Formation of heterogeneous multi-agent systems under min-weighted persistent graph

Yanlong Sun, Ziqiang Xu, Xiaoning Zhang and Jing Yan*

Institute of Electrical Engineering,
Yanshan University,
Qinhuangdao 066004, China
Email: yanlongsun@hotmail.com
Email: xuziqiang@163.com
Email: 122175314@qq.com
Email: jyan@ysu.edu.cn
*Corresponding author

Cailian Chen and Xinping Guan

Key Laboratory of System Control and Information Processing,
Ministry of Education of China,
Department of Automation,
Shanghai Jiao Tong University,
Shanghai 200240, China
Email: cailianchen@sjtu.edu.cn
Email: xpguan@sjtu.edu.cn

Abstract: In this paper, we develop a simple and efficient formation control framework for heterogeneous multi-agent systems under min-weighted persistent graph. As the ability of each agent may be different, the architecture of agents is considered to be heterogeneous. To reduce the communication complexity of keeping connectivity for agents, a topology optimisation scheme is proposed, which is based on min-weighted persistent graph. According to the topology of agents, a directed acyclic graph (DAG) is constructed to reflect the signal flow relation of agents, and then the corresponding formation control protocol is designed by using the transfer function model. Apply the proposed method, it is shown that the communication complexity of multi-agent systems is decreased, and the connection safety is improved. Based on signal flow graph analysis and Mason's rule, the convergence conditions are provided to show the agents can keep a formation. Finally, several simulations are worked out to illustrate the effectiveness of our theoretical results.

Keywords: formation; heterogeneous; min-weighted persistent graph; multi-agent systems

Reference to this paper should be made as follows: Sun, Y., Xu, Z., Zhang, X., Yan, J., Chen, C. and Guan, X. (2017) 'Formation of heterogeneous multi-agent systems under min-weighted persistent graph', *Int. J. System Control and Information Processing*, Vol. 2, No. 1, pp.14–30.

Biographical notes: Yanlong Sun received his BE in Measuring and Control Technology and Instrumentations from Yanshan University, Qinhuangdao, China, in 2010, where he is currently pursuing the PhD in Control Science and Engineering. His research interest covers in multi-agent systems and underwater sensor networks.

Ziqiang Xu received his BE in Automation from Yanshan University, Qinhuangdao, China, in 2015. Currently, he is pursuing the PhD of Control Theory and Control Engineering in Yanshan Univerisity. His research interest covers in estimation and localisation for underwater cyber-physic systems.

Xiaoning Zhang received his BE in Conrol Technology and Instrument Professional from Anhui University of Technology, Maanshan, China, in 2014. Currently, he is pursuing the ME of Pattern Recognition and Intelligent System in Yanshan University. His research interest covers in time synchronisation and localisation for underwater cyber-physic systems.

Jing Yan received his BE in Automation from Henan University, Kaifeng, China, in 2008, and the PhD in Control Theory and Control Engineering from Yanshan University, Qinhuangdao, China, in 2014. In 2014, he was a Research Assistant with the Key Laboratory of System Control and Information Processing, Ministry of Education, Beijing, China. He is currently a Lecturer with Yanshan University. His research interests cover in cyberphysical systems, multiagent systems and networked teleoperation systems.

Cailian Chen received his BE and ME in Automatic Control from Yanshan University, Qinhuangdao, China, in 2000 and 2002, respectively, and the PhD in Control and Systems from the City University of Hong Kong, Hong Kong, in 2006. She joined the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 2008, as an Associate Professor, where she is currently a Full Professor. Her research interests include distributed estimation and control of network systems, wireless sensor and actuator network, multiagent systems, and intelligent control systems.

Xinping Guan received his PhD in Control and Systems from the Harbin Institute of Technology, Harbin, China, in 1999. In 2007, he joined the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Distinguished University Professor, the Executive Deputy Dean of the University Office of Research Management. His current research interests include cyber-physical systems, multiagent systems, wireless networking and applications in smart city and smart factory, and underwater sensor networks.

1 Introduction

Since the pioneering works of Reynolds (1987) and Vicsek et al. (1995), the formation problem of multi-agent systems has attracted researchers from a wide range of disciplines, such as biology, physics, robotics, computer science, social science and control engineering, due to its applications in the cooperative control of mobile robots, the design of distributed sensor networks, and so on.

The main challenge in formation control is to design a decentralised control law by depending on local information interaction. Thus, fundamental questions about what local interaction rules are and how they work, are raised. To interpret these questions, some local interaction schemes were proposed, such as game-based method (Hu et al., 2015; Semsar-Kazerooni and Khorasani, 2009), graph-based method (Aguilar and Gharesifard, 2015; Olfati-Saber, 2006), etc. In these references, 'neighbour rule' was widely applied in the topology communication between agents, wherein each agent was required to communicate with its neighbours in the topology interaction graph to keep the connectivity of multi-agent systems. Analysing the action of 'neighbour rule' in formation control, we find that some interactions between agents can be removed during the formation process. Accordingly, the energy consumption with respect to communication between agents is increased. To reduce the communication complexity, Ding et al. (2010) schematically represented the neighbouring relationship of networked agents as a directed acyclic graph (DAG), wherein the cycles in topology interaction graph were deleted. On the other hand, some scholars studied the rigid and persistent configurations to decrease the communication complexity (Zhang et al., 2015). Yan et al. (2015) and Chen et al. (2015) studied optimally rigid formation, wherein the least communication number was required. Although the number of communication links is the least, these rigid graphs are not unique. If the edges of the graphs are weighted by the required communication energy, how to find the min-weighted graph becomes a challenging problem.

In addition to the references mentioned so far, most of the above works are used to deal with homogeneous multi-agent systems, i.e., the model of agents can be transformed into single-integrator or double-integrator dynamics through feedback linearisation. The homogeneous architecture is attractive because it is resilient to individual failures, but suffers from poor fundamental limits and performance (Gupta and Kumar, 2000). On the other hand, heterogeneous multi-agent systems have been shown to be superior to homogeneous ones due to their potential to increase systems lifetime and reliability experimentally (Yarvis et al., 2005).

This paper proposes a simple and efficient formation control framework for heterogeneous multi-agent systems. To reduce the energy consumption caused by the communication between agents, a topology optimisation scheme is proposed to decrease the communication complexity during the formation process. First, a min-weighted rigid graph is generated to delete the unnecessary interaction between agents. By directing the edges of the min-weighted rigid graph, we can easily obtain the min-weighted persistent graph to represent the neighbouring relationship of networked agents. To reflect the signal flow relation of agents, the min-weighted persistent graph is transformed into a DAG. Second, the model of each agent is provided, in which the architecture of the multi-agent systems is heterogeneous. Moreover, a formation control protocol is designed by constructing a feedback system. Finally, the convergence conditions are provided to show the agents can keep a formation by using signal flow graph analysis and Mason's rule.

2 Preliminaries and problem formulation

2.1 Directed acyclic graph

A *graph* $\mathcal{G}$ is a pair that consists of a set of vertices $\mathcal{V} = \{1, 2, \ldots, N\}$ and edges $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, j \neq i\}$ (i.e., the graph is in general directed and has no self-loop). The graph is said to be *undirected* if $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E}$, and the number of edges is defined by $|\mathcal{E}|$.

An edge (i, j) in a directed graph is said to be *incoming* with respect to j and *outgoing* with respect to i. Such an edge has vertex i as a *tail* and vertex j as a *head*. The *in-degree* of a vertex in a directed graph is defined as the number of edges that have this vertex as a head. If (i, j) is an edge, then i and j are adjacent. A *directed cycle* is a connected directed graph where every vertex is incident with one incoming and one outgoing edge. An *directed acyclic graph* (DAG) is a directed graph with no cycle. We define the nodes belong to the neighbour set of node i ($\mathcal{N}_i$) if they can receive information from node i, which is

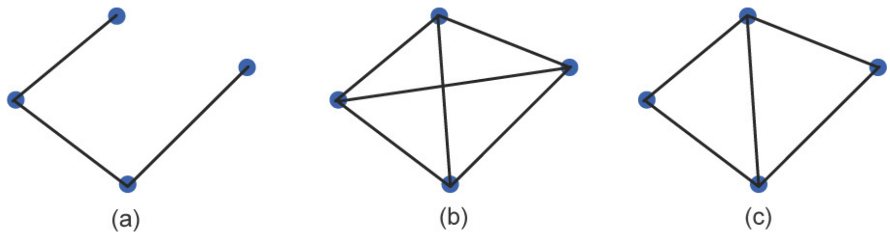
$$\mathcal{N}_i = \{j \in [1, 2, \ldots, N], (j, i) \in \mathcal{E}\}, \forall i = 1, 2, \ldots, N. \quad (1)$$

If node j is a neighbour of node i, we define $a_{ji} = 1$, otherwise, $a_{ji} = 0$.

2.2 Rigid graph

A graph is said to be *rigid* (Yan et al., 2015; Luo et al., 2009) if the corresponding set of distance constraint is sufficient to maintain the formation shape, i.e., a graph is *rigid* if it can provide that all prescribed distance constraints are satisfied during a continuous displacement. Or else, it is called *flexible*. A *minimally rigid graph* is a rigid graph where no edge can be removed without losing rigidity. Figure 1 is an example of the flexible, rigid and minimally rigid graphs.

Figure 1 An example of the flexible, rigid and minimally rigid graphs: (a) flexible graph; (b) rigid graph and (c) minimally rigid graph (see online version for colours)



A trajectory $z_i(t) \in \mathbb{R}^2$ represents the continuous motion of vertex i, and we assume $z_i(t)$ is continuously differential for $\forall i \in \mathcal{V}$. If $\forall (i, j) \in \mathcal{E}$ satisfies $||z_i(t) - z_j(t)|| = \eta$ (where $\eta > 0$ is a constant), and $(z_i(t) - z_j(t))^T(\dot{z}_i(t) - \dot{z}_j(t)) = 0$ when $t = 0$, we say that $\dot{z} = (\dot{z}_1, \dot{z}_2, \ldots, \dot{z}_N)$ is an *infinitesimal flex*. An infinitesimal flex is trivial if it results from a rigid motion of the framework. A framework is said to be *infinitesimally rigid* if it only has trivial infinitesimal flex. Here, we review the concept of *infinitesimal rigidity* as presented in Tay and Whitele (1985). The infinitesimal rigidity of a framework is a stronger condition than rigidity, where all infinitesimally rigid networks are rigid.

Next, we introduce an important matrix of rigid graph in two-dimensional coordinates. Order the coordinates of vertex $\mathcal{V}$ as follows

$$\{z_1^1, z_1^2, z_2^1, z_2^2, \ldots, z_N^1, z_N^2\}. \quad (2)$$

Then a rigidity matrix $M \in \mathbb{R}^{|\mathcal{E}| \times 2N}$ is defined. The elements in the row (i, j) and columns $2i - 1, 2i, 2j - 1, 2j$, are $z_i^1 - z_j^1, z_i^2 - z_j^2, z_j^1 - z_i^1, z_j^2 - z_i^2$, and zero elsewhere.

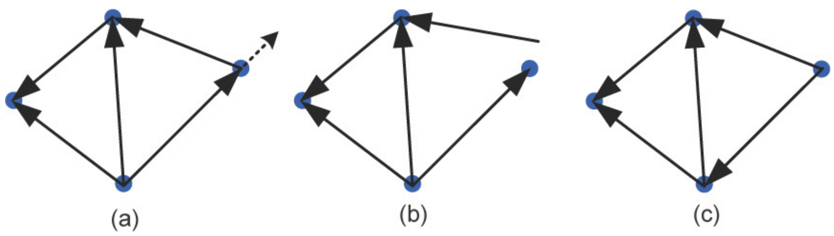
The following properties show the relationship between infinitesimally rigid graph and rigidity matrix.

Property 1 (Tay and Whitele, 1985)**:** Let M be the rigidity matrix of a generic framework of N vertices in $\mathbb{R}^2$. A framework with $N > 2$ vertices in $\mathbb{R}^2$ is infinitesimally rigid if and only if rank$(M) = 2N - 3$.

Property 2 (Luo et al., 2009)**:** An infinitesimally rigid framework $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N vertices and $2N - 3$ edges is *minimally rigid*. If every edge of the framework $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is weighted by its length, a *min-weighted rigid graph* is the minimally rigid graph that has the minimally weighted sum in all infinitesimally rigid graphs.

A *persistent graph* can be obtained from a relative rigid graph by adding directions for all edges. An undirected graph corresponding to the persistent graph must be a rigid graph, but we cannot generate a persistent graph by adding arbitrary directions to the edges of a relative rigid graph (see Figure 2). We define a directed graph to be a persistent graph if and only if any two vertices in the directed graph can maintain their relative distance during any motion. A graph is a *minimally persistent graph* if and only if it is a persistent graph and the relative undirected graph is a minimally rigid graph.

Figure 2 An example of the persistent graph. If we add directions to the edges shown in graph (a) and let it move along the direction of the dashed arrow, the graph will change into the configuration shown in graph (b). So, graph (a) is not a persistent graph. The persistent graph (c) shows that moving any one vertex will not change the configuration (see online version for colours)



Lemma 1 (Hendrickx et al, 2007)**:** *A graph is minimally persistent if and only if it is minimally rigid and no vertex has out-degree larger than 2. A graph is min-weighted persistent if and only if it is min-weighted rigid and no vertex has out-degree larger than 2.*

2.3 Signal flow graph

Subsequently, we use signal flow graph to illustrate the information transmission framework for multi-agent systems. A *signal flow graph* is a network of directed branches, which connects at nodes. Branch jk originates at node j and terminates upon node k, the direction from j to k being indicated by an arrowhead on the branch. Each branch jk is associated with a quantity called W_{kj}, which is a transfer function from agent j to k in this paper.

Each node contains an associated quantity, which is called node signal, and it represents the state of the agent in this paper. To simplify the analysis in Section 3, we define the quantity by W_{kj} but not W_{jk}. A node performs two functions:

- addition of the signals on all incoming branches
- transmission of the total node signal to all outgoing branches.

In a signal flow graph, *sources* are nodes that only have outgoing branches, *sinks* are nodes that only have incoming branches, and *mixed nodes* contains the two types of branches. A *forward path* is one that is from source to sink, where no node is encountered more than once. A *path gain* is the product of the branch gains along that path. A *feedback loop* is a path that forms a closed loop along which each node is encountered once. Loops are *nontouching* if they have no common node.

2.4 Heterogeneous multi-agent systems

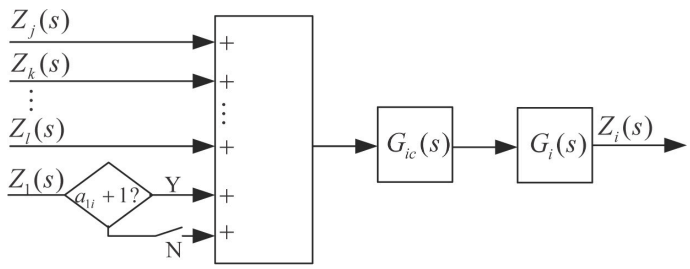
In this paper, we consider a connected heterogeneous multi-agent systems as $\Lambda = \{1, 2, \ldots, N\}$, in which the model of each agent may be different. Meanwhile, a leader-follower strategy is adopted. The agent with stronger communication and computing abilities is considered as leader, and the other ones are followers. In another word, the formation information is divided into two independent parts: global and local parts. The global information is responsible for determining the position of desired formation, in addition,the leader can obtain the global formation information. Local information is used to acquire the relative positions of followers with respect to the frame decided by leader.

At time t, the positions of the leader and follower $i(i \in \Lambda/\{1\})$ are denoted as $z_1(t) \in \mathbb{R}^2$ and $z_i(t) \in \mathbb{R}^2$, respectively. Thus the Laplace transforms of $z_1(t)$ and $z_i(t)$ can be represented by $Z_1(s)$ and $Z_i(s)$, respectively. Figure 3 shows the relation between inputs and outputs for agent i. The model of describing the relation between inputs and outputs for follower agent i is given as

$$Z_i(s) = \sum\nolimits_{j \in N_i} a_{ji} Z_j(s) G_{ic}(s) G_i(s), \tag{3}$$

where $G_i(s)$ is the transfer function which is used to describe the model of heterogeneous agent i. $G_{ic}(s)$ is the designed controller which satisfies different requirements. $\mathcal{N}_i$ is the neighbour set for agent i.

Figure 3 The relation between inputs and output for agent *i*



3 Main results

In this section, we design a formation control framework for heterogeneous multi-agent systems under min-weighted persistent graph. First, to reduce the communication complexity while maintaining connectivity in the formation process, a min-weighted persistent graph is presented to design a topology optimisation scheme. Second, a distributed formation protocol is given to achieve the formation task.

3.1 Topology optimisation scheme

The heterogeneous multi-agent systems are spaced in a bounded region. To guarantee the secure connection of multi-agent systems, each agent needs to receive or transmit information to another one. Several factors may influence the secure connectivity such as distance, channel and network key. Thus, these factors can form the secure connection degree (Jia et al, 2009). Based on this, the following assumption is given.

Assumption 1: *In view of the bounded region, we assume the secure connection degree for each agent is fixed, i.e., the motions of agents do not affect the secure connection degree.*

Based on this, the secure connection degree is denoted by an undirected graph $\mathcal{G}_s = (\mathcal{V}, \mathcal{E}_s)$ where $\mathcal{V} = \{1, 2, \ldots, N\}$ and $\mathcal{E}_s \subseteq \{(i, j) \cup (j, i) : i, j \in \mathcal{V}, j \neq i\}$, and the length of edge reflects the degree of secure. For simplicity of the analysis, we make the following rules: the shorter of the edge, the safer of the connection; if there is no edge between two nodes, the secure connection degree is defined as 0. Then, the problem to reduce communication complexity is transformed into minimising of the length sum for all edges. In view of the property of min-weighted persistent graph, minimising of the length sum for all edges is transformed into forming a min-weighted persistent graph by deleting some unnecessary edges of $\mathcal{G}_s$.

The main generation process of min-weighted persistent graph is summarised as follows.

- according to the secure connection relation, we build the secure connection degree graph $\mathcal{G}_s$
- order all the edges by length from short to long, and build the min-weighted rigid graph $\mathcal{G}_{mwg}$
- based on the rank relation, we add directions to the edges in $\mathcal{G}_{\mathrm{mwg}}$ and build the min-weighted persistent graph $\mathcal{G}_{\mathrm{mwp}}$.

Based on the particular properties and the algorithm of optimally rigid graph generation shown in Luo et al. (2009), we give the detailed generation algorithm of min-weighted persistent graph.

Lemma 2 (Hendrickx et al, 2007)**:** *A rigid graph (with more than one vertex) is minimally persistent if and only if one of the following two conditions is satisfied:*

- *three vertices have an out-degree* 1 *and all the others have an out-degree 2*
- *one vertex has an out-degree* 0*, one vertex has an out-degree* 1*, and all the others have an out-degree 2.*

Corrolary 1: *The min-weighted persistent graph $\mathcal{G}_{\rm mwp}$ presented in Algorithm 1 has the following property: the leader agent z_1 has an out-degree 0, one agent which in a triangle $\tilde{\triangle}$ with the leader has an out-degree 1, all the others have an out-degree 2, and the sum of degree for these agents is $2N-3$.*

Proof: Referring to the particular properties and the algorithm of optimally rigid graph generation provided in Luo et al. (2009), min-weighted persistent graph $\mathcal{G}_{\rm mwp}$ is obtained by using Algorithm 1. From Lemma 1, it is known that $\mathcal{G}_{\rm mwp}$ is also a minimally persistent graph. Notice in Algorithm 1, there are the following actions: add directions to the edges of the triangle, where vertex 1 has two incoming edges, $j(\text{or } k)$ has one incoming edge and $k(\text{or } j)$ does not has incoming edge; for all the rest vertices, each vertex is assigned 2 edges, and adding directions to these edges out of vertices where each vertex has two outcoming edges. Therefore, $\mathcal{G}_{\rm mwp}$ satisfies the second condition in Lemma 2. Compute the sum of degree, so the sum of degree for these agents is $2N-3$. That completes the proof. □

Corrolary 2: *The min-weighted persistent graph $\mathcal{G}_{\rm mwp}$ generated in Algorithm 1 can be denoted by a directed acyclic graph (DAG).*

Proof: The directed acyclic graph should satisfy the following conditions:

- directed
- acyclic.

Obviously, $\mathcal{G}_{\rm mwp}$ is directed. Next, we prove there is no cycle by using mathematical induction.

- Because the triangle which contains leader agent z_1 is the basement of $\mathcal{G}_{\rm mwp}$, we first consider if there exists a cycle in this triangle. When $n=3$, the leader agent has an out-degree 0, one agent has an out-degree 1, and the other one has an out-degree 2. Obviously, there is no cycle in this triangle.

  $\Longrightarrow$ Corrolary 2 holds when $n=3$.

- When $n=h$ $(h>3$ is an integer$)$, we assume there is no cycle. Add a new vertex ℓ, then we should prove there is no cycle when $n=h+1$. We can use apagoge to prove this judgement.

  Make the following assumption: at least one cycle is formed when a new vertex ℓ is added. Notice that Property 2, min-weighted persistent graph is minimally rigid, and minimally rigid graph with n vertices has $2n-3$ edges. Thus, min-weighted persistent graph with $h+1$ and h vertices have $2(h+1)-3$ and $2h-3$ edges, respectively. Therefore, two edges which connect with vertex ℓ are added, and at least one cycle contains vertex ℓ. In order to form the cycles, vertex ℓ should have an out-degree 1, and an in-degree 1. An example can be shown by Figure 4. However in Corrolary 1, it is known that the agents which are not in the triangle $\tilde{\triangle}$ have an out-degree 2. Obviously, the assumption does not hold. Therefore, there is no cycle when $n=h+1$.

  $\Longrightarrow$ Corrolary 2 holds when $n=h+1$.

- For $n\leq N$, there is no cycle by recursive method. That completes the proof. □

Algorithm 1 Generation of min-weighted persistent graph

Initiation: Give the secure connection degree graph $\mathcal{G}_s = (\mathcal{V}, \mathcal{E}_s)$ where $\mathcal{V} = \{1, 2, \ldots, N\}$, $\mathcal{E}_s \subseteq \{(i,j) \cup (j,i) : i, j \in \mathcal{V}, j \neq i\}$, and $|\mathcal{E}_s|$ is the number of edges.

1. Calculate all the length of edges corresponding to the set $\mathcal{E}_s$;
2. Sequence all the edges by increasing;
3. Build the rigidity matrix $\tilde{M}$ of the graph $\mathcal{G}_s$ according to the sequence;
4. Initialized M as the first row of $\tilde{M}$

for $1 : |\mathcal{E}_s|$
  while $rank(M) \leq 2N - 3$
    Add the next row of $\tilde{M}$ to M to form
     a new matrix $\vec{M}$;
   if $\vec{M}$ is full rank
   $M = \vec{M}$;
    %Record the edge corresponding to the row;
   end
  end
end

5. Draw the min-weighted rigid graph $\mathcal{G}_{mwg}$ according to M;
6. Choose a triangle $\tilde{\triangle}$ which contains the leader agent z_1;

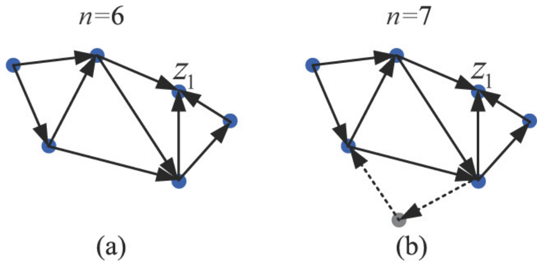
for all the non-zero elements in the column to z_1
  if find two non-zero elements corresponding to $(1, j)$
  and $(1, k)$, and row $(j, k) \in M$
  break;
  end
 set the rows corresponding to $(1, j)$ and $(1, k)$,
  and (j, k) to be 0;
Add directions to the edges of the triangle, where vertex 1 has two incoming edges, j(or k) has one incoming edge and k(or j) does not has incoming edge;
end
while $M \neq 0$
  for all the rest vertices
  Each vertex is assigned 2 edges;
  Add directions to these edges out of vertices, and each
  vertex has two outcoming edges;
  The rows corresponding to these edges are set to be 0;
  end
end
Obtain the min-weighted persistent graph $\mathcal{G}_{\rm mwp}$.

3.2 Formation protocol design for heterogeneous multi-agent systems

In Section 3.1, the secure connection degree relation is described by a directed min-weighted persistent graph $\mathcal{G}_{\rm mwp}$, and an arbitrary edge (i, j) in $\mathcal{G}_{\rm mwp}$ implies vertex j can transmit its information to vertex i. On the other hand, an arbitrary edge (i, j) in signal flow graph implies node j can receive information from i. To construct the relationship between $\mathcal{G}_{\rm mwp}$ and signal flow graph, we reverse the directions of edges in $\mathcal{G}_{\rm mwp}$, then a signal flow graph

$\mathcal{G}^f_{\mathrm{mwp}}$ which reflects the signal flow relationship between agents is obtained. In $\mathcal{G}_{\mathrm{mwp}}$, the leader agent has an out-degree 0, thus in $\mathcal{G}^f_{\mathrm{mwp}}$ the leader agent has an in-degree 0 and it can be considered as a source node, and the other ones are mixed or sink nodes. Besides, there is no cycle in $\mathcal{G}^f_{\mathrm{mwp}}$.

Figure 4 An example of cycle when a vertex is added. (a) denotes that there is no cycle and (b) shows that cycles are formed when a vertex is added (see online version for colours)



Next, we use feedback strategy to design the formation protocol. In $\mathcal{G}^f_{\mathrm{mwp}}$, the nodes represent the states of agents, the edges denote the direction of signal flows, and the gain of each edge is a transfer function from one state to another. First, we recall Mason's rule to calculate the overall transfer function from one source to one sink, i.e., the overall function from the leader to an arbitrary agent.

Lemma 3 (Mason's rule) (Mason, 1956)**:** *In a signal flow graph containing only one source and one sink, the overall gain T is given by*

$$T = \left(\sum T_i \triangle_i\right) / \triangle \tag{4}$$

with

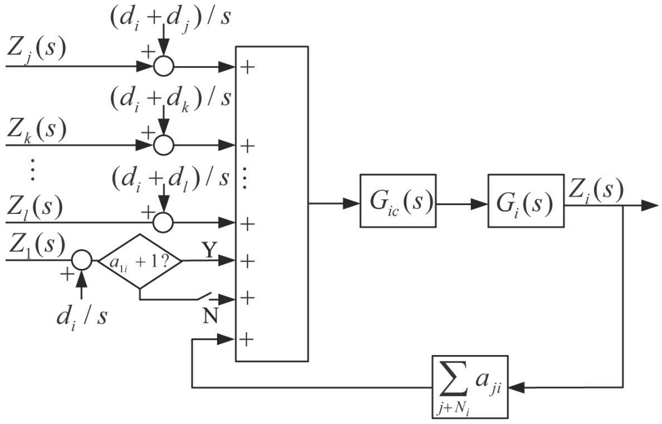
$$\triangle = 1 - \sum\nolimits_m P_{m1} + \sum\nolimits_m P_{m2} - \sum\nolimits_m P_{m3} + \dots, \tag{5}$$

where $\triangle$ is the graph determinant, P_{mr} is the gain product of the mth possible combination of r nontouching loops$(m, r \in N)$, $\triangle_i$ is the determinant of the remaining subgraph when the ith forward path is removed, and T_i is the forward path gain.

During the formation process, agent 1 is the leader, the other ones are followers, and each follower should keep a desired relative distance with the leader. Then the leader is considered as reference point for the formation, and we make the following design: agent i and j should keep relative position $d_i \in \mathbb{R}^2$ and $d_j \in \mathbb{R}^2$ with the leader respectively, where $d_i \neq d_j$. To achieve the formation task, a feedback control strategy is applied to the heterogeneous multi-agent systems. This feedback control strategy can be described by a block diagram as shown in Figure 5. Based on this feedback control strategy, we can get the formation protocol for agent $i(i \in \Lambda/\{1\})$ as followss.

$$Z_i(s) = \frac{\sum_{j \in N_i} a_{ji} G_{ic}(s) G_i(s) [Z_j(s) + (d_i - d_j)/s]}{1 + \sum_{j \in N_i} a_{ji} G_{ic}(s) G_i(s)}. \tag{6}$$

Figure 5 The feedback control strategy of the formation



Lemma 4 (Routh stability criteria) (Krishnamurthi, 1972)**:** *For systems stability, the primary requirement is that all of the roots of the characteristic equation have negative real parts. The Routh stability criterion states that all of the roots of the characteristic polynomial have negative real parts if and only if the following conditions are satisfied: all the elements in the first column of the Routh array are positive.*

Then the convergence condition of the formation for heterogeneous multi-agent systems can be given as follows.

Theorem 1: *Consider the heterogeneous multi-agent systems as shown in Section 2.4. The systems are stable and each follower can keep a desired relative position with the leader, if all the elements in the first column of the Routh array for characteristic polynomial* $(c+1)G_{ic}(s)G_i(s) + 1 = 0$ *have the same sign and* $G_{ic}(s)G_i(s)$ *contains the factor* $1/s$*, where* $G_i(s)$ *is used to describe the model of heterogeneous agent* i*,* $G_{ic}(s)$ *is the designed controller which satisfies different requirements,* $c = \sum_{j\in N_i} a_{ji}$*, and* $i \in \Lambda/\{1\}$*.*

Proof: Rearranging equation (6) and noting that $\breve{Z}_j(s) = Z_j(s) + (d_i - d_j)/s$, we have

$$\frac{Z_i(s)}{\sum_{j\in N_i} a_{ji}\breve{Z}_j(s)} = \frac{G_{ic}(s)G_i(s)}{1 + \sum_{j\in N_i} a_{ji}G_{ic}(s)G_i(s)}. \tag{7}$$

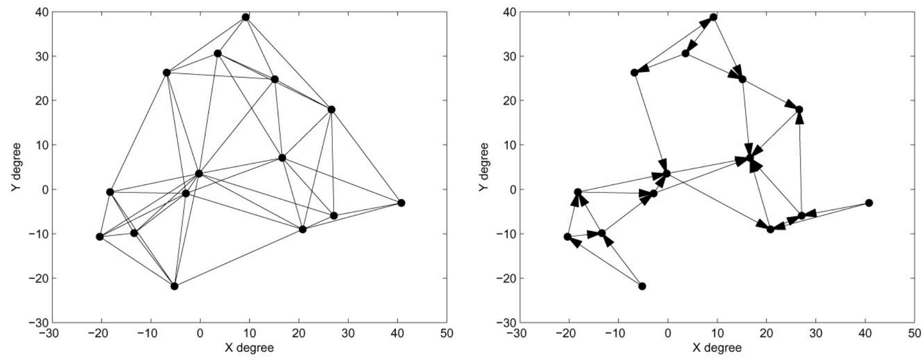
Thus, Figure 5 is refreshed as Figure 6, $\breve{Z}_j(s)$ can be considered as the new control input, $\breve{Z}_0(s) = Z_0(s) + d_i/s$, and the characteristic polynomial for systems (7) is constructed as follows.

$$(c+1)G_{ic}(s)G_i(s) + 1 = 0, \tag{8}$$

where $c = \sum_{j\in N_i} a_{ji}$, $i \in \Lambda/\{1\}$ and $j \in \{1, \ldots, N\}$.

Based on Routh stability criterion in Lemma 4, we can obtain the following conclusion: systems (7) are stable if and only if all the elements in the first column of the Routh array for characteristic polynomial $(c+1)G_{ic}(s)G_i(s) + 1 = 0$ are positive.

Figure 6 Topology optimisation process for secure connection degree: (a) the formation process for agents and (b) the min-weighted persistent graph



(a) (b)

Re-arranging equation (7), we have

$$Z_i(s) = \frac{\sum_{j\in N_i} a_{ji}\breve{Z}_j(s)G_{ic}(s)G_i(s)}{1+\sum_{j\in N_i} a_{ji}G_{ic}(s)G_i(s)} = \frac{\sum_{j\in N_i} a_{ji}\breve{Z}_j(s)}{1/G_{ic}(s)G_i(s)+\sum_{j\in N_i} a_{ji}}$$
$$= \sum\nolimits_{j\in N_i} \frac{a_{ji}}{1/G_{ic}(s)G_i(s)+c}\breve{Z}_j(s). \tag{9}$$

In DAG, there is no cycle. Therefore there is no loop in the signal flow graph, i.e., the determinant of the remaining subgraph and the graph determinant are always 1. Consider the signal flow graph with the leader as its source and agent l as its sink. According to Mason's rule in Lemma 3, the overall gain of this signal flow graph is given as

$$T = \sum T_p\triangle_p/\triangle = \sum T_p = \sum\nolimits_p \prod\nolimits_q \tilde{W}_{pq}, \tag{10}$$

where $\triangle_p = 1$ is the determinant of the remaining subgraph when the pth forward path is removed, T_p is the forward path gain, $\triangle$ is graph determinant which is equal to 1, $\tilde{W}_{pq}$ is the qth edge on the pth forward path in the signal flow graph. According to the superposition principle, the output $Z_l(s)$ is given by

$$Z_l(s) = T\breve{Z}_1(s), \tag{11}$$

where $\breve{Z}_1(s)$ is the refreshed state of the leader.

From equations (10) and (11), we have the following relation.

$$Z_l(s) = \sum\nolimits_p \prod\nolimits_q \tilde{W}_{pq}\breve{Z}_1(s). \tag{12}$$

If the qth edge on the pth forward path in the signal flow graph is labelled as (j,i), then $\tilde{W}_{pq} = W_{ij} = \frac{a_{ji}}{1/G_{ic}(s)G_i(s)+c}$. Moreover, equation (12) can be rearranged into

$$Z_l(s) = \sum_p \prod_q \frac{a_{ji}\breve{Z}_1(s)}{1/G_{ic}(s)G_i(s)+c} = \sum_p \prod_q \frac{a_{ji}[Z_1(s)+d_i/s]}{1/G_{ic}(s)G_i(s)+c}. \tag{13}$$

The output function for agent $\varsigma \in \Lambda/\{1\}$ where $(1,\varsigma)$ is an edge in $\mathcal{G}^f_{\rm mwp}$ can be denoted as

$$Z_\varsigma(s) = \frac{a_{1\varsigma}[Z_1(s) + d_\varsigma/s]}{1/G_{\varsigma c}(s)G_\varsigma(s) + a_{1\varsigma}}. \tag{14}$$

To keep a desired relative position vector with the leader, the output function for agent ς should be $Z_1(s) + d_\varsigma/s$ as $s \to 0$, i.e., $\lim\limits_{s\to 0} \frac{a_{1\varsigma}}{1/G_{\varsigma c}(s)G_\varsigma(s)+a_{1\varsigma}} = 1$. Thus we can obtain the following conclusion: $G_{ic}(s)G_i(s)$ should contain the factor $1/s$. Applying this argument to arbitrary agent in the DAG, we know the following argument: $G_{ic}(s)G_i(s)$ should contain a factor of $1/s$ for every follower.

Remark 1: In order to achieve the formation task, agent should communicate with the other ones. In this paper, wireless communication (or topology) relationship of agents is optimised by graph $\mathcal{G}_{\rm mwp}$, as provided by Section 3.1. Based on this, any agent i is required to communicate with the other slave ones in its neighbouring set.

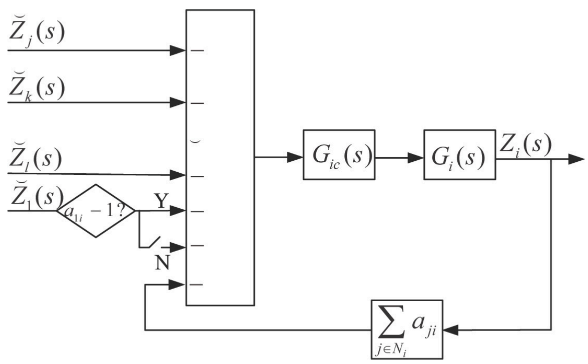
4 Simulation

In this section, we present the simulation studies of the proposed control scheme on a group of 15 heterogeneous agents. Under the formation control strategy in Section 3, each agent should keep a desired relative position with the leader agent 1. The position state for the leader agent 1 is designed as

$$z_0(t) = \begin{cases} (0.04t, 10\sin(0.02t))^T, & t \in [0, 50\pi) \\ (0.04t, 0)^T, & t \in (50\pi, 300] \end{cases}.$$

Initially, the minimally secure connection degree is 30, i.e., agents can be connected if the connection degree between agents is bigger than 30. Correspondingly, the connection relationship for all agents can form a secure connection degree graph as shown in Figure 7(a), wherein the length of each edge denotes the connection degree. By using the optimisation scheme in Section 3.1, we can obtain the min-weighted persistent graph for secure connection degree as shown in Figure 7(b).

Figure 7 The refreshed feedback control strategy of the formation



To verify the formation protocol proposed in Section 3.2, we use the following parameters as provided in Table 1, and these parameters meet the requirement in Theorem 1. The desired formation shape is a rectangular with 2 m long and 5 m wide, and the formation process is given by Figure 8(a). To show more clearly, the formation errors in X-axis and Y-axis are given by Figure 8(b) and (c), respectively. Obviously, the formation errors approximately converge to zero and the systems are stable. Noticing that the connection relationship of agents in this paper can reduce the communication complexity during the formation process, we verify this judgement through comparison experiment. For the research on 'neighbour rule', we choose Aguilar and Gharesifard (2015) as the representative. Besides, Chen et al. (2015) is chose as the representative of the research on 'persistent formation'. We make the following rules: the shorter of the edge, the safer of the connection; if there is no edge between two nodes, the secure connection degree is defined as 0. The result is provided by Figure 8(d). The number of communication edges and the sum of length for edges are the smallest with respect to Aguilar and Gharesifard (2015) and Chen et al. (2015) i.e., the communication complexity is less than the result in Aguilar and Gharesifard (2015) and Chen et al. (2015), and the connection between agents is safer than the result in Aguilar and Gharesifard (2015) and Chen et al. (2015).

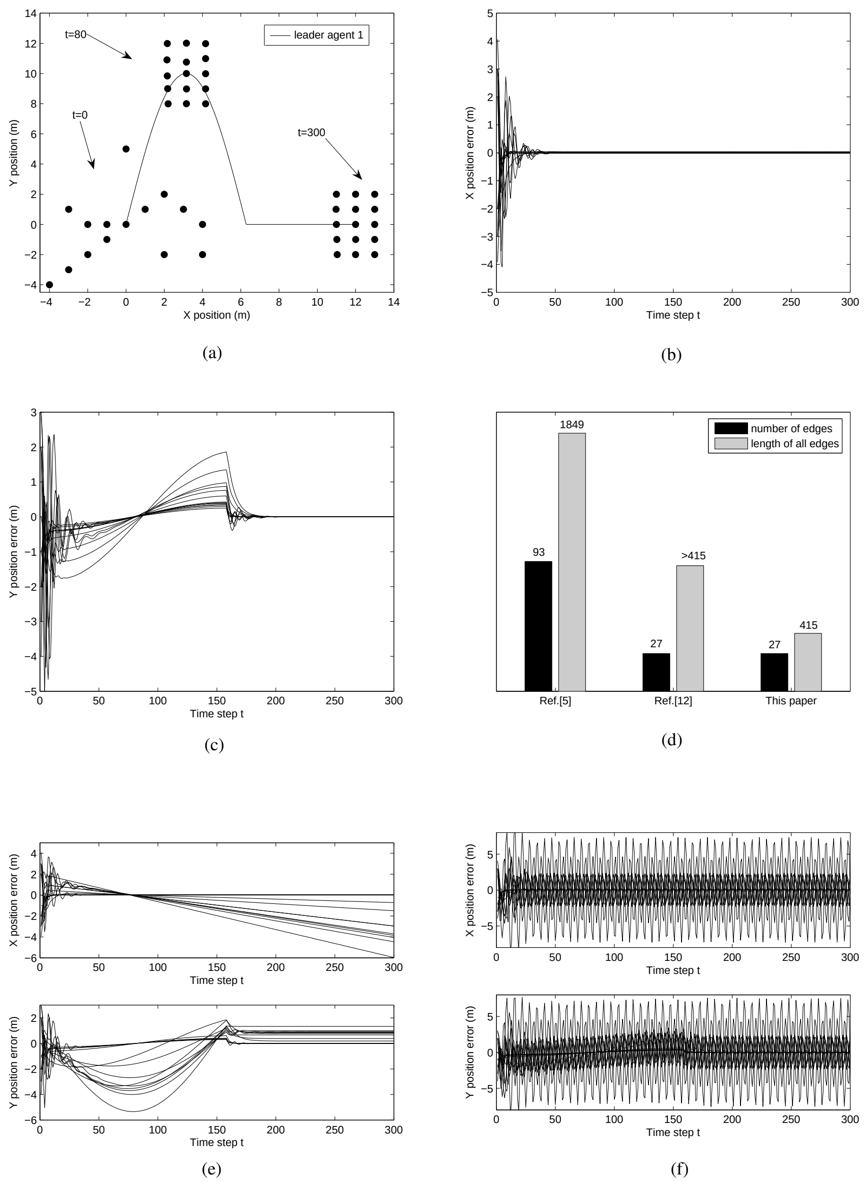
Table 1 Parameters of the formation control

i	$G_{ic}(s)$	$G_i(s)$	d_i	$z_i(0)$
2	$\frac{s^2+2s+1}{s^2+s}$	$\frac{1}{s^2+2s+1}$	$(1,0)^T$	$(-1,-1)^T$
3	$\frac{1}{s^2+2s+4}$	$\frac{1}{s}$	$(1,-1)^T$	$(-3,-3)^T$
4	$\frac{1}{s^2+2s+3}$	$\frac{1}{s}$	$(0,-1)^T$	$(4,-2)^T$
5	$\frac{s^2+2s+1}{s^2+s}$	$\frac{1}{s^2+2s+1}$	$(1,-2)^T$	$(4,0)^T$
6	$\frac{s+1}{s^2+2s+4}$	$\frac{1}{s^2}$	$(0,-2)^T$	$(3,1)^T$
7	$\frac{s+1}{s^2+2s+8}$	$\frac{1}{s^2}$	$(0,2)^T$	$(0,5)^T$
8	$\frac{1}{s^2+2s+3}$	$\frac{1}{s}$	$(1,2)^T$	$(-1,0)^T$
9	$\frac{1}{s^2+2s+4}$	$\frac{1}{s(s^2+s+3)}$	$(0,1)^T$	$(-3,1)^T$
10	$\frac{1}{s^2+2s+3}$	$\frac{1}{s}$	$(1,1)^T$	$(-2,0)^T$
11	$\frac{s^2+2s+1}{s^2+s}$	$\frac{1}{s^2+2s+1}$	$(-1,2)^T$	$(-2,-2)^T$
12	$\frac{s^2+2s+1}{s^2+s}$	$\frac{1}{s^2+2s+1}$	$(-1,1)^T$	$(2,-2)^T$
13	$\frac{1}{s^2+2s+3}$	$\frac{1}{s}$	$(-1,0)^T$	$(2,2)^T$
14	$\frac{s+1}{s^2+2s+4}$	$\frac{1}{s^2}$	$(-1,-1)^T$	$(1,1)^T$
15	$\frac{s^2+2s+1}{s^2+s}$	$\frac{1}{s^2+2s+1}$	$(-1,-2)^T$	$(-4,-4)^T$

To illustrate more clearly, we can test Theorem 1 by adjusting the parameters as follows: $G_{5c}(s) = \frac{s^2+2s+1}{s^2+s+4}, G_5(s) = \frac{1}{s^2+2s+1}$ and the others are fixed. Obviously, all the elements in the first column of the stability table have the same sign, i.e., the systems are stable, but

$G_{5c}(s)G_5(s)$ do not contain the factor $1/s$. So each agent cannot keep a desired relative position with the leader which is verified by Figure 8(e).

Figure 8 The simulation results during the formation process: (a) the formation process for agents; (b) the formation position error in X-axis; (c) the formation position error in Y-axis; (d) comparison on communication complexity and connection safety; (e) the formation error when $G_{5c}(s)G_5(s)$ do not contain the factor $1/s$ and (f) the formation error when the first column of the stability table does not have the same sign



In the same way, the parameters can be adjusted as follows: $G_{5c}(s) = \frac{s^2+2s+1}{s^2+s}, G_5(s) = \frac{1}{s^2+2s-1}$ and the others are fixed. Obviously, all the elements in the first column of the stability table do not have the same sign, i.e., the systems are unstable. Figure 8(f) can verify this argument.

5 Conclusion

We investigate the pursuit formation problem of heterogeneous multi-agent systems under min-weighted persistent graph. A topology optimisation scheme based on minweighted persistent graph is proposed to reduce the communication complexity of keeping connectivity for agents. According to transform function method, a formation control protocol is proposed. The convergence condition of the pursuit formation task for heterogeneous multi-agent systems can be given by using Mason's rule and Rouths stability criteria. It is shown that the number of communication edges and the sum of length for edges can be reduced. Simulation results illustrate the effectiveness of our designs.

Acknowledgements

The work was partially supported by NSF of China under 61503320, by China Postdoctoral Science Foundation Funded Project under 2015M570235, by Youth Foundation of Hebei Educational Committee under QN2015187, by Postdoctoral Science Foundation Funded Project of Hebei Province under B2015003018, by the Open Project Program of Key Laboratory of System Control and Information Processing, Ministry of Education under Scip201501, and by Yanshan University under B832 and 14LGA010.

References

Aguilar, C.O. and Gharesifard, B. (2015) 'Graph controllability classes for the Laplacian leader-follower dynamics', *IEEE Transactions on Automatic Control*, Vol. 60, No. 6, pp.1611–1623.

Chen, C.L., Yan, J., Ning, L., Wang, Y.Y., Yang, X. and Guan, X.P. (2015) 'Ubiquitous monitoring for industrial cyber-physical systems over relay assisted wireless sensor networks', *IEEE Transactions on Emerging Topics in Computing*, Vol. 3, No. 3, pp.352–362.

Ding, W., Yan, G.F. and Lin, Z.Y. (2010) 'Collective motions and formations under pursuit strategies on directed acyclic graphs', *Automatica*, Vol. 46, No. 1, pp.174–181.

Gupta, P. and Kumar, P.R. (2000) 'The capacity of wireless networks', *IEEE Transactions on Information Theory*, Vol. 46, No. 2, pp.388–404.

Hendrickx, J.M., Anderson, B, Delvenne, J.C. and Blondel, V.D. (2007) 'Directed graphs for the analysis of rigidity and persistence in autonomous agent systems', *International Journal of Robust and Nonlinear Control*, Vol. 17, Nos. 10–11, pp.960–981.

Hu, Y.J., Gao, Y. and An, B. (2015) 'Multiagent reinforcement learning with unshared value functions', *IEEE Transactions on Cybernetics*, Vol. 45, No. 4, pp.647–662.

Jia, J., Chen, J., Chang, G.R. and Wen, Y.Y. (2009) 'Optimal sensor deployment based on secure connection in mobile sensor network', *Journal of Software*, Vol. 20, No. 4, pp.1038–1047.

Krishnamurthi, V. (1972) 'Correlation between Routh's stability criterion and relative stability of linear systems', *IEEE Transactions on Automatic Control*, Vol. 17, No. 1, pp.144–145.

Luo, X.Y., Li, S.B. and Guan, X.P. (2009) 'Automatic generation of minweighted persistent formations', *Chinese Physics B*, Vol. 18, No. 8, pp.3104–3114.

Mason, S.J. (1956) 'Feedback theory: Further properties of signal flow graphs', *JProceedings of the IRE*, Vol. 44, No. 7, pp.920–926.

Olfati-Saber, R. (2006) 'Flocking for multi-agent dynamic systems: algorithms and theory', *IEEE Transactions on Automatic Control*, Vol. 51, No. 3, pp.401–420.

Reynolds, C.W. (1987) 'Flocks, herds, and schools: a distributed behavioral model', *Computer Graphics*, Vol. 21, No. 4, pp.25–34.

Semsar-Kazerooni, E. and Khorasani, K. (2009) 'Multi-agent team cooperation: A game theory approach', *Automatica*, Vol. 45, No. 10, pp.2205–2213.

Tay, T.S. and Whitele, W. (1985) 'Generating isostatic frameworks', *Structural Topology*, Vol. 11, pp.21–69.

Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I. and Shochet, O. (1995) 'Novel type of phase transition in a system of self-driven particles', *Physical Review Letters*, Vol. 75, No. 6, pp.1226–1229.

Yan, J., Chen, C.L., Luo, X.Y., Liang, H., Yang, X. and Guan, X.P. (2015) 'Topology optimization based distributed estimation in relay assisted wireless sensor networks', *IET Control Theory and Applications*, Vol. 8, No. 18, pp.2219–2229.

Yarvis, M., Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y. and Singh, S. (2005) 'Exploiting heterogeneity in sensor networks', *Processings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, USA, pp.878–890.

Zhang, P.P., De, Q.M. and Cai, X.Y. (2015) 'Three-dimensional dynamic formation control of multi-agent systems using rigid graphs', *Journal of Dynamic Systems, Measurement and Control*, Vol. 137, No. 11, pp.1–7.