Acta Automatica Sinica
Vol. 39, No. 6 June, 2013

Hierarchical Distributed Control Design for Multi-agent Systems Using Approximate Simulation

TANG Yu-Tao[1] HONG Yi-Guang[1]

Abstract In this paper, we consider a hierarchical control design for multi-agent systems based on approximate simulation. To reduce complexity, we first construct a simple abstract system to guide the agents, then we discuss the simulation relations between the abstract system and multiple agents. With the help of this abstract system, distributed hierarchical control is proposed to complete a coordination task. By virtue of a common Lyapunov function, we analyze the collective behaviors with switching multi-agent topology in light of simulation functions.

Key words Multi-agent systems, distributed hierarchical control, simulation function, switching topology

Citation Tang Yu-Tao, Hong Yi-Guang. Hierarchical distributed control design for multi-agent systems using approximate simulation. *Acta Automatica Sinica*, 2013, **39**(6): 868−874

Coordination of multi-agent systems has attracted more and more attention in recent years. A fundamental challenge in multi-agent systems is to cope with distributed design and large-scale information processing. However, the complexity in the analysis and synthesis is huge because of the large sizes and hierarchical/hybrid structures of multi-agent systems. In the coordination problem of multi-agent networks, the leader-follower framework has been widely used, where one or more (maybe virtual) leaders are selected to lead the whole group of agents to achieve the collective task[1−5].

Hierarchical control makes a great effort to deal with complex systems. For complexity reduction and effective design, an important task is to give simplified models as abstract systems for given practical systems or reduce the model complexity of existing models[6−7]. Different methods have been developed, including inclusion principle[8], aggregation[9], and approximate simulation[10]. In fact, the hierarchical control based on approximate simulation becomes a hot topic in these years. The hierarchical approach features the following steps: a simple abstraction is constructed for the complex plant (usually called concrete system); the abstract controller for the abstract system is designed simply; then the control law for the concrete system can be synthesized in hierarchical way by embedding an abstract controller, for example, the proposed hierarchical control method for linear systems with simulation function[10]. However, most of the existing results are for single systems. Note that approximate simulation can simplify some problems, but it may lose some accuracy. Therefore, tradeoff is needed between abstraction and accuracy.

Multi-agent systems are certainly complex, and the hierarchical control of multi-agent systems has been discussed to simplify the design complexity. For example, to simplify the change of formation pattern, formation hierarchy was employed and an effective design was given for multiple vehicles in [5], and a hierarchical formation control based on pseudo-rigid transformation was proposed for multi-robot system in [11]. However, those ideas highly rely on the communication topology among the robots and (virtual) leaders.

The objective of this paper is to investigate a new hierarchical control for multi-agent systems. Here we try to extend the centralized abstraction-based hierarchical controls to distributed ones when the network topology keeps switching. Since the agents are homogeneous, we only construct one abstract model for the agents. Then we treat the abstract model as a virtual leader and use the leader-follower framework to guide the group. In other words, to reduce the complexity to control agents directly, we build a simplified model of agents as a virtual leader of the multi-agent system, and we give a simple design for the abstract system. Then we propose the distributed control with two parts: one is the local control for coordination, while the other is from the virtual leader. The abstract model is set up in a remote control center, whose control is sent out as a command to each agent in the distance. In our problem, to solve the problems resulting from switching topology, a common Lyapunov function is constructed to guarantee the control design.

The paper is organized as follows. Section 1 provides preliminaries and the formulation of distributed hierarchical control. Section 2 carries out the discussion on approximate simulation and the abstract system. Section 3 gives the design of distributed hierarchical control of multi-agent systems with switching topology and then shows a numerical result afterwards. Finally, Section 4 presents the concluding remarks.

1 Preliminaries and formulation

First of all, we introduce some basic concepts and notations in graph theory (referring to [12] for details). A directed graph (or digraph) is usually denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \cdots, n\}$ is the set of nodes and $\mathcal{E}$ is the set of arcs, each element of which is an ordered pair of distinct nodes in $\mathcal{N}$. (i, j) denotes an arc leaving node i and entering node j. A walk in digraph $\mathcal{G}$ is an alternating sequence $i_1 e_1 i_2 e_2 \cdots e_{k-1} i_k$ of nodes i_l and arcs $e_m = (i_m, i_{m+1}) \in \mathcal{E}$ for $l = 1, 2, \cdots, k$. If there exists a walk from node i to node j then node j is said to be reachable from node i. In particular, each node is thought to be reachable by itself. A node that is reachable from any node of $\mathcal{G}$ is called a globally reachable node of $\mathcal{G}$.

I_n is an $n \times n$ identity matrix. A symmetric matrix $P > 0$ means that it is positive definite. For any vector $\boldsymbol{x}$, $\boldsymbol{x}^{\mathrm{T}}$ denotes its transpose. $\sigma(A)$ denotes the spectrum of matrix A. diag$\{\cdot\}$ denotes a diagonal matrix with a list of diagonal elements[13].

Here we consider a system consisting of m agents and a leader (denoted as node 0). In the sequel, the state of agent i is denoted by $\boldsymbol{x}_i$ for $i = 1, \cdots, m$. If m agents are regarded as the vertices in $\mathcal{V}$, the relationships between n agents can be conveniently described by a simple graph $\mathcal{G}$, where (i, j) defines one of the graph′s arcs if agent j is reachable from agent i. $N_i(t)$ denotes the set of labels

Manuscript received May 8, 2012; accepted September 18, 2012
Supported by National Natural Science Foundation of China (61174071)
Recommended by Associate Editor XIA Yuan-Qing
1. Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

of those agents, from which agent i ($i = 1, \cdots, m$) can be reachable at time t. The weighted adjacency matrix of $\mathcal{G}$ is denoted by $A_m = [a_{ij}] \in \mathbf{R}^{m\times m}$, where $a_{ii} = 0$ and $a_{ij} \geq 0$ ($a_{ij} > 0$ if there is a walk from agent i to agent j). Its degree matrix $D_m = \mathrm{diag}\{d_1, \cdots, d_m\} \in \mathbf{R}^{m\times m}$ is a diagonal matrix, where diagonal elements $d_i = \sum_{j=1}^{m} a_{ij}$ for $i = 1, \cdots, m$. Then the Laplacian of the graph is defined as $L = D_m - A_m$.

In what follows, we mainly concern another graph $\bar{\mathcal{G}}$ associated with the system consisting of m agents and the leader. In fact, $\bar{\mathcal{G}}$ contains m agents (related to graph $\mathcal{G}$) and the leader with directed edges from some agents to the leader by the connection weights $a_{i0} > 0$ (if agent i is connected to the leader, or equivalently, agent i is a neighbor of the leader, denoted by $i \in N_0$).

The following lemma shows an important property of Laplacian L[12] for a relationship between the graph connectivity and its Lapalcian.

Lemma 1. Laplacian L of $\mathcal{G}$ has a zero eigenvalue with eigenvector $\mathbf{1} = (1, \cdots, 1)^{\mathrm{T}} \in \mathbf{R}^m$.

Consider a set of m agents, whose dynamics is identified as follows:

$$\dot{\boldsymbol{x}}_i = A\boldsymbol{x}_i + B\boldsymbol{u}_i, \quad \boldsymbol{u}_i \in \mathbf{R}^k,\ \boldsymbol{x}_i \in \mathbf{R}^n \tag{1}$$

where $(A,\ B)$ is stabilizable, $\boldsymbol{x}_i, \boldsymbol{u}_i$ are the states and controls of agent $i,\ i = 1, \cdots, m$.

All the agents will complete a task given in a task set, denoted as $\mathcal{T}$. Here, for simplicity, we assume the desired trajectory is $\boldsymbol{w}_d(t) \in \mathbf{R}^q$, which can be obtained by motion planning. Our task here is to make $\lim_{t\to\infty}(\boldsymbol{y}_i - \boldsymbol{w}_d) = 0$ with $\boldsymbol{y}_i = C\boldsymbol{x}_i \in \mathbf{R}^q$. Without loss of generality, we assume that C is of full row rank.

Usually, it is not easy to control all the agents to complete a task, especially when the agent dynamics are complex or high dimensional. Certainly, we can use abstraction techniques to make an abstract system for each agent in order to simplify the practical design as people have done for single systems. However, since the agents are homogeneous, we take one simple abstract model for all of them for simplicity, and moreover, we add the inter-agent communication to improve the convergence (to trade off the requirement of information of the abstract system sometimes). In this case, we try hierarchical control to simply the control design as follows:

1) Take one simple abstract model for all the agents, described by

$$\begin{cases}\dot{\boldsymbol{w}} = S\boldsymbol{w} + D\boldsymbol{v}, & \boldsymbol{w} \in \mathbf{R}^l\\ \boldsymbol{y}_0 = F\boldsymbol{w} \in \mathbf{R}^q\end{cases} \tag{2}$$

which can be viewed as a virtual leader (node 0) to influence or guide the multi-agent network by its output $\boldsymbol{y}_0$. Note that (2) is artificial, and S, D, F can be selected for our task purpose. The dimension of $\boldsymbol{w}$ is much lower than that of $\boldsymbol{x}_i$ so that we can have a simpler controller for (2).

2) Control $\boldsymbol{v}$ of (2) can be easily obtained. Then we only send $\boldsymbol{v}$, instead of the task trajectory, to all the agents, and also send $\boldsymbol{y}_0$ to one or more agents which are treated as the neighbors of the leader. Then the problem becomes a leader-follower problem.

Although leader-follower framework here, the hierarchical formulation is different from the conventional leader-follower problems, where the leader dynamics is given. Here, the abstract model (viewed as leader) can be built on our own, which provides us more degrees of freedom to optimize the controllers′ performance.

A_0 is an $m \times m$ diagonal matrix whose ith diagonal element is a_{i0} (if agent i is connected to the leader or node 0, namely $i \in N_0$). With L still representing the Laplacian matrix of digraph $\mathcal{G}$, we define a matrix $H = L + A_0$ to describe the connectivity of the whole graph $\bar{\mathcal{G}}$. Obviously, $H\mathbf{1} = A_0\mathbf{1}$ with the help of Lemma 1. The next lemma about H was borrowed from [14].

Lemma 2[14]**.** For any undirected graph $\mathcal{G}$, H is positive definite if and only if node 0 of $\bar{\mathcal{G}}$ is globally reachable.

For the multi-agent system under consideration, the relationships between neighbors (and the interconnection topology) change over time. Suppose that there is an infinite sequence of bounded, non-overlapping, contiguous time-intervals $[t_i, t_{i+1}),\ i = 0, 1, \cdots$, starting at $t_0 = 0$.

Denote $\mathcal{S} = \{\bar{\mathcal{G}}_1, \bar{\mathcal{G}}_2, \cdots, \bar{\mathcal{G}}_\mu\}$ as a set of the graphs with all possible topologies, which includes all possible interconnection graphs (involving n agents and a leader), and denote $\mathcal{P} = \{1, 2, \cdots, \mu\}$ as its index set.

To describe the variable interconnection topology, we define a switching signal $\sigma : [0, \infty) \to \mathcal{P}$, which is piecewise-constant. Therefore, N_i and the connection weight a_{ij} ($i = 1, \cdots, n, j = 1, \cdots, n$) are time-varying, and moreover, Laplacian L_p ($p \in \mathcal{P}$) associated with the switching interconnection graph is also time-varying (switched at $t_i,\ i = 0, 1, \cdots$), though it is a time-invariant matrix in any interval $[t_i, t_{i+1})$. To avoid infinite-switching within a finite time interval and related non-smooth description, we assume that there is a constant $\tau_0 > 0$, often called dwell time, with $t_{i+1} - t_i \geq \tau_0,\ \forall i$.

To study the hierarchical control of switching multi-agent systems with approximate simulations, we give a standard assumption for coordination.

Assumption 1. Node 0 (abstract system) is always globally reachable in $\bar{\mathcal{G}}_{\sigma(t)}$.

This assumption is necessary in the coordination.

In this paper, the topology is time-varying with a given dwell time, but keeps connected. Then the graph of the whole system is time-varying, denoted as $\bar{\mathcal{G}}_{\sigma(t)} : [0, \infty) \to \mathcal{S}$. Consequently, the neighbor sets $\mathcal{N}_j^{\sigma(t)}$ ($j = 1, \cdots, m$) and $\mathcal{N}_0^{\sigma(t)}$, along with the connection weights a_{ij} ($i, j = 1, \cdots, m$) and a_{i0} ($i = 1, \cdots, N$) are time-varying. In other words, Laplacian $L_{\sigma(t)}$ associated with the switching interconnection graph $\mathcal{G}_{0,\sigma(t)}$ and $A_{0,\sigma(t)}$ associated with the connections between agents and the leader are time-varying (switched at $t_i,\ i = 0, 1, \cdots$). Obviously, $H_{\sigma(t)} = L_{\sigma(t)} + A_{0,\sigma(t)}$ is also time-varying.

However, $\bar{\mathcal{G}}_p$ ($p \in \mathcal{P}$) is a constant graph during the time interval $[t_i, t_{i+1})$, and then $L_{\sigma(t)} = L_p$, $H_{\sigma(t)} = H_p$ are constant matrices when $t \in [t_i, t_{i+1})$. According to Lemma 2 and Assumption 1, all eigenvalues of the matrices H_p ($p \in \mathcal{P}$) are positive, and then we denote $\bar{\lambda}$ as the minimal eigenvalue of these matrices H_p ($p \in \mathcal{P}$).

Since the set $\mathcal{P}$ is finite, $\bar{\lambda} > 0$.

For the considered agent system, we give an assumption.

Assumption 2. (A, B) is stabilizable.

The tracking error for agent i is defined as

$$\boldsymbol{e}_i = C\boldsymbol{x}_i - \boldsymbol{y}_0 \in \mathbf{R}^q, \quad i = 1, \cdots, m \tag{3}$$

In our formulation, the information of $\boldsymbol{y}_0$ (and therefore, e_i) may not be available for agent i in its design. Instead, to achieve the aim, distributed algorithms are proposed by considering relative measurements as follows:

$$\boldsymbol{z}_i = \sum_{j\in N_i} a_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j) + \sum_{i\in N_0} a_{i0}(\boldsymbol{x}_i - C^{+}\boldsymbol{y}_0) \tag{4}$$

for $i = 1, \cdots, m$, where C^+ is a generalized inverse of C satisfying $CC^+ = I_q$. Here C^+ assures that we can obtain $\boldsymbol{e}_i = C\boldsymbol{x}_i - \boldsymbol{y}_0$ via $\boldsymbol{z}_i$ when $\boldsymbol{y}_0$ is obtained, as was used in distributed output regulation problem[15]. With this relative information, control input u will be designed subject to abstract output variable $\boldsymbol{y}_0$.

Here we focus on the distributed control problem with the information exchanged between the followers once they are connected. In other words, the only available information in the distributed control for agent i includes $(\boldsymbol{x}_i - \boldsymbol{x}_j)$ $(j \in N_i)$ if it is not connected to the leader. Furthermore, if agent i is connected to the leader, then the output of the leader or $(\boldsymbol{x}_i - C^+\boldsymbol{y}_0)$ is also available.

The distributed control laws can be expressed in the following form:

$$\boldsymbol{u}_i = R\boldsymbol{v} + K_z\boldsymbol{z}_i, \quad i = 1, \cdots, m \tag{5}$$

where command $\boldsymbol{v}$ has been sent to and drive all the agents.

Note that the input to agent i includes two parts: a global command $\boldsymbol{v}$ to complete the collective task, and a relative information $\boldsymbol{z}_i$ for coordination between agents (and leaders); that is, agent i can be guided hierarchically by the abstract model via $\boldsymbol{v}$ and its output $\boldsymbol{y}_0$ (determined by the initial conditions); particularly, if we take $R = 0$, this control law comes out to be a pure local rule.

In practice, to complete a task, the input command $\boldsymbol{v}$ and initial conditions of the constructed abstract model will be planned or designed according to the task set $\mathcal{T}$. In other words, $\boldsymbol{v}$ cannot be designed independently of a given task. Here we mainly focus on the distributed design of the interface between two levels for multi-agent systems, and do not pay much attention on how to construct $\boldsymbol{v}$ based on a given task. In fact, once we obtain a simple abstract system, there are many methods to construct $\boldsymbol{v}$ for a given task trajectory.

Denote

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_m \end{bmatrix}, \ \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_m \end{bmatrix}, \ \boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_m \end{bmatrix}$$

Then the multi-agent system (1) and (2) becomes

$$\begin{cases} \dot{\boldsymbol{x}} = (I_m \otimes A)\boldsymbol{x} + (I_m \otimes B)\boldsymbol{u} \\ \dot{\boldsymbol{w}} = S\boldsymbol{w} + D\boldsymbol{v} \\ \boldsymbol{e} = (I_m \otimes C)\boldsymbol{x} - (\boldsymbol{1} \otimes F)\boldsymbol{w} \end{cases} \tag{6}$$

with $\boldsymbol{1} = (1 \cdots 1)^{\mathrm{T}} \in \mathbf{R}^N$.

Then the closed-loop system can be rewritten as

$$\begin{cases} \dot{\boldsymbol{x}} = A_c^\sigma\boldsymbol{x} + R_c\boldsymbol{v} + B_c^\sigma\boldsymbol{w} \\ \dot{\boldsymbol{w}} = S\boldsymbol{w} + D\boldsymbol{v} \\ \boldsymbol{e} = (I_m \otimes C)\boldsymbol{x} - \boldsymbol{1} \otimes \boldsymbol{y}_0 \end{cases} \tag{7}$$

with

$$A_c^\sigma = I_m \otimes A + H_\sigma \otimes (BK_z), \ R_c = \boldsymbol{1} \otimes BR$$

$$B_c^\sigma = -(H_\sigma\boldsymbol{1}) \otimes (BK_zC^+F)$$

by $H\boldsymbol{1} = A_0\boldsymbol{1}$. The superscript σ describes the switching resulting from variable interaction topologies. When there is no confusion, we may drop the superscript.

2 Simulation and abstract system

In this section, we discuss approximate simulation and related abstract system.

A function $\gamma : \mathbf{R}_+ \to \mathbf{R}_+$ is said to be a class-K function if it is strictly increasing and continuous with $\gamma(0) = 0$.

Simulation functions have been introduced and discussed in many publications including [7, 10, 16] to provide a measurement of approximation simulation. Roughly speaking, given two systems S and S', the (approximate) simulation relation of S' by S is a relation over their state spaces explaining how a state trajectory of S' can be transformed into a state trajectory of S with identical (approximate) output trajectories. A simulation function is to make sure the distance between the associated output trajectories remains within computable bounds under the transformation.

A simulation function was defined in [7] and we redefine a simulation function in this paper for distributed hierarchical control as follows:

Definition 1. Function $V(\boldsymbol{\xi}, \boldsymbol{w})$ is called a simulation function of a system

$$\dot{\boldsymbol{\xi}} = \bar{A}\boldsymbol{\xi} + \bar{B}\boldsymbol{u}, \quad \boldsymbol{\zeta} = \bar{C}\boldsymbol{\xi} \tag{8}$$

and its abstract model in the form of (2) if there are control $\boldsymbol{u}$ and class-K functions α and γ such that

$$V(\boldsymbol{\xi}, \boldsymbol{w}) \geq ||\boldsymbol{\zeta} - \boldsymbol{y}_0|| \tag{9}$$

and for any $\boldsymbol{v}$ with $\gamma(||\boldsymbol{v}||) \leq V(\boldsymbol{\xi}, \boldsymbol{w})$,

$$\frac{\mathrm{d}V}{\mathrm{d}t} \leq -\alpha(V(\boldsymbol{\xi}, \boldsymbol{w})) \tag{10}$$

Remark 1. This definition is slightly different from that given in [7]. In (10), we constrain an upper bound for $\frac{\mathrm{d}V}{\mathrm{d}t}$, and we will find that this $\mathcal{K}$ function allows us to obtain a more precise bound for simulation relationship between the output trajectories of two systems.

It is known from [7] that, if there is a simulation function of system S' by S and $\alpha(r) = 0$, then

$$||\boldsymbol{\zeta}(t) - \boldsymbol{y}_0(t)|| \leq \max\{V(\boldsymbol{w}(0), \boldsymbol{\xi}(0)), \gamma(||\boldsymbol{v}||_\infty)\} \tag{11}$$

With our modified definition, we can get a similar estimation via the comparison theorem:

$$||\boldsymbol{\zeta}(t) - \boldsymbol{y}_0(t)|| \leq \beta(V(\boldsymbol{w}(0), \boldsymbol{\xi}(0)), t) + \gamma(||\boldsymbol{v}||_\infty) \tag{12}$$

which is true when $\boldsymbol{v} = 0$ ($\boldsymbol{\zeta}$ can asymptotically track the output trajectory $\boldsymbol{y}_0$ of system (2)).

In this paper, the distributed hierarchical control design for system (1) is achieved under (5) if there is a simulation function for system (7). In fact, the above formulation can be viewed as an extension of the hierarchical control of single system (e.g., [7]) to networked systems. To reduce the complexity introduced by the dynamics of agents, we first construct a simple abstract system, and then use this abstract model to guide all the homogeneous agents whose dynamics may be complex to complete a task through the global command $\boldsymbol{v}$.

Remark 2. The difference between the hierarchical control and distributed output regulation (DOR)[15] is as follows: in DOR, the leader is given in advance without any input and tracking can be achieved asymptotically; in the hierarchical control defined here, the leader is artificial to achieve some given task and therefore it has its own control input (or command); the tracking may not be accurate (probably with bounds described by simulation functions).

Remark 3. In our abstract model, the initial conditions can also be selected. Even if taking $\boldsymbol{v} = 0$, this hierarchical method can still drive all the agents to complete some simple tasks like consensus via relative information by the proposed distributed control laws.

Before discussing the hierarchical control, we need to discuss how to construct the abstract system by determining S, D, and F. Moreover, D can be selected to optimize the controller's performance. Different from the conventional leader-follower problems, the abstract system (along with its control input $\boldsymbol{v}$) as the leader is not given in advance but constructed based on the given task.

To solve the distributed hierarchical control design, we give another assumption as follows.

Assumption 3. For given agent dynamics (1), there are matrices X and C^+ (with $CC^+ = I_q$) such that

$$XS = AX, \quad C^+F = X \tag{13}$$

To make Assumption 3 hold, we need to discuss the suitable matrices S and F. This assumption is not so restrictive as it appears because S, F, X can be freely selected. The freedom to construct different abstract systems may help us in the distributed design to achieve collective tasks.

Remark 4. Assumption 3 can be relaxed as: for given agent dynamics (1), there are matrices X and C^+ such that

$$XS = AX, \quad F = CX, \quad C^+F = X \tag{14}$$

Namely, C^+ needs not be a generalized inverse of C (with $CC^+ = I_q$). In fact, with (14), we can always find a C^+ such that Assumption 3 holds.

Lemma 3. $\mathrm{im}(X)$ is an A-invariant subspace.

Since A-invariant subspace can be determined using the Jordan normal form of A, possible X and S can be selected. Assumption 3 also gives $F = CX$. Moreover, we have the following lemma.

Lemma 4. Given X, let $F = CX$. Then there exists a C^+ such that $C^+F = X$ if and only if

$$\ker(X^{\mathrm{T}}) + \mathrm{im}(C^{\mathrm{T}}) = \mathbf{R}^n$$

A simple necessary condition is $\mathrm{rank}(X) \le \mathrm{rank}(C)$.

Proof. Suppose that there exists matrix C^+ such $C^+F = X$. Then we have $C^+CX = X$ and $(I_n - C^+C)X = 0$, or $X^{\mathrm{T}}(I_n - C^{\mathrm{T}}{C^+}^{\mathrm{T}}) = 0$. For all $\boldsymbol{x} \in \mathbf{R}^n$, we have $\boldsymbol{x} = C^{\mathrm{T}}{C^+}^{\mathrm{T}}\boldsymbol{x} + (I_n - C^{\mathrm{T}}{C^+}^{\mathrm{T}})\boldsymbol{x}$, since $(I_n - C^{\mathrm{T}}{C^+}^{\mathrm{T}})\boldsymbol{x} \in \ker(X^{\mathrm{T}})$, and $C^{\mathrm{T}}{C^+}^{\mathrm{T}}\boldsymbol{x} \in \mathrm{im}(C^{\mathrm{T}})$, we have $\ker(X^{\mathrm{T}}) + \mathrm{im}(C^{\mathrm{T}}) = \mathbf{R}^n$.

Conversely, let us assume that $\ker(X^{\mathrm{T}}) + \mathrm{im}(C^{\mathrm{T}}) = \mathbf{R}^n$. Then, there exists matrix M such that

$$\mathrm{im}(M) \subseteq \ker(X^{\mathrm{T}}), \text{ and } \mathrm{im}(M) \oplus \mathrm{im}(C^{\mathrm{T}}) = \mathbf{R}^n$$

Then, there exist two matrices C_1 and C_2 such that

$$I_n = MC_1 + C^{\mathrm{T}}C_2$$

It follows that $X^{\mathrm{T}} = X^{\mathrm{T}}(MC_1 + C^{\mathrm{T}}C_2) = X^{\mathrm{T}}C^{\mathrm{T}}C_2 = F^{\mathrm{T}}C_2$ because $\mathrm{im}(M) \subseteq \ker(X^{\mathrm{T}})$, or $C^+F = X$ if we take $C^+ = C_2^{\mathrm{T}}$.

The necessary condition is easy to verify by noting that $\dim(\ker(X^{\mathrm{T}})) + \dim(\mathrm{im}(C^{\mathrm{T}})) \ge n$, and $\dim(\ker(X^{\mathrm{T}})) = n - \mathrm{rank}(X)$, $\dim(\mathrm{im}(C^{\mathrm{T}})) = \mathrm{rank}(C)$. □

Combined with $CC^+ = I_q$, Lemmas 3 and 4 give us a method to find suitable S, X, D, R, C^+ and F. How to effectively obtain an abstract system for a distributed design is still a challenging problem. The selection of D and R is to make (S, D) be controllable and moreover, satisfy specific performance indexes.

3 Main results

In this section, we will show the simulation function for switching multi-agent systems. Clearly, the multi-agent coordination problem becomes much harder to solve when the considered multi-agent system keeps switching. Even if we can find a control law to stabilize the group of systems simultaneously, we may not stabilize the system that switches between the group of subsystems since the switched system may be unstable though all the subsystems are stable.

Common Lyapunov function (CLF) is known to be an effective tool to solve these problems. The following lemma is well known and provides a way to construct a common Lyapunov function.

Lemma 5[17]**.** Consider system $\dot{\boldsymbol{x}} = \tilde{A}\boldsymbol{x} + \tilde{B}\boldsymbol{u}$, $\boldsymbol{y} = \tilde{C}\boldsymbol{x}$. If $(\tilde{A}, \tilde{B})$ is stabilizable and matrix M is positive definite, then there is a unique solution P, in the class of symmetric positive semi-definite maps, to the Riccati equation:

$$\tilde{A}^{\mathrm{T}}P + P\tilde{A} - P\tilde{B}M^{-1}\tilde{B}^{\mathrm{T}}P + Q = 0$$

Furthermore, $\tilde{A} - \tilde{B}M^{-1}\tilde{B}^{\mathrm{T}}P$ is stable.

Based on Lemma 2, all the eigenvalues of H_σ are positive, denoted as λ_i^σ, $i = 1, \cdots, m$, for any $\sigma(t)$.

To deal with the problem, we first give a lemma for simultaneous design.

Lemma 6. With Assumption 2, there are a positive definite matrix P, matrix K_z, and positive constants $\bar{c}, \hat{c}$ such that

$$\begin{cases} P \ge \dfrac{C^{\mathrm{T}}C}{\bar{c}} \\ (A + \lambda_i^\sigma BK_z)^{\mathrm{T}}P + P(A + \lambda_i^\sigma BK_z) \le -\hat{c}P \end{cases} \tag{15}$$

where $\lambda_i^\sigma > 0$ for $i = 1, \cdots, m$ are the eigenvalues of H_σ.

Proof. Clearly, the minimum eigenvalue of H_σ is well-defined because the number of H_σ is finite, we denote it as $\bar{\lambda} > 0$. Because (A, B) is stabilizable, according to Lemma 5, we obtain that $A - BB^{\mathrm{T}}P$ is stable, where P is the unique solution of the Riccati equation $A^{\mathrm{T}}P + PA - PBB^{\mathrm{T}}P + I_n = 0$. Take $K_z = -\max\{1, 1/\bar{\lambda}\}B^{\mathrm{T}}P$, and then it is not hard to see that $A + B(\lambda_i^\sigma K_z)$ is stable under the state feedback $\boldsymbol{u}_i^0 = K_z\boldsymbol{z}_i$ because $A - \kappa BB^{\mathrm{T}}P$ is so for $\kappa \ge 1$. Note that there is a constant $\hat{c} > 0$ such that $I_n > \hat{c}P$. In this way, the second inequality can be proved.

Since P is positive definite, there is a sufficiently large constant $\bar{c}$ that makes the first inequality hold. □

In the following, we assume $\bar{c} = 1$ without loss of generality (otherwise, we can take $\bar{c}P$ as a new P). Obviously, $(I_m \otimes C)(\mathbf{1} \otimes X) = \mathbf{1} \otimes F$.

Here we have to find a simulation function based on the given distributed control (5).

The following theorem is our main result.

Theorem 1. Under Assumptions 1 $\sim$ 3, there is a hierarchical control law in the distributed form of (5) for the multi-agent systems when the interconnection topology keeps switching among $\mathcal{S} = \{\bar{\mathcal{G}}_1, \bar{\mathcal{G}}_2, \cdots, \bar{\mathcal{G}}_\mu\}$.

Proof. Take $P_c = I_m \otimes P$, where P is obtained from Lemma 6, then construct the common simulation function candidate, which is independent of the switching signal σ, as follows:

$$V(\boldsymbol{x}, \boldsymbol{w}) = \sqrt{(X_c\boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}}P_c(X_c\boldsymbol{w} - \boldsymbol{x})}, \quad X_c = \mathbf{1} \otimes X$$

where X is defined in Assumption 3. Obviously, condition (9) is satisfied. We only need to prove (10) for the switching system with distributed design.

As we mentioned, in the time interval $[t_i, t_{i+1})$ for any given $i = 0, 1, \cdots$, there is no switching and therefore, the interconnection graph associated with matrix H_p is unchanged with some $p \in \mathcal{P}$. Let us focus on the time-invariant system during $[t_i, t_{i+1})$. By Assumption 3 and Lemma 2, the constant matrix H_p is positive definite. Let us consider the derivative of V in the time interval $[t_i, t_{i+1})$, that is,

$$\dot{V}|_{(6)} = \frac{\partial V}{\partial \boldsymbol{w}}\dot{\boldsymbol{w}} + \frac{\partial V}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{(X_c\boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}}P_c[X_c(S\boldsymbol{w} + D\boldsymbol{v}) - (A_c^p\boldsymbol{x} + R_c\boldsymbol{v} + B_c^p\boldsymbol{w})]}{\sqrt{(X_c\boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}}P_c(X_c\boldsymbol{w} - \boldsymbol{x})}}$$

Due to Assumption 3, we have

$$X_c S = (I_m \otimes A) X_c, \quad \mathbf{1} \otimes C^{+} F = X_c$$

Then

$$\dot{V} = \frac{(X_c \boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}} P_c (I_m \otimes A + H_p \otimes (BK_z))(X_c \boldsymbol{w} - \boldsymbol{x})}{\sqrt{(X_c \boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}} P_c (X_c \boldsymbol{w} - \boldsymbol{x})}} + \frac{(X_c \boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}} P_c (X_c D - R_c) \boldsymbol{v}}{\sqrt{(X_c \boldsymbol{w} - \boldsymbol{x})^{\mathrm{T}} P_c (X_c \boldsymbol{w} - \boldsymbol{x})}}$$

From Lemma 6,

$$[I_m \otimes A + H_p \otimes (BK_z)]^{\mathrm{T}} P_c + P_c [I_m \otimes A + H_p \otimes (BK_z)] \le -\hat{c} P_c$$

Moreover, because $||X_c D - R_c|| \le ||XD - BR||$ and $||\sqrt{P_c}|| = ||\sqrt{P}||$,

$$\dot{V} \le -\frac{\hat{c}}{2} V + ||\sqrt{P}|| \cdot ||XD - BR|| \cdot ||\boldsymbol{v}||$$

when $t \in [t_i, t_{i+1})$. Therefore, there is a class-K function γ such that

$$\dot{V} \le -\frac{\hat{c}}{2} V + \gamma(||\boldsymbol{v}||), \quad \forall t \ge 0$$

Thus, the conclusion follows.

Moreover, when $\boldsymbol{v} \equiv \mathbf{0}$, $\lim_{t\to\infty} V(t) = 0$, and then

$$\lim_{t\to\infty} [X_c \boldsymbol{w}(t) - \boldsymbol{x}(t)] = \mathbf{0}$$

which implies the consensus of $\boldsymbol{x}_i$, $i = 1, \cdots, m$. □

Remark 5. The selection of D and R can be made to minimize $||XD - BR||$ with (S, D) being controllable.

In this case with $R = 0$, the result can also be obtained, but the simulation errors will be larger than the case with suitably selected R.

From Theorem 1, we have

$$||\boldsymbol{e}_i|| \le \max\{V((\boldsymbol{w}(0), \boldsymbol{\xi}(0)), t), \gamma(||\boldsymbol{v}||_\infty)\}$$

It is clear that a smaller $||\boldsymbol{v}||_\infty$ (which means a slower changing command) yields smaller errors.

Furthermore, from the above proof, we can obtain the following corollary.

Corollary 1. Under Assumptions 1 ∼ 3, all agents achieve consensus with control law (5) when $\boldsymbol{v} = \mathbf{0}$.

Generally, the output consensus error between two agent outputs is

$$||\boldsymbol{y}_i - \boldsymbol{y}_j|| \le 2\max\{V((\boldsymbol{w}(0), \boldsymbol{\xi}(0)), t), \gamma(||\boldsymbol{v}||_\infty)\}$$

Remark 6. Note that $\boldsymbol{v}$ can be tuned such that the simulation error remains within computable bounds. There is a tradeoff between exact fulfilment for specifications and the admissible set of global commands $\boldsymbol{v}$'s. Although we cannot drive all agents exactly follow the guidance if $\boldsymbol{v} \ne \mathbf{0}$, this provides an error tolerance for the task completion without redesigning controllers.

Then we give an illustrative example for consensusly tracking a desired trajectory or planned path.

Example 1. Consider 3 agents with the following dynamics:

$$\begin{cases} \dot{x}_i^1 = x_i^2 \\ \dot{x}_i^2 = x_i^3 \\ \dot{x}_i^3 = u_i \\ y_i = x_i^1 \in \mathbf{R} \end{cases} \tag{16}$$

where $C = (1, 0, 0)$. Obviously, Assumption 2 holds.

Here we consider a desired task trajectory $w_d(t) = 5 - 5\cos(0.1t)$. We implement this hierarchical control method step by step as follows:

1) Construct an abstract system as

$$\dot{\boldsymbol{w}} = \boldsymbol{v}, \quad y_0 = \boldsymbol{w} \in \mathbf{R} \tag{17}$$

Note that $F = 1$, $S = 0$, and $D = 1$. It is not hard to find that

$$X_c = \mathbf{1} \otimes X, \quad X^{\mathrm{T}} = (1, 0, 0)$$

and therefore, we can take $C^{+} = X$ and Assumption 3 is satisfied.

2) Design $\boldsymbol{v}$ and its initial conditions for the abstract system according to the task. In this simple case, the abstract system can complete the task by simply taking $w(0) = 0$ and the command controller v as follows:

$$\dot{\boldsymbol{w}} = \boldsymbol{v} = \dot{w}_d = 0.5\sin(0.1t)$$

3) The intra-agent topology is the switching between graph $\mathcal{G}_i$ $(i = 1, 2)$ described by the two Laplacians, respectively:

$$L_1 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad L_2 = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

and the diagonal matrices for the interconnection between the abstract system and the agents are

$$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

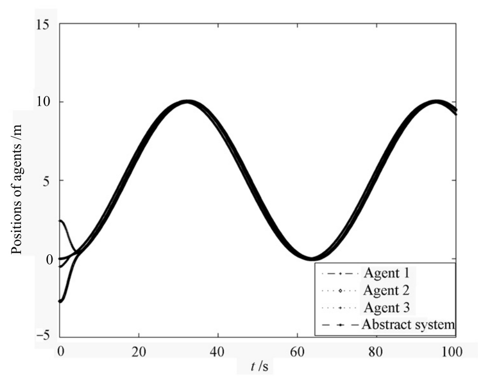
The switchings between the two topologies are periodically carried out in the following order $\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_1, \mathcal{G}_2, \cdots\}$ with periods $t = 5$, which satisfies Assumption 1.

4) Construct control input for agent i based on $\boldsymbol{v}$ and even y_0 (if available) by an interface function between two levels in the form of (5) as

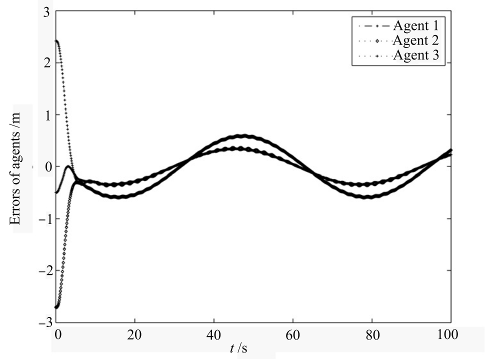
$$\begin{cases} u_i = \boldsymbol{v} + \boldsymbol{k}_1 \boldsymbol{z}_i, \\ \boldsymbol{z}_i = \sum\limits_{j \in N_i^{\sigma}} a_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j) + a_{i0}(\boldsymbol{x}_i - X^{\mathrm{T}} \boldsymbol{w}) \end{cases}$$

where $\boldsymbol{k}_1 = -(1, \sqrt{2}+1, \sqrt{2}+1)$.

The initial positions of all agents are generated randomly between $[-3, 3]$. Here the output trajectories of all agents are showed in Fig. 1 with tracking errors.



(a) Output trajectories of all agents



(b) Values of $y_i - \boldsymbol{w}$
($i = 1, 2, 3$)

Fig. 1 The simulation results when $w_d(t) = 5 - 5\cos(0.1t)$

This method can be easily extended to the formation problem if we send different commands to different agents. Here is an example for distributed formation control.

Example 2. Consider 3 planar agents with the following dynamics:

$$\begin{cases}\dot{\boldsymbol{x}}_i^1 = \boldsymbol{x}_i^2\\ \dot{\boldsymbol{x}}_i^2 = \boldsymbol{x}_i^3\\ \dot{\boldsymbol{x}}_i^3 = \boldsymbol{u}_i\\ \boldsymbol{y}_i = \boldsymbol{x}_i^1 \in \mathbf{R}^2\end{cases} \tag{18}$$

where $C = (I_2, 0, 0)$ and $\boldsymbol{x}_i^j \in \mathbf{R}^2$. Assumption 2 is satisfied.

For simplicity, we consider the formation discussed in [11]. The formations for $\boldsymbol{y}_i$ ($i = 1, 2, 3$) are obtained with a constant transformation matrix

$$F = \begin{bmatrix}\dfrac{\sqrt{3}}{2} & \dfrac{1}{2}\\ -\dfrac{1}{2} & \dfrac{\sqrt{3}}{2}\end{bmatrix}$$

and relative vectors from a given virtual center to the agents are $\boldsymbol{h}_1 = (\sqrt{2}, 1), \boldsymbol{h}_2 = (-\sqrt{3}, 1), \boldsymbol{h}_3 = (0, -2)$, respectively.

The motion of the virtual center for the formation is planned first so that the center of the agents in a special triangular formation follows a given smooth trajectory $\boldsymbol{y}(t)$.

Similar to Example 1, we construct an abstract system

$$\dot{\boldsymbol{w}} = \boldsymbol{v}, \quad \boldsymbol{y}_0 = \boldsymbol{w} \in \mathbf{R}^2 \tag{19}$$

with $X^{\mathrm{T}} = (I_2, 0, 0)$, $C^+ = X$ and Assumption 2 is satisfied. Obviously, Assumption 3 holds.

In our case, we can simply take $\boldsymbol{w}(0) = \boldsymbol{y}(0)$ and the command controller $\boldsymbol{v} = \dot{\boldsymbol{y}}(t)$ to make the abstract system moves along $\boldsymbol{y}(t)$. Here we let $\boldsymbol{y}(t)$ be a straight line, and still assume the topology as in Example 1, which satisfies Assumption 1.

To make all the agents follow the trajectory $\boldsymbol{y}(t)$ with the specific formation, a distributed control law is given for agent i:

$$\begin{cases}\boldsymbol{u}_i = \boldsymbol{v} + \boldsymbol{v}_i + \boldsymbol{k}_1\boldsymbol{z}_i\\ \boldsymbol{z}_i = \sum\limits_{j\in N_i^\sigma} a_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j) + a_{i0}(\boldsymbol{x}_i - X^{\mathrm{T}}\boldsymbol{w})\\ \boldsymbol{v}_i = -k_1(\sum\limits_{j\in N_i^\sigma} a_{ij}(\boldsymbol{h}_i - \boldsymbol{h}_j) + a_{i0}\boldsymbol{h}_i), \quad i = 1, 2, 3\end{cases}$$

where $\boldsymbol{k}_1 = -(1, \sqrt{2}+1, \sqrt{2}+1) \otimes I_2$.

The simulation result is shown in Fig. 2.

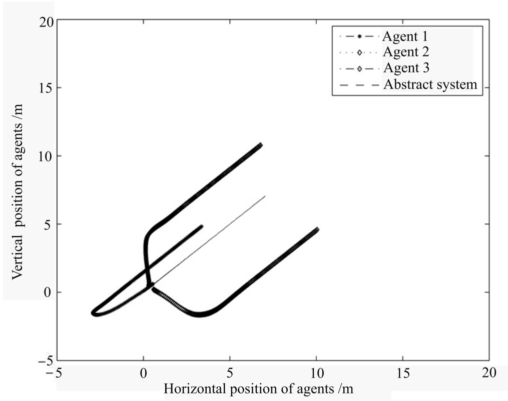


Fig. 2 Triangle formation of three agents

4 Conclusions

In this paper, the distributed multi-agent control problem was discussed with hierarchical control design. A simple abstract model was constructed to reduce complexity and complete our tasks. The solvability was solved with the help of simulation functions under some assumptions. Leader-follower techniques were employed with two distributed hierarchical control laws for the switching multi-agent topology case, and the stability analysis was given with the help of common quadratic Lyapunov functions. Our ongoing work includes the extension of our result to nonlinear cases and the accuracy for complex tasks.

References

1 Olfati-Saber R, Fax J A, Murray R M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 2007, **95**(1): 215−233

2 Ren W, Beard R W, Atkins E M. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 2007, **27**(2): 71−82

3 Wang X L, Hong Y G, Huang J, Jiang Z P. A distributed control approach to a robust output regulation problem for multi-agent linear systems. *IEEE Transactions on Automatic Control*, 2010, **55**(12): 2891−2895

4 Chen J, Yu M, Dou L H, Gan M G. A fast averaging synchronization algorithm for clock oscillators in nonlinear dynamical network with arbitrary time-delays. *Acta Automatica Sinica*, 2010, **36**(6): 873−880

5 Williams A, Glavaski S, Samad T. Formations of formations: hierarchy and stability. In: Proceedings of the 2004 American Control Conference. Boston, USA: IEEE, 2004. 2992−2997

6 Antoulas A C, Sorensen D C, Gugercin S. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 2000, **280**: 193−219

7 Pappas G J, Lafferriere G, Sastry S. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 2000, **45**(6): 1144−1160

8 Ikeda M, Siljak D, White D. An inclusion principle for dynamic systems. *IEEE Transactions on Automatic Control*, 1984, **29**(3): 244−249

9 Aoki M. Control of large-scale dynamic systems by aggregation. *IEEE Transactions on Automatic Control*, 1968, **13**(3): 246−253

10 Girard A, Pappas G J. Hierarchical control system design using approximate simulation. *Automatica*, 2009, **45**(2): 566−571

11 Tang Y T, Hong Y G. Control design for flexible hierarchical formation of multiple robots. In: Proceedings of the 10th World Congress on Intelligent Control and Automation. Beijing, China, 2012

12 Godsil C, Royle G. *Algebraic Graph Theory*. New York: Springer-Verlag, 2001

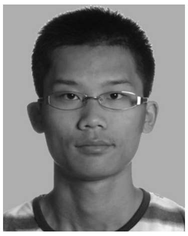
13 Horn R A, Johnson C R. *Matrix Theory*. Cambridge: Cambridge University Press, 1986

14 Hong Y G, Hu J P, Gao L X. Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 2006, **42**(7): 1177−1182

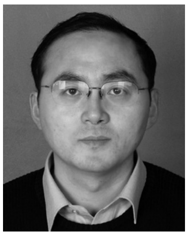
15 Hong Y G, Wang X L, Jiang Z P. Multi-agent coordination with general linear models: a distributed output regulation approach. In: Proceedings of the 8th IEEE International Conference on Control and Automation. Xiamen, China: IEEE, 2010. 137−142

16 Milner R. *Communication and Concurrency*. New York: Prentice-Hall, 1989

17 Wonham W M. *Linear Multivariable Control (Third edition)*. New York: Springer-Verlag, 1985



TANG Yu-Tao Ph. D. candidate at Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS). He received his B. S. degree in applied mathematics from Harbin Engineering University in 2009. His research interest covers multi-agent systems and robotic networks. Corresponding author of this paper. E-mail: yttang@amss.ac.cn



HONG Yi-Guang Professor at Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS). He received his Ph. D. degree from Institute of Systems Science, CAS. His research interest covers nonlinear control, multi-agent systems, and networked systems. E-mail: yghong@iss.ac.cn