Review:

Distributed coordination in multi-agent systems: a graph Laplacian perspective

Zhi-min HAN\(^1\), Zhi-yun LIN\(^{1,2}\), Min-yue FU\(^{2,3}\), Zhi-yong CHEN\(^2\)

\(^{1}\)State Key Laboratory of Industrial Control Technology, College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China

\(^{2}\)School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan NSW 2308, Australia

\(^{3}\)State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China

E-mail: hanzhimin@zju.edu.cn; linz@zju.edu.cn; minyue.fu@newcastle.edu.au; zhiyong.chen@newcastle.edu.au

Received Apr. 13, 2015; Revision accepted May 15, 2015; Crosschecked May 15, 2015

Abstract: This paper reviews some main results and progress in distributed multi-agent coordination from a graph Laplacian perspective. Distributed multi-agent coordination has been a very active subject studied extensively by the systems and control community in last decades, including distributed consensus, formation control, sensor localization, distributed optimization, etc. The aim of this paper is to provide both a comprehensive survey of existing literature in distributed multi-agent coordination and a new perspective in terms of graph Laplacian to categorize the fundamental mechanisms for distributed coordination. For different types of graph Laplacians, we summarize their inherent coordination features and specific research issues. This paper also highlights several promising research directions along with some open problems that are deemed important for future study.

Key words: Multi-agent systems, Distributed coordination, Graph Laplacian

doi: 10.1631/FITEE.1500118

1 Introduction

In different communities, the terms ‘agent’ and ‘multi-agent system’ have different connotations. But roughly speaking, the following interpretations are broadly admitted: An ‘agent’ is a computational mechanism that exhibits a high degree of autonomy, performing actions in its environment based on information received, via sensors and feedback, from the environment, and a ‘multi-agent system’ contains more than one agent interacting with one another with some constraints such that agents may not at any time know everything about the entire system.

In computer science, the research for multi-agent systems typically refers to software agents, which have been widely studied in the 1980s and 1990s. Multi-agent systems have replaced single agents as the computing paradigm in artificial intelligence (Weiss, 1999). On the other hand, the agents in a multi-agent system can be robots as well and thus multi-agent systems are also referred to as multi-robot systems in the robotic society. The study of multi-robot systems began in the early 1990s (for example, Sugihara and Suzuki (1990)). However, it is much later that researchers in the systems and control community started to investigate more general multi-agent systems. Since 2003, multi-agent
systems have become a very active research topic in systems and control, where a multi-agent system is usually considered to be a collection of autonomous or semi-autonomous, but interacting, dynamic systems. A schematic diagram of a multi-agent system is shown in Fig. 1, where the network represents the coupling structure among the agents. The coupling links can be communication channels, sensing information flow, or physical connections, and thus can be static or dynamic when links may be established or dropped over time.

![Fig. 1 A multi-agent system](image)

The agents in a multi-agent system have several important features.

1. Autonomy: The agents are at least semi-autonomous.
2. Local views: No agent has a full global view of the system, or the system is too complex for an agent to make practical use of such knowledge (e.g., the states of all agents).
3. Decentralization: Each agent interacts with only a few neighboring agents based on relative information from neighbors, in absence of designated controlling agents.
4. Time evolution: The state of each agent evolves according to certain local coordination protocols interacting with one another, which eventually leads to the occurrence of collective behaviors of the entire system.

This survey paper will focus mainly on recent progress on multi-agent systems in the systems and control community, considering both continuous- and discrete-time dynamics. Research issues include consensus, formation control, flocking, sensor localization, distributed optimization, etc. In the systems and control community, pioneering works on multi-agent systems started with the investigation of the consensus problem (Jadbabaie et al., 2003; Lin et al., 2004; 2005; Olfati-Saber and Murray, 2004; Moreau, 2005; Ren and Beard, 2005). After that, a huge number of works have appeared concerning a variety of control tasks, agent models, and control strategies in multi-agent systems. In addition, there have been several monographs on multi-agent systems from the system and control viewpoint (Lin, 2008; Ren and Beard, 2008; Bullo et al., 2009; Qu, 2009; Mesbahi and Egerstedt, 2010; Ren and Cao, 2011). Moreover, excellent surveys on distributed control of multi-agent systems were given in Leonard et al. (2007), Murray (2007), Olfati-Saber et al. (2007), Ren et al. (2007), Anderson et al. (2008), Dörfler and Bullo (2014), and Oh et al. (2015b). However, these survey papers focused either on one specific research problem in multi-agent systems such as consensus (Olfati-Saber et al., 2007; Ren et al., 2007), synchronization (Dörfler and Bullo, 2014), formation control (Anderson et al., 2008; Oh et al., 2015b), or ocean sampling (Leonard et al., 2007), or general multi-agent research problems in terms of applications (Murray, 2007).

This paper intends to present a survey from a new perspective in terms of graph Laplacian, which connects different research issues of multi-agent systems in one string for distributed coordination. Based on this motivation, we categorize the existing results in multi-agent systems into ordinary Laplacian, signed Laplacian, complex Laplacian, and generalized Laplacian based protocols according to the type of graph Laplacian.

1. Ordinary Laplacian based protocols: An ordinary Laplacian refers to a Laplacian matrix associated to a graph with positive and real weights. Consensus, translational formation control, flocking, and distributed resource allocation can all be solved by ordinary Laplacian based protocols.

2. Signed Laplacian based protocols: A signed Laplacian refers to a Laplacian matrix associated to a graph with positive or negative weights. Bipartite consensus, cluster consensus, optimization over convergence speed, affine formation control, and distance-based localization call for signed Laplacian based protocols.

3. Complex Laplacian based protocols: A complex Laplacian refers to a Laplacian matrix associated to a graph with complex weights. Similar formation control and relative position based localization use complex Laplacian based protocols.

4. Generalized Laplacian based protocols: A generalized Laplacian refers to a Laplacian matrix associated with a graph with weights that may be matrices, time-varying variables, or dynamic
systems. This appears in bearing-angle based localization, distributed coordination over switching topologies, and distributed coordination with dynamic gains. The rest of this paper is structured as follows: Preliminaries about graph and graph Laplacian are presented in Section 2. In Section 3, we survey the recent development of multi-agent research with respect to different categories of graph Laplacian. Section 4 highlights several promising research directions along with some open problems and Section 5 concludes the paper.

**Notations:** 1 stands for a vector of all 1 elements. $I_n$ represents the $n \times n$ identity matrix. For a vector $w = [w_1, w_2, \ldots, w_m]^T$, $\text{diag}(w)$ is the diagonal matrix with its diagonal entries being $w_1, w_2, \ldots, w_m$.

### 2 Preliminaries

This section introduces basic concepts and notations of graph and graph Laplacian.

#### 2.1 Graph concepts

Throughout the paper, a system of $n$ agents is modeled as a graph. Specifically, agents are represented as nodes of a graph and interactions due to sensing and communication are represented as edges of the graph. Next, we review basic notions from graph theory (Godsil and Royle, 2001) and several new notions (Lin et al., 2014; Wang et al., 2014b).

A directed graph $G = (\mathcal{V}, \mathcal{E})$ consists of a node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An edge of $G$ is denoted by an ordered pair of nodes $(j, i)$, which means that the edge has tail at node $j$ and head at node $i$. Alternatively, $(j, i)$ is called an ‘incoming edge’ of node $i$ and an ‘outgoing edge’ of node $j$. If $(j, i) \in \mathcal{E}$, node $j$ is called an ‘in-neighbor’ of node $i$ and node $i$ is called an ‘out-neighbor’ of node $j$. We define $\mathcal{N}_i$ as the in-neighbor set of agent $i$, i.e., $\mathcal{N}_i := \{j : (j, i) \in \mathcal{E}\}$.

A ‘walk’ in a directed graph $G$ is an alternating sequence $p : v_1e_1v_2e_2\ldots v_ke_k$ of nodes $v_i$ and edges $e_i$ such that $e_i = (v_i, v_{i+1})$ for every $i = 1, 2, \ldots, k - 1$. If there exists a walk from node $u$ to $v$ in $G$, then node $v$ is said to be ‘reachable’ from node $u$. A directed graph is said to be ‘strongly connected’ if every node is reachable from every other node. Moreover, a directed graph is said to be ‘rooted’ if there exists a node, from which every other node is reachable.

For a directed graph $G$, a node $v$ is said to be ‘2-reachable’ from a non-singleton subset of nodes $\{u_1, u_2, \ldots, u_k\}$ if there exists a walk from a node in $\{u_1, u_2, \ldots, u_k\}$ to $v$ after removing any one node except $v$. A directed graph $G$ is said to be ‘2-rooted’ if there exists a subset of two nodes, from which every other node is 2-reachable. The notions of $k$-reachable and $k$-rooted for $k \geq 2$ are defined in the same manner.

We consider undirected graphs as directed ones with special properties. That is, if a directed graph $G = (\mathcal{V}, \mathcal{E})$ satisfies the property that $(i, j) \in \mathcal{E}$ and $(j, i) \in \mathcal{E}$, then $G$ is said to be ‘undirected’.

#### 2.2 Graph Laplacian

A weighted ‘adjacency matrix’ $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ of a directed graph $G$ with $n$ nodes is defined such that $a_{ij}$ is the weight of edge $(j, i)$ satisfying $a_{ij} \neq 0$ if $(j, i)$ is an edge of $G$ and $a_{ij} = 0$ otherwise. The ‘degree matrix’ $D = [d_{ii}] \in \mathbb{R}^{n \times n}$ is a diagonal matrix defined as

$$d_{ii} = \begin{cases} \sum_{j \in \mathcal{N}_i} a_{ij}, & i = j, \\ 0, & \text{otherwise}. \end{cases}$$

The graph Laplacian is then expressed as

$$L = D - A.$$

From the definition, it is certain that $L \mathbf{1} = 0$. In other words, 0 is always an eigenvalue of $L$ with the associated eigenvector $\mathbf{1}$.

For an undirected graph, suppose that it has $m$ edges, labeled $1, 2, \ldots, m$ with weights $w_1, w_2, \ldots, w_m$. We can then arbitrarily assign a direction for each edge. The ‘incidence matrix’ $B = [b_{il}] \in \mathbb{R}^{n \times m}$ is defined as

$$b_{il} = \begin{cases} 1, & \text{edge } l \text{ starts from node } i, \\ -1, & \text{edge } l \text{ ends at node } i, \\ 0, & \text{otherwise}. \end{cases}$$

In this case, the Laplacian can be written as

$$L = B \text{diag}(w) B^T,$$

where $w = [w_1, w_2, \ldots, w_m]^T$. The Laplacian of an undirected graph is symmetric and positive semi-definite.
3 Survey of distributed multi-agent coordination

Due to the feature of local views in multi-agent systems, the agents can access only the relative information about a portion of other agents. That is, if we denote by $x_i$ the state of agent $i$ ($i = 1, 2, \ldots, n$), then the following information shall be available to agent $i$, where $j_1, j_2, \cdots, j_m \in N_i$. This relative information is then used by agent $i$ to make a control signal for local coordination. If a linear feedback control is considered, then it must be of the form given in Fig. 2. Such a control structure naturally leads to a Laplacian associated to a graph with weights $K_{ij}$, where $K_{ij}$ can be a static gain or a dynamics. Depending on the type of $K_{ij}$, we classify distributed multi-agent coordination into four categories.

3.1 Ordinary Laplacian based protocols

When the gains $K_{ij}$ in Fig. 2 (namely, the weights on the edges of the graph modeling a multi-agent system) are real scalars and positive, the resulting Laplacian is called the ‘ordinary Laplacian’. Consensus and its related extensions such as translational formation control, flocking, and distributed resource allocation adopt the ordinary Laplacian based protocols.

3.1.1 Consensus

Consensus is a basic distributed coordination problem in multi-agent systems. ‘Consensus’ means the agreement of all agents on some common features by negotiating with their neighbors from arbitrary initial states. The consensus features can be positions, velocities, attitudes, and many other quantities. The consensus problem was originally studied in management science (Degroot, 1974) and similar ideas were found in distributed computing (Tsitsiklis, 1984; Tsitsiklis et al., 1986). In recent years, some consensus algorithms were studied under various setups (Jadbabaie et al., 2003; Lin et al., 2004; 2005; Olfati-Saber and Murray, 2004; Moreau, 2005; Ren and Beard, 2005; Hong et al., 2006; Cortés, 2008; Ren, 2008; Tahbaz-Salehi and Jadbabaie, 2008; Stanković et al., 2009; Tian and Liu, 2009; Nedic et al., 2010; Li et al., 2011; Cai and Ishii et al., 2012; Hendrickx and Tsitsiklis, 2013; Fanti et al., 2015).

1. Continuous-time consensus

Consider agents with single-integrator dynamics given by

$$\dot{x}_i = u_i, \quad (1)$$

where $x_i \in \mathbb{R}$ and $u_i \in \mathbb{R}$ are the state and control input of agent $i$, respectively. A linear consensus law was studied in Jadbabaie et al. (2003), Lin et al. (2004), Olfati-Saber and Murray (2004), and Ren and Beard (2005) as

$$u_i = \sum_{j=1}^{n} a_{ij} (x_j - x_i), \quad i = 1, 2, \ldots, n, \quad (2)$$

where $a_{ij}$ is a positive real constant, which is the weight attributed to edge $(j, i)$ from the graph perspective. With control law (2), the multi-agent system can be written in a matrix form:

$$\dot{x} = -L x, \quad (3)$$

where $x = [x_1, x_2, \ldots, x_n]^T$ and $L$ is the ordinary Laplacian associated with graph $G$.

Consensus is said to be ‘achieved’ if for all $x_i(0)$ and all $i, j = 1, 2, \ldots, n$,

$$|x_i(t) - x_j(t)| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty.$$  

It is known that consensus is achieved for system (3) if and only if $L$ has a simple zero eigenvalue, or equivalently, the directed graph is rooted (having a spanning tree is an equivalent notion (Ren and Beard, 2005)).

To apply the consensus algorithms in practice, many factors should be taken into consideration,
such as link failures, communication delays, disturbances from the environment, and complicated agent dynamics. Therefore, the consensus problem has been further investigated (Olfati-Saber and Murray, 2004; Hong et al., 2006; Bliman and Ferrari-Trecate, 2008; Cortés, 2008; Tian and Liu, 2009; Zhu and Cheng, 2010; Cao et al., 2011; Li et al., 2011; Hendrickx and Tsitsiklis, 2013).

2. Discrete-time consensus

For implementation on digital platforms, a discrete-time counterpart of the consensus laws is considered. That is, in discrete time, the single-integrator (1) can be approximately written as

\[
\frac{x_i[k+1] - x_i[k]}{T} = u_i[k],
\]

where \( k \) is the discrete-time index, \( T \) is the sampling period, and \( x_i[k] \) and \( u_i[k] \) denote the state and control input of the \( i \)th agent at \( t = kT \), respectively. The consensus control law in discrete time takes the same form:

\[
u_i[k] = \sum_{j=1}^{n} a_{ij} (x_j[k] - x_i[k]), \quad i, j = 1, 2, \ldots, n,
\]

where \( a_{ij} \) is a positive real constant. Substituting Eq. (5) into Eq. (4), the multi-agent system can then be written in a matrix form:

\[
x[k+1] = (I_n - TL)x[k],
\]

where \( x[k] = [x_1[k], x_2[k], \ldots, x_n[k]]^T \) and \( L \) is the Laplacian associated to graph \( G \). Under the assumption that \( T < \frac{1}{\max_i l_{ii}} \), the same result can be established as the continuous-time counterpart. That is, consensus is achieved for system (6) if and only if \( L \) has a simple zero eigenvalue or equivalently the directed graph is rooted.

3.1.2 Translational formation control

Formation control refers to a control task that aims to steer a group of agents to form a specific relative configuration between each other. This problem is relatively straightforward in the centralized case, in which all team members know the desired shape, location, and orientation of the formation. However, in many situations, the agents cannot access the global information in a centralized way. As a result, distributed formation control attracts huge attention.

Suppose that the agents have a common sense of direction in the plane. Consensus control schemes can be modified by including displacement vectors to solve the formation control problem (Lin et al., 2004; 2005; Ren, 2007; Huang and Wu, 2010; Kuriki and Namerikawa, 2014). However, it should be noted that the formation achieved by the modified consensus control schemes has only translational freedom as compared to the desired formation specified in a global coordinate system, which is then referred to as ‘translational formation control’ in this study.

Regarding translational formation control, Lin et al. (2004) pointed out that if convergence to a point formation is feasible, then more general formations are achievable too. The simplest strategy for formation control in the plane may be the cyclic pursuit strategy given by

\[
\begin{cases}
z_i = (z_{i+1} + \xi_i) - z_i, & i = 1, 2, \ldots, n - 1, \\
z_n = (z_1 + \xi_n) - z_n,
\end{cases}
\]

where \( z_i \in \mathbb{C} \) is the position of agent \( i \), and \( (\xi_1, \xi_2, \ldots, \xi_n) : = \xi \) is the target configuration of the \( n \) agents satisfying that the centroid of them is at the origin. The vector form of Eq. (7) is

\[
\dot{z} = -Lz + \xi,
\]

where \( z \) is the vector of all \( z_i \)’s, and \( L \) is the ordinary Laplacian of the cycle graph that models the cyclic interactions. For a more general interaction graph, by analyzing the associated Laplacian, it can be concluded whether a translation formation can be achieved or not.

If the agents do not have a common sense of direction, a simultaneous orientation alignment and formation control strategy was studied in Oh and Ahn (2014). The control strategy has two components. The first component is concerned with orientation alignment, designed as

\[
\dot{\theta}_i = \sum_{j \in \mathcal{N}_i} a_{ij} (\theta_j - \theta_i),
\]

where \( \theta_i \) is the orientation angle of agent \( i \)’s local frame and \( a_{ij} \) is a positive real constant. Let \( \Theta = [\theta_1, \theta_2, \ldots, \theta_N]^T \). Then from Eq. (9), it is obtained that

\[
\dot{\Theta} = -L\Theta,
\]

which is the standard consensus control law given in Eq. (3). Therefore, there exists \( \Theta_\infty = \theta_\infty \mathbf{1} \) such that
\( \mathbf{u}_i^t = \sum_{j \in N_i} b_{ij} ((z_j^t - z_i^t) - (z_j^* - z_i^*)) \), \hspace{0.5cm} (11)

where \( u_i^t \in \mathbb{R}^2 \) and \( z_j^t \in \mathbb{R}^2 \) denote the control input and the position of agent \( j \) in agent \( i \)'s local frame respectively, \( z_j^t - z_i^* \) is the desired displacement between agents \( j \) and \( i \) with respect to some common frame, and \( b_{ij} \) is a positive real constant. Under the orientation alignment law (10) and the formation control law (11), the multi-agent system exponentially converges to the desired formation with its orientation determined by \( \theta^* \).

Due to distributed and linear features of these ordinary Laplacian based formation control laws, research focuses have also been extended to formation control with more specifications such as collision avoidance and robustness to disturbances (Cortés, 2009; Huang and Wu, 2010; Kuriki and Namerikawa, 2014).

3.1.3 Flocking

Flocking is an amazing natural phenomenon, e.g., flocking of birds, schooling of fish, and swarming of bacteria (Okubo, 1986), attracting much attention in biology, physics, and computer science (Reynolds, 1987). This phenomenon emerges from limited environmental information and simple protocols that organize a large number of agents into a coordinated motion. As a control problem, ‘flocking’ means that the same velocity is attained by all the agents and the distances between the agents are maintained (Moshtagh et al., 2006).

Flocking can be considered as a variant of the consensus problem. Thus, the ordinary Laplacian based protocols can be adopted to solve the flocking problem. The flocking problem has also been widely investigated from single-integrator kinematics to double-integrator dynamics, from timely communication to delayed communication, from fixed topology to switching topology, and from without robustness to robustness (Blondel et al., 2005; Moshtagh and Jadbabaie, 2007; Li et al., 2008; He et al., 2012; Wang and Peng, 2012; Martin, 2014; Semnani and Basir, 2015).

3.1.4 Distributed resource allocation

The distributed resource allocation problem deals with how to allocate available resources to a number of users, called agents, in a distributed manner, which can be found in many applications in financial markets, smart grids, wireless sensor networks, cloud systems, etc. The problem is commonly formulated as an optimization problem subject to a network structure constraint. The network is modeled as a directed graph of \( n \) nodes. Each node \( i \) is associated with a variable \( x_i \in \mathbb{R} \) and a corresponding convex cost function \( f_i : \mathbb{R} \rightarrow \mathbb{R} \). Then the following optimization problem represents a resource allocation problem:

\[
\min \sum_{i=1}^{n} f_i(x_i) \text{ s.t. } \sum_{i=1}^{n} x_i = c, \hspace{0.5cm} (12)
\]

where \( c \in \mathbb{R} \) is a given constant. The variable \( x_i \) can be thought of as the amount of some resources available to agent \( i \) and \( -f_i \) can be interpreted as the local concave utility function. The problem (12) is to find an allocation of the resource that maximizes the total utility \( -\sum_{i=1}^{n} f_i(x_i) \).

Assume that the cost functions \( f_i \) are convex and twice continuously differentiable with second derivatives that are bounded below and above. The optimization problem (12) has a unique optimal solution \( x^* = [x_1^*, x_2^*, \ldots, x_n^*]^T \). Let

\[
\nabla f(x) = [f'_1(x_1), f'_2(x_2), \ldots, f'_n(x_n)]
\]

denote the gradient of \( f \) at \( x \). The optimality conditions for this problem are

\[
1^T x^* = c, \hspace{0.5cm} \nabla f(x^*) = \lambda^* 1,
\]

where \( \lambda^* \) is the unique optimal Lagrange multiplier and \( 1 \) is the all-one-vector of a proper dimension.

A distributed iteration algorithm is proposed to solve problem (12) (Xiao and Boyd, 2006), which has the same idea as the ordinary Laplacian based protocols for consensus. That is, each node updates according to

\[
x_i[k+1] = x_i[k] + \sum_{j \in N_i} a_{ij} (f'_j(x_j[k]) - f'_i(x_i[k])),
\]

where \( a_{ij} > 0 \) is a real number. Aggregating all node updates together leads to the matrix form

\[
x[k+1] = x[k] - L \nabla f(x[k]), \hspace{0.5cm} (13)
\]
where \( L \) is the ordinary Laplacian matrix associated to the graph with weights \( a_{ij} \). Under the assumption that the Laplacian is balanced (i.e., \( 1^T L = L 1 = 0 \)), for an initial condition satisfying \( 1^T x[0] = c \), it follows that \( 1^T x[k] = c \) for all \( k \) by the update law (13), which ensures the equality constraint in (12). Moreover, the equilibrium \( x \) by the update law (13), which ensures the equal-

\[
\nabla f(x) = 0 \implies \nabla f(x) = \lambda 1 \text{ for some } \lambda.
\]

This means that the equilibrium \( x \) of Eq. (13) is the optimal solution \( x^* \) of problem (12). Therefore, the ordinary Laplacian based law given in Eq. (13) solves the distributed resource allocation problem.

In many applications, the resource allocation problem also includes an inequality constraint for each variable \( x_i \). That is, in addition to the linear equality constraint in problem (12), there are additional inequality constraints:

\[
\underline{x} \leq x_i \leq \bar{x}_i, \text{ for } i = 1, 2, \ldots, n, \tag{14}
\]

where \( \underline{x} \) and \( \bar{x} \) are two real numbers. In this case, Yang et al. (2013) introduced a surplus variable for each node to temporarily store the mismatch due to the inequality constraint by adopting the surplus idea of solving the averaging consensus problem for directed graphs (Cai and Ishii, 2014), and then solved the distributed resource allocation problem (12) with inequality constraints (14). The algorithm proposed in Yang et al. (2013) can deal with static directed graphs without the need of assuming the Laplacian to be balanced. To overcome the challenges caused by time-varying directed communication graphs, Xu et al. (2015) proposed a non-negative surplus scheme and applied the ordinary Laplacian based idea to solve the same distributed resource allocation problem (12) subject to the inequality constraints (14). Moreover, various consensus based algorithms have been developed to solve the distributed resource allocation problems. Here we provide some examples. Lakshmanan and de Farias (2008) proposed a decentralized, asynchronous gradient-descent method that is suitable for implementation in the case where the communication between agents is described in terms of a dynamic network. Dominguez-Garcia et al. (2012) addressed the problem of optimally dispatching a set of distributed energy resources in a distributed fashion, and showed how the ratio consensus algorithm, which is a linear-iterative algorithm, enables components in a multi-component system to achieve consensus on a certain quantity. Kar and Hug (2012) presented a fully distributed approach for economic dispatch in power systems. The approach is based on the consensus + innovation framework, in which each network agent participates in a collaborative process of neighborhood message exchange and local computation. Xing et al. (2015) also presented a fully distributed algorithm for the economic dispatch problem, with the goal of minimizing the aggregated cost of a network of generators, which cooperatively furnish a given amount of power within their individual capacity constraints.

### 3.2 Signed Laplacian based protocols

When the gains \( K_{ij} \) in Fig. 2 (namely, the weights on the edges of the graph modeling a multi-agent system) are real scalars but may be positive or negative, the resulting Laplacian is called the ‘signed Laplacian’. Bipartite consensus, cluster consensus, optimization over convergence speed, affine formation control, and distance-based localization call for a signed Laplacian based approach. A signed Laplacian can be thought of as a generalization of ordinary Laplacian, and many new treatments on signed Laplacian have to be developed due to its distinct features.

#### 3.2.1 Bipartite consensus

For a multi-agent system, ‘bipartite consensus’ means that the states of all the agents converge to a value which is the same in modulus but not in sign. For this problem, some edges of the graph are weighted by positive numbers while some others are weighted by negative numbers. A positive weight is associated to a friend relationship between two agents linked by an edge, while a negative weight is associated to an enemy relationship between two agents linked by an edge (Wasserman and Faust, 1994; Easley and Kleinberg, 2010). A graph with signed weights is said to be ‘structurally balanced’ if it admits a bipartition \((V_1 \text{ and } V_2)\) of the nodes such that (1) \( V_1 \cup V_2 = V \), (2) \( V_1 \cap V_2 = \emptyset \), (3) \( a_{ij} \geq 0 \) for \( i, j \in V_q \), \( q \in \{1, 2\} \), and (4) \( a_{ij} \leq 0 \) for \( i \in V_q \) and \( j \in V_r, q \neq r, q, r \in \{1, 2\} \).

It is said to be ‘structurally unbalanced’ otherwise.

For a graph with signed weights \( a_{ij} \), one type of signed Laplacian \( L^s = [l_{ij}^s] \in \mathbb{R}^{n \times n} \) is defined in the
following form (Kunegis et al., 2010):

$$p_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} |a_{ik}|, & j = i, \\ -a_{ij}, & j \neq i. \end{cases}$$  \quad (15)

For a symmetric $L^s$, it was shown in Kunegis et al. (2010) that the signed Laplacian $L^s$ is positive semi-definite and that it is positive definite if and only if the graph is structurally unbalanced.

In Altafini (2013), a control law was proposed to solve the bipartite consensus problem using signed Laplacian. Just like the ordinary Laplacian for the consensus problem, one can have the same gradient Laplacian. Just like the ordinary Laplacian for the connected and the weights on pairs of edges of the same nodes have the same sign when it is a directed graph.

The consensus condition in Section 3.1.1 is known as ‘complete’ in the sense that all the agents are required to converge to the same state. However, a real-world network may be composed of multiple smaller subnetworks, called clusters. As a result, agents in the network may reach more than one consistent state, while the agents in the same cluster reach consensus. Very recently, increasing attention has been paid to cluster consensus (Wu and Chen, 2009; Wu et al., 2009; Lu X et al., 2010a; 2010b; Yu and Wang, 2010; Liu and Chen, 2011; Xia and Cao, 2011; Han Y et al., 2013; Qin and Yu, 2013), by which it means that for any initial states of the nodes, not only all the nodes within the same cluster reach complete consensus, but also there is no consensus between any two different clusters. Cluster consensus can find examples in engineering control (Passino, 2002), distributed computation (Hwang et al., 2004), etc.

The cluster consensus problem is often considered in the following extensively studied model that consists of $n$ coupled agents in $m$ clusters:

$$\dot{x}_i = f_i(t, x_i) + c\Gamma \sum_{j=1, j \neq i}^n a_{ij}(x_j - x_i),$$  \quad (18)

where $x_i \in \mathbb{R}^p$ denotes the state of agent $i$ ($i = 1, 2, \ldots, n$), $f_i : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^p$ is continuous and globally Lipschitz, $c > 0$ is the coupling strength, $\Gamma = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n)$ with $\gamma_k \geq 0$ ($k = 1, 2, \ldots, n$) is a diagonal matrix denoting the inner coupling, and $a_{ij}$ is the coupling coefficient from agent $j$ to agent $i$ for $i \neq j$.

Denote the $m$ clusters as

$$\begin{cases} C_1 = \{1, 2, \ldots, r_1\}, \\ C_2 = \{r_1 + 1, r_1 + 2, \ldots, r_2\}, \\ \vdots \\ C_m = \{r_{m-1} + 1, r_{m-1} + 2, \ldots, n\}, \end{cases}$$

where $1 \leq r_1 < r_2 < \ldots < r_{m-1} < n$. Then the Laplacian matrix of the graph modeling the system can be written in the following block matrix form:

$$L = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1m} \\ L_{21} & L_{22} & \cdots & L_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ L_{m1} & L_{m2} & \cdots & L_{mm} \end{bmatrix},$$

where $L_{ij}$ $(1 \leq i, j \leq m)$ specifies the coupling from cluster $C_j$ to $C_i$. In order to make the cluster consensus problem solvable, it is often assumed that

$$\sum_{j \in C_l} a_{ij} = \text{constant}, \quad \forall i \in C_k, \ k \neq l.$$  

This means that for nodes within the same cluster, the sums of the incoming weights from the same other cluster are the same. A simple case is that the constant is 0 for any $k$ and $l$, which is also termed the ‘in-degree balanced’ condition. This in-degree balanced condition shows that the inter-cluster coupling may be either positively or negatively weighted and indeed both signs are required.
To guarantee cluster consensus, it is usually assumed that different clusters of nodes have different self-dynamics $f_i(t, x_i)$ and nodes in the same cluster have the same self-dynamics (Lu W et al., 2010; Xia and Cao, 2011), or that there is a leader for each cluster of nodes and such leaders have no coincidence with each other (Liu and Chen, 2011; Qin and Yu, 2013).

A comparison summary of consensus, bipartite consensus, and cluster consensus is given in Table 1.

### 3.2.3 Optimization over convergence speed

In distributed multi-agent coordination, to achieve some optimal features such as convergence speed, the edge weights of the network need to be selected to maximize or minimize specific cost functions. As pointed out in Boyd (2006), signed weights may improve the convergence speed such as in consensus. Some specific cases of this general problem have been addressed in a series of recent papers.

1. Fastest linear averaging. Find weights in a distributed averaging network, driven by random noise, that minimize the steady-state mean-square deviation of the node values (Xiao et al., 2007).

2. Absolute algebraic connectivity. Find edge weights that maximize the algebraic connectivity of the graph (i.e., the smallest positive eigenvalue of its Laplacian matrix). The optimal value is called the absolute algebraic connectivity by Fiedler (de Abreu, 2014).

3. Fastest mixing Markov chain. Find edge transition probabilities that give the fastest mixing Markov process on the graph (Sun et al., 2006).

4. Minimum total effective resistance. Find edge weights that minimize the total effective resistance of the graph. This is the same as minimizing the average commute time from any node to any other, in the associated Markov chain (Ghosh et al., 2008).

5. Least steady-state mean-square deviation. Find weights in a distributed averaging network, driven by random noise, that minimize the steady-state mean-square deviation of the node values (Xiao et al., 2007).

In many interesting cases, the problems are convex, involving minimizing a convex function (or maximizing a concave function) over a convex set. In Boyd (2006), a variety of standard methods were provided to effectively solve the aforementioned problems. We take one example here. For an undirected graph with $m$ edges, a Laplacian can be written in the following form:

$$L(w) = \sum_{l=1}^{m} w_l b_l b_l^T = B \text{diag}(w) B^T, \quad (19)$$

where $w_l$ is the weight of edge $l$, $\text{diag}(w) \in \mathbb{R}^{m \times m}$ is the diagonal matrix formed from $w = [w_1, w_2, \ldots, w_m]^T \in \mathbb{R}^m$, and $B = [b_1, b_2, \ldots, b_m] \in \mathbb{R}^{n \times m}$ is the incidence matrix of the graph. It is a fact that the Laplacian of any undirected graph is positive semi-definite and has the smallest eigenvalue at 0. We define the eigenvalues of the Laplacian matrix $L$ as

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$ 

Let $\phi$ be a symmetric closed convex function defined on a convex subset of $\mathbb{R}^{n-1}$. Then

$$\psi(w) = \phi(\lambda_2, \lambda_3, \ldots, \lambda_n)$$

### Table 1: A comparison summary of consensus, bipartite consensus, and cluster consensus

<table>
<thead>
<tr>
<th>Type</th>
<th>Dynamics</th>
<th>Laplacian</th>
<th>No_0</th>
<th>Geometric condition</th>
<th>Graphical condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bipartite</td>
<td>$\dot{x} = -L^s x$</td>
<td>Signed</td>
<td>A simple zero</td>
<td>$\ker(L^s) = \left{ a \begin{bmatrix} 1 &amp; 0 \ 0 &amp; 1 \end{bmatrix} - a \begin{bmatrix} 0 &amp; 0 \ 1 &amp; 1 \end{bmatrix} \right}$</td>
<td>Structurally balanced</td>
</tr>
<tr>
<td>Consensus</td>
<td>$\dot{x} = -L x$</td>
<td>Ordinary</td>
<td>A simple zero</td>
<td>$\ker(L) = {a1}$</td>
<td>Rooted</td>
</tr>
<tr>
<td>Cluster</td>
<td>$\dot{x} = -L x$</td>
<td>Signed</td>
<td>$m$ zeros</td>
<td>$\ker(L) = \left{ a_1 \begin{bmatrix} 1 \ 0 \ \vdots \ 0 \end{bmatrix} + \cdots + a_m \begin{bmatrix} 0 \ 1 \end{bmatrix} \right}$</td>
<td>Each cluster is rooted</td>
</tr>
</tbody>
</table>

*No_0: number of zero eigenvalues*
is a convex function of $w$. Thus, a symmetric convex function of positive eigenvalues yields a convex function of the edge weights.

Consider the optimization problem with the general form:

$$\min \psi(w) \quad \text{s.t.} \quad w \in W, \quad (20)$$

where $W$ is a closed convex set and the optimization variable is $w \in \mathbb{R}^n$. The problem (20) is to choose edge weights on the graph, subject to some constraints, to minimize a convex function of positive eigenvalues of the associated Laplacian matrix, which often leads to signed weights and signed Laplacian.

### 3.2.4 Affine formation control

Affine formation is a new type of collective pattern in multi-agent systems, which was introduced in Lin et al. (2013a) and Wang et al. (2014b). An ‘affine formation’ represents a class of collective configurations that preserve collinearity and ratios of distances (i.e., the agents lying on a line initially still lie on a line and maintain the ratio of distances after transformation). Identically, an affine formation of a target configuration $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ with each $\xi_i \in \mathbb{R}^d$ for $1 \leq i \leq n$ is any configuration in its ‘affine image’, defined as

$$A(\xi) := \{q = (q_1, q_2, \ldots, q_n) \mid q_i = A\xi_i + a, A \in \mathbb{R}^{d \times d}, a \in \mathbb{R}^d, i = 1, 2, \ldots, n\},$$

or equivalently,

$$A(\xi) := \{q = (I_n \otimes A)\xi + 1_n \otimes a \mid A \in \mathbb{R}^{d \times d}, a \in \mathbb{R}^d\}.$$

In Lin et al. (2013a), a signed Laplacian was introduced, which satisfies

$$L1_n = 0 \quad \text{and} \quad (L \otimes I_d)\xi = 0,$$

where $\xi$ is the target configuration and $d$ is the dimension of the ambient space that the agents lie in. In general, the signed Laplacian contains both positive and negative off-diagonal entries. The signed Laplacian is then used to solve the affine formation control problem. That is, for a group of $n$ agents whose states are denoted by $z = [z_1^T, z_2^T, \ldots, z_n^T]^T$ with $z_i \in \mathbb{R}^d$, the signed Laplacian based protocol

$$\dot{z} = -(L \otimes I_d)z$$

is able to steer the agents to form an affine formation of $p$ under certain conditions. It is shown that an affine formation is stabilizable over an undirected graph if and only if the undirected graph is universally rigid, while an affine formation is stabilizable over a directed graph if and only if the directed graph is $(d + 1)$-rooted.

For the undirected graph case, the signed Laplacian $L$ can be written in the form of Eq. (19). Thus, an effective Laplacian to solve the affine formation control problem can be found by solving the following convex optimization problem:

$$\min \psi(w) \quad \text{s.t.} \quad w \in W \quad \text{and} \quad (L(w) \otimes I_d)\xi = 0. \quad (22)$$

For the directed graph case, it was shown in Wang et al. (2014b) that for almost all signed Laplacian associated to a $(d + 1)$-rooted graph, a real diagonal matrix $D$ exists to assign the eigenvalues of $DL$ in the right-half complex plane. In other words, by proper scaling, an effective Laplacian can be found to solve the affine formation control problem. With the same idea, Han T et al. (2014a; 2014b) addressed the formation merging control problem in the 3D space under directed and switching topologies, which merges a group of followers and a group of leaders into a single rigid formation.

### 3.2.5 Distance-based localization

Network localization is one of the primary functions that are commonly desired in spatially distributed multi-agent systems (e.g., sensor networks or robotic networks), as the positional information may crucially help decide an agent’s behaviour or identify the meaning of the data collected by the agents. Localization is usually related to solve linear or nonlinear equations, which come from the constraints in terms of the Euclidean coordinates of all the agents and all the locally available inter-agent measurements.

For inter-agent distance measurements, Khan et al. (2009) developed a barycentric coordinate based localization approach, which converts the nonlinear distance constraint to a linear equation related to an ordinary Laplacian. Later, the idea was generalized in Diao et al. (2014) by relaxing the assumption that each sensor node lies inside the convex hull spanned by its neighbors and all sensor nodes lie inside the convex hull spanned by the anchor nodes,
which then calls for a signed Laplacian.

Consider a static sensor network in the plane, composed of \( m \) ‘anchor nodes’ (their coordinates are known in a global coordinate system \( \Sigma_b \)) and \( n \) ‘free nodes’ (their coordinates in \( \Sigma_b \) are unknown and need to be determined). A sensor network is commonly modeled as an undirected graph \( G = (V, E) \), with each vertex \( i \in V \) corresponding to a sensor node (either an anchor node or a free node) and each edge \( (i,j) \in E \) indicating that sensor nodes \( i \) and \( j \) are able to communicate with each other and that the distance between \( i \) and \( j \) is available to both sensor nodes.

Recall from Goldenberg et al. (2006) that if all free nodes in the network are localizable, then every free node in \( G \) has at least three disjoint paths from the set of anchor nodes. This implies that locally each free node has at least three neighbors in \( G \). For any free node \( i \in V \), denote by \( N_i \) the set of all its neighbors in \( G \). Moreover, denote by \( p_i \in \mathbb{C} \) the coordinate of sensor node \( i \) in \( \Sigma_b \) (it is represented as a complex number just for notation simplicity). We say that the real constants \( a_{ij} \) (\( j \in N_i \)) are barycentric coordinates of node \( i \) with respect to its neighbors if the following two properties hold:

\[
\text{linear precision: } p_i = \sum_{j \in N_i} a_{ij} p_j, \quad (23)
\]

\[
\text{constant precision: } \sum_{j \in N_i} a_{ij} = 1. \quad (24)
\]

Given inter-agent distance measurements, the barycentric coordinates \( a_{ij} \) can then be computed, based on which the aggregated constraint of (23) can be written as

\[
p = Ap, \quad (25)
\]

where \( p = [p_1, p_2, \ldots, p_{m+n}]^T \) and \( A \) is the matrix with the \((i,j)\)th entry being \( a_{ij} \). Due to property (24), it is then clear that \( L := I - A \) is a Laplacian matrix satisfying \( L1 = 0 \). Note that in general, the barycentric coordinates \( a_{ij} \) may be positive or negative. Thus, \( L \) is a signed Laplacian. Eq. (25) can be re-written as

\[
Lp = 0. \quad (26)
\]

Without loss of generality, write \( p = [p_s^T, p_a^T]^T \), where \( p_a \) is the vector of the Euclidean coordinates of all anchor nodes and \( p_s \) is the vector of the Euclidean coordinates of all free nodes. Then the localization problem can be solved by solving for \( p_s \) from the linear equation (26) for given \( p_a \).

In addition to the aforementioned localization scheme that relates to signed Laplacian, there are also other localization approaches related to graph Laplacian, see for example, the kernel locality preserving projection (KLPP) technique (Wang et al., 2009) and the semi-supervised Laplacian regularized least squares algorithm (Chen et al., 2011).

### 3.3 Complex Laplacian based protocols

When the gains \( K_{ij} \) in Fig. 2 (namely, the weights on the edges of the graph modeling a multi-agent system) are complex numbers, the resulting Laplacian is called the ‘complex Laplacian’. Compared with real-valued Laplacian, complex Laplacian exhibits more freedoms and thus can be used to solve formation shape control and sensor localization in the plane without requiring all the agents to have a common sense of direction.

#### 3.3.1 Similar formation control

Similar formation control refers to the control task that aims to steer a group of agents to form a geometry pattern of the same shape as desired regardless of its size. A similar formation is one obtained from the target configuration via rotation, translation (horizontal and vertical), and scaling, and thus has four degrees of freedom. Similar formation control using complex Laplacian was proposed in Ding et al. (2010), and then extended and generalized in Ding et al. (2012), Han et al. (2012), Han Z et al. (2013; 2014), Wang et al. (2012a; 2012b; 2014a), and Lin et al. (2013b; 2014). The goal is to drive a network of agents in the plane to form a formation shape as desired while the size of the target formation is not a concern. This is motivated mainly by the observation that if the size of the formation can be varied, the whole formation can dynamically adapt to changes in the environment such as passing through a narrow area, adapt to changes of their ongoing tasks, and respond to unseen threats.

First, several notions related to similar formation are presented. In the plane, a tuple of \( n \) complex numbers

\[
\xi = [\xi_1, \xi_2, \cdots, \xi_n]^T
\]

is called a ‘target configuration’ for \( n \) agents, which
defines a formation pattern in a specific coordinate system. Usually, two agents are not expected to overlap each other, and thus we assume that

$$\xi_i \neq \xi_j \quad \text{for} \ i \neq j.$$  

A similar formation has four degrees of freedom, namely, translation (horizontal and vertical), rotation, and scaling, which can be defined as

$$F_\xi = c_1 1_n + c_2 \xi,$$

where $c_1, c_2 \in \mathbb{C}$.

In Wang et al. (2012b), the formation control law based on complex Laplacian for single-integrator agents was given as

$$\dot{z}_i = \sum_{j \in \mathcal{N}_i} w_{ij}(z_j - z_i), \quad i = 1, 2, \ldots, n, \quad (27)$$

where $z_i \in \mathbb{C}$ represents the position of agent $i$, and $w_{ij} = k_{ij} e^{\alpha_{ij}}$ is a complex weight, for which $k_{ij} > 0$, $\alpha_{ij} \in [-\pi, \pi]$, and $\iota = \sqrt{-1}$ is the imaginary unit.

The aggregated dynamics of the $n$ agents under control law (27) turns out to be

$$\dot{z} = -Lz, \quad (28)$$

where $z = [z_1, z_2, \ldots, z_n]^T \in \mathbb{C}^n$ and $L$ is the complex Laplacian.

Unlike real-valued Laplacian, complex Laplacian may not have all eigenvalues in the right-half complex plane. Therefore, to stabilize system (28), a pre-multiplication of a diagonal complex matrix $D$ may be necessary. Thus, system (28) changes to

$$\dot{z} = -DLz, \quad (29)$$

where $D = \text{diag}(d_1, d_2, \ldots, d_n)$ is diagonal and invertible. It is certain that the null space of $DL$ is the same as the one of $L$. Thus, the two systems have the same equilibrium formation, and the basic idea of solving the formation control problem is as follows: First, find a complex Laplacian $L$ such that the set of all configurations with the desired formation shape is exactly the null space of $L$. Second, find an invertible and diagonal matrix $D$ to assign the eigenvalues of $DL$ such that all trajectories converge to form the desired formation shape.

As shown in Wang et al. (2012a; 2012b; 2013) and Lin et al. (2014), if the graph is undirected and 2-rooted, then for any formation vector $\xi$, a complex Laplacian exists such that its null space equals the set of all configurations with the desired formation shape as $\xi$; if the graph is directed and 2-rooted, then for any ‘generic’ formation vector $\xi$ (a configuration $\xi$ is said to be generic if the coordinates $\xi_1, \xi_2, \ldots, \xi_n$ do not satisfy any non-trivial algebraic equation with integer coefficients (Gortler et al., 2010)), a complex Laplacian exists such that its null space equals the set of all configurations with the desired formation shape as $\xi$. Moreover, for both cases, an invertible, complex, and diagonal matrix $D$ exists, which can arbitrarily assign the eigenvalues of $DL$ if the graph is 2-rooted.

With a small number of knowledgeable agents in the group knowing the desired size of the target formation, Lin et al. (2014) showed how a formation with the desired size can be accomplished. In addition, Han et al. (2012) and Lin et al. (2013b) solved the similar formation control problems over a leader-follower network based on complex Laplacian. The formation manoeuvring problem with a constant velocity was addressed in Han et al. (2013; 2014).

A comparison summary of translational formation control, affine formation control, and similar formation control is given in Table 2.

### 3.3.2 Relative position based localization

Complex Laplacian also plays a very important role in sensor network localization. In particular, for sensor nodes with relative position measurements on non-consistent local frames (i.e., the orientations of local frames on different nodes are different and are not known), the localization problem of determining all node positions has been rarely investigated. Diao et al. (2013) first addressed the relative position based localization problem by adopting the idea of using complex Laplacian.

For a sensor network $G$ containing $m$ location-known anchors and $n$ sensor nodes to be localized, called free nodes, denote by $p_i \in \mathbb{C}$ the coordinate of node $i$ (either an anchor or a free node) in a global coordinate system $\Sigma_g$. For every free node $i$, suppose that it measures the relative positions of its neighbors in its own frame $\Sigma_i$. We denote by $\theta_i$ the orientation difference between $\Sigma_i$ and $\Sigma_g$. Then the relative position information in node $i$’s local frame $\Sigma_i$ can be represented as

$$y_{ij} = e^{i\theta_i}(p_j - p_i), \quad j \in \mathcal{N}_i,$$
which is available to node $i$. Thus, with a sufficient number of neighbors, sensor node $i$ can solve a set of complex coefficients $w_{ij}$ to satisfy
$$
\sum_{j \in N_i} w_{ij} y_{ij} = 0.
$$
(30)

Notice that for the same set of complex coefficients $w_{ij}$, Eq. (30) implies
$$
\sum_{j \in N_i} w_{ij} (p_j - p_i) = 0.
$$
(31)

Writing down all these equations for all nodes, we have
$$
Lp = 0,
$$
(32)
where $p = [p_a^T, p_s^T]^T$ with $p_a$ and $p_s$ being the aggregated coordinate vectors of all anchor nodes and all free nodes respectively, and $L$ is an $(m+n) \times (m+n)$ complex Laplacian, which associates to the graph with complex weights $w_{ij}$ solved from Eq. (30). Note that the anchor nodes do not need to measure the relative positions of their neighbors. Thus, the complex Laplacian must be of the following form:
$$
L = \begin{bmatrix}
0 & 0 \\
B & H
\end{bmatrix},
$$
where $B$ indicates the links from the anchor nodes to free nodes and $H$ indicates the links from other free nodes. Thus, Eq. (32) turns out to be
$$
Bp_a + Hp_s = 0,
$$
(33)
and the localization problem becomes to find a solution $p_s$ from linear equation (33).

For the localization problem using relative position measurements, a necessary and sufficient condition was presented for localizability in terms of 2-reachability of the sensing graph (Diao et al., 2013; Lin et al., 2015). Moreover, a distributed and iterative localization algorithm was provided as well to compute the coordinates of each sensor node in the global coordinate system $\Sigma_g$, which requires only communication between neighbors.

### 3.4 Generalized Laplacian based protocols

Besides positive real numbers, both positive and negative real numbers, and complex numbers, the gains $K_{ij}$ in Fig. 2 can be matrices, variables, or even dynamic systems. With these generalized weights and generalized Laplacian, many more realistic scenarios can be taken into account and many more complicated control tasks can be addressed in the same framework. We will survey matrix-valued Laplacian, time-varying Laplacian, and dynamic Laplacian as well as relevant distributed coordination problems.

#### 3.4.1 Bearing-angle based localization

In some applications such as bearing-angle based localization, a matrix-valued Laplacian will be adopted to solve the localization problem. The work of Zhu and Hu (2014) is an example, which aims to determine the locations of all sensor nodes in a network given the angle-of-arrival (AOA) measurements among neighboring nodes together with the absolute coordinates of several anchor nodes. To solve the AOA based localization problem, a matrix-valued Laplacian $L$ was constructed using locally available AOA measurements, which was called the ‘stiffness matrix’ in Zhu and Hu (2014). To be more specific, the matrix-valued Laplacian used in AOA localization has the block matrix form $L = [L_{ij}]$ with $L_{ij} \in \mathbb{R}^{2 \times 2}$ given by
$$
L_{ij} = \begin{cases}
\sum_{k \in N_i} a_{ik} P_{sk}, & i = j, \\
-a_{ij} P_{ij}, & i \neq j,
\end{cases}
$$

<table>
<thead>
<tr>
<th>Formation</th>
<th>Dynamics</th>
<th>Dimension</th>
<th>Laplacian</th>
<th>No_0</th>
<th>Geometric condition</th>
<th>Graphical condition</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translational</td>
<td>$\dot{z} = -Lz + \xi$, $z \in \mathbb{C}^n$</td>
<td>2 or $d$</td>
<td>Ordinate, $L \in \mathbb{R}^{n \times n}$</td>
<td>1</td>
<td>$\ker(L) = {a1 : a \in \mathbb{R}}$</td>
<td>Rooted</td>
<td>Stable</td>
</tr>
<tr>
<td>Similar</td>
<td>$\dot{z} = -Lz_i$, $z \in \mathbb{C}^n$</td>
<td>2</td>
<td>Complex, $L \in \mathbb{C}^{n \times n}$</td>
<td>2</td>
<td>$\ker(L) = {c_1 1 + c_2 \xi : c_1, c_2 \in \mathbb{C}}$</td>
<td>2-rooted</td>
<td>Stabilizable by a complex diagonal $D$</td>
</tr>
<tr>
<td>Affine</td>
<td>$\dot{z} = -(L \otimes I_d)z$, $z \in \mathbb{R}^{nd}$</td>
<td>$d$</td>
<td>Signed, $L \in \mathbb{R}^{n \times n}$</td>
<td>$d + 1$</td>
<td>$\ker(L) = {(I_n \otimes A)\xi + 1 \otimes a : A \in \mathbb{R}^{d \times d}, a \in \mathbb{R}^d}$</td>
<td>$(d + 1)$-rooted</td>
<td>Stabilizable by a real diagonal $D$</td>
</tr>
</tbody>
</table>

No_0: number of zero eigenvalues
where $a_{ij} > 0$ and $P_{ij} \triangleq e_{ij}e_{ij}^T \in \mathbb{R}^{2 \times 2}$ is a projection matrix which can be computed by node $i$ using its AOA measurement about node $j$. If the graph modeling the sensing relationship of the sensor network is an undirected graph, $L$ has positive semi-definite diagonal blocks and negative semi-definite off-diagonal blocks. Moreover, each row and column add up to zeros.

With the matrix weights $a_{ij}P_{ij}$, a distributed protocol was then given in Zhu and Hu (2014) for undirected networks:

$$\dot{\hat{p}}_i = -\sum_{j \in N_i} a_{ij} P_{ij}(\hat{p}_j - \hat{p}_i), \quad \forall i,$$

where $\hat{p}_i \in \mathbb{R}^2$ represents the current estimate of node $i$’s position $p_i$. This dynamics is similar to the continuous-time consensus protocol (2), with the difference being the matrix-valued weights instead of scalars.

Zhong et al. (2014) also developed effective alternatives for sensor localization using AOA measurements. Assuming that AOA information can be mutually measured by pairs of sensor nodes in their local frames, which may not share the same orientation, a distributed localization scheme was then proposed based on matrix-valued Laplacian.

### 3.4.2 Distributed coordination over switching topologies

Instead of fixed constants or matrices, the weights can be time-varying variables, which are often used to solve distributed coordination problems under time-varying settings. A typical case is the consensus problem over a switching topology. That is, the topology switches over time, so is the Laplacian (Lin et al., 2004; Olfati-Saber and Murray, 2004; Ren and Beard, 2004; Moreau, 2005; Cao et al., 2011; Proskurnikov, 2013; Wei and Fang, 2014).

For a multi-agent system with $n$ agents, it is associated with a time-varying weighted graph $G(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the vertex set consisting of all agents in the system and $\mathcal{E}(t)$ is the edge set at time $t$. The time-varying Laplacian is then defined as $L(t) = [l_{ij}(t)]$, with

$$l_{ij}(t) = \begin{cases} -a_{ij}(t), & i \neq j \text{ and } j \in N_i, \\ 0, & i \neq j \text{ and } j \notin N_i, \\ \sum_{k \in N_i} a_{ik}(t), & i = j, \end{cases}$$

where $a_{ij}(t)$ is a time-varying weight.

Thus, the multi-agent system governed by a time-varying Laplacian is as follows:

$$\dot{x} = -L(t)x,$$

where $x = [x_1, x_2, \ldots, x_n]^T$. For such a system, it is well known that the system reaches consensus if there exists $T > 0$ such that for any $t$, the graph associated to $\int_t^{t+T} L(\tau) d\tau$ is connected (for undirected graphs) or rooted (for directed graphs).

With the similar way, the time-varying Laplacian was adopted to solve the consensus problem for double-integrator agents (Kingston and Beard, 2006; Casbeer et al., 2008; Ren and Cao, 2008; Qin et al., 2011; Zhang et al., 2011).

### 3.4.3 Distributed coordination with dynamic Laplacian

For more complex agent models, distributed multi-agent coordination naturally leads to the use of dynamic Laplacian (i.e., the entries of the Laplacian are also dynamic systems). However, very few works are concerned with dynamic Laplacian.

In Oh et al. (2015a), a consensus problem was considered for a network of linear systems, whose models are represented by the multiplication of positive real systems and a single integrator in the $s$-domain. This can be considered as a generalization of single-integrator consensus networks. That is, in the $s$-domain, the distributed multi-agent coordination can be described by

$$sX_i(s) = -\sum_{j \in N_i} a_{ij}G_i(s)(X_i(s) - X_j(s)),$$

where $X_i(s)$ is the Laplace transform of $x_i(t)$. In this scenario, $a_{ij}G_i(s)$ is the weight attributed to edge $(j, i)$, which is a dynamic system. The corresponding Laplacian $L(s)$ can then be defined as follows with $l_{ij}$ being the $(i, j)$th entry:

$$l_{ij} = \begin{cases} -a_{ij}G_i(s), & i \neq j \text{ and } j \in N_i, \\ 0, & i \neq j \text{ and } j \notin N_i, \\ \sum_{k \in N_i} a_{ik}G_i(s), & i = j. \end{cases}$$

Let $X(s) = [X_1(s), X_2(s), \ldots, X_n(s)]^T$. Then the aggregated system of Eq. (36) can be written as

$$sX(s) = -L(s)X(s).$$
Certainly, the properties of the dynamic Laplacian $L(s)$ determine the collective behaviors of the multi-agent system.

The system (36) was also considered in Wang and Elia (2010) to model the consensus network with dynamic communication channels. It was shown that for an undirected graph, system (36) asymptotically reaches consensus if and only if $G$ is connected and the characteristic equation $\det(sI_n + L(s)) = 0$ has a distinct root at zero, and all the other roots are in the open-left-half complex plane.

4 Future research directions

Although there has been substantial progress in multi-agent systems, many fundamental yet challenging problems remain unsolved. Summary and discussion on further issues are provided in the following.

4.1 Spectrum of variant graph Laplacians

Though the spectrum of ordinary graph Laplacian has been well studied, the variants including signed Laplacian, complex Laplacian, and generalized Laplacian have not been fully explored. However, the spectrum of these variant graph Laplacians is very important in understanding how collective behaviors emerge from local coordination and on how to design effective distributed coordination schemes for engineering applications. As reviewed in this paper, some basic links between graph connectivity and the number of zero eigenvalues for variant graph Laplacians have been established, which provide fundamental solutions to a variety of distributed multi-agent coordination problems. However, unlike ordinary Laplacian, signed Laplacian, complex Laplacian, and generalized Laplacian may have eigenvalues in the whole complex plane and exhibit more complicated phenomena. In particular, it is still unclear how the weights of different types affect the spectrum of the corresponding Laplacian. Moreover, it is more desirable to have a distributed approach to find proper weights in some constrained set such that the resulting multi-agent system meets certain specifications, while at the present stage, centralized computation based on global knowledge of the network may still be required. An example is how to find a (block) diagonal matrix $D$ to stabilize a signed Laplacian, complex Laplacian, or matrix-valued Laplacian in a distributed manner.

4.2 Distributed multi-agent coordination over directed and time-varying graphs

In the analysis of distributed multi-agent coordination, the directed graph case shows much more challenges than the undirected graph case and the time-varying graph case leads to more difficulties than the static graph case. However, the nature of a multi-agent network is often directed and time-varying. Within the directed and time-varying setup, many multi-agent coordination problems including formation control, sensor localization, and distributed optimization remain open and relevant research is in its infant stage. To address these challenging issues, further study on variant graph Laplacians associated to directed and time-varying graphs is necessary. New tools have to be developed in the future such that some breakthrough can be made.

4.3 Distributed multi-agent coordination with interaction dynamics

As seen in this survey paper, most up-to-date works still focus on static weights (either a scalar, a complex number, or a matrix) for distributed multi-agent coordination. On the one hand, from the control viewpoint, dynamic feedback can solve some problems that are not able to be solved by static feedback; that is to say, if the weights $K_{ij}$ in Fig. 2 are a dynamic system rather than a static gain, the multi-agent system may have better coordination performance. On the other hand, the dynamics on the edges may also represent the dynamic behaviors of wireless communication channels or data pre-processing techniques such as filters. Thus, the multi-agent systems with dynamics on interaction links present a more general framework and can unify many realistic systems. Current study such as Oh et al. (2015a) considered a very special case, for which the dynamics $G_i(s)$ in system (36) can be taken out from the summation expression such that $L(s)$ can be decomposed into a product of a diagonal matrix and an ordinary Laplacian.

4.4 Nonlinear multi-agent coordination

Graph Laplacian based approaches are linear approaches for distributed multi-agent coordination. However, the world is nonlinear. For example, the
agent model may be nonlinear such as unicycle. The measurement output may be nonlinear such as formation control with only distance measurements. The control specification may also be nonlinear, e.g., to maintain desired distances between pairs of agents. Moreover, the coordination law may have to call for a nonlinear one, as a linear one may not be competent such as in solving the rendezvous problem (Lin et al., 2007). Graph Laplacian based linear approaches, however, serve a starting point for nonlinear multi-agent coordination research. Thus, it is fundamental and systematic to study nonlinear multi-agent coordination by moving the coordination results from linear setup to nonlinear setup.

4.5 Distributed coordination of heterogeneous agents

Heterogeneous agent networks are a common form of multi-agent systems, meaning that the agents in a network may have different sensing and communication capabilities, different dynamic models, and different autonomy. One of the challenges in heterogeneous agent networks is the missing of a unified framework and analysis tool in determining the system’s overall performance and capabilities when the agents are non-homogeneous and equipped with different resources. Interesting example problems include sensor localization and formation control, for which a combination of different measurements (e.g., inter-agent distances, inter-agent bearings, and inter-agent relative positions) is used in a network by different agents. Another interesting example is the synchronization problem with heterogeneous dynamics, whose individual systems are different and in particular the state dimensions may be different.

5 Conclusions

Throughout the paper, we come to understand that the graph Laplacian plays a significant role in distributed multi-agent coordination, including consensus, formation control, sensor localization, distributed optimization, etc. Though with different focuses on different research issues in multi-agent systems, they are commonly based on graph Laplacians that may be of different types but have the same structure. Thus, the analysis of coordination behaviors can be transformed to the analysis of variant graph Laplacians. This paper surveyed recent developments in multi-agent systems, particularly related to graph Laplacian based approaches, and highlighted several open fundamental yet challenging research problems. We expect that this paper provides a helpful overview of distributed multi-agent coordination principles for anyone who will conduct research in multi-agent systems.

References


