v}_2^{(k+1)} = \hat{v}_2^{(k)}$, where $E = (\hat{L} - \mu I)$. Hence based on the above work, a decentralized algebraic connectivity estimation scheme is exploited here as conclusion.

Scheme 1. (Decentralized algebraic connectivity estimation)

1. Firstly, each elements of the corresponding row of the preconditioned matrix $E$ is calculated by agent $i$, where
\[ E_{ij} = \left( L_{ij} + \frac{N + \delta}{N} - \zeta \mu \right), \quad j = 1, \ldots, N \] (16)

and \( \zeta = \begin{cases} 0, & j \neq i \\ 1, & j = i \end{cases} \).

2. For each agent \( i \), the Algorithm 2 is applied to deal with the corresponding work in solving the non-homogeneous linear equation \( E\hat{v}_{2,i}^{(k+1)} = \hat{v}_{2,i}^{(k)} \). At the each iteration step \( k + 1 \), the agent \( i \) can obtain the corresponding component \( \hat{v}_{2,i}^{(k+1)} \) of the estimated solution vector.

3. The \( \hat{\lambda}_2^{(k+1),i} \) can be estimated by agent \( i \) as

\[ \hat{\lambda}_2^{(k+1),i} = \frac{\sum L_{ij} \hat{v}_{2,j}^{(k+1)}}{\hat{v}_{2,i}^{(k+1)}} \] (17)

4. When the scheme step 3 is finished, set \( k = k + 1 \).

**Theorem 2.** Agent \( i \) can complete the estimation of its corresponding element \( \hat{v}_{2,i}^{(k+1)} \) in the estimated eigenvector \( \hat{v}_{2,i}^{(k+1)} \) at the inverse power iteration step \( k + 1 \) in a decentralized way by the inverse power iteration estimation Scheme 1, then the overall eigen-pair estimation scheme runs successfully. So we can show that the entire algebraic connectivity estimation scheme runs successfully. And

\[
\hat{\lambda}_2 = \hat{\lambda}_4 = \cdots = \hat{\lambda}_N.
\]

**Proof.** According to the proof of Theorem 1, since the PI average consensus estimators of Algorithm 1 can run much faster than the step 2 in Algorithm 2, the decentralized conjugate gradient Algorithm 2 will agree with the standard CG algorithm. So with the values of \( \alpha^{(i)} \) and \( \beta^{(i+1)} \) obtained by Algorithm 1, we can accomplish Algorithm 2 in the decentralized way, allowing each corresponding agent \( i \) to obtain the final \( \hat{v}_{2,i}^{(k+1)} \) at the inverse power iteration step \( k + 1 \). As it can be guaranteed that the results of every decentralized inverse power iteration step converge, it can be shown that the entire algebraic connectivity estimation scheme runs successfully. So we can get the estimation value of eigenvector \( \hat{v}_2 \) and eigenvalue \( \hat{\lambda}_2 \) in a decentralized way. The power iteration method in [16] can be described as

\[
\left( I - \theta L - \frac{1}{N} 11^T \right)^{(k+1)} \chi^{(0)} = \chi^{(k+1)}
\] (18)

where

\[
\chi^{(0)} = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_N u_N
\]

\( u_i \) is the corresponding eigenvector of \( \lambda_i \), \( \alpha_i \) is the scale coefficient of \( u_i \) in (19).

So we can obtain that

\[
\chi^{(k+1)} = (1 - \frac{\theta \lambda_2}{\lambda_2 - \mu}) \left[ a_2 u_2 + \frac{1 - \theta \lambda_3}{1 - \theta \lambda_2} \right] \chi^{(k+1)}
\]

\[ + \cdots + \frac{1 - \theta \lambda_N}{1 - \theta \lambda_2} \alpha_N u_N + \left( \frac{1}{1 - \theta \lambda_2} \right) a_1 u_1
\]

which means the convergence rate of (18) is inversely proportional to \( \frac{\lambda_2 - \mu}{\lambda_2 - \mu} \).

Similar to (18), the inverse power iteration Scheme 1 can be seen as

\[
\left( L + \frac{N + \delta}{N} 11^T - \mu I \right)^{(k+1)} \chi^{(0)} = \chi^{(k+1)}
\] (21)

So we can obtain that

\[
\chi^{(k+1)} = \left( \frac{1}{\lambda_2 - \mu} \right) \left[ a_2 u_2 + \frac{\lambda_2 - \mu}{\lambda_3 - \mu} \right] \chi^{(k+1)}
\]

\[ + \cdots + \frac{\lambda_2 - \mu}{\lambda_N - \mu} \alpha_N u_N + \left( \frac{\lambda_2 - \mu}{N + \delta - \mu} \right) a_1 u_1
\]

(22)

which means the convergence rate of (21) is inversely proportional to \( \frac{\lambda_2 - \mu}{\lambda_2 - \mu} \).

Therefore if

\[
1 - \frac{\theta \lambda_2}{\lambda_2 - \mu} > \frac{\lambda_2 - \mu}{\lambda_2 - \mu}
\]

(23)

hold, the convergence rate of Scheme 1 will be faster than that of the power iteration estimation algorithm.

From (18), the precondition matrix \( I - \theta L - \frac{1}{N} 11^T \) should be a positive semi-definite matrix, which means that \( \theta \leq \frac{1}{\lambda_2} \). So the convergence rate of the power iteration estimation algorithm reaches the upper bound when \( \theta = \frac{1}{\lambda_2} \). Taking \( \theta = \frac{1}{\lambda_2} \) in (23), we get the inequality that

\[
\frac{1 - \frac{\lambda_2}{\lambda_2}}{1 - \frac{\lambda_2}{\lambda_N}} > \frac{\lambda_2 - \mu}{\lambda_2 - \mu}
\]

(24)

From (24), we can derive that

\[
\lambda_2 > \mu > \lambda_2 - (\lambda_N - \lambda_2)
\]

(25)
It is concluded that for any given $\theta$ in the power iteration estimation algorithm, there always exists $\mu$ which can guarantee that the convergence rate of the Scheme 1 is much faster than that of the power iteration estimation algorithm except for the condition $\lambda_3 = \lambda_4 = \cdots = \lambda_N$.

**Remark 2.** For the heterogeneous MAS networks with directed topology, the necessary condition to run the overall algorithm is that the directed graph is strongly connected. Then the proposed scheme should be improved to consider that for the directed graph of $\mathcal{A}$, the matrix $A$ is not irreducible, which is the future work to be carried out next.

**IV. SIMULATIONS**

In this section, simulations are performed to validate our proposed decentralized estimation algorithm with effective performance.

We assume that a team of seven agents are steered autonomously to execute a given task. In order to make the whole team work effectively, all the agents should be controlled to maintain the connectivity by keeping the algebraic connectivity $\lambda_2$ larger than 0, therefore $\lambda_2$ has to be estimated before the connectivity maintenance controller is carried out. We set the time constant of the PI average consensus estimation in Algorithm 1 significantly less than that of the eigenvector estimation, and the time constant of the eigenvector estimation is even greatly less than that of the connectivity maintenance controller[16]. So in each time period of the connectivity maintenance controller the Laplacian matrix $L$ is treated as a constant matrix. As in the following, the initial position of a seven-agent network was depicted in Fig.1 along with inter-agent communication links represented by solid lines. The Laplacian spectrum of the initial topology is $\{0, 0.555, 0.634, 2.045, 2.778, 3.632, 4.746\}$, where the eigenvector associated with $\lambda_2$ is $v_2 = \{0.156, 0.496, -0.136, 0.221, -0.487, 0.320, -0.569\}$. For the weighted adjacency matrix, the communication radius is set to be $R = 8m$, the corresponding parameters are properly chosen as $\tau = 0.8$ in (3), $\gamma = 100$, $K_I = 100$, $K_P = 200$ in (6), $\sigma = 0.7$ in (12), $\mu = 0.55$ in (14).

First, in order to illustrate the advantage of our conjugate gradient method over the relaxed Jacobi iteration in [17], the comparative estimation results of the solution vector of $E\hat{v}_2^{(1)} = \hat{v}_2^{(0)}$ with the iteration steps

![Fig. 1. Initially connected team position and topology of Seven-agents network. [Color figure can be viewed at wileyonlinelibrary.com]](image1)

![Fig. 2. Estimation of $v_2^{(1)}$ using different methods. [Color figure can be viewed at wileyonlinelibrary.com]](image2)
7 for Algorithm 2 and the estimation accuracy $e_t = \|v^{(1)}(1+k+1)\| - \|v^{(1)}(k)\| = 10^{-2}$ for relaxed Jacobi iteration are shown in Fig. 2, where the actual values of $v^{(1)}$ are denoted by black dashed lines, and their estimates are denoted by coloured solid lines. The relaxation parameter is chosen as $\varphi = 0.4$.

It can be clearly seen that the conjugate gradient method exhibits superior convergence property to the relaxed Jacobi iteration. The iteration step and consumed time of the conjugate gradient method are, respectively, 7 and 0.121 s, then the iteration step and consumed time of the relaxed Jacobi iteration are respectively 3488 and 19.717 s.

Next, the comparative convergence results of the estimated $\hat{v}_2$ and $\hat{\lambda}_2$ by using both the power iteration in [16] and the inverse power iteration are demonstrated in Fig. 3 and Fig. 4, the actual values of the eigenvector and eigenvalue are denoted by black dashed lines, and their estimates are denoted by coloured solid lines. In this situation, $K_1 = 2.4, K_2 = 2, K_3 = 2.6$, sampling period $t = 0.05s$.

With the estimation accuracy $e = \|\hat{\lambda}_2^{(k+1)}\| - \|\hat{\lambda}_2^{(k)}\| = 10^{-6}$, it is clear that by exploiting our proposed inverse power iteration algorithm, each components of the estimated eigenvector $\hat{v}_2$ and $\hat{\lambda}_2$ converges to the fine value $v_2$ and $\lambda_2$ much faster than the power iteration. The iteration step and cost time of our algorithm are respectively 5 and 0.753 s, the the iteration step and cost time of the power iteration are respectively 419 and 0.926 s.

![Fig. 3. Estimation of $\lambda_2$ and $v_2$ using inverse power iteration.](image1)

![Fig. 4. Estimation of $\lambda_2$ and $v_2$ using power iteration.](image2)
V. CONCLUSIONS AND FUTURE WORK

The problem of estimation of algebraic connectivity and its corresponding eigenvector for multi-agent systems is investigated. The proposed estimation scheme could successfully provide the estimated connectivity $\hat{\lambda}_2$ and its corresponding eigenvector $\hat{v}_2$ in a fully decentralized way. The higher convergence rate of the estimation can be achieved compared with the classical power iteration scheme. While there are still some issues need to be addressed. It will be interesting and challenging to consider more realistic communication models, especially failures of communication links or directed communication graphs as well. And the method to choose $\mu$ more effectively will be our future work. It will also be interesting to extend the proposed method to account for more complex application situations.

REFERENCES