Int. J. Appl. Math. Comput. Sci., 2009, Vol. 19, No. 4, 639–646
DOI: 10.2478/v10006-009-0051-1

amcs

A MATRIX INEQUALITY BASED DESIGN METHOD FOR CONSENSUS PROBLEMS IN MULTI–AGENT SYSTEMS

GUISHENG ZHAI, SHOHEI OKUNO, JOE IMAE, TOMOAKI KOBAYASHI

Department of Mechanical Engineering
Osaka Prefecture University, Sakai, Osaka 599–8531, Japan
e-mail: zhai@me.osakafu-u.ac.jp

In this paper, we study a consensus problem in multi-agent systems, where the entire system is decentralized in the sense that each agent can only obtain information (states or outputs) from its neighbor agents. The existing design methods found in the literature are mostly based on a graph Laplacian of the graph which describes the interconnection structure among the agents, and such methods cannot deal with complicated control specification. For this purpose, we propose to reduce the consensus problem at hand to the solving of a strict matrix inequality with respect to a Lyapunov matrix and a controller gain matrix, and we propose two algorithms for solving the matrix inequality. It turns out that this method includes the existing Laplacian based method as a special case and can deal with various additional control requirements such as the convergence rate and actuator constraints.

Keywords: multi-agent systems, consensus, decentralized control, graph Laplacian, matrix inequality, LMI.

1. Introduction

For multi-agent systems, the notion "consensus" means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents (Olfati-Saber *et al*., 2007). The theoretical framework for posing and solving consensus problems in networked dynamic systems was introduced in (Olfati-Saber and Murray, 2003; 2004) based on the earlier work of Fax and Murray (Fax, 2001; Fax and Murray, 2004). In recent years, there has been much interest in problems related to multi-agent systems with a close relation to consensus problems, including collective behavior of flocks and swarms, sensor fusion, random networks, the synchronization of coupled oscillators, formation control of multi-robots, optimization-based cooperative control, etc. For more detailed information on this line, see the survey paper (Olfati-Saber *et al*., 2007) and the references therein.

Focusing on the basic consensus problem requiring that all agents' states converge to the same vector, the well known existing method is to describe the agents' interconnection structure as a directed or undirected graph and to use the graph Laplacian as a state feedback gain. In that context, the proof of the states' convergence is usually made using LaSalle's invariant principle (Khalil, 2002). However, to the best of our knowledge, such a Laplacian based method is generally limited to the case that each agent has a low dimension and the control specification is simple. Pogromsky *et al.* (2002) studied partial synchronization, which is closely related to the consensus problem, and proposed designing global symmetric coupling among agents (subsystems) so that the system's stability could be examined by Lyapunov's direct method. In that context, the requirement of global symmetry limits the range of systems, and the feedback gain is basically the interconnection strength among subsystems (similar to the graph Laplacian). Recently, Wang *et al.* (2008) established a decentralized control method for achieving the consensus of multi-agent systems, but the assumption was made that all subsystems should be controllable, and the differences in the agents' states are equally used in the feedback, which generally leads to conservativeness. The controller proposed in (Pogromsky *et al*., 2002) cannot deal with convergence rate specification or actuator constraints, and the one in (Wang *et al*., 2008) cannot attack actuator constraints. To deal with the case that agents' dynamics are in a general form, and to incorporate these additional control specifications, we need to seek a new method.

This paper is motivated by the above observation. We study a basic consensus problem in multi-agent systems, where the entire system is decentralized in the sense that

each agent can only obtain information (states or outputs) from its neighbor agents. Realizing the fact that the limitation of the existing Laplacian based method is originated from the use of LaSalle's invariant principle in the convergence proof, we reduce the consensus problem to solving a *strict matrix inequality* with respect to a Lyapunov matrix and a controller gain matrix, which is a necessary and sufficient condition for the consensus problem. The controller gain matrix has a structure constraint corresponding to the interconnection among the agents. Since the matrix inequality is bilinear with respect to the variables, we propose two algorithms for solving the matrix inequality effectively. It turns out that this method includes the existing Laplacian based method as a special case and can deal with various additional control requirements.

The remainder of this paper is organized as follows: In Section 2, we give some preliminaries about graph theory together with the existing Laplacian based method for consensus. Section 3 establishes the matrix inequality based method by reducing the consensus problem to a matrix inequality with an algorithm and discusses how to solve the matrix inequality effectively. Two numerical examples are given in Section 4 to show the validity of the proposed method, and an extension is made to the case of static output feedback in Section 5. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Graph Laplacian. Let us review some basic definitions for consensus in a network. The interconnection structure of a family of agents can be represented by using a *directed graph* (or *digraph*) $G = (\mathcal{V}, \mathcal{E})$ with the set of nodes $\mathcal{V} = \{1, 2, \ldots, N\}$ and edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The edge $(i, j) \in \mathcal{E}$ or $i \to j$ means that the information of the i-th agent is available for the j-th agent.

The neighbor agents set of the i-th agent is defined as

$$\mathcal{N}_i := \left\{ j \in \mathcal{V} \middle| (j, i) \in \mathcal{E} \right\}, \tag{1}$$

which is the index set of the agents from which the i-th agent can obtain necessary information. Then, the *graph Laplacian* of the agents' structure is defined as $L = [l_{ij}]_{N \times N}$, where

$$l_{ij} = \begin{cases} -1 & \text{if } j \in \mathcal{N}_i, \\ |\mathcal{N}_i| & \text{if } j = i, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

and $|\mathcal{N}_i|$ denotes the number of neighbors of the i-th agent (or the *in-degree* of agent i). For example, using the above definition, the graph Laplacians of the structures in Fig. 1 are, respectively,

$$\begin{bmatrix} 2 & 0 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}. \tag{3}$$

It is easy to see from the definition (2) that all row-sums of L are zero, and thus L always has a zero eigenvalue and the corresponding eigenvector $\mathbf{1} = [1\, 1\, \cdots\, 1]^T$.

For other spectral properties of the graph Laplacian, see, e.g., (Mohar, 1991; Godsil and Royle, 2001).

2.2. Consensus via the graph Laplacian. For simplicity, consider the case where all agents are integrators described by

$$\dot{x}_i = u_i\,, \quad x_i \in \mathbb{R}. \tag{4}$$

The *consensus* problem is to design the control input u_i, depending on states of its neighbor agents, so that all agents' states converge to the same value, i.e.,

$$\lim_{t \to \infty} |x_i(t) - x_j(t)| = 0\,, \quad \forall i, j\,. \tag{5}$$

To solve the consensus problem, the existing method is to construct the control input as

$$u_i = \sum_{j \in \mathcal{N}_i} (x_j - x_i), \tag{6}$$

which is based on the idea of proportionally reducing the errors between two agents' states. In fact, with the definitions $x = [x_1\, x_2\, \cdots\, x_N]^T$ and $u = [u_1\, u_2\, \cdots\, u_N]^T$, the control input (6) can be compactly written as

$$u = -Lx\,. \tag{7}$$

In this sense, we call (6) a *(graph) Laplacian based method*.

The closed-loop system composed of (4) and (6) (or (7)) is

$$\dot{x} = -Lx, \tag{8}$$

and in the literature LaSalle's invariant principle (Khalil, 2002) is usually used to show that all states converge to the same value as required in (5) .

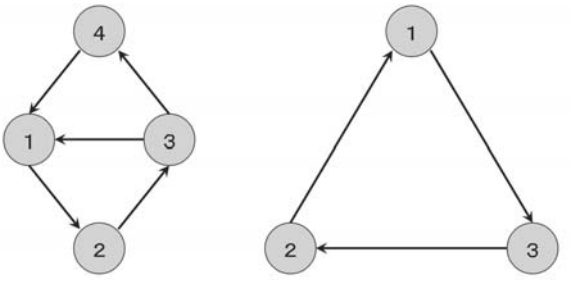

Fig. 1. Directed graph examples.

2.3. Kronecker product. A tool that is very useful in modeling and manipulating equations governing group motion is the *Kronecker product* $\otimes$ (Lancaster and Tismenetsky, 1985), which is defined between two matrices $P = [p_{ij}]$ and Q as

$$P \otimes Q = [p_{ij}Q]\,. \tag{9}$$

For example, if $\dot{x}_i = Ax_i$ represents the dynamics of a single agent, the dynamics of $\mathcal{N}$ identical agents can be represented as $\dot{x} = (I_{\mathcal{N}} \otimes A)x$. Another important case is when A is an $\mathcal{N}\times\mathcal{N}$ matrix representing the manipulation of scalar data from $\mathcal{N}$ agents, and that the manipulation needs to be applied to each value of a vector of length n. In that case, the manipulation can be represented by concatenating the $\mathcal{N}$ vectors of length n into a single vector of length $\mathcal{N}n$, and multiplying it by $A \otimes I_n$.

The following property of the Kronecker product:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{10}$$

can be proved (Lancaster and Tismenetsky, 1985) when all matrix operations are well defined. In particular, if X is an $r \times s$ matrix, and Y is an $N \times N$ matrix, then

$$(I_N \otimes X)(Y \otimes I_s) = (Y \otimes I_r)(I_N \otimes X) = Y \otimes X\,. \tag{11}$$

3. Matrix inequality based design method

3.1. Problem formulation. Consider the case where all agents have the same dynamics described as

$$\dot{x}_i = Ax_i + Bu_i, \tag{12}$$

where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ is the control input, and A, B are constant matrices of an appropriate dimension. Since it is known that consensus among agents is possible if and only if the interconnection graph includes a directed spanning tree (Olfati-Saber *et al.*, 2007), we assume that the present system graph also has this property. Moreover, for the benefit of dealing with all agents in a collective manner, we write the entire system compactly as

$$\dot{x} = A_D x + B_D u, \quad x \in \mathbb{R}^{nN}, \quad u \in \mathbb{R}^{mN}, \tag{13}$$

where $x = [x_1^T\ x_2^T\ \cdots\ x_N^T]^T \in \mathbb{R}^{nN}$ is the group state and $u = [u_1^T\ u_2^T\ \cdots\ u_N^T]^T \in \mathbb{R}^{mN}$ is the group control input, and $A_D = I_N \otimes A\,,\ B_D = I_N \otimes B\,.$

The consensus problem is to design the decentralized state feedback u, i.e., to design each u_i depending on states of its neighbor agents and itself, so that all agents' states converge to the same vector, i.e.,

$$\lim_{t\to\infty} \|x_i(t) - x_j(t)\| = 0\,, \quad \forall i, j\,. \tag{14}$$

Here, the symbol $\|\cdot\|$ denotes the Euclidean norm of a vector.

The following lemma plays an important role in the forthcoming discussion.

Lemma 1. *Let $L_C = L \otimes I_n$. Then,*

$$L_C A_D = A_D L_C\,. \tag{15}$$

Proof. Using (10) or (11), we obtain

$$\begin{aligned} L_C A_D &= (L \otimes I_n)(I_N \otimes A) \\ &= (LI_N) \otimes (I_n A) = (I_N L) \otimes (AI_n) \\ &= (I_N \otimes A)(L \otimes I_n) = A_D L_C. \end{aligned} \tag{16}$$

■

3.2. Stabilization condition and the consensus algorithm. In this section, we propose a matrix inequality based method for the above-mentioned consensus problem. The basic idea is to reduce the consensus problem to a stabilization issue.

As formulated in the previous section, the control input has a decentralized structure. In other words, the control input of the i-th agent only depends on states of its neighbor agents and itself. To meet this requirement, we propose the following control input:

$$u = K_D L_C x, \tag{17}$$

where $K_D = \text{diag}\{K_1, K_2, \ldots, K_N\}$, $K_i \in \mathbb{R}^{n\times n}$. To see that the controller (17) has the desired decentralized structure, we take the left graph structure in Fig. 1 as an example and suppose all agents' dynamics dimension is one. Then, (17) leads to

$$\begin{aligned} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} &= K_D \begin{bmatrix} 2 & 0 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\ &= K_D \begin{bmatrix} 2x_1 - x_3 - x_4 \\ x_2 - x_1 \\ x_3 - x_2 \\ x_4 - x_3 \end{bmatrix} \\ &= \begin{bmatrix} k_1(2x_1 - x_3 - x_4) \\ k_2(x_2 - x_1) \\ k_3(x_3 - x_2) \\ k_4(x_4 - x_3) \end{bmatrix}, \end{aligned} \tag{18}$$

which is obviously consistent with the interconnection structure described on the left side of Fig. 1. For example, the control input of Agent 1 depends on x_1, x_3 and x_4.

The closed-loop system composed of (13) and (17) is

$$\dot{x} = (A_D + B_D K_D L_C)x\,. \tag{19}$$

Note that the objective of designing K_D is *not* to drive all states to zeros, but to drive all states to the same vector. Having this in mind, we further observe that

$$x_C = L_C x = 0 \Longleftrightarrow x = \alpha\mathbf{1}\,. \tag{20}$$

Thus, the control problem is reduced to considering the stability/stabilization of x_C whose dynamics can be described as

$$\begin{aligned}\dot{x}_C &= L_C\dot{x} = L_C A_D x + L_C B_D K_D L_C x \\ &= A_D L_C x + L_C B_D K_D x_C \\ &= (A_D + L_C B_D K_D) x_C .\end{aligned} \tag{21}$$

If all elements of x_C are independent, we can use any existing design method (Lyapunov equation, matrix inequality) for the above system and obtain a stabilization condition with respect to the unknown gain matrix K_D. However, since the matrix L is not full rank, L_C is not full rank either, and thus the elements of the vector x_C are not independent. For example, consider the right graph structure in Fig. 1 with all agents' dynamics dimension being one. Then, $n = 1$, $N = 3$, and

$$x_C = L_C x = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} x_1 - x_2 \\ x_2 - x_3 \\ x_3 - x_1 \end{bmatrix}. \tag{22}$$

Obviously, $x_1 - x_2 \to 0$ and $x_2 - x_3 \to 0$ lead to $x_3 - x_1 \to 0$, which means we only need to take care of two elements of the vector x_C.

Based on the above observation, we extract the linearly independent rows of L and denote it by $\tilde{L}$. Then, let $\tilde{L}_C = \tilde{L} \otimes I_n$, and let $\tilde{x}_C = \tilde{L}_C x$. It is easy to see that $\tilde{x}_C$ is in fact a subvector of x_C and the elements of $\tilde{x}_C$ are independent. Therefore, from now on we will focus our attention on the stabilization of $\tilde{x}_C$.

The dynamics equation of $\tilde{x}_C$ is

$$\dot{\tilde{x}}_C = (\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D)\tilde{x}_C \,, \tag{23}$$

where $\tilde{A}_D = I_{N-1} \otimes A$, and $\tilde{K}_D$ is computed from K_D satisfying $\tilde{K}_D \tilde{L}_C = K_D L_C$. The procedure is as follows: Since $\tilde{L}$ is extracted from L, suppose that the extracted row number is $j_1, j_2, \ldots, j_{N-1}$. Defining e_i as the i-th column of an identity matrix, we obtain

$$\tilde{L} = E_C L, \quad E_C \triangleq \begin{bmatrix} e_{j_1}^T \\ e_{j_2}^T \\ \vdots \\ e_{j_{N-1}}^T \end{bmatrix}. \tag{24}$$

Then, using (10), we obtain

$$\begin{aligned}\tilde{L}_C &= \tilde{L} \otimes I_n = (E_C L) \otimes (I_n \times I_n) \\ &= (E_C \otimes I_n)(L \otimes I_n) = (E_C \otimes I_n) L_C \,.\end{aligned} \tag{25}$$

Thus, the relation between K_D and $\tilde{K}_D$ is obtained from $\tilde{K}_D \tilde{L}_C = K_D L_C$ such that

$$\tilde{K}_D (E_C \otimes I_n) L_C = K_D L_C \,. \tag{26}$$

Although the matrix L_C is singular and thus cannot be dropped in the above equation, we can use a kind of pseudo-inverse matrix or another direct method to obtain $\tilde{K}_D$ including the unknown gain matrices in K_D.

For example, consider again the right graph structure in Fig. 1 with all agents' dynamics dimension being one. We can easily obtain $\tilde{L}_C$ from L_C and E_C as

$$\tilde{L}_C = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}, \; E_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{27}$$

Since the original gain matrix is $K_D = \text{diag}\{k_1, k_2, k_3\}$, from (26) we obtain

$$\begin{aligned}&\tilde{K}_D \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \\ &\quad = \tilde{K}_D \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \\ &\quad = K_D L_C = \begin{bmatrix} k_1 & -k_1 & 0 \\ 0 & k_2 & -k_2 \\ -k_3 & 0 & k_3 \end{bmatrix},\end{aligned} \tag{28}$$

and thus

$$\begin{aligned}\tilde{K}_D &= \begin{bmatrix} k_1 & -k_1 & 0 \\ 0 & k_2 & -k_2 \\ -k_3 & 0 & k_3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \\ -k_3 & -k_3 \end{bmatrix}.\end{aligned} \tag{29}$$

Notice that $\tilde{K}_D$ has a different size from K_D, but it inherits all the variables in K_D in a linear form.

To summarize the above discussion, we have reduced the consensus problem to the stabilization of the system (23) by the feedback gain matrix $\tilde{K}_D$. The original feedback gain matrices K_i are included in $\tilde{K}_D$ and thus can be extracted easily.

Theorem 1. *The controller (17) solves the consensus problem in the system (12) or (13) if and only if there is a positive definite matrix P satisfying*

$$(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D)^T P + P(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D) < 0 \,, \tag{30}$$

where $\tilde{K}_D$ has the decentralized structure including the feedback gains K_i as variables.

Remark 1. From the above discussion it is easy to see that the existing Laplacian based method is a special case of Theorem 1, where the feedback gains are set to a fixed vector with the agents' dimension being one.

Remark 2. Although the convergence rate issue is mentioned in the literature using the name of *algebraic connectivity*, there is no practical design method for it. In

contrast, Theorem 1 can design the convergence rate of the agents' states, e.g., by specifying an appropriate positive scalar ζ in the condition (30) as

$$\begin{aligned}(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D + \zeta I)^T P \\ + P(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D + \zeta I) < 0\,.\end{aligned} \tag{31}$$

Remark 3. Theorem 1 can also deal with actuator constraints directly, e.g., by specifying an appropriate positive scalar k_{lim} and constructing an additional linear matrix inequality

$$\begin{bmatrix} k_{lim} I & \tilde{K}_D \\ \tilde{K}_D^T & k_{lim} I \end{bmatrix} > 0, \tag{32}$$

since the matrix inequality (32) requires necessarily that $\tilde{K}_D^T \tilde{K}_D < k_{lim}^2 I$ and thus $K_i^T K_i < k_{lim}^2 I$.

Remark 4. Although it is assumed in the problem formulation that all agents have the same dynamic differential equation, it can be seen that the result is the same even if B_i's are different. In that case, $B_D = \mathrm{diag}\{B_1, B_2, \ldots, B_N\}$, where B_i is the control input matrix of the i-th agent. Concerning the system matrix part, it can be relaxed to the assumption that $L_C A_D = A_D L_C$ holds with $A_D = \mathrm{diag}\{A_1, A_2, \ldots, A_N\}$, where A_i is the system matrix of the i-th agent. To conclude in the end of this subsection, we summarize the consensus algorithm as follows.

Consensus Algorithm via Decentralized State Feedback

Step 1. Extract the linear independent rows of L and denote it by $\tilde{L}$. Let $\tilde{L}_C = \tilde{L} \otimes I_n$, $\tilde{A}_D = I_{N-1} \otimes A$.

Step 2. Solve $\tilde{K}_D \tilde{L}_C = K_D L_C$ or $\tilde{K}_D (E_C \otimes I_n) L_C = K_D L_C$ to obtain $\tilde{K}_D$ including the unknown feedback gains as parameters.

Step 3. Solve the matrix inequality (30) with respect to $\tilde{K}_D$ and $P > 0$.

Step 4. Extract the controller gain K_i's from $\tilde{K}_D$.

3.3. Discussion on solving (30). Theorem 1 gives a necessary and sufficient condition under which the consensus problem is solved. However, the matrix inequality (30) is a bilinear matrix inequality (BMI) with respect to the variables $\tilde{K}_D$ and $P > 0$. If there is no constraint on $\tilde{K}_D$, we can transform (30) into an equivalent LMI (Boyd *et al.*, 1994). This is not the case right now, and $\tilde{K}_D$ has a fixed structure, as discussed in the previous section. Thus, as pointed out in the literature, it is generally difficult to solve (30) globally.

We first propose a two stage method for solving (30). Although $\tilde{A}_D$ is not stable, we can always find a positive scalar λ such that $\tilde{A}_D - \lambda I$ is stable. For example, we can choose λ larger than the largest real part of the eigenvalues of $\tilde{A}_D$, i.e., $\lambda > \max\{\mathrm{Re}\lambda(\tilde{A}_D)\}$. Then, there exists a positive definite matrix P_λ satisfying

$$(\tilde{A}_D - \lambda I)^T P_\lambda + P_\lambda (\tilde{A}_D - \lambda I) < 0\,. \tag{33}$$

The next stage is to solve (30) with P fixed as P_λ, i.e.,

$$(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D)^T P_\lambda + P_\lambda (\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D) < 0, \tag{34}$$

which is an LMI with respect to $\tilde{K}_D$, thus easily solvable with any existing LMI software such as the LMI Control Toolbox in Matlab (Gahinet *et al.*, 1994).

If the above method does not provide a solution, this means that we have to consider how to fix the variable P so that the resultant LMI is feasible. For this purpose, we propose to use the homotopy based method established in (Zhai *et al.*, 2001). More precisely, we introduce a real number μ varying from 0 to 1 and define a matrix function

$$F(P, \tilde{K}_D, \mu) = F_1(P) + \mu F_2(P, \tilde{K}_D), \tag{35}$$

where

$$\begin{aligned} F_1(P) &= (\tilde{A}_D - \lambda I)^T P + P(\tilde{A}_D - \lambda I), \\ F_2(P, \tilde{K}_D) &= P\tilde{L}_C B_D \tilde{K}_D + \tilde{K}_D^T B_D^T \tilde{L}_C^T P + 2\lambda P \end{aligned} \tag{36}$$

and λ is computed as above. Then

$$F(P, \tilde{K}_D, \mu) = \begin{cases} F_1(P) & \text{if } \mu = 0, \\ \text{the LHS of (30)} & \text{if } \mu = 1, \end{cases} \tag{37}$$

and the problem of finding a solution to (30) is embedded in the parametrized family of problems

$$F(P, \tilde{K}_D, \mu) < 0\,, \quad \mu \in [0, 1]\,. \tag{38}$$

Next, we start solving (38) with $\mu = 0$, which is very easy when using the LMI software. Then, we increase μ gradually (for example, let $\mu = k/M (k = 1, 2, \ldots, M)$ with a large M) and solve (38) gradually with P and $\tilde{K}_D$ being fixed alternately, until μ reaches 1.

For a more detailed description of the homotopy based algorithm for solving BMIs, refer to (Zhai *et al.*, 2001).

Remark 5. Although, for simplicity, we used λI in (33) and (36), which makes $\tilde{A}_D - \lambda I$ stable, it can be replaced by any matrix W provided that $\tilde{A}_D - W$ is stable. In that case, $F_1(P) = (\tilde{A}_D - W)^T P + P(\tilde{A}_D - W)$, $F_2(P, \tilde{K}_D) = P\tilde{L}_C B_D \tilde{K}_D + \tilde{K}_D^T B_D^T \tilde{L}_C^T P + W^T P + PW$.

4. Numerical example

In this section, we provide two examples showing the effectiveness of our method.

4.1. Example 1. Consider the right graph structure in Fig. 1 with all agents' dynamics being a double integrator, described as

$$\begin{bmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i \,. \tag{39}$$

Setting

$$\tilde{L} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix},$$

$\tilde{L}_C = \tilde{L} \otimes I_2$ and $\lambda = 10$, the first (two stage) algorithm gives us the solution

$$P = \begin{bmatrix} 52.90 & 2.54 & 0 & 0 \\ 2.54 & 53.16 & 0 & 0 \\ 0 & 0 & 52.90 & 2.54 \\ 0 & 0 & 2.54 & 53.16 \end{bmatrix},$$
$$\tilde{K}_D = \begin{bmatrix} -1.10 & -1.08 & 0 & 0 \\ 0 & 0 & -0.32 & -0.28 \\ 0.38 & 1.36 & 0.38 & 1.36 \end{bmatrix}. \tag{40}$$

Then, from (29) and the above solution it is obtained that

$$\begin{aligned} K_1 &= \begin{bmatrix} -1.10 & -1.08 \end{bmatrix}, \\ K_2 &= \begin{bmatrix} -0.32 & -0.28 \end{bmatrix}, \\ K_3 &= \begin{bmatrix} -0.38 & -1.36 \end{bmatrix}. \end{aligned} \tag{41}$$

With the above gains, the elements of $x_i - x_j \in \mathbb{R}^2$ $(i, j = 1, 2, 3, i \neq j)$ are described in Fig. 2, which shows that a consensus among all agents has been achieved. ♦

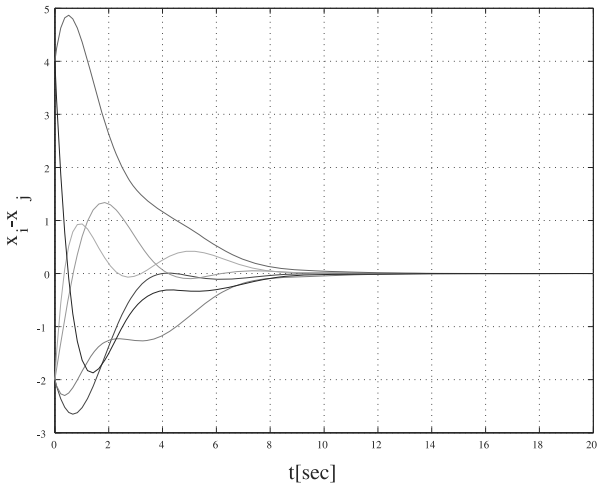

Fig. 2. Consensus achieved in Example 1.

4.2. Example 2. Consider a more complex system (a one-link arm) and the interconnection structure in Fig. 3.

The mechanics equation of each one link arm is

$$J\ddot{y}_i + B\dot{y}_i = k_t u_i, \tag{42}$$

and thus

$$\dot{x}_i = \begin{bmatrix} 0 & 1 \\ 0 & -B/J \end{bmatrix} x_i + \begin{bmatrix} 0 \\ k_t/J \end{bmatrix} u_i, \tag{43}$$

where $x_i = [y_i \;\; \dot{y}_i]^T$, J is the inertia moment of the arm, B is the viscous friction coefficient of the arm axis of rotation, k_t is the motor torque coefficient. These parameters are set to $J = 0.03[\mathrm{kgm}^2]$, $B = 0.1[\mathrm{kgm}^2/\mathrm{s}]$, and $k_t = 0.5[\mathrm{Nm/V}]$ for numerical computation.

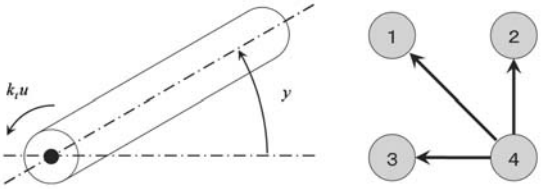

Fig. 3. One-link arm and the associated interconnection structure.

There are four agents now and the graph Laplacian is

$$L = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{44}$$

and thus $\tilde{L}$ is obtained as

$$\tilde{L} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}. \tag{45}$$

Setting $\tilde{L}_C = \tilde{L} \otimes I_2$ and $\lambda = 1$, the first (two stage) algorithm gives us the solution

$$P = \begin{bmatrix} 1.72 & 0.22 & 0 & 0 & 0 & 0 \\ 0.22 & 1.18 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.72 & 0.22 & 0 & 0 \\ 0 & 0 & 0.22 & 1.18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.72 & 0.22 \\ 0 & 0 & 0 & 0 & 0.22 & 1.18 \end{bmatrix},$$
$$\tilde{K}_D = \begin{bmatrix} -0.09 & 0.15 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.09 & 0.15 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.09 & 0.15 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{46}$$

The solution $\tilde{K}_D$ is reasonable since the fourth agent is the leader and thus it does not get any information from other agents. The other three agents are in the same position (only get information from the fourth agent), and thus their gains are the same.

With the above controller gains, the elements of $x_i - x_j \in \mathbb{R}^2$ $(i, j = 1, 2, 3, 4, i \neq j)$ are described in Fig. 4, which shows that a consensus among all agents has been achieved. ♦

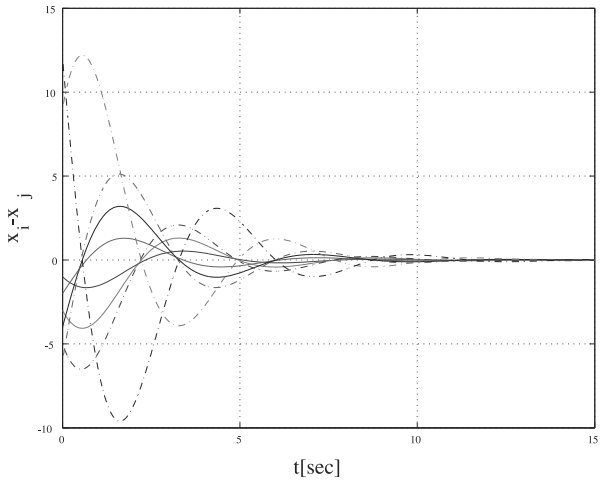

Fig. 4. Consensus achieved in Example 2

5. Extension to static output feedback

In this section, we extend the discussion to the case of static output feedback. Consider the case in which all agents have the same dynamics described as

$$\dot{x}_i = Ax_i + Bu_i, \quad y_i = Cx_i, \tag{47}$$

where x_i and u_i are the same as before, and $y_i \in \mathbb{R}^q$ is the measurement output. Similarly, the entire system is written as

$$\dot{x} = A_D x + B_D u, \quad y = C_D x, \tag{48}$$

where x and u are the same as before, $y = [y_1^T \; y_2^T \; \cdots y_N^T]^T \in \mathbb{R}^{qN}$, and $C_D = I_N \otimes C$.

Now, the consensus problem is to design the decentralized static output feedback

$$u = K_D L_{\bar{C}} y, \tag{49}$$

instead of (17), so that all agents' states converge to the same vector. Here, note that $L_{\bar{C}} = L \otimes I_q$ is different from the previous L_C.

The closed-loop system composed of (48) and (49) is

$$\dot{x} = (A_D + B_D K_D L_{\bar{C}} C_D)x. \tag{50}$$

To proceed, we need the following lemma.

Lemma 2.

$$L_{\bar{C}} C_D = C_D L_C. \tag{51}$$

Proof. Using (10), we obtain

$$\begin{aligned} L_{\bar{C}} C_D &= (L \otimes I_q)(I_N \otimes C) \\ &= (LI_N) \otimes (I_q C) = (I_N L) \otimes (CI_n) \\ &= (I_N \otimes C)(L \otimes I_n) = C_D L_C. \end{aligned} \tag{52}$$

■

According to Lemma 2, the closed-loop system (50) is written as

$$\dot{x} = (A_D + B_D K_D C_D L_C)x. \tag{53}$$

Note that the above is almost the same as (19) and thus the remaining discussion is similar to that in Section 3.

We obtain the dynamics equations of x_C and $\tilde{x}_C$ as

$$\begin{aligned} \dot{x}_C &= (A_D + L_C B_D K_D C_D)x_C, \\ \dot{\tilde{x}}_C &= (\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D \tilde{C}_D)\tilde{x}_C, \end{aligned} \tag{54}$$

where $\tilde{C}_D = I_{N-1} \otimes C$, and $\tilde{K}_D$ is determined from K_D satisfying $\tilde{K}_D \tilde{C}_D \tilde{L}_C = K_D C_D L_C$. The matrix variable $\tilde{K}_D$ has a different size from K_D, but it inherits all the variables in K_D in a linear form.

The above discussion is summarized in the following theorem.

Theorem 2. *The controller (49) solves the consensus problem in the system (47) or (48) if and only if there is a positive definite matrix P satisfying*

$$\begin{aligned} &(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D \tilde{C}_D)^T P \\ &\quad + P(\tilde{A}_D + \tilde{L}_C B_D \tilde{K}_D \tilde{C}_D) < 0. \end{aligned} \tag{55}$$

For integrity, we state the algorithm in accordance with Theorem 2 as follows.

Consensus Algorithm via Decentralized Output Feedback

Step 1. Extract the linear independent rows of L and denote it by $\tilde{L}$. Let $\tilde{L}_C = \tilde{L} \otimes I_n$, $\tilde{A}_D = I_{N-1} \otimes A$.

Step 2. Solve $\tilde{K}_D \tilde{C}_D \tilde{L}_C = K_D C_D L_C$ to obtain $\tilde{K}_D$ including the unknown feedback gains as parameters.

Step 3. Solve the matrix inequality (55) with respect to $\tilde{K}_D$ and $P > 0$.

Step 4. Extract the controller gain K_i's from $\tilde{K}_D$.

Remark 6. In much the same manner, the assumption that the output matrices of all agents are the same can be relaxed to the one that $L_{\bar{C}} C_D = C_D L_C$ with $C_D = \mathrm{diag}\{C_1, C_2, \ldots, C_N\}$, where C_i is the output matrix of the i-th agent.

6. Concluding remarks

For the basic consensus problem in multi-agent systems, we have proposed to reduce the control problem to solving a matrix inequality with respect to a Lyapunov matrix and a controller gain matrix. To solve the matrix inequality which is bilinear with respect to the variables, we proposed two algorithms which can be switched in accordance with the computing situation. The proposed method can deal with agents with general linear system dynamics of any dimension and incorporate additional control requirements such as the convergence rate and actuator constraints.

References

Boyd, S., El Ghaoui, L., Feron, E. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA.

Fax, J. A. (2001). *Optimal and Cooperative Control of Vehicle Formations*, Ph.D. dissertation, Control Dynamical Systems, California Institute of Technology, Pasadena, CA.

Fax, J. A. and Murray, R. M. (2004). Information flow and cooperative control of vehicle formations, *IEEE Transactions on Automatic Control* **49**(9): 1465–1476.

Gahinet, P., Nemirovskii, A., Laub, A. and Chilali, M. (1994). The LMI control toolbox, *Proceedings of the 33rd IEEE Conference on Decision and Control, Orlando, FL, USA*, pp. 2038–2041.

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*, Springer-Verlag, Berlin.

Khalil, H. K. (2002). *Nonlinear Systems, 2nd Edn.*, Prentice Hall, Upper Saddle River, NJ.

Lancaster, P. and Tismenetsky, M. (1985). *The Theory of Matrices with Applications, 2nd Edn.*, Academic Press, Orlando, FL.

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

Olfati-Saber, R. and Murray, R. M. (2003). Consensus protocols for networks of dynamic agents, *Proceedings of the 2003 American Control Conference, Denver, CO, USA*, pp. 951–956.

Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control* **49**(9): 1520–1533.

Mohar, B. (1991). The Laplacian spectrum of graphs, *in* Y. Alavi, G. Chartrand, O. Ollermann and A. Schwenk (Eds.), *Graph Theory, Combinatorics, and Applications*, Wiley, New York, NY, pp. 871–898.

Pogromsky, A., Santoboni, G. and Nijmeijer, H. (2002). Partial synchronization: From symmetry towards stability, *Physica D* **172**(1): 65–87.

Wang, J., Cheng, D. and Hu, X. (2008). Consensus of multi-agent linear dynamic systems, *Asian Journal of Control* **10**(2): 144–155.

Zhai, G., Ikeda, M. and Fujisaki, Y. (2001). Decentralized H_∞ controller design: A matrix inequality approach using a homotopy method, *Automatica* **37**(4): 565–572.

Guisheng Zhai received the B.S. degree from Fudan University, China, in 1988, and the M.E. and Ph.D. degrees, both in system science, from Kobe University, Japan, in 1993 and 1996, respectively. Currently, he is an associate professor of mechanical engineering at Osaka Prefecture University, Japan. His research interests include large scale and decentralized control systems, robust control, switched systems and switching control, networked control systems, neural networks and signal processing, etc. He has published more than 180 international journal and conference papers and has been on editorial boards of several academic journals. He is a Senior Member of the IEEE, a member of the ISCIE, SICE, JSST and JSME.

Shohei Okuno received the B.S. and M.E. degrees both in mechanical engineering from Osaka Prefecture University, Japan, in 2007 and 2009, respectively. His research interests include formation control, matrix inequality, hybrid and switching control.

Joe Imae received the B.S. degree in precision mechanical engineering from Utsunomiya University, Japan, in 1973, and the M.S. and Ph.D. degrees in precision mechanical engineering from Tohoku University, Japan, in 1975 and 1981, respectively. Currently, he is a professor at the Department of Mechanical Engineering of Osaka Prefecture University, where he has been since 2001. His research interests include constrained dynamic optimization, evolutionary computation, and non-differentiable optimal control. He spent a sabbatical year as a visiting fellow at Australian National University from 1989 to 1990, and two months as an academic visitor at Imperial College in 1991. He is a member of the IEEE and others.

Tomoaki Kobayashi received the B. Eng., M. Eng., and Dr. Eng. degrees in 1998, 2000 and 2003, respectively, from the University of Tsukuba, Japan. He is currently an assistant professor at the Department of Mechanical Engineering, Graduate School of Engineering, Osaka Prefecture University. His research interests include optimal control, real time control systems and their applications. He is a member of the IEEE, SICE, RSJ, JSME and ISCIE.

Received: 17 November 2008
Revised: 14 March 2009
Re-revised: 7 April 2009