Kai Cai and Zhiyun Lin

Directed Cooperation:

Distributed Control of Multi-Agent Systems over Directed Graphs

ver. 1.2

December 30, 2022

Preface

Cooperative control of multi-agent systems has been actively studied in the field of systems and control in the past two decades. Such systems typically consist of a large number of distributed agents, which locally interact with one another such that they jointly pursue a global goal. Research results on cooperative control of multi-agent systems have found wide applications in robotics (swarms of vehicles/drones) [CWRKG20, MC19, SVC$^+$16], engineering (sensor/power networks) [CAYM15, DB10, OS07], physics (systems of oscillators) [DCB13, PR11, SS08], epidemics (spreading processes) [YLAC21, KBG14, OGNK13], and social/political science (opinion dynamics) [YLA$^+$18, FJB16, AL15]. The literature has grown in near-intractable volumes, but excellent textbooks (e.g. [Bul22, FM16, BAW11, ME10, RB08]) and surveys (e.g. [OPA15, DB14, CYRC13, GS10, OSFM07]) have kept the content in organized manners.

This monograph aims to provide a new perspective of the research work on cooperative control of multi-agent systems. This perspective is based on different types of *graph Laplacian matrices*. The standard (conventional) Laplacian matrix is defined based on a nonnegative *adjacency matrix* [Bap10, GR00], which describes the interaction (graph) topology of a multi-agent system. This type of Laplacian matrix is fundamental in describing the dynamics of a number of multi-agent cooperative control problems including consensus, averaging, synchronization, regulation flocking, and optimization [JLM03, INK19, CI11, CI12, Ren08, Lun12, WSA11, KCK20, OS06, XHC$^+$17, ZYC20]. The algebraic properties of this type of Laplacian matrix have been found to characterize stability and performance of the corresponding cooperative control algorithms. These algebraic properties are also closely related to the connectivity properties of the interaction graph.

More recently, two other types of Laplacian matrices have been proposed in designing cooperative control algorithms. One type is defined from a complex-valued (entry-wise) adjacency matrix, and is called *complex Laplacian*. A complex Laplacian matrix has been found useful in solving a class of formation control and localization in the 2D plane [LDY$^+$13, LWHF14, LFD15, LHZF16, LWHF16]. The other type of Laplacian matrix is defined from a general real adjacency matrix which need not be nonnegative. This type of Laplacian matrix is called *signed Lapalcian*, and has been found effective in designing cooperative control algorithms to solve formation control and localization in 3D and higher-dimensional spaces [LWC$^+$16, Zha18, HLZ$^+$17, CWL$^+$17, CLC$^+$16]. For both types – complex and signed Laplacian matrices – their algebraic properties are again essential in characterizing stability and performance of the corresponding cooperative control algorithms. In addition, these algebraic properties are also related to certain connectivity properties of the interaction graph.

The above works based on different types of Laplacian matrices thus provide us with a new angle to overview the relevant literature on multi-agent cooperative control. Although there are many different cooperative control problems in their appearances, they have a few basic points in common. The interaction topology of the agents can be described by graphs, the dynamics of multi-agent systems is hence underlied by Laplacian matrices, and the algebraic properties of these Laplacian matrices dictate stability/performance of the corresponding cooperative control algorithms. These common points therefore allow us to interlink and organize different cooperative control problems and their solutions by different types of Laplacian matrices and the corresponding algebraic properties.

Organization

There are three ways we have in mind in organizing the content of this monograph. In all cases, Part I (including Chapter 1) presents the required mathematical preliminaries for the rest of the monograph. This part is expected to be read first if the reader is not familiar with the content.

The rest of the monograph can be viewed in three ways:

First, eight different cooperative control problems are presented through Chapters 2–9.

Second, Parts II and III (including Chapters 2–5) are based on standard Laplacians, Part IV (Chapters 6–7) on complex Laplacians, and Part V (Chapters 8–9) on signed Laplacians.

Third, Parts II, III, IV, and V are based respectively on different connectivity conditions of directed graphs.

These different views are to provide flexibility to the reader with different purposes. One may choose to read different problems independently, or different types of graph Laplacians independently, or graph connectivity conditions progressively.

Focus

The focus of this monograph is on linear dynamics, time-invariant, and directed graphs. Relevant work on nonlinear dynamics, time-varying, and undirected graphs is introduced in the section "Notes and References" at the end of each chapter.

Intended Audience

This monograph is written for applied science and engineering students in the graduate level or higher undergraduate levels, as a textbook or a reference for a relevant course. The monograph is also intended for researchers in systems control, robotics, signal processing, and computer engineering with interests in multi-agent systems, networked control, and cooperative behaviors.

Kai Cai and Zhiyun Lin
2022.04.05

Contents

Part I
Mathematical Preliminaries

This part introduces the basic concepts of directed graphs and their associated matrices. Three types of graph Laplacian matrices are defined, and their algebraic properties presented. These concepts and properties lay a theoretical foundation for the multi-agent cooperative control problems introduced later in the book.

Chapter 1

Graphs and Laplacian Matrices

We introduce basic elements of directed graphs, including nodes, edges, subgraphs, neighbors, and degrees. Then graph connectivity concepts that are key for distributed control problems are introduced; these concepts include strongly connectedness, strong components, spanning trees, and spanning multiple trees. We then introduce relevant matrices of directed graphs, including adjacency matrices, degree matrices, and Laplacian matrices. In particular, we define three types of Laplacian matrices and analyze their algebraic properties (eigenstructures and ranks). Key relations between these algebraic properties of graph matrices and graph connectivity conditions are established.

1.1 Directed graphs

A *directed graph* (or simply *digraph*) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a non-empty finite set $\mathcal{V}$ of elements called *nodes*, and a finite set $\mathcal{E}$ of ordered pairs of nodes called *edges*. Thus $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The set $\mathcal{V}$ is called the *node set* and $\mathcal{E}$ the *edge set* of digraph $\mathcal{G}$.

Three examples of digraphs are displayed in Fig. 1.1:

$$\begin{aligned}
\mathcal{G}_1 &= (\{v_1, v_2, v_3, v_4\}, \{(v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_2), (v_3, v_4), (v_4, v_1), (v_4, v_2)\}) \\
\mathcal{G}_2 &= (\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_1, v_3), (v_3, v_2)\}) \\
\mathcal{G}_3 &= (\{v_1, v_2, v_3\}, \{(v_1, v_1), (v_1, v_2), (v_1, v_3), (v_3, v_2)\})
\end{aligned}$$

For an edge (u, v) the first node u is its *tail* and the second node v is its *head*. The edge (u, v) is said to *leave* u and *enter* v. The head and tail of an edge are its *end-nodes*. A *loop* is an edge whose end-nodes are the same node. An edge is *multiple* if there is another edge with the same end-nodes. A digraph is *simple* if it has no loops or multiple edges.[1]

[1] In this book, unless otherwise specified, only simple digraphs are considered.

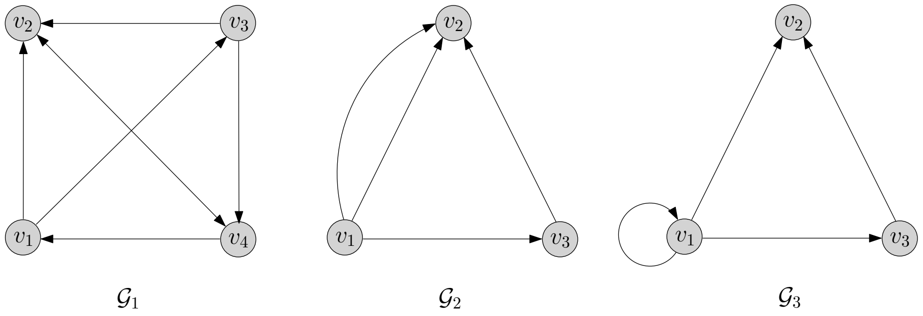


Figure 1.1: Directed graphs (digraphs)

For example, consider the digraphs in Fig. 1.1. Here, digraph $\mathcal{G}_1$ is simple; digraph $\mathcal{G}_2$ has multiple edges, namely (v_1, v_2); and digraph $\mathcal{G}_3$ has a loop, namely (v_1, v_1).

In the special case where for every edge $(u, v) \in \mathcal{E}$, the edge (v, u) of opposite direction satisfies $(v, u) \in \mathcal{E}$ as well, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called an *undirected* graph. Two examples are given in Fig. 1.2, where the edges are customly drawn without arrows.

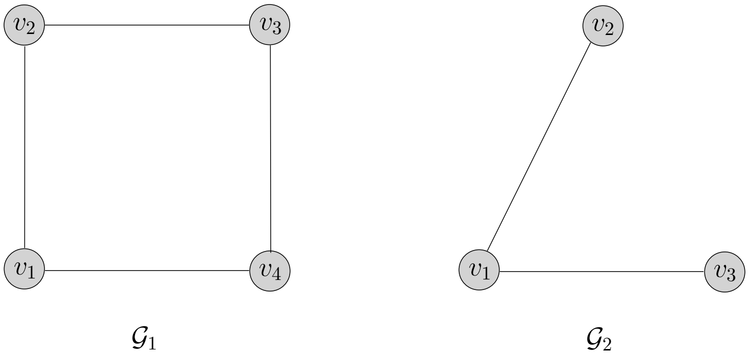
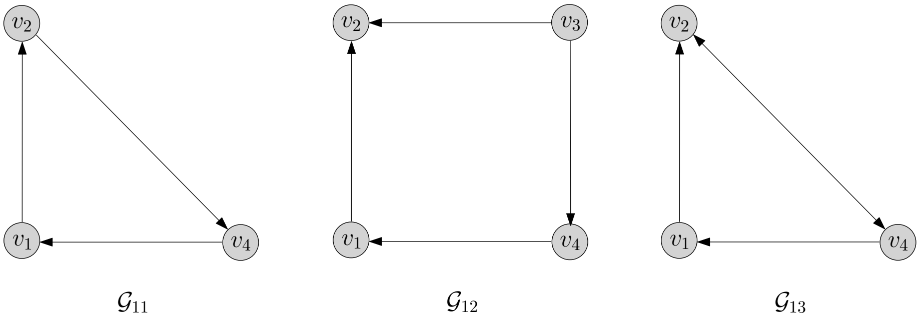


Figure 1.2: Undirected graphs

Subdigraphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph. $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is said to be a *subdigraph* of $\mathcal{G}$ if $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. If moreover $\mathcal{V}' = \mathcal{V}$, then $\mathcal{G}'$ is a *spanning subdigraph* of $\mathcal{G}$. For a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a non-empty subset $\mathcal{V}' \subseteq \mathcal{V}$, the *induced subdigraph* by $\mathcal{V}'$ is $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, with $\mathcal{E}' = \mathcal{E} \cap (\mathcal{V}' \times \mathcal{V}')$.

For example, consider the digraphs displayed in Fig. 1.3. Here $\mathcal{G}_{11}$, $\mathcal{G}_{12}$, and $\mathcal{G}_{13}$ are all subdigraphs of $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E})$ in Fig. 1.1. Only $\mathcal{G}_{12}$ is a spanning subdigraph, while only $\mathcal{G}_{13}$ is the induced subdigraph by $\mathcal{V}' = \{v_1, v_2, v_4\} \subseteq \mathcal{V}$. Note that $\mathcal{G}_{11}$ is not the induced subdigraph by $\mathcal{V}' = \{v_1, v_2, v_4\}$ because edge (v_4, v_2) is absent and $\mathcal{E}' \subsetneqq \mathcal{E} \cap (\mathcal{V}' \times \mathcal{V}')$.



$\mathcal{G}_{11}$ $\mathcal{G}_{12}$ $\mathcal{G}_{13}$

Figure 1.3: Subdigraphs

Neighbors and degrees

The local structure of a digraph is described by the neighbors and the degrees of its nodes. For a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a node $v \in \mathcal{V}$, the *neighbor set* of v is $\mathcal{N}_v := \{u \in \mathcal{V} \mid (u, v) \in \mathcal{E}\}$, while the *out-neighbor set* of v is $\mathcal{N}_v^o := \{u \in \mathcal{V} \mid (v, u) \in \mathcal{E}\}$. The nodes in $\mathcal{N}_v$ and $\mathcal{N}_v^o$ are respectively the *(in-)neighbors* and *out-neighbors* of v.

The *degree*, d_v, of a node v is the cardinality of the neighbor set $\mathcal{N}_v$, written $d_v = |\mathcal{N}_v|$. Similarly, the *out-degree*, d_v^o, of a node v is the cardinality of the out-neighbor set $\mathcal{N}_v^o$, i.e. $d_v^o = |\mathcal{N}_v^o|$.

A node v with $d_v = d_v^o$ is called *balanced*. A digraph $\mathcal{G}$ is *balanced* if every node is balanced. Every undirected graph is balanced.

As an illustration, consider the digraph $\mathcal{G}_1$ displayed in Fig. 1.1. For node v_1, its neighbor set is $\mathcal{N}_{v_1} = \{v_4\}$ and out-neighbor set $\mathcal{N}_{v_1}^o = \{v_2, v_3\}$; hence its degree is $d_{v_1} = 1$ and out-degree $d_{v_1}^o = 2$. As a result, v_1 is not balanced. Next consider the digraph $\mathcal{G}_{11}$ in Fig. 1.3. Observe that every node has degree 1 and out-degree 1, so every node is balanced and digraph $\mathcal{G}_{11}$ is balanced.

1.2 Connectivity of digraphs

A (directed) *path* in a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a sequence of nodes

$$v_1 v_2 \cdots v_k \quad (k \geq 1)$$

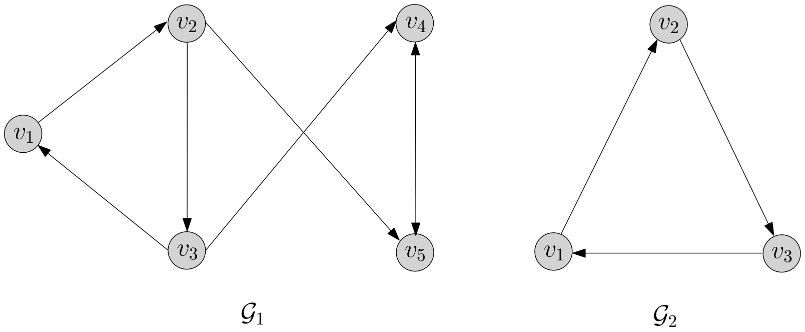
such that $(v_i, v_{i+1}) \in \mathcal{E}$ for every $i = 1, 2, \ldots, k-1$. The path is said to be *from* v_1 *to* v_k. If $v_1 = v_k$, the path is called a *cycle*. The *length* of a path is the number of the consisting edges. Hence the path above has length $k-1$. It is allowed that $k = 1$, in which case the path is of length 0. Also note that a loop is a cycle of length 1.

Let $u, v \in \mathcal{V}$ be two nodes of $\mathcal{G}$. We say that v is *reachable* from u if there is a path from u to v; written $u \to v$. If v is *not* reachable from u, we write $u \not\to v$. Every node v is reachable from itself, i.e. $v \to v$, by the (trivial) path of length 0. For any node v, the set of nodes reachable from v is $\mathcal{V}(v^{\rightarrow}) = \{v' \in \mathcal{V} \mid v \to v'\}$, while the set of nodes from which v is reachable is $\mathcal{V}({}^{\rightarrow}v) = \{v' \in \mathcal{V} \mid v' \to v\}$. We call $\mathcal{V}(v^{\rightarrow})$ the *reachable set* of v, and $\mathcal{V}({}^{\rightarrow}v)$ the *backward reachable set* of v. Both $\mathcal{V}(v^{\rightarrow})$ and $\mathcal{V}({}^{\rightarrow}v)$ are nonempty, because v belongs to both.

A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *strongly connected* if

$$(\forall u, v \in \mathcal{V}) u \to v$$

namely every node is reachable from every other node. In this case, $\mathcal{V}(v^{\rightarrow}) = \mathcal{V}({}^{\rightarrow}v) = \mathcal{V}$ for every node $v \in \mathcal{V}$.



$\mathcal{G}_1$ $\mathcal{G}_2$

Figure 1.4: Reachability and strongly connected digraphs

For example, consider digraph $\mathcal{G}_1$ in Fig. 1.4. Although for $i = 1, 2, 3$ there holds $\mathcal{V}(v_i^{\rightarrow}) = \mathcal{V}({}^{\rightarrow}v_i) = \mathcal{V}$, for $i = 4, 5$ only $\mathcal{V}(v_i^{\rightarrow}) = \{v_4, v_5\} \subsetneqq \mathcal{V}$. The latter means that nodes v_4, v_5 cannot reach v_1, v_2, v_3. Hence $\mathcal{G}_1$ is not strongly connected. By contrast, $\mathcal{G}_2$ is strongly connected: $\mathcal{V}(v_i^{\rightarrow}) = \mathcal{V}(^{\rightarrow}v_i) = \mathcal{V}$ for all $i = 1, 2, 3$.

A strongly connected digraph $\mathcal{G}$ contains at least one cycle. Given a strongly connected digraph $\mathcal{G}$ containing $m(\geq 1)$ cycles, let $l_1, \dots, l_m$ be the lengths of all these cycles and denote by p their greatest common divisor, i.e.

$$p := \text{g.c.d.}\{l_1, \dots, l_m\}.$$

If $p > 1$, we say that $\mathcal{G}$ is *periodic* with period p. Otherwise ($p = 1$), we say that $\mathcal{G}$ is *aperiodic*. Note that a strongly connected digraph with a loop is aperiodic (as in this case $p = 1$).

In a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $r \in \mathcal{V}$ is called a *root* if

$$(\forall v \in \mathcal{V}) r \rightarrow v$$

that is, every node is reachable from r (equivalently $\mathcal{V}(r^{\rightarrow}) = \mathcal{V}$). Note that in a strongly connected digraph $\mathcal{G}$, every node is a root.

Let r be a root of digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A spanning subdigraph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ is called a *spanning tree (with root r)* if

- r has no neighbor, i.e. $\mathcal{N}_r = \emptyset$;
- every node $v \in \mathcal{V} \setminus \{r\}$ has exactly one neighbor, i.e. $d_v = 1$.

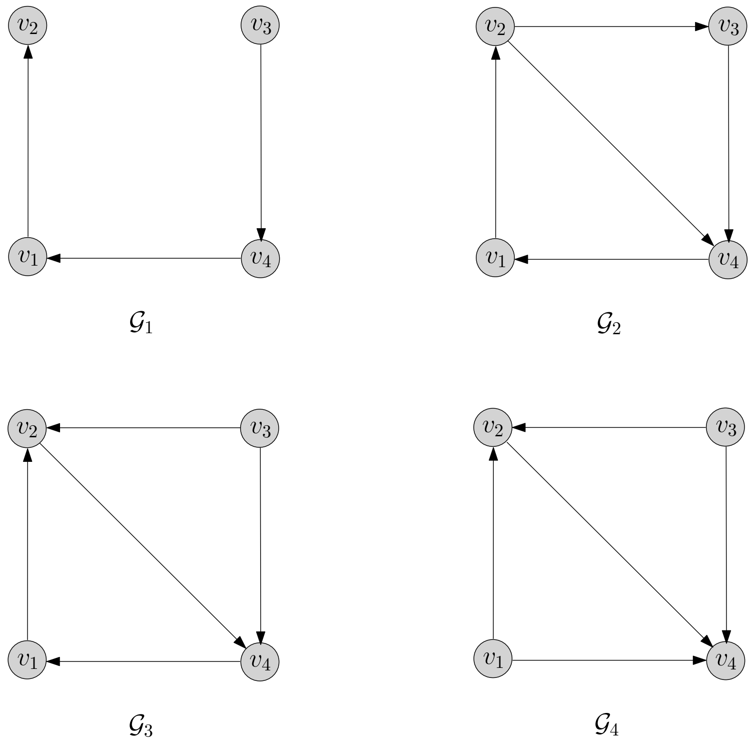
Definition 1.1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph. We say that $\mathcal{G}$ contains a spanning tree if there exists a spanning subdigraph of $\mathcal{G}$ that is a spanning tree.*

Consider the digraphs displayed in Fig. 1.5. Digraph $\mathcal{G}_1$ is a spanning tree with root v_3. $\mathcal{G}_2$ is strongly connected, and (so) it contains a spanning tree (say $\mathcal{G}_1$). $\mathcal{G}_3$ is not strongly connected, but contains a spanning tree ($\mathcal{G}_1$). Finally $\mathcal{G}_4$ neither is strongly connected nor contains a spanning tree.

Note that if $\mathcal{G}$ is strongly connected, then $\mathcal{G}$ contains a spanning tree; but the reverse need not hold. Nevertheless whether or not $\mathcal{G}$ contains a spanning tree may be verified by inspecting its strongly connected subdigraphs.

Strong components

Let $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ be a subdigraph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\emptyset \neq \mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' = \mathcal{E} \cap (\mathcal{V}' \times \mathcal{V}')$. Namely $\mathcal{G}'$ is an induced subdigraph of $\mathcal{G}$ by $\mathcal{V}'$. We say that $\mathcal{G}'$ is a *strong component* of $\mathcal{G}$ if $\mathcal{G}'$



$\mathcal{G}_1$ $\mathcal{G}_2$ $\mathcal{G}_3$ $\mathcal{G}_4$

Figure 1.5: Strongly connected digraphs and spanning trees

is strongly connected and for every other induced subdigraph $\mathcal{G}'' = (\mathcal{V}'', \mathcal{E}'')$ with $\mathcal{V}' \subseteq \mathcal{V}''$ and $\mathcal{E}' \subseteq \mathcal{E}''$, $\mathcal{G}''$ is not strongly connected. In other words, $\mathcal{G}'$ is a *maximal* strongly connected induced subdigraph of $\mathcal{G}$ (which need not be unique). Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be two strong components of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Then they are either identical (i.e. $\mathcal{V}_1 = \mathcal{V}_2, \mathcal{E}_1 = \mathcal{E}_2$) or disjoint (i.e. $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset, \mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$).

A strong component $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is said to be *closed* if

$$(\forall u \in \mathcal{V}')(\forall v \in \mathcal{V} \setminus \mathcal{V}')v \not\to u$$

namely no edge enters any node in $\mathcal{V}'$. In this case, $\mathcal{V}' = \mathcal{V}({}^{\to}u) \subseteq \mathcal{V}(u^{\to})$ for every node $u \in \mathcal{V}'$.

Fig. 1.6 provides examples of induced subdigraphs, $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$, of the first digraph $\mathcal{G}$, where $\mathcal{G}_1$ is not a strong component, $\mathcal{G}_2$ is a closed strong component, and $\mathcal{G}_3$ is a strong component but not closed.

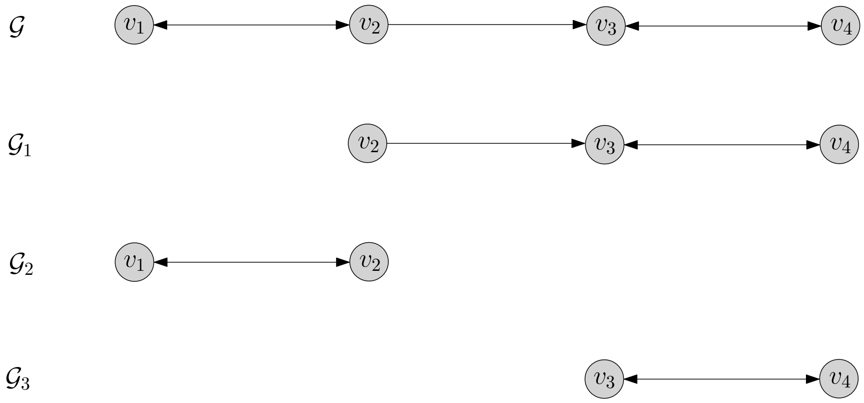


Figure 1.6: Strong components and closed strong components

Theorem 1.1 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph. The following are equivalent:*

(i) *$\mathcal{G}$ contains a spanning tree;*

(ii) *$\mathcal{G}$ contains a unique closed strong component.*

Proof. (i) ⇒ (ii). Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning tree. Let $\mathcal{V}_r$ be the subset of all roots, i.e.

$$\mathcal{V}_r := \{r \in \mathcal{V} \mid \mathcal{V}(r^{\rightarrow}) = \mathcal{V}\}.$$

Thus $\mathcal{V}_r \neq \emptyset$. Let $\mathcal{G}_r$ be the induced subdigraph by $\mathcal{V}_r$. It will be shown that $\mathcal{G}_r$ is the unique closed strong component of $\mathcal{G}$.

If $\mathcal{V}_r = \mathcal{V}$, namely every node is a root, then $\mathcal{G}_r = \mathcal{G}$ is strongly connected, and maximality, closedness, and uniqueness follow trivially.

If $\mathcal{V}_r \subsetneqq \mathcal{V}$ (i.e. $\mathcal{V}_r$ is a strict subset of $\mathcal{V}$), first note that $\mathcal{G}_r$ is closed. To see this, suppose on the contrary that there exist $r \in \mathcal{V}_r$ and $v \in \mathcal{V} \setminus \mathcal{V}_r$ such that $v \to r$. Since r is a root, v is also a root, but this contradicts $v \notin \mathcal{V}_r$. Next, note that $\mathcal{G}_r$ is strongly connected. This follows from the fact that every node in $\mathcal{V}_r$ is a root and $\mathcal{G}_r$ is closed. Moreover, no node in $\mathcal{V} \setminus \mathcal{V}_r$ (i.e. non-root) can be added to $\mathcal{V}_r$ while preserving strongly connectedness, so $\mathcal{G}_r$ is a strong component of $\mathcal{G}$. Finally, we prove that $\mathcal{G}_r$ is unique. Let $\mathcal{G}'_r = (\mathcal{V}'_r, \mathcal{E}'_r)$ be another closed strong component of $\mathcal{G}$. Then either $\mathcal{V}'_r \cap \mathcal{V}_r = \emptyset$ or $\mathcal{V}'_r = \mathcal{V}_r$. Since all nodes $\mathcal{V}_r$ are roots, they can reach all nodes in $\mathcal{V}'_r$, but this contradicts closedness of $\mathcal{G}'_r$. Hence, it is only possible that $\mathcal{V}'_r = \mathcal{V}_r$, and $\mathcal{G}'_r = \mathcal{G}_r$ after all. This establishes that $\mathcal{G}_r$ is the unique closed strong component of $\mathcal{G}$.

(ii) $\Rightarrow$ (i). Suppose that $\mathcal{G}$ contains a unique closed strong component $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$. We will prove that $\mathcal{G}$ contains a spanning tree by showing that every node in $\mathcal{V}_r$ is a root.

Suppose on the contrary that there is a node $r \in \mathcal{V}_r$ such that it is not a root. Then $\mathcal{V}(r^{\rightarrow}) \subsetneqq \mathcal{V}$. Let $\mathcal{U} := \mathcal{V} \setminus \mathcal{V}(r^{\rightarrow})$; thus $\mathcal{U} \neq \emptyset$. Note that no node in $\mathcal{V}(r^{\rightarrow})$ can reach any node in $\mathcal{U}$, because otherwise r could also reach some node in $\mathcal{U}$. Hence the induced subdigraph $\mathcal{G}_u$ by $\mathcal{U}$ is closed. In the following, it will be shown that $\mathcal{G}_u$ contains at least one closed strong component.

Select an arbitrary node $u_1 \in \mathcal{U}$, and check if $\mathcal{V}(^{\rightarrow}u_1) \subseteq \mathcal{V}(u_1^{\rightarrow})$. If so, it follows that the induced subdigraph $\mathcal{G}_1$ by $\mathcal{V}(^{\rightarrow}u_1)$ is a closed strong component of $\mathcal{G}_u$. If the condition fails, then select another arbitrary node $u_2 \in \mathcal{V} \setminus \mathcal{V}(^{\rightarrow}u_1)$, and check if $\mathcal{V}(^{\rightarrow}u_2) \subseteq \mathcal{V}(u_2^{\rightarrow})$. Note that here $\mathcal{V}(^{\rightarrow}u_2) \subseteq \mathcal{V} \setminus \mathcal{V}(^{\rightarrow}u_1)$ necessarily holds, for otherwise u_1 could be reached from u_2. If the condition holds, then the induced subdigraph $\mathcal{G}_2$ by $\mathcal{V}(^{\rightarrow}u_2)$ is a closed strong component of $\mathcal{G}_u$. If not, repeat the above procedure. Since the node set $\mathcal{U}$ is finite, in the worst case after (say) k repetitions and check failures, the subset $\mathcal{V}(^{\rightarrow}u_{k+1}) \subseteq \mathcal{V} \setminus \mathcal{V}(^{\rightarrow}u_1) \setminus \cdots \setminus \mathcal{V}(^{\rightarrow}u_k)$ contains a singleton node u_{k+1}. Since $\mathcal{V}(^{\rightarrow}u_{k+1}) \subseteq \mathcal{V}(u_{k+1}^{\rightarrow})$ holds trivially, the induced subdigraph $\mathcal{G}_{k+1}$ by $\mathcal{V}(^{\rightarrow}u_{k+1})$ is a closed strong component of $\mathcal{G}_u$.

We have thus proved that $\mathcal{G}_u$ contains a closed strong component, say $\mathcal{G}'_u$. Since $\mathcal{G}_u$ is closed in $\mathcal{G}$, $\mathcal{G}'_u$ is also a closed strong component of $\mathcal{G}$. But $\mathcal{G}'_u$ is different from $\mathcal{G}_r$, which is a contradiction to the assumed unique strong component of $\mathcal{G}$. Therefore, every node in $\mathcal{V}_r$ is a root and $\mathcal{G}$ contains at least one spanning tree. □

To illustrate Theorem 1.1, consider the digraphs in Fig. 1.4. $\mathcal{G}_1$ contains two strong components, but only the one induced by $\{v_1, v_2, v_3\}$ is closed. Hence $\mathcal{G}_1$ contains a spanning tree with root (say) v_1. $\mathcal{G}_2$ contains only one strong component, namely itself, which is (trivially) closed. So again $\mathcal{G}_2$ contains a spanning tree with root (say) v_3. On the other hand, consider digraph $\mathcal{G}_4$ in Fig. 1.5. We have identified that $\mathcal{G}_4$ does not contain a spanning tree. Indeed, this digraph contains 4 strong components, two of which are closed: one induced by $\{v_1\}$ and the other by $\{v_3\}$.

Spanning multiple trees

Let us now generalize the concept of spanning trees to allow multiple roots.

Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{R} \subseteq \mathcal{V}$ be a subset of nodes, and $k := |\mathcal{R}|$. Consider $k \geq 2$, i.e. $\mathcal{R}$ contains at least two nodes. Let $v \in \mathcal{V} \setminus \mathcal{R}$. We say that v is *k-reachable* from $\mathcal{R}$ if there is a path from a node in $\mathcal{R}$ to v after removing arbitrary $k-1$ nodes except for v itself; written $\mathcal{R} \rightarrow_k v$. More formally, $\mathcal{R} \rightarrow_k v$ if

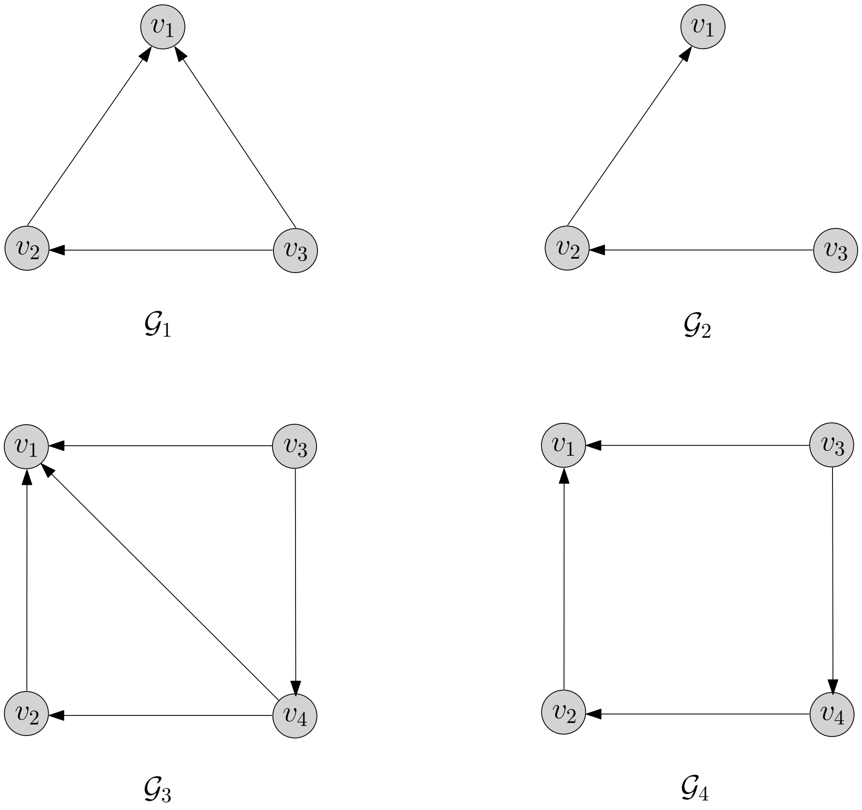
$$(\forall \mathcal{U} \subseteq \mathcal{V} \setminus \{v\}) |\mathcal{U}| = k-1 \Rightarrow (\exists r \in \mathcal{R} \cap (\mathcal{V} \setminus \mathcal{U})) r \rightarrow v \text{ in } \mathcal{G}' \text{ induced by } \mathcal{V} \setminus \mathcal{U}.$$

If v is *not* k-reachable from $\mathcal{R}$, we write $\mathcal{R} \not\to_k v$.

The subset $\mathcal{R}$ of $k(\geq 2)$ nodes is called a *k-root subset* if

$$(\forall v \in \mathcal{V} \setminus \mathcal{R})\mathcal{R} \to_k v$$

that is, every node (not in $\mathcal{R}$) is k-reachable from $\mathcal{R}$. Note that in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if $\mathcal{R}$ is a k-root subset, then for every $r \in \mathcal{R}$, $\mathcal{R} \setminus \{r\}$ is a $(k-1)$-root subset in the induced subgraph by $\mathcal{V} \setminus \{r\}$. In the special case $k = 2$, i.e. $\mathcal{R} = \{r_1, r_2\}$, r_1 (resp. r_2) is a root of the induced subgraph by $\mathcal{V} \setminus \{r_2\}$ (resp. by $\mathcal{V} \setminus \{r_1\}$).



$\mathcal{G}_1$ $\mathcal{G}_2$

$\mathcal{G}_3$ $\mathcal{G}_4$

Figure 1.7: k-reachability

Consider the digraphs in Fig. 1.7. In $\mathcal{G}_1$, v_1 is 2-reachable from $\{v_2, v_3\}$, and $\{v_2, v_3\}$ is a 2-root set. By contrast, in $\mathcal{G}_2$, v_1 is not 2-reachable from $\{v_2, v_3\}$, because after removing v_2, v_1 is no longer reachable from v_3. Similarly, in $\mathcal{G}_3$, v_1 is 3-reachable from $\{v_2, v_3, v_4\}$, and

$\{v_2, v_3, v_4\}$ is a 3-root set. But in $\mathcal{G}_4$, v_1 is not 3-reachable, because after removing v_2 and v_3, v_1 is not reachable from v_4. Finally, removing v_2 in $\mathcal{G}_1$, v_3 is a root of the induced subgraph by $\{v_1, v_3\}$; also removing v_4 in $\mathcal{G}_3$, $\{v_2, v_3\}$ is a 2-root subset of the induced subgraph by $\{v_1, v_2, v_3\}$.

Let $\mathcal{R}$ be a k-root subset of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A spanning subdigraph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ is called a *spanning k-tree (with k-root subset $\mathcal{R}$)* if

- every root $r \in \mathcal{R}$ has no neighbor, i.e. $\mathcal{N}_r = \emptyset$;
- every node $v \in \mathcal{V} \setminus \mathcal{R}$ has exactly k neighbors, i.e. $d_v = k$.

Definition 1.2 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph and $k \geq 2$. We say that $\mathcal{G}$ contains a spanning k-tree if there exists a spanning subdigraph of $\mathcal{G}$ that is a spanning k-tree.*

As an illustration, $\mathcal{G}_1$ in Fig. 1.7 contains a spanning 2-tree $\mathcal{G}_1'$, which is displayed in Fig. 1.8. For another example, $\mathcal{G}_3$ in Fig. 1.7 contains a spanning 3-tree $\mathcal{G}_2'$ in Fig. 1.8.

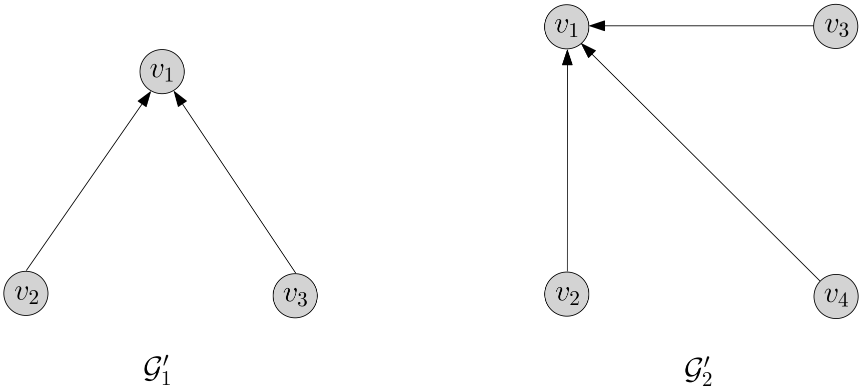


Figure 1.8: Spanning k-tree

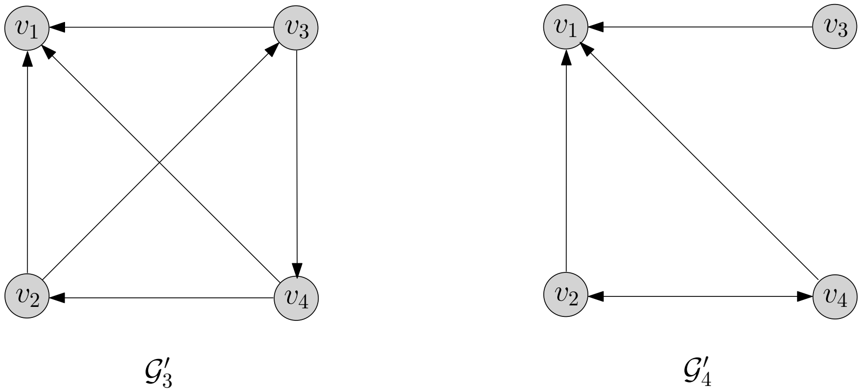
A counterpart of Theorem 1.1 is the following, which establishes the relation between $\mathcal{G}$ containing a spanning k-tree and the number of closed strong components.

Theorem 1.2 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph and $k \geq 2$. If $\mathcal{G}$ contains a spanning k-tree, then $\mathcal{G}$ contains $l \in [1, k]$ closed strong components.*

Proof. Suppose on the contrary that $\mathcal{G}$ contains $k+1$ closed strong components: $\mathcal{G}_1, \dots, \mathcal{G}_k, \mathcal{G}_{k+1}$. It will be shown that there cannot exist a k-root subset, and consequently $\mathcal{G}$ does not contain a spanning k-tree.

Consider an arbitrary subset $\mathcal{V}'$ of k nodes in $\mathcal{G}$. Then there exists a closed strong component $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ ($i \in [1, k+1]$) such that $\mathcal{V}' \cap \mathcal{V}_i = \emptyset$. Namely $\mathcal{G}_i$ does not contain any node in $\mathcal{V}'$. Now choose a node v_i in $\mathcal{G}_i$, so $v_i \in \mathcal{V}_i$ and $v_i \notin \mathcal{V}'$. Then remove $k-1$ nodes from the other k closed strong components ($\mathcal{G}_i$ excluded). Since $\mathcal{G}_i$ is closed, the chosen node v_i cannot be reached from the subset $\mathcal{V}'$. This by definition means that $\mathcal{V}'$ is not a k-root subset. Since $\mathcal{V}'$ is arbitrary, we conclude that there cannot exist a k-root subset in $\mathcal{G}$. This completes the proof. □

To illustrate Theorem 1.2, first consider $k = 2$. Both $\mathcal{G}_1$ in Fig. 1.7 and $\mathcal{G}'_1$ in Fig. 1.8 contain a spanning 2-tree. While $\mathcal{G}_1$ contains 1 closed strong component (induced by $\{v_3\}$), $\mathcal{G}'_1$ contains 2 closed strong components (induced respectively by $\{v_2\}$ and $\{v_3\}$). Next consider $k = 3$. The digraphs in Fig. 1.9 contain a spanning 3-tree. $\mathcal{G}'_3$ has 1 closed strong component (induced by $\{v_2, v_3, v_4\}$), while $\mathcal{G}'_4$ has 2 closed strong components (induced respectively by $\{v_2, v_4\}$ and $\{v_3\}$). In addition, the spanning 3-tree $\mathcal{G}'_2$ in Fig. 1.8 has 3 closed strong components (induced respectively by $\{v_2\}$, $\{v_3\}$, and $\{v_4\}$).



$\mathcal{G}'_3$

$\mathcal{G}'_4$

Figure 1.9: Number of closed strong components in digraphs containing a spanning multiple tree

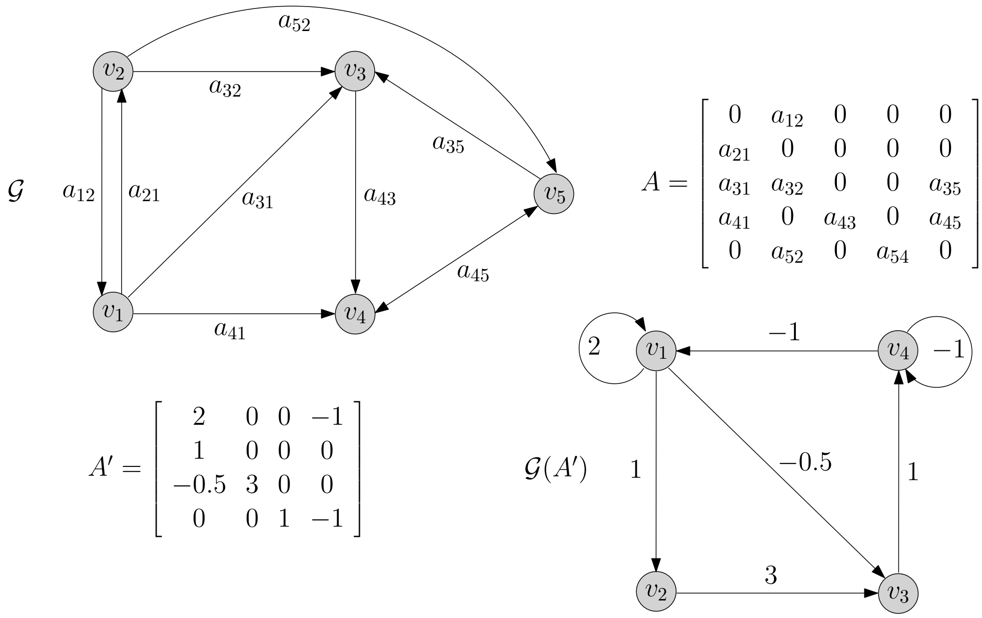
1.3 Matrices of digraphs

Given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \dots, v_n\}$, we may assign each edge $(v_j, v_i) \in \mathcal{E}$ a *weight* a_{ij}. Otherwise the pair $(v_j, v_i) \notin \mathcal{E}$ is associated with $a_{ij} = 0$. The weight a_{ij} may be a positive real number, or any real number, or even a complex number. With weights assigned, the digraph

$\mathcal{G}$ is called a *weighted digraph*.

The *adjacency matrix* of a weighted digraph $\mathcal{G}$ is an $n \times n$ matrix $A = (a_{ij})$. Depending on the field where a_{ij} belongs, A may be a nonnegative matrix (entry-wise nonnegative) if $a_{ij} > 0$, an arbitrary real matrix if $a_{ij} \in \mathbb{R}$, or a complex matrix if $a_{ij} \in \mathbb{C}$. In the special case of undirected weighted graphs, the adjacency matrix A is symmetric, i.e. $A = A^\top$.

Conversely for a given $n \times n$ matrix $A = (a_{ij})$, we may construct a weighted digraph $\mathcal{G}(A)$ of n nodes such that an edge (v_j, v_i) exists with weight a_{ij} if and only if $a_{ij} \neq 0$.



$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & a_{35} \\ a_{41} & 0 & a_{43} & 0 & a_{45} \\ 0 & a_{52} & 0 & a_{54} & 0 \end{bmatrix}$$

$$A' = \begin{bmatrix} 2 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ -0.5 & 3 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

Figure 1.10: Adjacency matrices

Illustration of adjacency matrices is provided in Fig. 1.10. Given a weighted digraph $\mathcal{G}$ of five nodes, its adjacency matrix A is a 5×5 matrix with each entry a_{ij} the weight on edge (v_j, v_i). Conversely for a given 4×4 matrix A', its corresponding digraph $\mathcal{G}(A')$ has four nodes, and an edge (v_j, v_i) with weight a_{ij} exists whenever $a_{ij} \neq 0$. Note that the two loops in $\mathcal{G}(A')$ are due to the nonzero diagonal entries a_{11} and a_{44}.

We write $A \geq 0$ if A is a nonnegative matrix, and $A > 0$ if A is a positive matrix (entry-wise positive). The same notation is used for nonnegative and positive vectors (which are special one-column matrices).

When the adjacency matrix A is a nonnegative matrix (i.e. $A \geq 0$), there are several important properties concerning its spectrum (the Perron-Frobenius Theorem) that we shall introduce in the sequel. To this end, we introduce two types of nonnegative matrices in order: irreducible matrices and primitive matrices.

A square matrix P is a *permutation matrix* if for each row and each column, there is exactly one entry equal to 1. That is, the columns of a permutation matrix are a reordering of the standard basis vectors. Indeed, if P is a permutation matrix and M an arbitrary matrix, then the operation $M \mapsto PM$ amounts to reordering the rows of M; further $PM \mapsto PMP^\top$ amounts to doing the same reordering of the columns of PM. A permutation matrix P is *orthogonal*: $P^\top P = PP^\top = I$.

Let $A \in \mathbb{R}^{n\times n}$ be a nonnegative matrix, i.e. $A \geq 0$. We say that A is *reducible* if either $n = 1$ and $A = 0$, or there exists a permutation matrix P such that $PAP^\top$ is block upper triangular, as follows:

$$\begin{bmatrix} B & C \\ 0 & D \end{bmatrix}$$

where B and D are square matrices. Otherwise A is *irreducible*.

For example, consider two nonnegative matrices

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 4 & 5 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 4 & 5 & 0 \end{bmatrix}.$$

A_1 is reducible because there exists the following permutation matrix

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{such that} \quad PA_1P^\top = \left[\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 3 \\ 0 & 4 & 0 & 5 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

On the other hand, A_2 is irreducible: no permutation matrix P can render $PA_2P^\top$ in the block upper triangular form.

Irreducibility of matrices is elegantly characterized by connectivity of digraphs.

Theorem 1.3 *Let $\mathcal{G}$ be a weighted digraph with n nodes and $A \geq 0$ the corresponding nonnegative adjacency matrix. Then A is irreducible if and only if $\mathcal{G}$ is strongly connected.*

For the example A_1, A_2 above, they are respectively the nonnegative adjacency matrices of digraphs $\mathcal{G}_1$ and $\mathcal{G}_2$ in Fig. 1.11. A_1 is reducible and digraph $\mathcal{G}_1$ is not strongly connected; whereas A_2 is irreducible and digraph $\mathcal{G}_2$ is strongly connected.

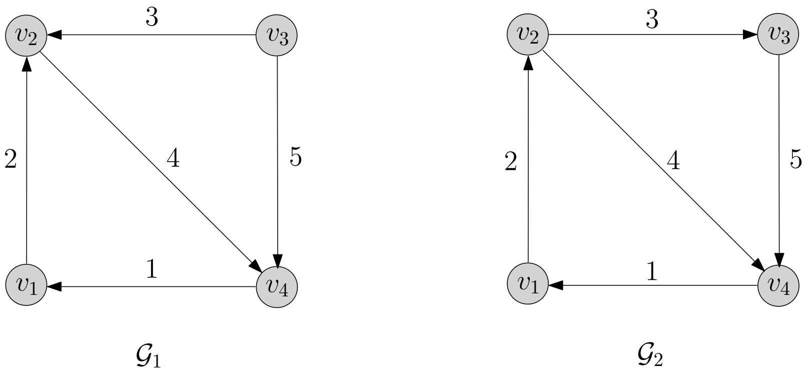


Figure 1.11: Irreducibility of nonnegative matrices characterized by graph connectivity

To prove Theorem 1.3, the following lemma is useful, which establishes a link between positivity of entries in adjacency matrix powers and reachability of the corresponding nodes. For an arbitrary positive integer $k \geq 1$, denote by a_{ij}^k the (i, j)-entry of the matrix A^k.

Lemma 1.1 *Let $\mathcal{G}$ be a weighted digraph with n nodes and $A \geq 0$ the corresponding nonnegative adjacency matrix. Then for every $i, j \in \{1, \ldots, n\}$ and every positive integer $k \geq 1$, $a_{ij}^k > 0$ if and only if there exists a path of length k from node v_j to node v_i.*

Proof. The proof is by induction on $k \geq 1$. For the base case where $k = 1$, the assertion holds by the definition of nonnegative adjacency matrix A. Namely, $a_{ij} > 0$ if and only if there is an edge $(v_j, v_i) \in \mathcal{E}$ (i.e. path of length 1 from v_j to v_i).

For the induction step, suppose that the assertion holds for $k - 1$. Note from $A^k = A^{k-1}A$ that

$$a_{ij}^k = \sum_{m=1}^{n} a_{im}^{k-1} a_{mj}.$$

Thus $a_{ij}^k > 0$ if and only if there is $m \in \{1, \ldots, n\}$ such that $a_{im}^{k-1} > 0$ and $a_{mj} > 0$. That is, there exist a path of length $k - 1$ from node v_m to v_i and a path of length 1 from v_j to v_m. These two paths constitute a path of length k from v_j to v_i. This finishes the induction step, and thereby establishes the assertion for any positive integer $k \geq 1$. □

Proof of Theorem 1.3 (If) Suppose on the contrary that A is reducible. By definition, there is a permutation matrix P such that

$$PAP^\top = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix} =: \tilde{A}.$$

Then the matrix $I + \tilde{A}$ is also block upper triangular, and so is its $n-1$ powers $(I+\tilde{A})^{n-1}$. Consequently $(I+\tilde{A})^{n-1}$ is not a positive matrix. Note that

$$(I+\tilde{A})^{n-1} = P(I+A)^{n-1}P^\top$$

so neither is $(I+A)^{n-1}$ positive. Since in general

$$(I+A)^{n-1} = I + c_1 A + c_2 A^2 + \cdots + c_{n-1}A^{n-1}$$

and the combinatorial coefficients $c_1, \ldots, c_{n-1}$ are all positive, there exist $i, j \in \{1, \ldots, n\}$ $(i \neq j)$ such that for every $k \in \{1, \ldots, n-1\}$ it holds that $a_{ij}^k = 0$. But this means (by Lemma 1.1) that there is no path of any length $k \in \{1, \ldots, n-1\}$ from node v_j to node v_i. Namely $v_j \not\to v_i$; hence digraph $\mathcal{G}$ is not strongly connected.

(Only if) Suppose on the contrary that $\mathcal{G}$ is not strongly connected. By definition, there exist two nodes v_i, v_j such that $v_j \not\to v_i$. Thus the set of nodes that cannot reach v_i is nonempty, i.e. $\mathcal{V} \setminus \mathcal{V}(^{\to}v_i) \neq \emptyset$ (v_j belongs). In fact, there does not exist any path from any node in $\mathcal{V} \setminus \mathcal{V}(^{\to}v_i)$ to any node in $\mathcal{V}(^{\to}v_i)$. To see this, suppose that there exist $v_l \in \mathcal{V} \setminus \mathcal{V}(^{\to}v_i)$ and $v_m \in \mathcal{V}(^{\to}v_i)$ such that $v_l \to v_m$. Since $v_m \to v_i$, we have $v_l \to v_i$, but this contradicts $v_l \notin \mathcal{V}(^{\to}v_i)$. By this fact, we reorder the nodes according to the partition of the node set: $\{\mathcal{V} \setminus \mathcal{V}(^{\to}v_i), \mathcal{V}(^{\to}v_i)\}$. The reordering amounts to a permutation of the indices of nodes, and correspondingly there is a permutation matrix P such that

$$PAP^\top = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}$$

But this means that A is reducible. □

Next we introduce primitive matrices. Let $A \in \mathbb{R}^{n \times n}$ be a nonnegative matrix, i.e. $A \geq 0$. We say that A is *primitive* if

$$(\exists k \geq 1) A^k > 0.$$

A primitive matrix is irreducible, but the converse need not hold. This is evident from the following graphical characterization of primitive matrices, as compared to that of irreducible matrices in Theorem 1.3.

Theorem 1.4 *An $n \times n$ nonnegative matrix A is primitive if and only if $\mathcal{G}(A)$ is strongly connected and aperiodic.*

Consider again the matrix A_2 which is the adjacency matrix of digraph $\mathcal{G}_2$ in Fig. 1.11. We have analyzed that A_2 is irreducible, as $\mathcal{G}_2$ is strongly connected. Moreover $\mathcal{G}_2$ is aperiodic: there are two cycles in $\mathcal{G}_2$ of length 3 and 4, respectively; hence $p = \text{g.c.d.}\{3, 4\} = 1$. By Theorem 1.4, A_2 is primitive. Indeed, it is checked that A_2^{10} is a positive matrix.
Let us consider two more matrices

$$A_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 4 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \end{bmatrix}.$$

First, A_3 is not primitive because digraph $\mathcal{G}(A_3)$ in Fig. 1.12 is not aperiodic. Indeed $\mathcal{G}(A_3)$ is a strongly connected digraph of period 4. Hence A_3 is irreducible but not primitive. On the other hand, A_4 is the same as A_3 except for the positive $(1, 1)$ entry. This diagonal entry is crucial, however, since digraph $\mathcal{G}(A_4)$ in Fig. 1.12 is aperiodic due to the loop at v_1. Therefore A_4 is primitive (in fact $A_4^6 > 0$).

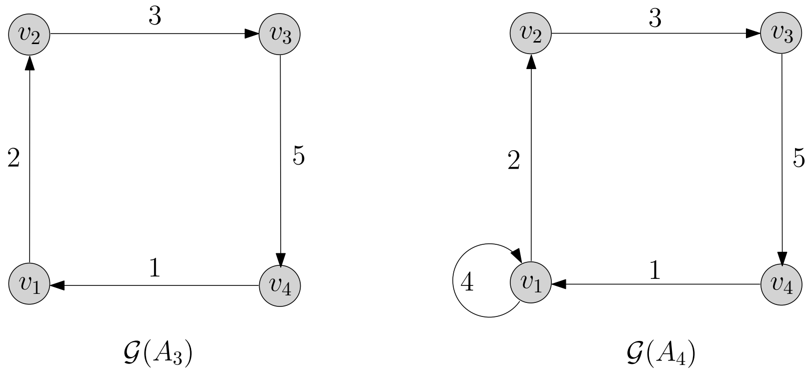


Figure 1.12: Primitivity of nonnegative matrices characterized by graph connectivity

The proof of Theorem 1.4 requires the following lemmas.

Lemma 1.2 *Let $m_1, m_2 \geq 1$ be two positive integers. If* g.c.d.$\{m_1, m_2\} = 1$*, then there is an integer $\bar{k} \geq 0$ such that for any integer $k \geq \bar{k}$,*

$$k = \alpha m_1 + \beta m_2$$

for some nonnegative integers α, β.

Proof. Since

$$\text{g.c.d.}\{m_1, m_2\} = 1,$$

1 is an integer combination of m_1 and m_2, i.e.

$$1 = \alpha_1 m_1 - \beta_1 m_2$$

for some nonnegative integers α_1, β_1. Let $\bar{k} := \beta_1 m_2^2$. Thus $\bar{k} \geq 0$ and for all $k \geq \bar{k}$,

$$k = \beta_1 m_2^2 + i \cdot m_2 + j$$

for some integers i, j satisfying $i \geq 0$ and $0 \leq j < m_2$. Substituting $1 = \alpha_1 m_1 - \beta_1 m_2$ into the above equation yields

$$\begin{aligned} k &= \beta_1 m_2^2 + i \cdot m_2 + j \cdot (\alpha_1 m_1 - \beta_1 m_2) \\ &= (j \cdot \alpha_1) \cdot m_1 + (\beta_1 (m_2 - j) + i) \cdot m_2. \end{aligned}$$

Let

$$\alpha := j \cdot \alpha_1 \text{ and } \beta := \beta_1 (m_2 - j) + i.$$

Then α, β are nonnegative integers due to $j < m_2$. Therefore, the conclusion follows. □

The next result shows the relationship between the period of a strongly connected digraph and the period of each node in the digraph. For an arbitrary node v in a strongly connected digraph $\mathcal{G}$, let $l_{v,1}, \ldots, l_{v,m}$ be the lengths of all $m (\geq 1)$ cycles from v to v. Denote by p_v their greatest common divisor, i.e.

$$p_v := \text{g.c.d.}\{l_{v,1}, \ldots, l_{v,m}\}$$

and we say that p_v is the period of node v.

Lemma 1.3 *Consider a strongly connected digraph $\mathcal{G}$. Let p be the period of a digraph $\mathcal{G}$ and p_i be the period of node v_i, $i \in \{1, \ldots, n\}$. Then $p = p_1 = \cdots = p_n$.*

Proof. Let $i \in \{1, \dots, n\}$. We will establish $p = p_i$ by showing that p divides p_i and p_i divides p.

First let $\mathcal{L} := \{l_1, \dots, l_k\}$ be the set of all the lengths of all $k(\geq 1)$ cycles in digraph $\mathcal{G}$. Then by definition, p is the greatest common divisor of the elements in $\mathcal{L}$. Note that for every path from v_i to v_i, it is either a (simple) cycle or consists of a number of cycles. So the length l_{v_i} of any path from v_i to v_i is an integer combination of l_j, $j \in \{1, \dots, k\}$, with nonnegative integer coefficients. This means that every $l_j \in \mathcal{L}$ divides l_{v_i}. Therefore p divides l_{v_i}, which further implies p divides p_i.

On the other hand, consider an arbitrary cycle in digraph $\mathcal{G}$, and let its length be $l_j \in \mathcal{L}$. If the cycle goes through v_i, then p_i divides l_j. If not, then the cycle necessarily goes through some other node, say v_m. Since $\mathcal{G}$ is strongly connected, there must exist a cycle going through v_i and v_m. Denote by $l_{i,m}$ the length of this cycle. Thus p_i divides $l_{i,m}$. Note that these two cycles constitute a path of length $l_{i,m} + l_j$ from v_i to v_i. So p_i divides $l_{i,m} + l_j$ and therefore p_i divides l_j. Hence, p_i divides any l_j in $\mathcal{L}$. This means that p_i divides p.

Based on the above established two facts that p_i divides p and p divides p_i, we conclude that $p = p_i$ for every $i \in \{1, \dots, n\}$. □

Lemma 1.4 *Let A be an $n \times n$ nonnegative matrix. If $\mathcal{G}(A)$ is strongly connected and p-periodic, then $a_{ii}^k = 0$ for any $i \in \{1, \dots, n\}$ and for any k that is not a multiple of p.*

Proof. Let p_i, $i \in \{1, \dots, n\}$, be the period of the node v_i in $\mathcal{G}(A)$. Thus by Lemma 1.3

$$p = p_1 = \cdots = p_n$$

since $\mathcal{G}(A)$ is strongly connected. Hence the length of any path from v_i to v_i is a multiple of p. Namely there is no path from v_i to v_i with length k that is not a multiple of p. So it follows from Lemma 1.1 that $a_{ii}^k = 0$ for every $i \in \{1, \dots, n\}$ and any k that is not a multiple of p. □

With the three lemmas above, we present the proof of Theorem 1.4.

Proof of Theorem 1.4. (If) Since $\mathcal{G}(A)$ is strongly connected and aperiodic, by Lemma 1.3 the period of $\mathcal{G}(A)$ and the period of each node v_i are equal to 1. For any node v_i, let $l_{v_i}^1, l_{v_i}^2$ $(l_{v_i}^1 \neq l_{v_i}^2)$ be the lengths of two paths from v_i to v_i. By Lemma 1.2 there is sufficiently large $\bar{k}_i$ such that for any $k \geq \bar{k}_i$, k may be expressed by a nonnegative integer combination of $l_{v_i}^1$ and $l_{v_i}^2$, which means that there is a path of length k from v_i to v_i. Let v_j be another node. Since $\mathcal{G}(A)$ is strongly connected, there is a path from v_i to v_j; let its length be l_{ij}. Thus for any $k \geq q_{ij} := \bar{k}_i + l_{ij}$ there is a path of length k from v_i to v_j. It follows from Lemma 1.1 that $a_{ij}^k > 0$ for all $k \geq q_{ij}$. Let

$$q := \max\{q_{ij} \mid i, j = 1, \dots, n\}.$$

Then we have $a_{ij}^k > 0$ for all $i, j = 1, \dots, n$ and $k \geq q$. Therefore by definition, A is a primitive matrix.

(Only if) Suppose on the contrary that $\mathcal{G}(A)$ is not strongly connected, or that it is strongly connected but not aperiodic. For the first case that $\mathcal{G}(A)$ is not strongly connected, there is a pair of nodes v_i and v_j such that v_j is not reachable from v_i. So by Lemma 1.1, $a_{ij}^k = 0$ for all $k > 0$. Hence there is no positive integer k such that A^k is positive and consequently A is not primitive.

For the second case, $\mathcal{G}(A)$ is strongly connected but not aperiodic, that is, it is p-periodic where $p > 1$. It follows from Lemma 1.4 that $a_{ii}^{k'} = 0$ for any positive integer k' that is not a multiple of p. Hence there is no positive integer k such that A^k is positive, as otherwise if there were a positive integer k^* such that A^{k^*} is positive, then A^k is positive for any $k \geq k^*$, which contradicts $a_{ii}^{k'} = 0$ for any positive integer k' that is not a multiple of p. Therefore, A is not primitive. □

We are now ready to introduce the Perron-Frobenius Theorem. Denote by $\sigma(A)$ the *spectrum* of matrix A, i.e. the set of all eigenvalues of A, and $\rho(A)$ the *spectrum radius* of A, i.e. the maximum magnitude of the eigenvalues of A.

Theorem 1.5 (Perron-Frobenius Theorem) *Consider a nonnegative matrix A. If A is irreducible, then*

- $\rho(A) > 0$;
- *$\rho(A)$ is a simple eigenvalue of A;*
- *$\rho(A)$ has a positive eigenvector and a positive left-eigenvector.*[a]

Moreover, if A is primitive, then

- $(\forall \lambda \in \sigma(A)) \lambda \neq \rho(A) \Rightarrow |\lambda| < \rho(A)$.

[a] Left-eigenvector w corresponding to an eigenvalue λ of A satisfies $w^\top A = \lambda w^\top$.

Of particular interest is specialization of the Perron-Frobenius Theorem to a special class of nonnegative matrices. A nonnegative matrix A is called row-stochastic (resp. column-stochastic) if every row (resp. every column) of A sums up to one; if A is both row-stochastic and column-stochastic, it is called doubly-stochastic.

Lemma 1.5 *If A is a row-stochastic (column-stochastic, doubly stochastic) matrix, then $\rho(A) = 1$.*

Proof. We prove the statement for row-stochastic matrices; the proofs for column-stochastic and doubly-stochastic matrices are similar.

Since A is row-stochastic, we have $A\mathbf{1} = \mathbf{1}$. This means that 1 is an eigenvalue of A. Hence

$\rho(A) \geq 1$. On the other hand,

$$\begin{aligned}
\rho(A) &= \max\{|\lambda| \mid \lambda \text{ is an eigenvalue of } A\} \\
&= \max\{\|\lambda x\|_\infty \mid \lambda \text{ is an eigenvalue of } A, x \text{ is a corresponding eigenvector, } \|x\|_\infty = 1\} \\
&= \max\{\|\lambda x\|_\infty \mid \lambda \text{ is an eigenvalue of } A, x \text{ is a corresponding eigenvector, } \|x\|_\infty = 1\} \\
&= \max\{\|Ax\|_\infty \mid x \text{ is an eigenvector of } A, \|x\|_\infty = 1\} \\
&\leq \max\{\|Ax\|_\infty \mid \|x\|_\infty = 1\} \\
&= \|A\|_\infty \\
&= \max_i \sum_j |a_{ij}| = 1.
\end{aligned}$$

The last equality follows from the fact that every row of A sums to one. Therefore $\rho(A) = 1$. □

Theorem 1.6 (Perron-Frobenius Theorem for Stochastic Matrices) *Consider a row-stochastic (column-stochastic, doubly stochastic) matrix A. If A is irreducible, then $\rho(A) = 1$ is a simple eigenvalue of A, with a positive eigenvector and a positive left-eigenvector. Specifically:*

- *if A is row-stochastic, then eigenvalue 1 has a positive eigenvector $\mathbf{1}$ ($A\mathbf{1} = \mathbf{1}$) and a positive left eigenvector π_l ($\pi_l^\top A = \pi_l^\top$);*
- *if A is column-stochastic, then eigenvalue 1 has a positive eigenvector π_r ($A\pi_r = \pi_r$) and a positive left eigenvector $\mathbf{1}$ ($\mathbf{1}^\top A = \mathbf{1}^\top$);*
- *if A is doubly-stochastic, then eigenvalue 1 has a positive eigenvector $\mathbf{1}$ ($A\mathbf{1} = \mathbf{1}$) and a positive left eigenvector $\mathbf{1}$ ($\mathbf{1}^\top A = \mathbf{1}^\top$).*

Moreover, if A is primitive, then

- $(\forall \lambda \in \sigma(A))\lambda \neq 1 \Rightarrow |\lambda| < 1$.

Laplacian matrix

For a weighted digraph $\mathcal{G}$, the *weighted degree* d_i of a node i is the sum of the weights of all edges entering i, i.e. $d_i = \sum_{j=1}^n a_{ij}$. Similarly, the *weighted out-degree* d_i^o of a node i is the sum of the weights of all edges leaving i, i.e. $d_i^o = \sum_{j=1}^n a_{ji}$. A node i with $d_i = d_i^o$ is called *weight-balanced*. A digraph $\mathcal{G}$ is *weight-balanced* if every node is weight-balanced.

The *degree matrix* of a weighted digraph $\mathcal{G}$ is $D := \operatorname{diag}(d_1, \ldots, d_n)$. Let A be the adjacent matrix of $\mathcal{G}$; then $D = \operatorname{diag}(A\mathbf{1})$, where $\mathbf{1}$ is the vector of all ones.

The *Laplacian matrix* of a weighted digraph $\mathcal{G}$ is $L := D - A$. By definition $L\mathbf{1} = 0$; namely each row of L sums to zero. Thus 0 is an eigenvalue of L, with a corresponding eigenvector $\mathbf{1}$.

We distinguish three types of Laplacian matrices depending on their entries. Each type is useful for a set of cooperative control problems.

- If A is nonnegative, then L has nonnegative diagonal entries and nonpositive off-diagonal entries. This L is called *standard Laplacian matrix*.
- If A is (arbitrary) real, then L is called *signed Laplacian matrix*.
- If A is complex, then L is called *complex Laplacian matrix*.

Continuing the example in Fig. 1.10, the degree matrix is $D := \text{diag}(d_1, d_2, d_3, d_4, d_5)$, where $d_1 = a_{12}$, $d_2 = a_{21}$, $d = a_{31} + a_{32} + a_{35}$, $d_4 = a_{41} + a_{43} + a_{45}$, and $d_5 = a_{52} + a_{54}$. Thus the Laplacian matrix is

$$L := \begin{bmatrix} d_1 & -a_{12} & 0 & 0 & 0 \\ -a_{21} & d_2 & 0 & 0 & 0 \\ -a_{31} & -a_{32} & d_3 & 0 & -a_{35} \\ -a_{41} & 0 & -a_{43} & d_4 & -a_{45} \\ 0 & -a_{52} & 0 & -a_{54} & d_5 \end{bmatrix}.$$

Since 0 is by definition an eigenvalue of Laplacian matrix L, its *kernel* (i.e. *null space*) is at least one-dimensional. It turns out that the dimensions of the kernel of Laplacian matrices play a central role in characterizing the types of allowable cooperative behaviors.

Remark 1.1 *It is sometimes convenient to define degree matrix and Laplacian matrix with respect to the out-degrees of nodes. Consider a weighted digraph $\mathcal{G}$ and its adjacency matrix A. The* out-degree matrix *of $\mathcal{G}$ is $D^o := \text{diag}(d_1^o, \ldots, d_n^o)$; hence $D^o = \text{diag}(\mathbf{1}^\top A)$. Correspondingly, the* out-degree Laplacian matrix *of $\mathcal{G}$ is $L^o := D^o - A$. By this definition $\mathbf{1}^\top L^o = 0$; namely each column of L^o sums to zero. Thus 0 is again an eigenvalue of L^o, with a corresponding left-eigenvector $\mathbf{1}$.*

1.4 Standard Laplacian Matrices

Let $\mathcal{G}$ be a weighted digraph with n nodes, A the associated adjacency matrix, and $D(= \text{diag}(A\mathbf{1}))$ the degree matrix. In this section we consider that A is nonnegative, and $L = D - A$ the standard Laplacian matrix.

The null space of L is at least one-dimensional, for L has at least one eigenvalue 0. The following is a graphical condition that characterizes when the null space of L is exactly one-dimensional (namely the 0 eigenvalue of L is simple).

Theorem 1.7 *Let $\mathcal{G}$ be a weighted digraph with n nodes and L the standard Laplacian matrix. Then* $\dim(\ker L) = 1$ *if and only if $\mathcal{G}$ contains a spanning tree.*

Note that $\dim(\ker L) = 1$ is equivalent to $\operatorname{rank}(L) = n - 1$. To prove Theorem 1.7, it is useful to first present the following sufficient condition for $\operatorname{rank}(L) = n - 1$.

Lemma 1.6 *Let $\mathcal{G}$ be a weighted digraph with n nodes and L the standard Laplacian matrix. If $\mathcal{G}$ is strongly connected, then rank$(L) = n - 1$.*

Proof. Suppose that $\mathcal{G}$ is strongly connected. Then by Theorem 1.3, the nonnegative adjacency matrix A of $\mathcal{G}$ is irreducible and the degree matrix D is invertible. As a result, the Laplacian matrix $L = D - A$ can be written as

$$L = D(I - D^{-1}A).$$

Let $\tilde{A} := D^{-1}A$ and $\tilde{L} := D^{-1}L = I - \tilde{A}$. Then $\tilde{A}$ is also nonnegative and has zeros at the same locations as A does; the latter means that $\tilde{A}$ is irreducible too.

Note moreover that every row of $\tilde{A}$ sums up to 1. Thus $\tilde{A}$ is row-stochastic and its spectral radius equals one by Lemma 1.5, i.e. $\rho(\tilde{A}) = 1$. It then follows from the Perron-Frobenius Theorem for Stochastic Matrices (Theorem 1.6) that $\rho(\tilde{A}) = 1$ is a simple eigenvalue of $\tilde{A}$. By spectrum mapping, we derive that 0 is a simple eigenvalue of $\tilde{L} = I - \tilde{A}$, i.e. $\operatorname{rank}(\tilde{L}) = n - 1$. Therefore $\operatorname{rank}(L) = \operatorname{rank}(D\tilde{L}) = n - 1$. □

Remark 1.2 *In the proof of Lemma 1.6, the Perron-Frobenius Theorem for Stochastic Matrices (Theorem 1.6) is invoked to show that rank$(L) = n - 1$, namely the eigenvalue 0 of L is simple. Not needed in the above proof but will be useful later (in Chapters 2 and 3 of averaging/optimization problems), the Perron-Frobenius Theorem for Stochastic Matrices also asserts that the simple eigenvalue 0 of L has a positive left-eigenvector. That is, there exists $\pi_l > 0$ such that $\pi_l^\top L = 0$.*

Similarly for the standard out-degree Laplacian matrix L^o in Remark 1.1, if $\mathcal{G}$ is strongly connected, then the eigenvalue 0 of L^o is simple (hence rank$(L^o) = n - 1$) and has a positive eigenvector. That is, there exists $\pi_r > 0$ such that $L^o \pi_r = 0$.

Now we prove Theorem 1.7.

Proof of Theorem 1.7. (Only if) Suppose on the contrary that that the (weighted) digraph $\mathcal{G}$ does not contain a spanning tree. Then by Theorem 1.1, $\mathcal{G}$ contains at least two (disjoint) closed strong components, say $\mathcal{G}_1$ and $\mathcal{G}_2$. It follows from Lemma 1.6 that their Laplacian matrices L_1 and L_2 (say) each have a simple eigenvalue 0. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ are closed, the Laplacian matrix L

of $\mathcal{G}$ has the following structure:

$$L = \begin{bmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ * & * & * \end{bmatrix}.$$

Consequently L has at least two eigenvalues 0, which implies $\operatorname{rank}(L) < n-1$.

(If) Suppose that $\mathcal{G}$ contains a spanning tree. Let $\mathcal{V}_r$ be the subset of all possible roots, i.e.

$$\mathcal{V}_r := \{r \in \mathcal{V} \mid \mathcal{V}(r^{\rightarrow}) = \mathcal{V}\}.$$

Thus $\mathcal{V}_r \neq \emptyset$.

If $\mathcal{V}_r = \mathcal{V}$, namely every node is a root, then $\mathcal{G}$ is strongly connected, and by Lemma 1.6 we have $\operatorname{rank}(L) = n-1$.

If $\mathcal{V}_r \subsetneqq \mathcal{V}$ (i.e. $\mathcal{V}_r$ is a strict subset of $\mathcal{V}$), then the induced subdigraph $\mathcal{G}_r$ is the unique closed strong component of $\mathcal{G}$ (by Theorem 1.1). Thus every node in $\mathcal{V}_r$ can reach every node in $\mathcal{V} \setminus \mathcal{V}_r$, whereas no node in $\mathcal{V} \setminus \mathcal{V}_r$ can reach any node in $\mathcal{V}_r$. Consider without loss of generality the case that the nodes are ordered according to the partition $\mathcal{V}_r \cup (\mathcal{V} \setminus \mathcal{V}_r)$ (re-ordering corresponds merely to a permutation of node indices and the associated similarity transformation does not change spectrum of the matrices involved). Then the nonnegative adjacency matrix A and degree matrix D have the following forms:

$$A = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_3 \end{bmatrix}.$$

Note that $A_1 = D_1 = 0$ if and only if $\mathcal{V}_r$ is a singleton set (i.e. containing a single node). Accordingly the Laplacian matrix L is block (lower) triangular:

$$L = D - A = \begin{bmatrix} D_1 & 0 \\ 0 & D_3 \end{bmatrix} - \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix} =: \begin{bmatrix} L_1 & 0 \\ L_2 & L_3 \end{bmatrix}.$$

Since $\mathcal{G}_r$ is strongly connected, its Laplacian matrix L_1 has $\operatorname{rank}(L_1) = n-1$ (by Lemma 1.6). Thus 0 is a simple eigenvalue of L_1, and it remains to show that L_3 does not have an eigenvalue 0.

To that end, let $\tilde{D} := D$ if $\mathcal{V}_r$ contains more than one node; and

$$\tilde{D} := \begin{bmatrix} 1 & 0 \\ 0 & D_3 \end{bmatrix}$$

if $\mathcal{V}_r$ contains exactly one node. Thus the defined $\tilde{D}$ is invertible. Use $\tilde{D}^{-1}$ to define

$$\tilde{A} := \tilde{D}^{-1}A = \begin{bmatrix} \tilde{A}_1 & 0 \\ \tilde{A}_2 & \tilde{A}_3 \end{bmatrix}, \quad \tilde{L} := \tilde{D}^{-1}L = I - \tilde{A} = \begin{bmatrix} \tilde{L}_1 & 0 \\ \tilde{L}_2 & \tilde{L}_3 \end{bmatrix}.$$

Note that $\tilde{A}$ is nonnegative and every row sums up to 1. Hence for every integer $k \geq 1$, it holds that $\tilde{A}^k$ is nonnegative and every row sums up to 1. Let us focus on $\tilde{A}^n$ (i.e. $k = n$), which has the form

$$\tilde{A}^n := \begin{bmatrix} \tilde{A}_1^n & 0 \\ X & \tilde{A}_3^n \end{bmatrix}.$$

Since every node in $\mathcal{V}_r$ can reach every node in $\mathcal{V} \setminus \mathcal{V}_r$, it follows from Lemma 1.1 that all the entries of the (2, 1)-block X are positive. Hence the largest row sum of $\tilde{A}_3^n$ is smaller than one, i.e. $\|\tilde{A}_3^n\|_\infty < 1$. By the same proof of Lemma 1.5, we derive $\rho(\tilde{A}_3^n) \leq \|\tilde{A}_3^n\|_\infty < 1$. Therefore $\rho(\tilde{A}_3) < 1$ and $\tilde{L}_3 = I - \tilde{A}_3$ has no eigenvalue 0. This implies that $\tilde{L}_3$ has full rank, and so does $L_3 = D_3\tilde{L}_3$. The latter means that L_3 has no eigenvalue 0. Therefore L has a simple eigenvalue 0 (which is from L_1), and $\text{rank}(L) = n - 1$. □

We end this section with a result which is a generalization of Theorem 1.7. The result states that the dimension of $\ker L$ is equal to the number of (disjoint) closed strong components in $\mathcal{G}$.

Theorem 1.8 *Let $\mathcal{G}$ be a weighted digraph with n nodes and L the standard Laplacian matrix. Consider an integer $k \in [1, n]$. Then $\dim(\ker L) = k$ if and only if $\mathcal{G}$ contains k closed strong components.*

Proof. (If) Suppose that $\mathcal{G}$ contains k ($\in [1, n]$) closed strong components, denoted by $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1), \ldots, \mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$. Let $\mathcal{V}_{k+1}$ be the set of remaining nodes (if any), i.e. $\mathcal{V}_{k+1} := \mathcal{V} \setminus (\mathcal{V}_1 \cup \cdots \cup \mathcal{V}_k)$. To show $\dim(\ker L) = k$, it is equivalent to show $\text{rank}(L) = n - k$.

Renumber (if necessary) the nodes in the order of $\mathcal{V}_1, \ldots, \mathcal{V}_k, \mathcal{V}_{k+1}$, and permute the corresponding rows and columns in the Laplacian matrix L. Since the k strong components $\mathcal{G}_1, \ldots, \mathcal{G}_k$ are closed, the above permutation yields a matrix $\hat{L}$ (similarly transformed from L) of the following form:

$$\hat{L} := \begin{bmatrix} \hat{L}_1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{L}_k & 0 \\ X_1 & X_2 & \cdots & X_k & \hat{L}_{k+1} \end{bmatrix}.$$

Since L and $\hat{L}$ are similar via a permutation matrix, $\text{rank}(L) = \text{rank}(\hat{L})$. Moreover, since every strong component $\mathcal{G}_i$ $(i \in [1,k])$ is strongly connected, its Laplacian matrix L_i has $\text{rank}(L_i) = n-1$ (by Lemma 1.6); hence $\text{rank}(\hat{L}_i) = n-1$ for all $i \in [1,k]$. Given the block lower triangular structure of $\hat{L}$, to show $\text{rank}(L) = \text{rank}(\hat{L}) = n-k$, it suffices to establish that $\hat{L}_{k+1}$ does not have an eigenvalue 0. This is along the same lines as the sufficiency proof of Theorem 1.7, but with a higher dimension in general.

To proceed, let $\hat{A}$ and $\hat{D}$ be the adjacency matrix and degree matrix corresponding to $\hat{L}$:

$$\hat{A} := \begin{bmatrix} \hat{A}_1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{A}_k & 0 \\ Y_1 & Y_2 & \cdots & Y_k & \hat{A}_{k+1} \end{bmatrix}, \quad \hat{D} := \begin{bmatrix} \hat{D}_1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{D}_k & 0 \\ 0 & 0 & \cdots & 0 & \hat{D}_{k+1} \end{bmatrix}.$$

Let $\tilde{D}_i := \hat{D}_i$ $(i \in [1,k])$ if $\mathcal{V}_i$ contains more than one node; $\tilde{D}_i := 1$ if $\mathcal{V}_i$ contains exactly one node. Also let $\tilde{D}_{k+1} := \hat{D}_{k+1}$. Note that $\tilde{D}_{k+1} \neq 0$ regardless of the number of nodes in $\mathcal{V}_{k+1}$ (as long as $\mathcal{V}_{k+1} \neq \emptyset$). This is because the induced digraph $\mathcal{G}_{k+1}$ by $\mathcal{V}_{k+1}$ is not closed; otherwise $\mathcal{G}_{k+1}$ would contain a closed strong component (as shown in the proof of Theorem 1.1). Thus define

$$\tilde{D} := \begin{bmatrix} \tilde{D}_1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{D}_k & 0 \\ 0 & 0 & \cdots & 0 & \tilde{D}_{k+1} \end{bmatrix}$$

which is invertible. Now use $\tilde{D}^{-1}$ to define

$$\tilde{A} := \tilde{D}^{-1}\hat{A} = \begin{bmatrix} \tilde{A}_1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{A}_k & 0 \\ \tilde{Y}_1 & \tilde{Y}_2 & \cdots & \tilde{Y}_k & \tilde{A}_{k+1} \end{bmatrix}, \quad \tilde{L} := \tilde{D}^{-1}\hat{L} = I - \tilde{A} = \begin{bmatrix} \tilde{L}_1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{L}_k & 0 \\ \tilde{X}_1 & \tilde{X}_2 & \cdots & \tilde{X}_k & \tilde{L}_{k+1} \end{bmatrix}.$$

Note that $\tilde{A}$ is nonnegative and every row sums up to 1. Hence for every integer $m \geq 1$, it holds that $\tilde{A}^m$ is nonnegative and every row sums up to 1. Let us focus on $\tilde{A}^m$ for $m \geq n-k$, which has the form

$$\tilde{A}^m := \begin{bmatrix} \tilde{A}_1^m & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{A}_k^m & 0 \\ Z_1 & Z_2 & \cdots & Z_k & \tilde{A}_{k+1}^m \end{bmatrix}.$$

We claim that for every row of $Z := [Z_1 \ \cdots \ Z_k]$, there exists at least one positive entry.

First since $\mathcal{G}_{k+1}$ is not closed, there is a node $u_1 \in \mathcal{V}_{k+1}$ and $v_i \in \mathcal{V}_i$ for some $i \in [1, k]$ such that $(v_i, u_1) \in \mathcal{E}$ (i.e. an edge exists with tail v and head u_1). It then follows from $\mathcal{G}_i$ being a strong component that there is a node $v_i' \in \mathcal{V}_i$ such that $v_i' \to u_1$ with a path of any length $l \geq 1$. Next let $\mathcal{V}_{k+2} := \mathcal{V}_{k+1} \setminus \{u_1\}$. If $\mathcal{V}_{k+2} \neq \emptyset$, then the induced digraph $\mathcal{G}_{k+2}$ is again not closed. Thus there is a node $u_2 \in \mathcal{V}_{k+2}$ and $v \in \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_k \cup \{u_1\}$ such that $(v, u_2) \in \mathcal{E}$. Since there is an edge (v_i, u_1), it follows from a similar argument to above that there is a node $v' \in \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_k$ such that $v' \to u_2$ with a path of any length $l \geq 2$. Note that $\mathcal{V}_{k+1}$ has at most $n - k$ nodes. Repeating the above argument at most $n - k$ times leads to the conclusion that for every $m \geq n - k$, there is a path of length m from some node in $\mathcal{V}_1 \cup \cdots \cup \mathcal{V}_k$ to every node in $\mathcal{V}_{k+1}$. This proves our claim by invoking Lemma 1.1.

Therefore the largest row sum of $\tilde{A}_{k+1}^m$ is smaller than one, i.e. $\|\tilde{A}_{k+1}^m\|_\infty < 1$, which implies that $\rho(\tilde{A}_{k+1}^m) \leq \|\tilde{A}_{k+1}^m\|_\infty < 1$. Hence $\rho(\tilde{A}_{k+1}) < 1$ and $\tilde{L}_{k+1} = I - \tilde{A}_{k+1}$ has no eigenvalue 0. It follows that $\tilde{L}_{k+1}$ has full rank, and so does $\hat{L}_{k+1} = \tilde{D}_{k+1}\tilde{L}_{k+1}$. The latter means that $\hat{L}_{k+1}$ has no eigenvalue 0. The sufficiency proof is now complete.

(Only if) Suppose that $\mathcal{G}$ contains $k' \in [1, n]$ closed strong components and $k' \neq k$. Then by the above proved sufficiency result, $\dim(\ker L) = k' \neq k$. □

1.5 Complex Laplacian Matrices

Let $\mathcal{G}$ be a weighted digraph with n nodes, A the associated adjacency matrix, and $D(= \mathrm{diag}(A\mathbf{1}))$ the degree matrix. In this section we consider the second type that A is a complex matrix, and $L = D - A$ is the complex Laplacian matrix.

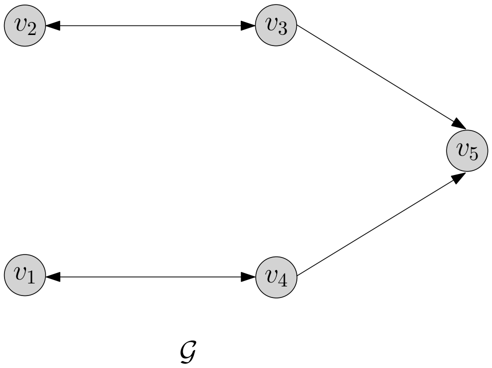
The following is a graphical condition that suffices to ensure that the null space of L is at most 2-dimensional.

Theorem 1.9 *Let $\mathcal{G}$ be a weighted digraph with n nodes and L the complex Laplacian matrix. If $\mathcal{G}$ contains a spanning 2-tree, then* $\dim(\ker L) \leq 2$ *for L with almost all complex entries.*

The phrase "*almost all* complex entries" means *for all complex entries except for those in some set of zero Lebesgue measure.*

Unlike Theorem 1.7, the graphical condition in Theorem 1.9 that $\mathcal{G}$ contains a spanning 2-tree is sufficient but not necessary to establish $\dim(\ker L) \leq 2$ (for L with almost all complex entries). The reason that the condition is not necessary follows from Theorem 1.8: a digraph $\mathcal{G}$ containing two closed strong components also gives rise to $\dim(\ker L) = 2$ for standard Laplacian matrix L which is a special case of complex Laplacian matrix; however, such a digraph need not contain a spanning 2-tree.

An example to illustrate this point is given in Fig. 1.13. Here the digraph $\mathcal{G}$ contains two closed strong components, but it does not contain a spanning 2-tree. Consider the unit weight for all edges in $\mathcal{G}$. Then the Laplacian matrix is displayed in Fig. 1.13, which has rank 3. Hence we indeed have $\dim(\ker L) = 2$, but $\mathcal{G}$ does not contain a spanning 2-tree.



$$L = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

$\text{rank}(L) = 3$

Figure 1.13: Digraph $\mathcal{G}$ contains two closed strong components but does not contain a spanning 2-tree

Note that $\dim(\ker L) \leq 2$ means that $\text{rank}(L) \geq n - 2$. To show this lower bound on $\text{rank}(L)$, it is sufficient to show that there exists a non-zero minor of L with size $n - 2$.

A *minor* with size $k \in [1, n]$ of L is the determinant of a $k \times k$ submatrix of L (by deleting $n - k$ rows and columns). If a minor with size k is non-zero, it implies that there are at least k linearly independent columns of L, hence giving a lower-bound k on the rank of L. In fact, $\text{rank}(L)$ is equal to the maximum size of a non-zero minor of L.

To prove Theorem 1.9, it is convenient to establish the following lemma.

Lemma 1.7 *Let $\mathcal{G}$ be a weighted digraph with n nodes and L the complex Laplacian matrix. If $\mathcal{G}$ contains a spanning tree, then* $\text{rank}(L) = n - 1$ *for L with almost all complex entries.*

The conclusion of this lemma is analogous to the sufficiency part of Theorem 1.7. But since we are dealing with complex L, the proof for Theorem 1.7 does not apply here, and a new proof technique is needed.

Proof of Lemma 1.7. Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$. Here $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$. Without loss of generality let $v_1 \in \mathcal{V}$ be the root. Then the standard Laplacian matrix T of $\mathcal{T}$ has the following form:

$$T := \begin{bmatrix} 0 & 0 \\ T_1 & T_2 \end{bmatrix}.$$

Since $\mathcal{T}$ is a spanning tree, by Theorem 1.7 we have $\operatorname{rank}(T) = n - 1$. Hence the determinant of T_2 is non-zero, i.e. $\det(T_2) \neq 0$. This is a non-zero minor with size $n - 1$.

Now consider the complex Laplacian matrix L' of $\mathcal{T}$, which has the same form as T: namely

$$L' := \begin{bmatrix} 0 & 0 \\ L_1' & L_2' \end{bmatrix}.$$

However, the entries of L_1', L_2' are complex numbers. According to the fact that a polynomial is either constantly zero or non-zero almost everywhere, it follows from $\det(T_2) \neq 0$ that $\det(L_2') \neq 0$ for L_2' with almost all complex entries.

Finally consider the complex Laplacian matrix L of $\mathcal{G}$ which generally has more edges than $\mathcal{T}$ (i.e. $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$). As a result, L generally contains more non-zeros entries than L':

$$L := \begin{bmatrix} * & * \\ L_1 & L_2 \end{bmatrix}.$$

Again according to the fact that a polynomial is either constantly zero or non-zero almost everywhere, it follows from $\det(L_2') \neq 0$ that $\det(L_2) \neq 0$ for L_2 with almost all complex entries. This means that for L with almost all complex entries, there is a non-zero minor with size $n - 1$, equivalently $\operatorname{rank}(L)$ is at least $n - 1$. On the other hand, since 0 is an eigenvalue of L, $\operatorname{rank}(L)$ can be at most $n - 1$. This concludes that $\operatorname{rank}(L) = n - 1$ for L with almost all complex entries. □

Now we prove Theorem 1.9.

Proof of Theorem 1.9. Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning 2-tree. Without loss of generality let $v_1, v_2 \in \mathcal{V}$ be the two roots, and write the complex Laplacian matrix L of $\mathcal{G}$ as follows:

$$L := \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ l_{21} & l_{22} & l_{23} & \cdots & l_{2n} \\ l_{31} & l_{32} & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}.$$

Remove the first row and the first column of L (all the following holds if the second row and the second column of L are removed). Denote the resulting submatrix by

$$L' := \begin{bmatrix} l_{22} & l_{23} & \cdots & l_{2n} \\ l_{32} & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}.$$

The above removal corresponds to removing from the digraph $\mathcal{G}$ the root v_1 and all those edges where v_1 is head or tail. Denote the resulting subdigraph $\mathcal{G}'$. Since $\mathcal{G}$ contains a spanning 2-tree, $\mathcal{G}'$ contains a spanning tree. Then it follows from Lemma 1.7 that $\text{rank}(L') = n-2$ for L' with almost all complex entries. This means that for L' with almost all complex entries, there is a non-zero minor of L' with size $n-2$. Since L' is a submatrix of L, we derive that for L with almost all complex entries, there is a non-zero minor of L with size $n-2$, equivalently $\text{rank}(L) \geq n-2$. This establishes the conclusion that $\dim(\ker L) \leq 2$ for L with almost all complex entries. □

Combining the conclusion of Theorem 1.9 and the fact that 0 is an eigenvalue of an arbitrary complex Laplacian L, we derive that if $\mathcal{G}$ contains a spanning 2-tree, then either $\dim(\ker L) = 1$ or $\dim(\ker L) = 2$ holds for L with almost all complex entries. For the special case that the digraph $\mathcal{G}$ is a spanning 2-tree, the following corollary asserts that the null space of its complex Laplacian matrix L is exactly 2 for L with almost all complex entries.

Corollary 1.1 *Let $\mathcal{G}$ be a weighted digraph with n nodes and L the complex Laplacian matrix. If $\mathcal{G}$ is a spanning 2-tree, then $\dim(\ker L) = 2$ for L with almost all complex entries.*

Proof. By Theorem 1.9, we know that $\dim(\ker L) \leq 2$. Without loss of generality let $v_1, v_2 \in \mathcal{V}$

be the two roots; thus the complex Laplacian matrix L of $\mathcal{G}$ has the following form:

$$L := \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & l_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}.$$

It follows from the above structure that $\dim(\ker L) \geq 2$. Therefore $\dim(\ker L) = 2$ after all. □

1.6 Signed Laplacian Matrices

Let $\mathcal{G}$ be a weighted digraph with n nodes, A the associated adjacency matrix, and $D(= \text{diag}(A\mathbf{1}))$ the degree matrix. In this section we consider the third type that A is an arbitrary real matrix, and $L = D - A$ is the signed Laplacian matrix.

Let $k \in [2, n-1]$ be an integer. The following is a graphical condition that is sufficient to ensure that the null space of L is at most k-dimensional.

Theorem 1.10 *Let $\mathcal{G}$ be a weighted digraph with n nodes, L the signed Laplacian matrix, and $k \in [2, n-1]$ an integer. If $\mathcal{G}$ contains a spanning k-tree, then* $\dim(\ker L) \leq k$ *for L with almost all real entries.*

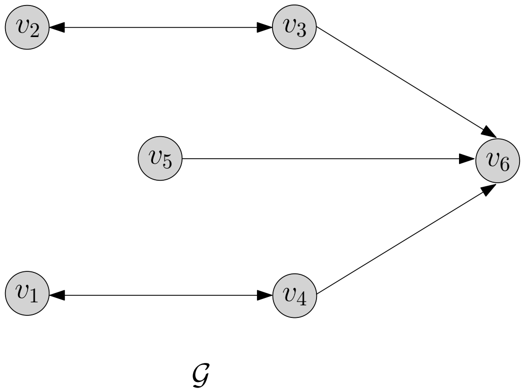
The conclusion is a generalization of Theorem 1.9 for k not only equal to 2 but also greater than 2; meanwhile a restriction, however, to the case of real entries.

Like Theorem 1.9, the graphical condition that $\mathcal{G}$ contains a spanning k-tree is only sufficient but not necessary to establish $\dim(\ker L) \leq k$ (for L with almost all real entries). The reason that the condition is not necessary again follows from Theorem 1.8: a digraph $\mathcal{G}$ containing k closed strong components also gives rise to $\dim(\ker L) = k$ for standard Laplacian matrix L which is a special case of signed Laplacian matrix; however, such a digraph need not contain a spanning k-tree.

For example, consider the digraph in Fig. 1.14. This digraph $\mathcal{G}$ contains three closed strong components, but it does not contain a spanning 3-tree. Consider the unit weight for all edges in $\mathcal{G}$. Then the Laplacian matrix is displayed in Fig. 1.14, which has rank 3. Hence we indeed have $\dim(\ker L) = 3$, but $\mathcal{G}$ does not contain a spanning 3-tree.

Note that $\dim(\ker L) \leq k$ means that $\text{rank}(L) \geq n - k$. To show this lower bound on $\text{rank}(L)$, it will be shown that there exists a non-zero minor of L with size $n - k$.

Proof of Theorem 1.10. The proof is by induction on $k \in [2, n-1]$.



$$L = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 3 \end{bmatrix}$$

$$\operatorname{rank}(L) = 3$$

Figure 1.14: Digraph $\mathcal{G}$ contains three closed strong components but does not contain a spanning 3-tree

Base case. Suppose that $\mathcal{G}$ contains a spanning 2-tree. Since a signed Laplacian matrix is a special complex Laplacian matrix, the conclusion for this case follows from Theorem 1.9.

Induction step. Suppose that if $\mathcal{G}$ contains a spanning k-tree ($k \in [2, n-2]$), then $\dim(\ker L) \leq k$ for L with almost all real entries. The latter means that $\operatorname{rank}(L) \geq n-k$ for L with almost all real entries, and equivalently there exists a non-zero minor of L with size $n-k$. Let $\mathcal{G}$ contain a spanning $(k+1)$-tree; without loss of generality let $v_1, \ldots, v_{k+1} \in \mathcal{V}$ be the $k+1$ roots, and write the signed Laplacian matrix L of $\mathcal{G}$ as follows:

$$L := \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1n} \\ l_{21} & l_{22} & l_{23} & \cdots & l_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{(k+1)1} & l_{(k+1)2} & l_{(k+1)3} & \cdots & l_{(k+1)n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}.$$

Remove the first row and the first column of L (all the following holds if the ith row and the ith column of L are removed for any $i \in [2, k+1]$). Denote the resulting submatrix by

$$L' := \begin{bmatrix} l_{22} & l_{23} & \cdots & l_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ l_{(k+1)2} & l_{(k+1)3} & \cdots & l_{(k+1)n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}.$$

The above removal corresponds to removing from the digraph $\mathcal{G}$ the root v_1 and all those edges where v_1 is head or tail. Denote the resulting subdigraph $\mathcal{G}'$. Since $\mathcal{G}$ contains a spanning $(k+1)$-tree, $\mathcal{G}'$ contains a spanning k-tree. Then it follows from the hypothesis that for L' with almost all real entries, there is a non-zero minor of L' with size $n-1-k = n-(k+1)$. Since L' is a submatrix of L, we derive that for L with almost all real entries, there is a non-zero minor of L with size $n-(k+1)$, equivalently $\text{rank}(L) \geq n-(k+1)$. This establishes $\dim(\ker L) \leq k+1$ for L with almost all real entries.

Following the above induction on $k \in [2, n-1]$, the proof is now complete. □

Now combining the conclusion of Theorem 1.10 and the fact that 0 is an eigenvalue of an arbitrary signed Laplacian L, we derive that if $\mathcal{G}$ contains a spanning k-tree, then $\dim(\ker L) \in [1, k]$ for L with almost all real entries. For the special case that the digraph $\mathcal{G}$ is a spanning k-tree, the following corollary asserts that the null space of its signed Laplacian matrix L is exactly k for L with almost all real entries.

Corollary 1.2 *Let $\mathcal{G}$ be a weighted digraph with n nodes, L the signed Laplacian matrix, and $k \in [2, n-1]$ an integer. If $\mathcal{G}$ is a spanning k-tree, then $\dim(\ker L) = k$ for L with almost all real entries.*

Proof. By Theorem 1.10, we know that $\dim(\ker L) \leq k$. Without loss of generality let $v_1, \ldots, v_k \in \mathcal{V}$ be the k roots; thus the signed Laplacian matrix L of $\mathcal{G}$ has the following form:

$$L := \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ l_{(k+1)1} & l_{(k+1)2} & l_{(k+1)3} & \cdots & l_{(k+1)n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix}.$$

It follows from the above structure that $\dim(\ker L) \geq k$. Therefore $\dim(\ker L) = k$ after all. □

We end this section by noting that the proofs above for Theorem 1.10 and Corollary 1.2 hold true even if "real entries" are replaced by "complex entries". This gives rise to the following theorem, which is a general result subsuming Theorems 1.9, 1.10 and Corollaries 1.1, 1.2.

Theorem 1.11 *Let $\mathcal{G}$ be a weighted digraph with n nodes, L the complex Laplacian matrix, and $k \in [2, n-1]$ an integer.*

- *If $\mathcal{G}$ contains a spanning k-tree, then $\dim(\ker L) \leq k$ for L with almost all complex entries.*
- *If $\mathcal{G}$ is a spanning k-tree, then $\dim(\ker L) = k$ for L with almost all complex entries.*

1.7 Notes and References

The material on digraphs, their connectivity and associated matrices is standard, and can be found in textbooks on graph theory, e.g.

- C. Godsil and G. Royle, Algebraic Graph Theory, Springer, 2001
- R.B. Bapat, Graphs and Matrices, Springer, 2010

The concepts of spanning multiple trees, complex and signed Laplacian matrices originate from

- Z. Lin, L. Wang, Z. Han, M. Fu, A graph laplacian approach to coordinate-free formation stabilization for directed networks, IEEE Transactions on Automatic Control, vol.61, pp.1269–1280, 2016
- Z. Lin, L. Wang, Z. Chen, M. Fu, Necessary and sufficient graphical conditions for affine formation control, IEEE Transactions on Automatic Control, vol.61, pp.2877–2891, 2016

Theorems 1.9, 1.10, and 1.11 are also adapted from the above.

Theorems 1.1, 1.2, 1.3, 1.4, 1.7, and 1.8 are adapted from

- Z. Lin, Distributed Control and Analysis of Coupled Cell Systems, VDM Verlag, 2008
- F. Bullo, Network Systems, Kindle Direct Publishing, 2020

Theorems 1.5 and 1.6 (Perron-Frobenius Theorem) can be found in e.g.

- R.A. Horn and C.R. Johnson, Matrix Analysis, 2nd ed., Cambridge University Press, 2013

Part II
Strongly Connected Digraphs: Averaging and Optimization

This part introduces two basic cooperative control problems — distributed averaging and optimization over digraphs. The necessary graphical condition for solving these two problems is that digraphs are strongly connected. The type of Laplacian matrices involved in these two problems is the standard Laplacian matrices. For agent dynamics, discrete-time linear time-invariant first-order systems are considered.

Chapter 2

Averaging

The first cooperative control problem we introduce is distributed averaging. Averaging is simple and useful in many contexts of networked systems. One example is *load balancing*: if there are five machines and ten jobs, having each machine process two jobs is the most efficient. Another example is environment measuring by *sensor networks*: if each sensor has measured an environment parameter, say temperature, contaminated by white noise, then the average of these measurements is the unbiased, minimum mean-squared error estimate of the true temperature. Other examples include cyclic pursuit, clock synchronization, and social influencing.

Networked systems and the interactions among component agents (via sensing or communication) are naturally modeled by digraphs. In this chapter, we show that a necessary graphical condition to achieve distributed averaging is that the digraph is *strongly connected*, namely every agent is reachable from every other agent. This is intuitively evident, as for locally computing the global average, each agent needs a 'channel', direct or indirect, to receive information from every other agent.

If the digraph is furthermore *balanced*, meaning roughly that each agent receives equal amount of in-flow information and out-going information, then averaging is easily solvable by a distributed algorithm (the consensus algorithm to be introduced in Chapter 4). However, *balanced* is neither a mild graphical condition nor a necessary condition for averaging. Hence we will assume only strongly connected digraphs (possibly unbalanced), and design a distributed algorithm that achieves averaging.

2.1 Problem Statement

Consider a network of n (> 1) agents. Each agent i $(\in [1, n])$ has a *state* variable $x_i(k) \in \mathbb{R}$, where $k \geq 0$ is a nonnegative integer and denotes the *discrete* time.

We model the interconnection structure of the networked agents by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Each node in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each (directed) edge (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent j communicates to agent i (namely, the information flow is from j to i). The (in-)neighbor set of agent i is $\mathcal{N}_i := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$, while the out-neighbor set $\mathcal{N}_i^o := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$.

We say that an algorithm is *distributed* if every agent i updates its state $x_i(k)$ based only on the information received from $\mathcal{N}_i$, and sends information only to $\mathcal{N}_i^o$.

Averaging Problem:

Consider a network of n agents interconnected through a digraph $\mathcal{G}$. Design a distributed algorithm to update the agents' states $x_i(k)$, $i = 1, \ldots, n$, such that

$$(\forall i \in [1, n])(\forall x_i(0) \in \mathbb{R}) \lim_{k \to \infty} x_i(k) = \frac{1}{n} \sum_{i=1}^{n} x_i(0).$$

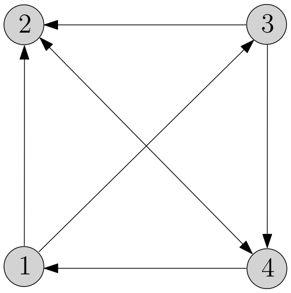


Figure 2.1: Illustrating example of averaging problem with four agents

Example 2.1 *We provide an example to illustrate the averaging problem. As displayed in Fig. 2.1, four agents are interconnected through a digraph $\mathcal{G}$. The (in-)neighbor sets of the agents are $\mathcal{N}_1 = \{4\}$, $\mathcal{N}_2 = \{1, 3, 4\}$, $\mathcal{N}_3 = \{1\}$, $\mathcal{N}_4 = \{2, 3\}$; and the out-neighbor sets are $\mathcal{N}_1^o = \{2, 3\}$, $\mathcal{N}_2^o = \{4\}$, $\mathcal{N}_3^o = \{2, 4\}$, $\mathcal{N}_4^o = \{1, 2\}$.*
Suppose that the initial states of the agents are $x_1(0) = 1$, $x_2(0) = 2$, $x_3(0) = 3$, $x_4(0) = 4$. Then the average is 2.5. The averaging problem is to design a distributed algorithm such that each agent's state asymptotically converges to the average value 2.5.

A necessary graphical condition for solving the averaging problem is given below.

Proposition 2.1 *Suppose that there exists a distributed algorithm that solves the averaging problem. Then the digraph $\mathcal{G}$ is strongly connected.*

Proof. The proof is by contradiction. Suppose that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *not* strongly connected. Then at least one node (agent) in $\mathcal{V}$ is not a root of $\mathcal{G}$. Let $\mathcal{R}$ denote the set of roots. Then $\mathcal{R} \neq \mathcal{V}$. We consider two cases separately: $\mathcal{R} = \emptyset$ and $\mathcal{R} \neq \emptyset$.

If $\mathcal{R} = \emptyset$, i.e. $\mathcal{G}$ does not contain a spanning tree, then it follows from Theorem 1.1 that $\mathcal{G}$ has at least two (distinct) closed strong components (say) $\mathcal{G}_1, \mathcal{G}_2$. In this case, consider an initial condition such that the agents in $\mathcal{G}_1$ have initial state $c_1 \in \mathbb{R}$, those in $\mathcal{G}_2$ have $c_2 \in \mathbb{R}$, and $c_1 \neq c_2$. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ are closed, information cannot be communicated from one to the other. Consequently, there exists no distributed algorithm that can solve the averaging problem.

It is left to consider $\mathcal{R} \neq \emptyset$. In this case, $\mathcal{G}$ contains a spanning tree, and again by Theorem 1.1 that $\mathcal{R}$ is the unique closed strong component in $\mathcal{G}$. Consider an initial condition such that all agents in $\mathcal{R}$ have initial state $c \in \mathbb{R}$, those in $\mathcal{V} \setminus \mathcal{R}$ have $c' \in \mathbb{R}$, and $c \neq c'$. Since $\mathcal{R}$ is closed, information cannot be communicated from $\mathcal{V} \setminus \mathcal{R}$ to $\mathcal{R}$. Consequently, there exists no distributed algorithm that can solve the averaging problem. □

Owing to Proposition 2.1, we shall henceforth assume that the digraph $\mathcal{G}$ is strongly connected.

Assumption 2.1 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents is strongly connected.*

2.2 Distributed Algorithm

Example 2.2 *Consider again Example 2.1. To achieve averaging, a natural idea is that each agent iteratively computes the (local) average of the state values received from neighbors and its own state value. Namely, for $i \in [1, 4]$*

$$\begin{aligned} x_i(k+1) &= \frac{1}{|\mathcal{N}_i|+1}(x_i(k) + \sum_{j \in \mathcal{N}_i} x_j(k)) \\ &= x_i(k) + \sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|+1}(x_j(k) - x_i(k)). \end{aligned}$$

For the initial states of the agents $x_1(0) = 1$, $x_2(0) = 2$, $x_3(0) = 3$, $x_4(0) = 4$, let us compute by the above equation the new states at $k = 1$:

$$\begin{aligned} x_1(1) &= x_1(0) + \frac{1}{2}(x_4(0) - x_1(0)) = \frac{1}{2}x_1(0) + \frac{1}{2}x_4(0) = 2.5 \\ x_2(1) &= x_2(0) + \frac{1}{4}(x_1(0) - x_2(0)) + \frac{1}{4}(x_3(0) - x_2(0)) + \frac{1}{4}(x_4(0) - x_2(0)) = 2.5 \\ x_3(1) &= x_3(0) + \frac{1}{2}(x_1(0) - x_3(0)) = \frac{7}{3} \\ x_4(1) &= x_4(0) + \frac{1}{3}(x_2(0) - x_4(0)) + \frac{1}{3}(x_3(0) - x_4(0)) = 3. \end{aligned}$$

Observe that the state sum at time $k = 1$ is $\sum_{i=1}^4 x_i(1) = \frac{31}{3}$, while the initial state sum $\sum_{i=1}^4 x_i(0) = 10$. The state sum has changed (by $\frac{1}{3}$) after one update, and this is in fact due to unbalanced structure of the digraph $\mathcal{G}$ in Fig. 2.1. Indeed, let $a_{ij} = \frac{1}{|\mathcal{N}_i|+1}$ be the *(positive) weight of edge $(j,i) \in \mathcal{E}$; then the weighted degrees are $d_1 = \frac{1}{2}$, $d_2 = \frac{3}{4}$, $d_3 = \frac{1}{2}$, $d_4 = \frac{2}{3}$, while the weighted out-degrees $d_1^o = \frac{3}{4}$, $d_2^o = \frac{1}{3}$, $d_3^o = \frac{7}{12}$, $d_4^o = \frac{3}{4}$ — the weighted digraph is thus not weight-balanced.*

Note that the adjacency matrix and standard Laplacian matrix of the weighted digraph $\mathcal{G}$ are:

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 \end{bmatrix}, \quad L = \begin{bmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

Hence the above state-update scheme may be written in vector form:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = (I - L) \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}.$$

The matrix $I - L$ is nonnegative and every row sums up to one; thus $I - L$ is a row stochastic *matrix. On the other hand, not every column of $I - L$ sums up to one, so $I - L$ is not column stochastic (and this is caused by non-weight-balancedness of the weighted digraph $\mathcal{G}$). This means that the initial sum is not kept invariant during each state update, and consequently asymptotic convergence to the initial average is not achievable. This is illustrated in Fig. 2.2.*

The problem illustrated by Example 2.2 suggests a plausible remedy: equip each agent i with an additional variable $s_i(k)$ to record the changes in state $x_i(k)$, such that the sum of $x_i(k)$ and $s_i(k)$ is a constant, i.e.

$$(\forall k \geq 0) \sum_{i=1}^{n} (x_i(k+1) + s_i(k+1)) = \sum_{i=1}^{n} (x_i(k) + s_i(k)).$$

We call $s_i(k)$ the *surplus* variable of agent i at time k. At $k = 0$, set $s_i(0) = 0$ for all i; this is intuitive because there is no change yet in state $x_i(0)$ to be recorded. Hence for every $k \geq 0$, there holds

$$\sum_{i=1}^{n} (x_i(k) + s_i(k)) = \sum_{i=1}^{n} (x_i(0) + s_i(0)) = \sum_{i=1}^{n} x_i(0). \tag{2.1}$$

Namely the initial state sum is kept invariant using the surplus variables.

In the following, we describe a distributed algorithm that updates the state $x_i(k)$ and the surplus

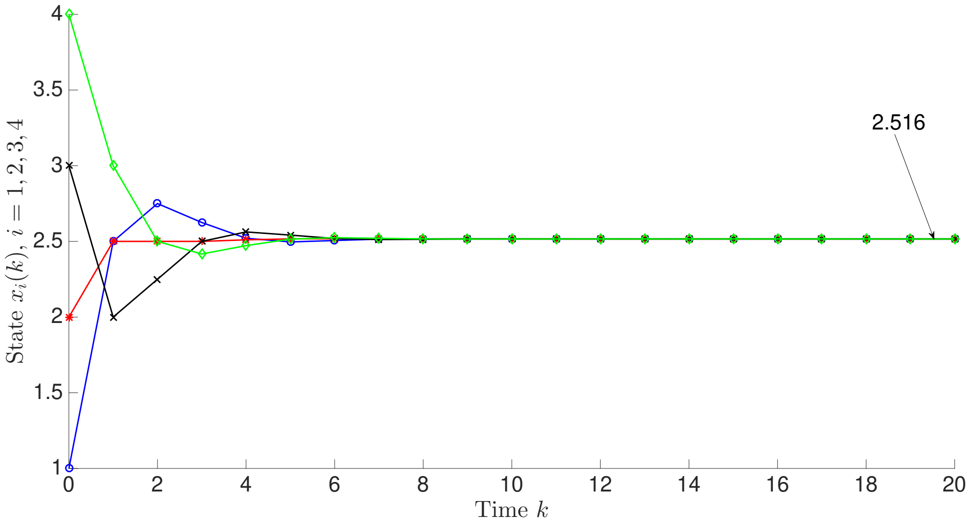


Figure 2.2: Failure to achieve averaging

$s_i(k)$ such that (2.1) holds.

Surplus-based Averaging Algorithm (SAA):

Every agent i has a state variable $x_i(k)$ whose initial value is an arbitrary real number, and a surplus variable $s_i(k)$ whose initial value is 0. At each time $k \geq 0$, every agent i performs three operations:

1) Agent i sends its state $x_i(k)$ and weighted surplus $a_{ji}s_i(k)$ to each out-neighbor $j \in \mathcal{N}_i^o$. The weights a_{ji} satisfy $\sum_{j\in\mathcal{N}_i^o} a_{ji} < 1$.

2) Agent i receives the state $x_j(k)$ and weighted surplus $a_{ij}s_j(k)$ from each (in-)neighbor $j \in \mathcal{N}_i$. The weights a_{ij} satisfy $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$.

3) Agent i updates its state $x_i(k)$ and surplus $s_i(k)$ as follows:

$$x_i(k+1) = x_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)) + \varepsilon s_i(k) \tag{2.2}$$

$$s_i(k+1) = (1 - \sum_{j\in\mathcal{N}_i^o} a_{ji})s_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}s_j(k) - \Big(x_i(k+1) - x_i(k)\Big). \tag{2.3}$$

The parameter ε in (2.2) is a positive real number, i.e. $\varepsilon > 0$.

Remark 2.1 *In SAA, (2.2) is the state update where the first two terms on the right constitute the averaging scheme in Example 2.2, and the last term specifies a certain amount of surplus used to influence the state update. On the other hand, (2.3) is the surplus update where the first two terms* on the right represent sending (resp. receiving) surplus to out-neighbor (resp. from in-neighbor) agents, and the third term records the change of state. Summing up (2.3) from $i = 1$ to n on both sides, we derive

$$\sum_{i=1}^{n} s_i(k+1) = \sum_{i=1}^{n} \left((1 - \sum_{j\in\mathcal{N}_i^o} a_{ji}) s_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij} s_j(k) \right) - \sum_{i=1}^{n} \Big(x_i(k+1) - x_i(k) \Big)$$
$$\Rightarrow \sum_{i=1}^{n} s_i(k+1) + \sum_{i=1}^{n} x_i(k+1)) = \sum_{i=1}^{n} s_i(k) + \sum_{i=1}^{n} x_i(k).$$

Hence SAA ensures constant sum of states and surpluses for all time as in (2.1).

Remark 2.2 In SAA, the weights a_{ij} are required to satisfy two conditions: $\sum_{j\in\mathcal{N}_i^o} a_{ji} < 1$ and $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$. In Example 2.2 the weights are chosen to be $a_{ij} = \frac{1}{|\mathcal{N}_i|+1}$ for every $j \in \mathcal{N}_i$, and for that example the two conditions are satisfied. However, in general this choice only ensures $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$ but not necessarily $\sum_{j\in\mathcal{N}_i^o} a_{ji} < 1$. An example illustrating this point is a variant of the digraph in Fig. 2.1 with an additional edge $(4,3)$: in this case $\sum_{j\in\mathcal{N}_4^o} a_{j4} = \frac{1}{2} + \frac{1}{4} + \frac{1}{3} > 1$. A simple choice that does ensure both conditions is the following:

$$a_{ij} = \min\left\{ \frac{1}{|\mathcal{N}_i|+1}, \frac{1}{|\mathcal{N}_i^o|+1} \right\}.$$

Another simple choice that requires the knowledge of the number of agents is $a_{ij} = \frac{1}{n}$.

Remark 2.3 Let $x := [x_1 \cdots x_n]^\top \in \mathbb{R}^n$ and $s := [s_1 \cdots s_n]^\top \in \mathbb{R}^n$ be the aggregated state and surplus, respectively, of the networked agents. Then the n equations of (2.2) become

$$x(k+1) = (I - L)x(k) + \varepsilon s(k).$$

Since $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$, $I - L$ is nonnegative. Moreover, since L has zero row sums, $I - L$ is row stochastic. On the other hand, the n equations of (2.3) become

$$\begin{aligned} s(k+1) &= (I - L^o)s(k) - (x(k+1) - x(k)) \\ &= Lx(k) + (I - L^o - \varepsilon I)s(k) \end{aligned}$$

where L^o is the out-degree Laplacian matrix. Since $\sum_{j\in\mathcal{N}_i^o} b_{ij} < 1$, $I - L^o$ is nonnegative. Moreover, since L^o has zero column sums, $I - L^o$ is column stochastic. Together, SAA is written compactly as follows:

$$\begin{bmatrix} x(k+1) \\ s(k+1) \end{bmatrix} = M \begin{bmatrix} x(k) \\ s(k) \end{bmatrix}, \text{ where } M := \begin{bmatrix} I - L & \varepsilon I \\ L & I - L^o - \varepsilon I \end{bmatrix}. \tag{2.4}$$

The initial conditions are $x(0) \in \mathbb{R}^n$ (arbitrary) and $s(0) = 0$. Notice that (i) the matrix M has negative entries due to the presence of the Laplacian matrix L in the $(2,1)$-block; (ii) the column sums of M are equal to one, which implies that the quantity $\mathbf{1}^T(x(k)+s(k))$ is a constant for all $k \geq 0$ (cf. (2.1)); and (iii) the state evolution specified by the $(1,1)$-block of M, i.e. $x(k+1) = (I-L)x(k)$ is the averaging scheme in Example 2.2.

Example 2.3 *Let us revisit Example 2.2. It is checked that the weights a_{ij} satisfy the two conditions $\sum_{j\in\mathcal{N}_i^o} a_{ji} < 1$ and $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$. We have seen the standard Laplacian matrix L and the row-stochastic $I - L$. The following are the out-degree Laplacian matrix L^o and the column-stochastic $I - L^o$:*

$$L^o = \begin{bmatrix} \frac{3}{4} & 0 & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{3} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{2} & 0 & \frac{7}{12} & 0 \\ 0 & -\frac{1}{3} & -\frac{1}{3} & \frac{3}{4} \end{bmatrix}, \quad I - L^o = \begin{bmatrix} \frac{1}{4} & 0 & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{2}{3} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{5}{12} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{4} \end{bmatrix}.$$

With these matrices, the matrix M in (2.4) may be constructed. Fig. 2.3 displays the case in which averaging is achieved when the parameter $\varepsilon = 0.1$; while Fig. 2.4 shows that when $\varepsilon = 0.5$, convergence does not occur. Hence the parameter ε needs to be carefully chosen (to be small enough) so as to achieve averaging.

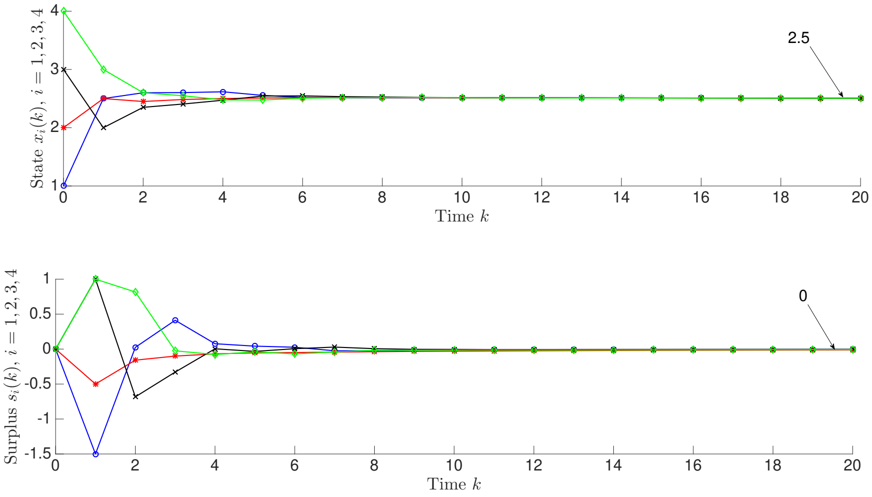


Figure 2.3: Convergence to average consensus when $\varepsilon = 0.1$

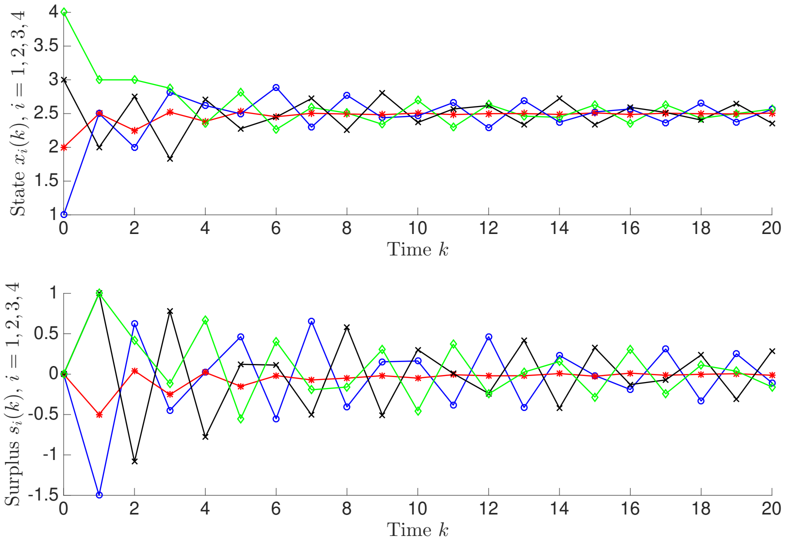


Figure 2.4: Failure to converge when $\varepsilon = 0.5$

2.3 Convergence Result

The following is the main result of this section.

Theorem 2.1 *Suppose that Assumption 2.1 holds. If the parameter $\varepsilon > 0$ is sufficiently small, then SAA solves the averaging problem.*

To prove Theorem 2.1, we will analyze the eigenvalues and eigenvectors of matrix M in (2.4). Write M in two parts: $M = M_0 + \varepsilon E$, where

$$M_0 := \begin{bmatrix} I - L & 0 \\ L & I - L^o \end{bmatrix}, \quad E := \begin{bmatrix} 0 & I \\ 0 & -I \end{bmatrix}.$$

The proof Theorem 2.1 is structured in two steps. First, we analyze the eigenvalues and eigenvectors of M_0. Second, we analyze the (infinitesimal) movement of M_0's eigenvalues upon being perturbed by εE.

Let us introduce two lemmas corresponding to the two steps outlined above.

Lemma 2.1 *Suppose that Assumption 2.1 holds. Then*

- *$I - L$ has a simple eigenvalue 1, with a positive eigenvector $\mathbf{1}$ and a positive left-*

eigenvector π_l; all the other eigenvalues λ satisfy $|\lambda| < 1$.

- *$I - L^o$ has a simple eigenvalue 1, with a positive eigenvector π_r and a positive left-eigenvector $\mathbf{1}$; all the other eigenvalues λ satisfy $|\lambda| < 1$.*

Proof. Under Assumption 2.1, it follows from Lemma 1.6 that the standard Laplacian matrix L has a simple eigenvalue 0. By spectrum mapping, $I - L$ has a simple eigenvalue 1. Since $I - L$ is row stochastic, $\rho(I - L) = 1$ (as shown in the proof of Lemma 1.6). Note also that the digraph $\mathcal{G}(I - L)$ constructed according to $I - L$ is strong connected and aperiodic, since all nodes have loops. Therefore by the Perron-Frobenius Theorem for Stochastic Matrices (Theorem 1.6), all the other eigenvalues λ of $I - L$ satisfy $|\lambda| < 1$.

Again under Assumption 2.1, the simple eigenvalue 0 of the standard Laplacian matrix L has a positive eigenvector $\mathbf{1}$ and a positive left-eigenvector π_l (Remark 1.2). It follows from

$$\begin{gathered}(I - L)\mathbf{1} = \mathbf{1} - L\mathbf{1} = \mathbf{1} \\ \pi_l^\top (I - L) = \pi_l^\top - \pi_l^\top L = \pi_l^\top\end{gathered}$$

that the simple eigenvalue 1 of $I - L$ has a positive eigenvector $\mathbf{1}$ and a positive left-eigenvector π_l.

The second statement concerning the out-degree Laplacian matrix can be proved similarly. □

Lemma 2.2 *Consider $M = M_0 + \varepsilon E$ and $\varepsilon > 0$. Let λ be a semi-simple double eigenvalue of M_0 (i.e. algebraic and geometric multiplicities of λ are both two), with (linearly independent) eigenvectors v_1, v_2 and (linearly independent) left-eigenvectors u_1, u_2 such that the following normalization condition holds:*

$$\begin{bmatrix} u_1^T \\ u_2^T \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

If ε is sufficiently small, then the two (perturbed) eigenvalues $\lambda(\varepsilon)$ of M corresponding to λ are $\lambda(\varepsilon) = \lambda + \varepsilon\lambda' + O(\varepsilon^2)$, where λ' has two values which are the eigenvalues of the following matrix:

$$\begin{bmatrix} u_1^T E v_1 & u_1^T E v_2 \\ u_2^T E v_1 & u_2^T E v_2 \end{bmatrix}. \tag{2.5}$$

Proof. Suppose that the positive perturbation parameter ε is sufficiently small. Then the two perturbed eigenvalues $\lambda(\varepsilon)$ of M corresponding to the semi-simple double eigenvalue λ of M_0 and the corresponding two perturbed eigenvectors $v(\varepsilon)$ may be expressed in terms of the following power series:

$$\lambda(\varepsilon) = \lambda + \varepsilon\lambda' + \varepsilon^2\lambda'' + \cdots = \lambda + \varepsilon\lambda' + O(\varepsilon^2)$$
$$v(\varepsilon) = v + \varepsilon v' + \varepsilon^2 v'' + \cdots = v + \varepsilon v' + O(\varepsilon^2).$$

It is left to show that λ' has two values which are the eigenvalues of the matrix in (2.5). Substituting the above two power series and $M = M_0 + \varepsilon E$ into the eigenvalue-eigenvector equation $M(\varepsilon)v(\varepsilon) = \lambda(\varepsilon)v(\varepsilon)$ yields

$$\begin{aligned}(M_0 + \varepsilon E)(v + \varepsilon v' + O(\varepsilon^2)) &= (\lambda + \varepsilon\lambda' + O(\varepsilon^2))(v + \varepsilon v' + O(\varepsilon^2)) \\ \Rightarrow M_0 v + \varepsilon(Mv' + Ev) + O(\varepsilon^2) &= \lambda v + \varepsilon(\lambda v' + \lambda' v) + O(\varepsilon^2).\end{aligned}$$

Hence we obtain

$$M_0 v = \lambda v \tag{2.6}$$
$$Mv' + Ev = \lambda v' + \lambda' v. \tag{2.7}$$

It follows from (2.6) that v is an eigenvector corresponding to the eigenvalue λ of M_0; thus there exist $c_1, c_2 \in \mathbb{R}$ such that $v = c_1 v_1 + c_2 v_2$. Note that at least one of c_1, c_2 is nonzero. Next multiply (2.7) by $u_1^\top$ from the left:

$$\begin{aligned}&u_1^\top(Mv' + Ev) = u_1^\top(\lambda v' + \lambda' v) \\ &\Rightarrow u_1^\top Mv' + u_1^\top Ev = \lambda u_1^\top v' + \lambda' u_1^\top v \\ &\Rightarrow u_1^\top Ev = \lambda' u_1^\top v \\ &\Rightarrow u_1^\top E(c_1 v_1 + c_2 v_2) = \lambda' u_1^\top (c_1 v_1 + c_2 v_2) \\ &\Rightarrow c_1 u_1^\top E v_1 + c_2 u_1^\top E v_2 = c_1 \lambda'.\end{aligned}$$

Similarly, multiplying (2.7) by $u_2^\top$ from the left yields:

$$c_1 u_2^\top E v_1 + c_2 u_2^\top E v_2 = c_2 \lambda'.$$

The above two equations may be written in matrix form:

$$\begin{bmatrix} u_1^T E v_1 & u_1^T E v_2 \\ u_2^T E v_1 & u_2^T E v_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \lambda' \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

The above matrix is the one in (2.5). Since c_1, c_2 are not both zero, we conclude that λ' has two values which are the two eigenvalues of this matrix. This completes our proof. □

Now we are ready to prove Theorem 2.1.

Proof of Theorem 2.1. Suppose that Assumption 2.1 holds and the parameter $\varepsilon > 0$ is sufficiently small. Write M in (2.4) as $M = M_0 + \varepsilon E$, where

$$M_0 := \begin{bmatrix} I - L & 0 \\ L & I - L^o \end{bmatrix}, \quad E := \begin{bmatrix} 0 & I \\ 0 & -I \end{bmatrix}.$$

The proof is structured into the following two steps.

Step 1: analyze the eigenvalues of M_0. Since M_0 is block (lower) triangular, its spectrum is $\sigma(M_0) = \sigma(I - L) \cup \sigma(I - L^o)$. By Lemma 2.1, 1 is a simple eigenvalue of $I - L$ (resp. $I - L^o$) and all the other eigenvalues λ of $I - L$ (resp. $I - L^o$) satisfy $|\lambda| < 1$. Hence M_0 has a double eigenvalue 1 (i.e. with algebraic multiplicity two), denoted by $\lambda_1 = \lambda_2 = 1$; and all the other $2n - 2$ eigenvalues have absolute values smaller than 1: $1 > |\lambda_3| \geq \cdots \geq |\lambda_{2n}|$.

Step 2: analyze the (infinitesimal) movement $\lambda_1 = \lambda_2 = 1$ of M_0 upon being perturbed by εE; for this we invoke Lemma 2.2. First we verify that the double eigenvalue 1 is semi-simple, namely with geometric multiplicity two. This may be done by checking the rank of

$$M_0 - I = \begin{bmatrix} I - L & 0 \\ L & I - L^o \end{bmatrix} - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} -L & 0 \\ L & -L^o \end{bmatrix}.$$

By elementary row operations — adding rows $1, \dots, n$ respectively to rows $n+1, \dots, 2n$ — the above matrix is transformed to

$$\begin{bmatrix} -L & 0 \\ 0 & -L^o \end{bmatrix}$$

and this matrix has rank $2n - 2$. The latter follows from Lemma 1.6 and Remark 1.2 that rank$(-L)$ =rank$(-L^o) = n - 1$ under Assumption 2.1). Since elementary row operations do not change rank, it holds that rank$(M_0 - I) = 2n - 2$. This means that the eigenspace of 1 is two-dimensional, namely the geometric multiplicity of eigenvalue 1 is two. This verifies that the double eigenvalue 1 is semi-simple.

Next we need to find (linearly independent) eigenvectors v_1, v_2 and left-eigenvectors u_1, u_2. Recall from Lemma 2.1 that the simple eigenvalue 1 of $I - L$ (resp. $I - L^o$) has a positive eigenvector $\mathbf{1}$ (resp. π_r) and a positive left-eigenvector π_l (resp. $\mathbf{1}$). Scale π_l, π_r (if necessary) such that $\mathbf{1}^\top \pi_l = 1$ and $\mathbf{1}^\top \pi_r = 1$, and consider the following:

$$v_1 = \begin{bmatrix} 0 \\ \pi_r \end{bmatrix}, \quad v_2 = \begin{bmatrix} \mathbf{1} \\ -n\pi_r \end{bmatrix}, \quad u_1 = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad u_2 = \begin{bmatrix} \pi_l \\ 0 \end{bmatrix}.$$

It is verified that

$$\begin{bmatrix} u_1^T \\ u_2^T \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

With the above preparations, we may qualify the changes of the semi-simple eigenvalue $\lambda_1 = \lambda_2 = 1$ of M_0 under a small perturbation εE by computing $\lambda_1(\varepsilon)$ and $\lambda_2(\varepsilon)$ according to Lemma 2.2; here $\lambda_1(\varepsilon)$ and $\lambda_2(\varepsilon)$ are the eigenvalues of M corresponding respectively to λ_1 and λ_2. It follows from Lemma 2.2 that for sufficiently small $\varepsilon > 0$, $\lambda_1(\varepsilon) = \lambda_1 + \varepsilon\lambda_1' + O(\varepsilon^2)$ and $\lambda_2(\varepsilon) = \lambda_2 + \varepsilon\lambda_2' + O(\varepsilon^2)$ where λ_1', λ_2' are the eigenvalues of the following matrix

$$\begin{bmatrix} u_1^\top E v_1 & u_1^\top E v_2 \\ u_2^\top E v_1 & u_2^\top E v_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \pi_l^\top \pi_r & -n\pi_l^\top \pi_r \end{bmatrix}.$$

Hence $\lambda_1' = 0$ and $\lambda_2' = -n\pi_l^\top \pi_r < 0$. This implies that $\lambda_1(\varepsilon)$ stays put at 1, while $\lambda_2(\varepsilon)$ moves to the left along the real axis. Then by continuity, there must exist a positive δ_1 such that $\lambda_1(\delta_1) = 1$ and $\lambda_2(\delta_1) < 1$. On the other hand, since eigenvalues are continuous functions of matrix entries, there must exist a positive δ_2 such that $|\lambda_i(\delta_2)| < 1$ for all $i \in \{3, \dots, 2n\}$. Thus for any sufficiently small $\epsilon \in (0, \min\{\delta_1, \delta_2\})$, the matrix M has a simple eigenvalue 1 and all other eigenvalues have absolute values smaller than one. For the simple eigenvalue 1, it follows from

$$M \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{1}^\top & \mathbf{1}^\top \end{bmatrix} M = \begin{bmatrix} \mathbf{1}^\top & \mathbf{1}^\top \end{bmatrix}$$

that its eigenvector and left-eigenvector are

$$y_1 := \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}, \quad z_1 := \frac{1}{n} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}.$$

We scale z_1 with $\frac{1}{n}$ such that $z_1^\top y_1 = 1$.

Now write M in Jordan canonical form as

$$M = WJW^{-1} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{2n} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & J' \end{bmatrix} \begin{bmatrix} z_1^\top \\ z_2^\top \\ \vdots \\ z_{2n}^\top \end{bmatrix}$$

where $y_i, z_i \in \mathbb{C}^{2n}$ ($i \in \{1, \dots, 2n\}$) are respectively the (generalized) right and left eigenvectors of M; and $J' \in \mathbb{C}^{(2n-1)\times(2n-1)}$ is a block diagonal matrix consisting of the Jordan blocks corresponding to those eigenvalues with absolute values smaller than one. Hence the kth power of M is

$$\begin{aligned} M^k = WJ^kW^{-1} &= W \begin{bmatrix} 1 & 0 \\ 0 & (J')^k \end{bmatrix} W^{-1} \\ &\to y_1 z_1^\top = \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^\top & \frac{1}{n}\mathbf{1}\mathbf{1}^\top \\ 0 & 0 \end{bmatrix}, \quad \text{as } k \to \infty. \end{aligned}$$

Therefore based on the SAA in (2.4):

$$\begin{aligned} \begin{bmatrix} x(k) \\ s(k) \end{bmatrix} &= M^k \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} \\ &\to \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^\top & \frac{1}{n}\mathbf{1}\mathbf{1}^\top \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^\top x(0) \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n x_i(0)\mathbf{1} \\ 0 \end{bmatrix}, \quad \text{as } k \to \infty. \end{aligned}$$

That is,

$$\lim_{k\to\infty} x_i(k) = \frac{1}{n}\sum_{i=1}^n x_i(0), \qquad \lim_{k\to\infty} s_i(k) = 0$$

i.e. SAA solves the averaging problem. □

2.4 Parameter Bound and Convergence Speed

Having shown that SAA solves the averaging problem for sufficiently small parameter $\varepsilon > 0$, in this section we aim to derive an upper bound on ε. As before write the matrix M in (2.4) as $M = M_0 + \varepsilon E$, where

$$M_0 := \begin{bmatrix} I - L & 0 \\ L & I - L^o \end{bmatrix}, \quad E := \begin{bmatrix} 0 & I \\ 0 & -I \end{bmatrix}.$$

We have shown that the spectrum of M_0 satisfies

$$1 = \lambda_1 = \lambda_2 > |\lambda_3| \geq \cdots \geq |\lambda_{2n}|.$$

The following is the main result of this section.

Theorem 2.2 *Suppose that Assumption 2.1 holds. SAA solves the averaging problem if the parameter ε satisfies $\varepsilon \in (0, \bar{\varepsilon})$, where*

$$\bar{\varepsilon} := \left(\frac{1 - |\lambda_3|}{32} \right)^{2n}. \tag{2.8}$$

Remark 2.4 *(Convergence Speed) By Theorem 2.2 if the parameter $\varepsilon \in (0, \bar{\varepsilon})$ with $\bar{\varepsilon}$ in (2.8), then SAA converges to the initial average. The speed of convergence is governed by the second largest (in terms of absolute value) eigenvalue of the updating matrix M, i.e. $|\lambda_2(\varepsilon)|$. We refer to $|\lambda_2(\varepsilon)|$ as the* convergence factor *of SAA; that is, SAA converges linearly at the rate of $O(|\lambda_2(\varepsilon)|^k)$. Note that $|\lambda_2(\varepsilon)| < 1$ is equivalent to averaging (as in the proof of Theorem 2.1); and the value of $|\lambda_2(\varepsilon)|$ depends not only on the digraph topology $\mathcal{G}$ but also on the parameter ε. We will illustrate this latter point using simulation examples in Section 2.5.*

To prove Theorem 2.2, we will relate the parameter ε to the distance between perturbed eigenvalues of M and unperturbed eigenvalues of M_0. To this end, we begin by introducing a metric for the distance between their spectra. Let $\sigma(M_0) := \{\lambda_1, \ldots, \lambda_{2n}\}$ and $\sigma(M) := \{\lambda_1(\varepsilon), \ldots, \lambda_{2n}(\varepsilon)\}$. The *optimal matching distance* $d\left(\sigma(M_0), \sigma(M)\right))$ is defined by

$$d\left(\sigma(M_0), \sigma(M)\right)) := \min_{\pi} \max_{i \in [1,2n]} |\lambda_i - \lambda_{\pi(i)}(\epsilon)| \tag{2.9}$$

where π is taken over all permutations of $\{1, \ldots, 2n\}$. Thus if we draw $2n$ identical circles centered respectively at $\lambda_1, \ldots, \lambda_{2n}$, then $d\left(\sigma(M_0), \sigma(M)\right))$ is the smallest radius such that these circles include all $\lambda_1(\varepsilon), \ldots, \lambda_{2n}(\varepsilon)$. Here is an upper bound on the optimal matching distance.

Lemma 2.3 *Consider $M \in \mathbb{R}^{n \times n}$ and $M = M_0 + \varepsilon E$. Then*

$$d\left(\sigma(M_0), \sigma(M)\right) \leq 2^{2-\frac{1}{2n}} (\|M_0\| + \|M\|)^{1-\frac{1}{2n}} \|\varepsilon E\|^{\frac{1}{2n}}.$$

Proof. Let $c \in [0, 1]$ and $N(c) := (1 - c)M_0 + cM$. Thus the eigenvalues of $N(c)$ trace $2n$ continuous curves in the complex plane as c changes from 0 to 1. The starting points of these curves are the eigenvalues of M_0 and the ending points are those of M. To prove the upper bound on $d\left(\sigma(M_0), \sigma(M)\right))$, it suffices to show that if Γ is any one of these curves, and a, b are the starting and ending points of Γ, then $|a - b|$ is bounded by the upper bound.

Without loss of generality assume that $\|M_0\| \leq \|M\|$ (the other case is symmetric). Let $\mathcal{L}$ be the straight line through a, b, and $\mathcal{S}$ be the segment of $\mathcal{L}$ between a, b; namely

$$\begin{aligned}\mathcal{L} &= \{z \mid z = a + l(b-a), l \in \mathbb{R}\}\\ \mathcal{S} &= \{z \mid z = a + l(b-a), l \in [0,1]\}.\end{aligned}$$

For each eigenvalue λ_i ($i \in \{1, \ldots, 2n\}$) of M_0, let $\lambda'_i = a + l_i(b-a)$, $l_i \in \mathbb{R}$, be the orthogonal projection of λ_i on the straight line $\mathcal{L}$. Also let $z = a + l(b-a)$ be an arbitrary point on $\mathcal{L}$. Then

$$\prod_{i=1}^{2n} |z - \lambda'_i| = \prod_{i=1}^{2n} |(l - l_i)(b-a)| = |a-b|^{2n} \prod_{i=1}^{2n} |l - l_i|.$$

By Chebyshev's inequality

$$\max_{l \in [0,1]} \prod_{i=1}^{2n} |l - l_i| \geq \frac{1}{2^{4n-1}}$$

there exists a point $z_0 = a + l_0(b-a)$ on the segment $\mathcal{S}$, for some $l_0 \in [0,1]$, such that

$$\prod_{i=1}^{n} |z_0 - \lambda'_i| \geq \frac{|a-b|^{2n}}{2^{4n-1}}.$$

Since Γ is a continuous curve between a and b, there exists a point λ_0 on Γ such that its orthogonal projection $\lambda'_0 = z_0$ on $\mathcal{S}$. It follows from the projection relation that for every $i \in \{1, \ldots, 2n\}$, $|\lambda_0 - \lambda_i| \geq |\lambda'_0 - \lambda'_i|$; hence

$$|\det(M_0 - \lambda_0 I)| = \prod_{i=1}^{2n} |\lambda_0 - \lambda_i| \geq \prod_{i=1}^{2n} |z_0 - \lambda'_i| \geq \frac{|a-b|^{2n}}{2^{4n-1}}.$$

Since λ_0 is a point on Γ, there exists $c_0 \in [0,1]$ such that λ_0 is an eigenvalue of $N(c_0) = (1-c_0)M_0 + c_0 M$. Choose an orthonormal basis $e_1, \ldots, e_{2n}$ such that $N(c_0)e_1 = \lambda_0 e_1$. Then it follows from Hadamard's inequality that

$$|\det(M_0 - \lambda_0 I)| \leq \prod_{i=1}^{2n} \|(M_0 - \lambda_0 I)e_i\|.$$

Owing to the chosen basis, $\|(M_0 - \lambda_0 I)e_1\| = \|(M_0 - N(t_0))e_1\| \leq \|M_0 - N(t_0)\|$. For $i = 2, \ldots, 2n$,

$$\|(M_0 - \lambda_0 I)e_i\| \leq \|M_0 e_i\| + |\lambda_0| \leq \|M_0\| + \|N(t_0)\|.$$

Hence

$$\begin{aligned}|\det(M_0-\lambda_0 I)| &\le \|M_0-N(t_0)\|(\|M_0\|+\|N(t_0)\|)^{2n-1}\\ &\le c_0\|M_0-M\|(\|M_0\|+(1-c_0)\|M_0\|+c_0\|M\|)^{2n-1}\\ &\le \|M_0-M\|(\|M_0\|+\|M\|)^{2n-1}.\end{aligned}$$

The last inequality is due to $\|M_0\| \le \|M\|$. From the above two inequalities of $|\det(M_0-\lambda_0 I)|$, we derive

$$\frac{|a-b|^{2n}}{2^{4n-1}} \le \|M_0-M\|(\|M_0\|+\|M\|)^{2n-1}$$

Taking $2n$th root yields

$$\begin{aligned}|a-b| &\le 2^{2-\frac{1}{2n}}\|M_0-M\|^{\frac{1}{2n}}(\|M_0\|+\|M\|)^{1-\frac{1}{2n}}\\ &= 2^{2-\frac{1}{2n}}(\|M_0\|+\|M\|)^{1-\frac{1}{2n}}\|\varepsilon E\|^{\frac{1}{2n}}.\end{aligned}$$

This is the upper bound on $d\,(\sigma(M_0),\sigma(M))$, and the proof is complete. □

Now we are ready to prove Theorem 2.2.

Proof of Theorem 2.2. Suppose that the parameter $\varepsilon \in (0,\bar{\varepsilon})$ with $\bar{\varepsilon}$ in (2.8). The proof is divided into two steps.

Step 1: we show that $|\lambda_3(\varepsilon)|,\dots,|\lambda_{2n}(\varepsilon)| < 1$.

Recall the two conditions on the weights a_{ij} of SAA: $\sum_{j=1}^n a_{ji} < 1$ and $\sum_{j=1}^n a_{ij} < 1$. Since the Laplacian matrix L is defined as $L = D - A$, we derive $\|L\|_\infty = 2\max_i \sum_{j=1}^n a_{ij} < 2$. On the other hand, by the definition of out-degree Laplacian matrix $L^o = D^o - A$ we have $\|I-L^o\|_\infty = \|(I-D^o)+A\|_\infty \le \max_i(1-\sum_{j=1}^n a_{ji})+\max_i \sum_{j=1}^n a_{ij} < 2$. Hence $\|M_0\|_\infty \le \|L\|_\infty + \|I-L^o\|_\infty < 4$ and $\|E\|_\infty \le 1$. It then follows from Lemma 2.3 that

$$\begin{aligned}d\,(\sigma(M_0),\sigma(M)) \le& 2^{2-\frac{1}{2n}}(\|M_0\|+\|M\|)^{1-\frac{1}{2n}}\|\varepsilon E\|^{\frac{1}{2n}}\\ &\le 2^{2-\frac{1}{2n}}(2\|M_0\|+\varepsilon\|E\|)^{1-\frac{1}{2n}}\|\varepsilon E\|^{\frac{1}{2n}}\\ &< 2^{2-\frac{1}{2n}}(8+\epsilon)^{1-\frac{1}{2n}}\epsilon^{\frac{1}{2n}}\\ &< 4(8+\epsilon)\epsilon^{\frac{1}{2n}}\\ &< 1-|\lambda_3|.\end{aligned}$$

The last inequality is due to $\varepsilon < \bar{\varepsilon}$ in (2.8). Now recall from the proof of Theorem 2.1 that the unperturbed eigenvalues $\lambda_3,\dots,\lambda_{2n}$ of M_0 lie strictly inside the unit circle. Therefore, perturbing the eigenvalues $\lambda_3,\dots,\lambda_{2n}$ by an amount less than $\bar{\varepsilon}$, the resulting eigenvalues $\lambda_3(\varepsilon),\dots,\lambda_{2n}(\varepsilon)$ will remain inside the unit circle.

Step 2: we show that $|\lambda_2(\varepsilon)| < 1$.

This is established by contraposition. First recall from the proof of Theorem 2.1 that $\lambda_2 = 1$ and for sufficiently small $\varepsilon > 0$, it holds that $|\lambda_2(\varepsilon)| < 1$. Now suppose that there exists $\delta \in (0, \bar{\varepsilon})$ such that $|\lambda_2(\delta)| \geq 1$. Owing to the continuity of eigenvalues, it suffices to consider $|\lambda_2(\delta)| = 1$. There are three such cases; for each we derive a contradiction.

Case 1: $\lambda_2(\delta)$ is a complex number with nonzero imaginary part and $|\lambda_2(\delta)| = 1$. Since M is a real matrix, there must exists another eigenvalue $\lambda_i(\delta)$, for some $i \in [3, 2n]$, such that $\lambda_i(\delta)$ is the complex conjugate of $\lambda_2(\delta)$. Then $|\lambda_i(\delta)| = |\lambda_2(\delta)| = 1$, which is in contradiction to the conclusion established in Step 1 above: all the eigenvalues $\lambda_3(\delta), \dots, \lambda_{2n}(\delta)$ stay inside the unit circle as $\delta \in (0, \bar{\varepsilon})$.

Case 2: $\lambda_2(\delta) = -1$. This implies that the optimal matching distance $d\left(\sigma(M_0), \sigma(M)\right) = 2$, which contradicts $d\left(\sigma(M_0), \sigma(M)\right) < 1 - |\lambda_3| < 1$ when (2.8) holds.

Case 3: $\lambda_2(\delta) = 1$. This means that the algebraic multiplicity of eigenvalue 1 equals two. The corresponding geometric multiplicity, however, equals one because $\operatorname{rank}(M - I) = 2n - 1$. To see this, write

$$M - I = \begin{bmatrix} I - L & \varepsilon I \\ L & I - L^o - \varepsilon I \end{bmatrix} - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} -L & \varepsilon I \\ L & -L^o - \varepsilon I \end{bmatrix}.$$

By elementary row operations — adding rows $1, \dots, n$ respectively to rows $n+1, \dots, 2n$ — the above matrix is transformed to

$$\begin{bmatrix} -L & \varepsilon I \\ 0 & -L^o \end{bmatrix}$$

and this matrix has rank $2n - 1$ (since $\operatorname{rank}(-L^o) = n - 1$ under Assumption 2.1 as stated in Remark 1.2). Thus there exists a generalized eigenvector $u = [u_1^\top \ u_2^\top]^\top \in \mathbb{R}^{2n}$ such that $(M-I)^2 u = 0$, and $(M-I)u$ is an eigenvector with respect to the eigenvalue 1. Since $[\mathbf{1}^\top \ 0]^\top$ is also an eigenvector corresponding to the eigenvalue 1, it must hold that

$$\begin{aligned}
& (M - I)u = c[\mathbf{1}^\top \ 0]^\top, \quad \text{for some scalar } c \neq 0 \\
\Rightarrow \ & \begin{bmatrix} -L & \varepsilon I \\ L & -L^o - \varepsilon I \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = c \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \\
\Rightarrow \ & \begin{cases} -Lu_1 + \epsilon u_2 = c\mathbf{1} \\ Lu_1 - L^o u_2 - \epsilon u_2 = 0 \end{cases} \\
\Rightarrow \ & - L^o u_2 = c\mathbf{1}.
\end{aligned}$$

Since $\operatorname{rank}(L^o) = n - 1$ but $\operatorname{rank}([L^o \ \ c\mathbf{1}]) = n$, there is no solution for u_2, which in turn implies that the generalized eigenvector u cannot exist. Therefore the eigenvalue 1 of M is simple, which contradicts that the algebraic multiplicity of eigenvalue 1 equals two.

Based on the impossibility of the above three cases, we conclude that for all $\varepsilon \in (0, \bar{\varepsilon})$, the eigenvalues of M satisfy

$$1 = \lambda_1(\varepsilon) > |\lambda_2(\varepsilon)| \geq |\lambda_3(\varepsilon)| \geq \cdots \geq |\lambda_{2n}(\varepsilon)|.$$

Following the same lines as in the proof of Theorem 2.1, the conclusion that SAA solves the averaging problem ensues. □

2.5 Simulation Examples

Let us illustrate, by simulation examples, that using SAA the states of the agents indeed converge to the desired average value, as well as how the convergence speed is affected by the parameter ε.

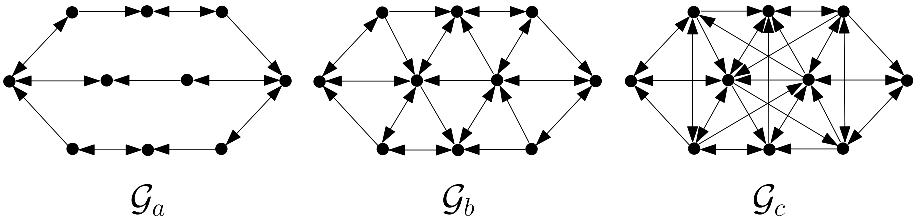


Figure 2.5: Three examples of strongly connected but unbalanced digraphs

Table 2.1: Convergence factor $|\lambda_2(\varepsilon)|$ with respect to different values of parameter ε

	$\varepsilon = 0.01$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.3$	$\varepsilon = 0.4$	$\varepsilon = 0.45$	$\varepsilon = 0.5$
$\mathcal{G}_a$	0.9915	0.9567	0.9754	0.9838	0.9990	1.0000	1.0487
$\mathcal{G}_b$	0.9909	0.9188	0.9203	0.9316	0.9400	0.9931	1.0611
$\mathcal{G}_c$	0.9906	0.9057	0.9062	0.9224	0.9333	0.9777	1.0000

Example 2.4 *Consider the three digraphs displayed in Fig. 2.5, with* 10 *nodes and respectively* 17, 29, *and* 38 *edges. All the digraphs are strongly connected, but they are unbalanced (indeed, no single node is balanced). We apply SAA by setting weights a_{ij} as in Remark 2.2; with these weights, these weighted digraphs are not weight-balanced.*
The convergence factor $|\lambda_2(\varepsilon)|$ for seven different values of the parameter ε are summarized in Table 2.1. Observe that small ε ensures convergence of SAA ($|\lambda_2(\varepsilon)| < 1$), whereas large ε can lead to instability. Moreover, in those converging cases the factor $|\lambda_2(\varepsilon)|$ decreases as the number of edges increases from $\mathcal{G}_a$ to $\mathcal{G}_c$, which indicates faster convergence when there

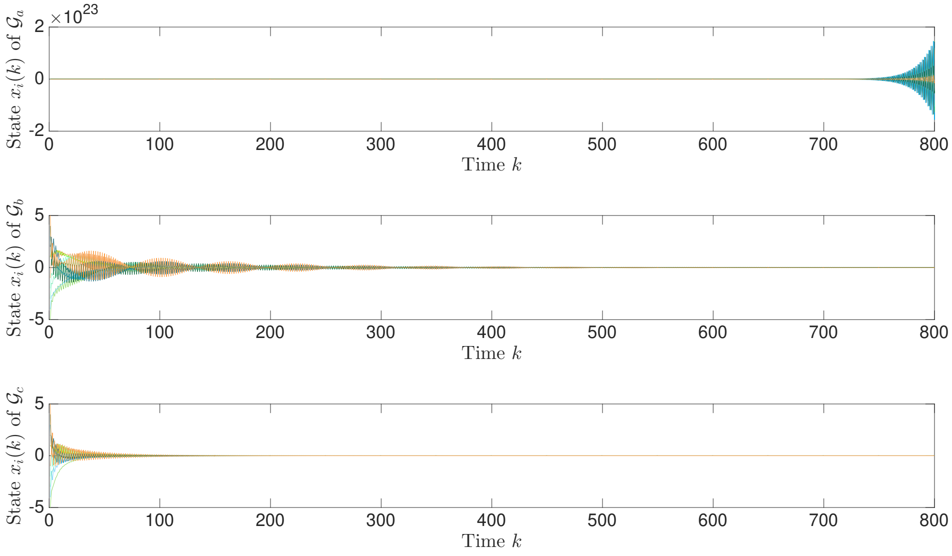


Figure 2.6: State trajectories when $\varepsilon = 0.45$

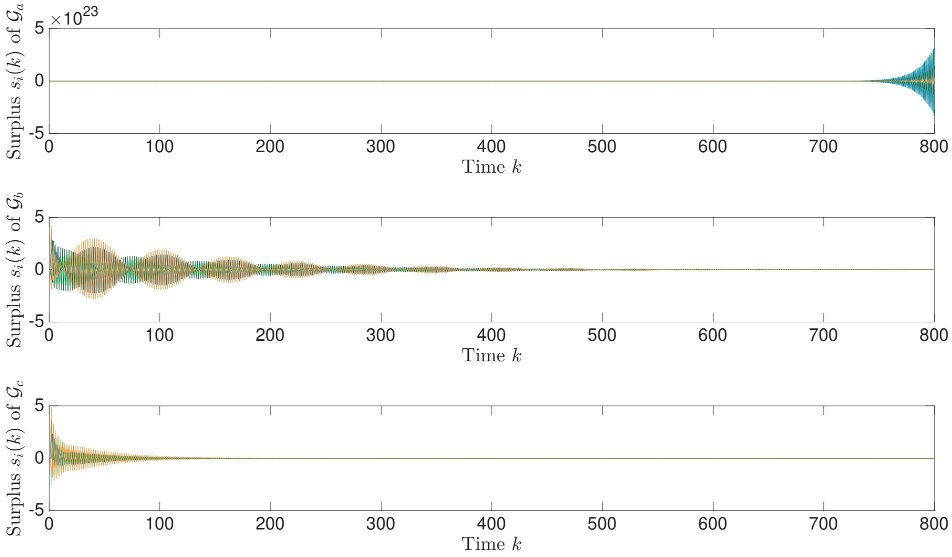


Figure 2.7: Surplus trajectories when $\varepsilon = 0.45$

are more communication channels available for information exchange. We also see that SAA is more robust on digraphs with more edges, in the sense that a larger range of values of ε is allowed.
For $\varepsilon = 0.45$, we display in Figs. 2.6 and 2.7 the trajectories of both states and surpluses when SAA is applied on digraphs $\mathcal{G}_a, \mathcal{G}_b, \mathcal{G}_c$ (with $x(0) = [-5\ \ -4\ \ -3\ \ -2\ \ -1\ 1\ 2\ 3\ 4\ 5]^\top$ and $s(0) = 0$). Consistent with the stability properties indicated by $|\lambda_2(\varepsilon)|$, $\mathcal{G}_a$ results in divergence, $\mathcal{G}_b$ convergence to the initial average 0 but with oscillatory transient behavior (since $|\lambda_2(\varepsilon)|$ is close to 1), and $\mathcal{G}_b$ convergence to the initial average 0 most smoothly.

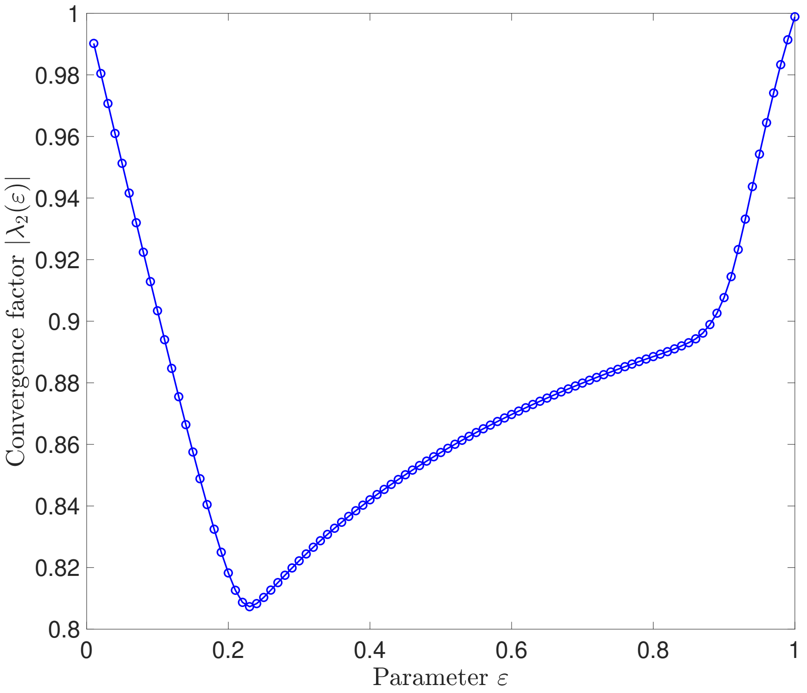


Figure 2.8: Convergence factor $|\lambda_2(\varepsilon)|$ versus parameter ε

Example 2.5 *We demonstrate the influence of parameter ε on the speed of convergence, specifically the convergence factor $|\lambda_2(\varepsilon)|$. To reduce the effect of network topology in this demonstration, we employ the Erdos-Reyni random digraph model: an edge between every pair of nodes can exist with probability $p = 1/2$, independent across the network and invariant over time; we take only those digraphs that are strongly connected.*

For SAA, consider Erdos-Reyni random digraphs of 100 *nodes and uniform weights* $1/100$ *(uniform weights are valid for SAA as asserted in Remark 2.2). Fig. 2.8 displays the curve of convergence factor* $|\lambda_2(\varepsilon)|$ *with respect to the parameter* ε*, each plotted point being the mean value of* $|\lambda_2(\varepsilon)|$ *over* 100 *random digraphs.*
To account for the trend of this curve, first recall from the perturbation argument in Theorem 2.1 that the matrix M *in (2.4) has two (maximum) eigenvalues* 1 *when* $\varepsilon = 0$*, and small* ε *causes that one of them (denote its absolute value by* λ_{in}*) moves into the unit circle. Meanwhile, some other eigenvalues of* M *inside the unit circle move outward; denote the maximum absolute value among these by* λ_{out}*. In Fig. 2.8 it is observed that when* ε *is small,* $|\lambda_2(\varepsilon)| = \lambda_{\text{in}}(> \lambda_{\text{out}})$ *and* λ_{in} *moves further inside as perturbation becomes larger; so* $|\lambda_2(\varepsilon)|$ *decreases (faster convergence) as* ε *increases in the beginning. Since the eigenvalues move continuously, there exists some* ε *such that* $\lambda_{\text{in}} = \lambda_{\text{out}}$*, corresponding to the fastest convergence speed. After that,* $|\lambda_2(\varepsilon)|$ *switches to* $\lambda_{\text{out}}(> \lambda_{\text{in}})$ *and* λ_{out} *moves further outside as* ε *increases; hence* $|\lambda_2(\varepsilon)|$ *increases and convergence becomes slower, and eventually divergence occurs (when* $|\lambda_2(\varepsilon)| \geq 1$*).*

2.6 Notes and References

The surplus-based averaging algorithm (SAA) is originated in

- K. Cai and H. Ishii, Average consensus on general strongly connected digraphs, Automatica, vol.48, pp.2750–2761, 2012

Eigenvalue perturbation result of Lemma 2.2 is due to

- A.P. Seyranian and A.A. Mailybaev, Multiparameter Stability Theory with Mechanical Applications, World Scientific, 2004

Bound on optimal matching distance in Lemma 2.3 is adapted from

- R. Bhatia, Matrix Analysis, Springer, 1996

In the proof of Lemma 2.3, Chebyshev's inequality can be found in standard texts e.g.

- T.J. Rivlin, An Introduction to the Approximation of Functions, Dover, 1981

and Hadamard's inequality in e.g.

- F. Riesz and B. Szokefalvi-Nagy, Functional Analysis, Dover, 1990

SAA has been generalized to address a number of other issues including time-varying and random digraphs as well as quantization of state values.

- K. Cai and H. Ishii, Average consensus on arbitrary strongly connected digraphs with time-varying topologies, IEEE Transactions on Automatic Control, vol.59, pp.1066–1071, 2014
- K. Cai, Averaging over general random networks, IEEE Transactions on Automatic Control, vol.57, pp.3186–3191, 2012
- K. Cai and H. Ishii, Quantized consensus and averaging on gossip digraphs, IEEE Transactions on Automatic Control, vol.56, pp.2087–2100, 2011

Chapter 3

Optimization

The second cooperative control problem we introduce is distributed optimization. Optimization is an important subject across mathematics, science, and engineering. Motivation of performing optimization over networked systems in a distributed fashion is driven by one or several combined factors including large scales, decentralized data collections, distributed computing technologies, and privacy concerns. One example of distributed optimization is large-scale machine learning, where big image/video data are collected and stored at different data centers, and multiple workstations in these centers perform optimization computation for global data classification or model prediction. Another example is economic dispatching in grid-connected smart buildings, where individual buildings process data of local energy generation and consumption which may be privacy-sensitive, and these buildings perform optimization computation for minimizing grid-wide generation costs subject to the constraint of meeting all consumption demands. Other application domains include power networks, smart grids, smart cities, transportation networks, and the Internet of Things (IoT).

In this chapter, we show that a necessary graphical condition to achieve distributed optimization is that the digraph is *strongly connected*. This is the same as the necessary condition for distributed averaging in the preceding chapter. Indeed, distributed optimization requires tracking the average value of the iteratively updated local optima, which intuitively demands that every agent possess a direct or indirect 'channel' in order to receive information from every other agent.

Owing to this close relation to averaging, we design a distributed optimization algorithm based on the surplus-based one presented for achieving averaging over strongly connected digraphs (which need not be balanced). Further, we will relate the distributed optimization problem to a widely studied problem of distributed resource allocation. Hence the latter may also be solved by the same distributed optimization algorithm.

3.1 Problem Statement

Consider a network of n (> 1) agents. Each agent i $(\in [1, n])$ has a *state* variable $x_i(k) \in \mathbb{R}$, and a local cost function $f_i : \mathbb{R} \to \mathbb{R}$.[1] The goal of distributed optimization is that the agents cooperatively solve the following problem:

$$\begin{aligned} \min_{x_1,\dots,x_n\in\mathbb{R}} \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to } x_1 = \cdots = x_n. \end{aligned} \tag{3.1}$$

Let $F(\xi) := \sum_{i=1}^{n} f_i(\xi)$ be the global cost function for the multi-agent network. Thus problem (3.1) means that every agent minimizes the global cost function. We shall restrict our attention to the case where F has a unique optimal solution $\xi^* \in \mathbb{R}$. Denote the optimal value by

$$F^* := F(\xi^*) = \min_{\xi\in\mathbb{R}} F(\xi).$$

Under the following assumption, F indeed admits a unique optimal solution ξ^* (see Lemma 3.8 in Appendix) and a reasonable rate of convergence to the solution ξ^* is ensured.

Assumption 3.1 *Every local cost function f_i $(i \in [1, n])$*

- *is* continuously differentiable *with gradient ∇f_i (which is derivative for one-dimensional f_i);*
- *is* strongly convex *with parameter $m_i > 0$ (or simply m_i-strongly convex), i.e.*

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) f_i(\xi_1) \geq f_i(\xi_2) + \nabla f_i(\xi_2)(\xi_1 - \xi_2) + \frac{m_i}{2}\|\xi_1 - \xi_2\|_2^2; \tag{3.2}$$

- *has a* Lipschitz-continuous gradient *with parameter $l_i > 0$ (or l_i-smooth), i.e.*

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) \|\nabla f_i(\xi_1) - \nabla f_i(\xi_2)\|_2 \leq l_i \|\xi_1 - \xi_2\|_2. \tag{3.3}$$

A straightforward characterization of the latter two conditions in Assumption 3.1 in the case that the inverse of the Hessian $\nabla^2 f_i$ (which is the reciprocal of the second derivative for one-dimensional f_i) exists is: $m_i \leq \nabla^2 f_i \leq l_i$. Namely, strong convexity and smoothness provide respectively lower and upper bounds on $\nabla^2 f_i$. As a result, $m_i \leq l_i$ always holds. Let

$$\bar{l} := \max_{i\in[1,n]} l_i, \quad l := \sum_{i=1}^{n} l_i, \quad m := \sum_{i=1}^{n} m_i. \tag{3.4}$$

[1] The choice of one-dimensional domain of function f is made deliberately for simplicity of presentation, and the essential ideas and techniques are the same for functions of multi-dimensional domain.

Then under Assumption 3.1, the global cost function F is m-strongly convex and l-smooth, with the *condition number* $Q := \frac{l}{m} \geq 1$.

Optimization Problem:

Consider a network of n agents interconnected through a digraph $\mathcal{G}$. Suppose that Assumption 3.1 holds and ξ^* is the (unique) optimal solution to $\min_{\xi \in \mathbb{R}} F(\xi)$. Design a distributed algorithm to update the agents' states $x_i(k)$, $i = 1, \dots, n$, such that

$$(\forall i \in [1, n])(\forall x_i(0) \in \mathbb{R}) \lim_{k \to \infty} x_i(k) = \xi^*.$$

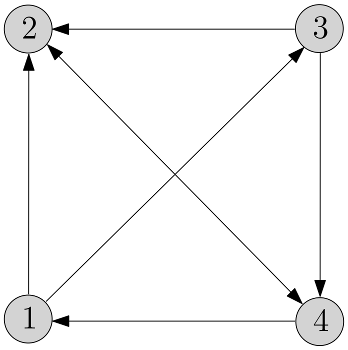


Figure 3.1: Illustrating example of optimization problem with four agents

Example 3.1 *We provide an example to illustrate the optimization problem. As displayed in Fig. 3.1, four agents are interconnected through a digraph $\mathcal{G}$. The (in-)neighbor sets of the agents are $\mathcal{N}_1 = \{4\}$, $\mathcal{N}_2 = \{1, 3, 4\}$, $\mathcal{N}_3 = \{1\}$, $\mathcal{N}_4 = \{2, 3\}$; and the out-neighbor sets are $\mathcal{N}_1^o = \{2, 3\}$, $\mathcal{N}_2^o = \{4\}$, $\mathcal{N}_3^o = \{2, 4\}$, $\mathcal{N}_4^o = \{1, 2\}$.*
Let the local cost functions of the agents be

$$\begin{aligned} f_1(\xi) &= \log(1 + \mathrm{e}^{-\xi}) + 2\xi^2 \\ f_2(\xi) &= 3\log(1 + \mathrm{e}^{-\xi}) + \xi^2 \\ f_3(\xi) &= 2\log(1 + \mathrm{e}^{-\xi}) + 2\xi^2 + 4 \\ f_4(\xi) &= \log(1 + \mathrm{e}^{-\xi}) + \xi^2 + \xi. \end{aligned}$$

Compute $\nabla^2 f_1(\xi) = \frac{\mathrm{e}^{\xi}}{(\mathrm{e}^{\xi}+1)^2} + 4$, which lies in the interval $(4, 4.25]$; thus f_1 is 4.05-strongly convex and 4.25-smooth. Similarly, f_2 is 2.05-strongly convex and 2.75-smooth; f_3 is 4.05-strongly convex and 4.5-smooth; and f_4 is 2.05-strongly convex and 2.25-smooth. Hence

Assumption 3.1 holds.
The global cost function F is

$$F(\xi) = \sum_{i=1}^{4} f_i(\xi) = 7\log(1+\mathrm{e}^{-\xi}) + 6\xi^2 + \xi + 4$$

which is 12.05*-strongly convex and* 13.75*-smooth. The unique optimal solution to* $\min_{\xi\in\mathbb{R}} F(\xi)$ *is* $\xi^* = 0.1819$*, and the optimal value is* $F^* = 8.6247$*.*
Suppose that the initial states of the agents are $x_1(0) = 1$*,* $x_2(0) = 2$*,* $x_3(0) = 3$*,* $x_4(0) = 4$*. The optimization problem is to design a distributed algorithm such that each agent's state asymptotically converges to the optimal solution* $\xi^* = 0.1819$*.*

A necessary graphical condition for solving the optimization problem is that the digraph $\mathcal{G}$ is strongly connected (this is the same as that for solving the averaging problem).

Proposition 3.1 *Suppose that there exists a distributed algorithm that solves the optimization problem. Then the digraph $\mathcal{G}$ is strongly connected.*

Proof. The proof is by contradiction. Suppose that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *not* strongly connected. Then at least one node (agent) in $\mathcal{V}$ is not a root of $\mathcal{G}$. Let $\mathcal{R}$ denote the set of roots. Then $\mathcal{R} \neq \mathcal{V}$. We consider two cases separately: $\mathcal{R} = \emptyset$ and $\mathcal{R} \neq \emptyset$.

If $\mathcal{R} = \emptyset$, i.e. $\mathcal{G}$ does not contain a spanning tree, then it follows from Theorem 1.1 that $\mathcal{G}$ has at least two (distinct) closed strong components (say) $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1), \mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$. In this case, consider local cost functions f_i and an initial condition such that the agents in $\mathcal{G}_1$ have initial state $c_1 \in \mathbb{R}$ that minimizes $\sum_{i\in\mathcal{V}_1} f_i(\cdot)$, those in $\mathcal{G}_2$ have $c_2 \in \mathbb{R}$ that minimizes $\sum_{i\in\mathcal{V}_2} f_i(\cdot)$, and $c_1 \neq c_2$. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ are closed (i.e. information cannot be communicated from one to the other) and the nodes in $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) have the same state value that minimizes $\sum_{i\in\mathcal{V}_1} f_i(\cdot)$ (resp. $\sum_{i\in\mathcal{V}_2} f_i(\cdot)$), there cannot exist any distributed algorithm that can update the states of the nodes in $\mathcal{G}_1$ or $\mathcal{G}_2$. Consequently, no distributed algorithm can solve the optimization problem.

It is left to consider $\mathcal{R} \neq \emptyset$. In this case, $\mathcal{G}$ contains a spanning tree, and again by Theorem 1.1 that $\mathcal{R}$ is the unique closed strong component in $\mathcal{G}$. Consider local cost functions f_i and an initial condition such that all nodes in $\mathcal{R}$ have the same state $c \in \mathbb{R}$, which minimizes $\sum_{i\in\mathcal{R}} f_i(\cdot)$; but $c \neq \xi^*$ where ξ^* is the optimal solution for $\sum_{i\in\mathcal{V}} f_i(\cdot)$. Since $\mathcal{R}$ is closed (i.e. information cannot be communicated from $\mathcal{V} \setminus \mathcal{R}$ to $\mathcal{R}$) and the nodes therein have the same state value that minimizes $\sum_{i\in\mathcal{R}} f_i(\cdot)$, there cannot exist any distributed algorithm that can update the states of the nodes in $\mathcal{R}$. Consequently, no distributed algorithm can solve the optimization problem. □

Owing to Proposition 3.1, we shall henceforth assume that the digraph $\mathcal{G}$ is strongly connected.

Assumption 3.2 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents is strongly connected.*

3.2 Distributed Algorithm

Example 3.2 *Consider again Example 3.1. To converge to the optimal solution ξ^*, a natural idea is that each agent employs gradient descent with respect to its local cost function, while iteratively computes the average of the state values received from neighbors. Namely, for $i \in [1,4]$*

$$x_i(k+1) = x_i(k) + \sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|+1}(x_j(k) - x_i(k)) - \varepsilon \nabla f_i(x_i(k))$$

where $\varepsilon > 0$ is a (small or diminishing) stepsize. In vector form we have

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} - \begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & \varepsilon & 0 \\ 0 & 0 & 0 & \varepsilon \end{bmatrix} \begin{bmatrix} \nabla f_1(x_1(k)) \\ \nabla f_2(x_2(k)) \\ \nabla f_3(x_3(k)) \\ \nabla f_4(x_4(k)) \end{bmatrix} \tag{3.5}$$

Denote by L the standard Laplacian matrix of the weighted digraph $\mathcal{G}$ in Fig. 3.1. Note that the first matrix above is $I - L$, which is row stochastic but is not column stochastic. The four eigenvalues of $I - L$ are:

$$1, 0.1667, 0.125 \pm 0.2602j$$

namely there is a simple eigenvalue 1 and other eigenvalues lie within the unit circle. Thus the spectral radius of $I - L$ is $\rho(I - L) = 1$. The (normalized) left eigenvector corresponding to the simple eigenvalue 1 is: $\pi_l := [0.4615\ 0.3077\ 0.4615\ 0.6923]^\top$; thus $\pi_l^\top(I - L) = \pi_l^\top$. Multiplying $\pi_l^\top$ on both sides of (3.5) above yields:

$$\sum_{i=1}^{4} \pi_i x_i(k+1) = \sum_{i=1}^{4} \pi_i x_i(k) - \varepsilon \sum_{i=1}^{4} \pi_i \nabla f_i(x_i(k)).$$

This is a gradient descent algorithm with a different global function $F'(\xi) := \sum_{i=1}^{4} \pi_i f_i(\xi)$, weighted by the left eigenvector π_l (for a different global state $x' := \sum_{i=1}^{4} \pi_l x_i$). Hence the above scheme does not solve the optimization of $F(\xi) = \sum_{i=1}^{4} f_i(\xi)$, i.e. the states do not *asymptotically converge to the optimal solution of F. This is illustrated in Fig. 3.2; here $\varepsilon = 0.1$ and the states converge to a vector* $[0.1035\ 0.2331\ 0.1599\ 0.0911]^\top$, *no component of which equals the optimal solution* $\xi^* = 0.1819$.

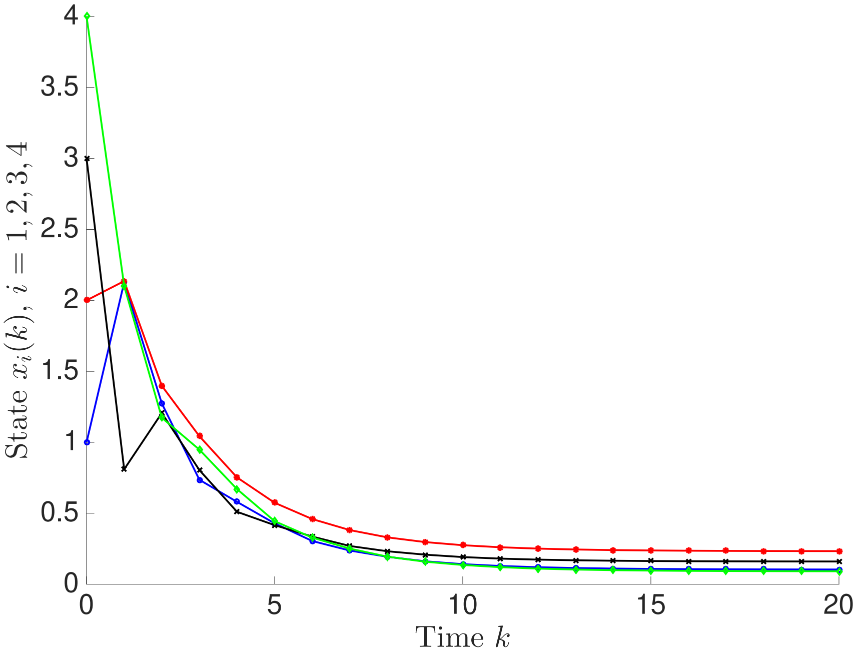


Figure 3.2: States fail to converge to the optimal solution of global cost function

Since our global function $F(\xi) = \sum_{i=1}^n f_i(\xi)$ is equally weighted over the local cost functions, if the left eigenvector π_l with respect to eigenvalue 1 of $I - L$ was $\mathbf{1}$ (the vector of all ones), then the scheme in Example 3.2 would have worked. In general, however, $\pi_l \neq \mathbf{1}$ for strongly connected digraphs (unless weight-balanced); instead we resort again to using surplus variables to achieve the same effect of uniform weights. Specifically, we equip each agent i with a surplus variable $s_i(k)$ to record the changes in the gradient of the local cost function, i.e. $\nabla f_i(x_i(k))$. At $k = 0$, we set $s_i(0) = \nabla f_i(x_i(0))$ for all i.

In the following, we describe a distributed algorithm that updates the state $x_i(k)$ and the surplus $s_i(k)$.

Surplus-based Optimization Algorithm (SOA):

Every agent i has a state variable $x_i(k)$ whose initial value is an arbitrary real number, and a surplus variable $s_i(k)$ whose initial value is $\nabla f_i(x_i(0))$. At each time $k \geq 0$, every agent i performs three operations:

1) Agent i sends its state $x_i(k)$ and weighted surplus $a_{ji}s_i(k)$ to each out-neighbor $j \in \mathcal{N}_i^o$. The weights a_{ij} satisfy $\sum_{j\in\mathcal{N}_i^o} a_{ij} < 1$.

2) Agent i receives the state $x_j(k)$ and weighted surplus $a_{ij}s_j(k)$ from each (in-)neighbor $j \in \mathcal{N}_i$. The weights a_{ij} satisfy $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$.

3) Agent i updates its state $x_i(k)$ and surplus $s_i(k)$ as follows:

$$x_i(k+1) = x_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)) - \varepsilon s_i(k) \tag{3.6}$$

$$s_i(k+1) = (1 - \sum_{j\in\mathcal{N}_i^o} a_{ji})s_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}s_j(k) + \Big(\nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k))\Big). \tag{3.7}$$

The parameter ε in (2.2) is a positive real number, i.e. $\varepsilon > 0$. The weights may be chosen as in Remark 2.2 to satisfy the two conditions $\sum_{j\in\mathcal{N}_i^o} a_{ij} < 1$ and $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$.

Remark 3.1 *In SOA, (3.6) is state update by the gradient descent scheme as described in Example 3.2, by treating $s_i(k)$ as the estimate of gradient of the local cost function. On the other hand, (3.7) is the surplus update where the first two terms represent sending (resp. receiving) surplus to out-neighbor (resp. from in-neighbor) agents, and the third term records the change in gradients. Summing up (3.7) from $i = 1$ to n on both sides, we derive*

$$\sum_{i=1}^n s_i(k+1) = \sum_{i=1}^n \left((1 - \sum_{j\in\mathcal{N}_i^o} a_{ji})s_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}s_j(k) \right) + \sum_{i=1}^n \Big(\nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k))\Big)$$

$$\Rightarrow \sum_{i=1}^n s_i(k+1) - \sum_{i=1}^n s_i(k) = \sum_{i=1}^n \nabla f_i(x_i(k+1)) - \sum_{i=1}^n \nabla f_i(x_i(k)).$$

Since $s_i(0) = \nabla f_i(x_i(0))$, we conclude for every $k \geq 0$ that $\sum_{i=1}^n s_i(k) = \sum_{i=1}^n \nabla f_i(x_i(k))$. Thus the sum of surplus variables $s_i(k)$ is the sum of gradients of the local cost functions at time k.

Remark 3.2 *(Relation with SAA) Consider (i) the special quadratic cost function $f_i(x_i) := \frac{1}{2}x_i^2$ (thus $\nabla f_i(x_i) = x_i$); and (ii) change of variable $\hat{s}_i := -s_i$. Substituting these into SOA, we obtain SAA with surplus variable $\hat{s}_i$. Note that $s_i \to 0$ if and only if $\hat{s}_i \to 0$. Owing to this relation, SOA is a generalization of SAA.*

Remark 3.3 *Let $x := [x_1 \cdots x_n]^\top \in \mathbb{R}^n$, $s := [s_1 \cdots s_n]^\top \in \mathbb{R}^n$, and $\nabla f(x) := [\nabla f_1(x_1) \cdots \nabla f_n(x_n)]^\top$ $\in \mathbb{R}^n$ be respectively the aggregated state, surplus, and gradients of the networked agents. Then*

SOA is written compactly as follows:

$$
\begin{aligned}
x(k+1) &= (I-L)x(k) - \varepsilon s(k)\\
s(k+1) &= (I-L^o)s(k) + (\nabla f(x(k+1)) - \nabla f(x(k)))
\end{aligned}
\tag{3.8}
$$

where $I-L$ is row stochastic and $I-L^o$ column stochastic. The initial conditions are $x(0)\in\mathbb{R}^n$ arbitrary and $s(0)=\nabla f(x(0))$.

Example 3.3 *Let us revisit Example 3.2. It is checked that the weights a_{ij} satisfy the two conditions $\sum_{j\in\mathcal{N}_i^o} a_{ij} < 1$ and $\sum_{j\in\mathcal{N}_i} a_{ij} < 1$. Then SOA in vector form is:*

$$
\begin{bmatrix} x_1(k+1)\\ x_2(k+1)\\ x_3(k+1)\\ x_4(k+1)\end{bmatrix}
=
\begin{bmatrix} \frac12 & 0 & 0 & \frac12\\ \frac14 & \frac14 & \frac14 & \frac14\\ \frac12 & 0 & \frac12 & 0\\ 0 & \frac13 & \frac13 & \frac13 \end{bmatrix}
\begin{bmatrix} x_1(k)\\ x_2(k)\\ x_3(k)\\ x_4(k)\end{bmatrix}
-
\begin{bmatrix} \varepsilon & 0 & 0 & 0\\ 0 & \varepsilon & 0 & 0\\ 0 & 0 & \varepsilon & 0\\ 0 & 0 & 0 & \varepsilon \end{bmatrix}
\begin{bmatrix} s_1(k)\\ s_2(k)\\ s_3(k)\\ s_4(k)\end{bmatrix}
$$

$$
\begin{bmatrix} s_1(k+1)\\ s_2(k+1)\\ s_3(k+1)\\ s_4(k+1)\end{bmatrix}
=
\begin{bmatrix} \frac14 & 0 & 0 & \frac12\\ \frac14 & \frac23 & \frac14 & \frac14\\ \frac12 & 0 & \frac{5}{12} & 0\\ 0 & \frac13 & \frac13 & \frac14 \end{bmatrix}
\begin{bmatrix} s_1(k)\\ s_2(k)\\ s_3(k)\\ s_4(k)\end{bmatrix}
+
\begin{bmatrix} \nabla f_1(x_1(k+1)) - \nabla f_1(x_1(k))\\ \nabla f_2(x_2(k+1)) - \nabla f_2(x_2(k))\\ \nabla f_3(x_3(k+1)) - \nabla f_3(x_3(k))\\ \nabla f_4(x_4(k+1)) - \nabla f_4(x_4(k)) \end{bmatrix}
$$

Fig. 3.3 displays the case in which all states converge to the optimal solution $\xi^* = 0.1819$ when the parameter $\varepsilon = 0.1$; while Fig. 3.4 shows that when $\varepsilon = 0.2$, convergence does not occur. Hence similar to SAA for the averaging problem, the parameter ε needs to be carefully chosen (to be small enough) so as to ensure convergence.

3.3 Convergence Result

The following is the main result of this section.

Theorem 3.1 *Suppose that Assumptions 3.1 and 3.2 hold. If the parameter $\varepsilon > 0$ is sufficiently small, then SOA solves the optimization problem.*

Consider the two matrices $I-L$ and $I-L^o$. Under Assumption 3.2 and by Lemma 2.1, the spectral radius $\rho(I-L) = 1$ is a simple eigenvalue with a positive left-eigenvector π_l such that $\pi_l^\top \mathbf{1} = 1$; and $\rho(I-L^o) = 1$ is also a simple eigenvalue with a positive eigenvector π_r such that

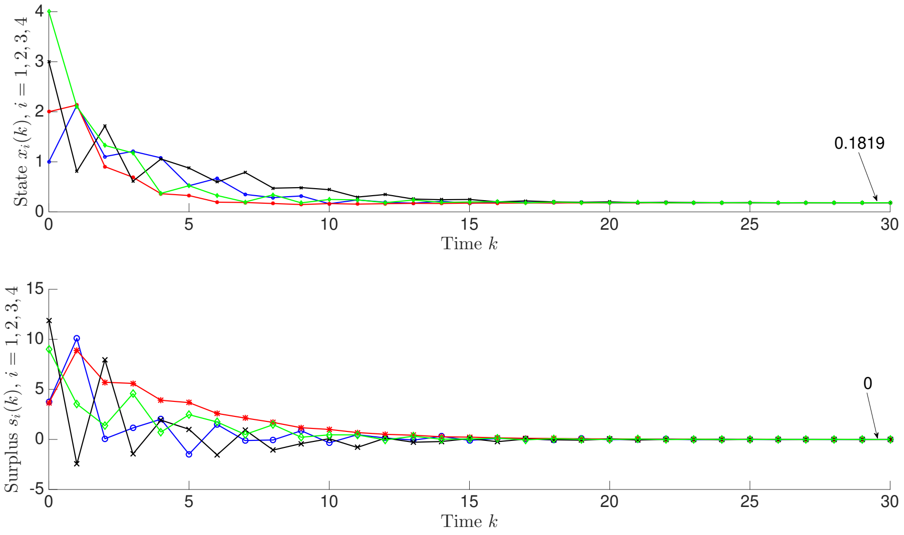


Figure 3.3: Convergence to optimal solution when $\varepsilon = 0.1$

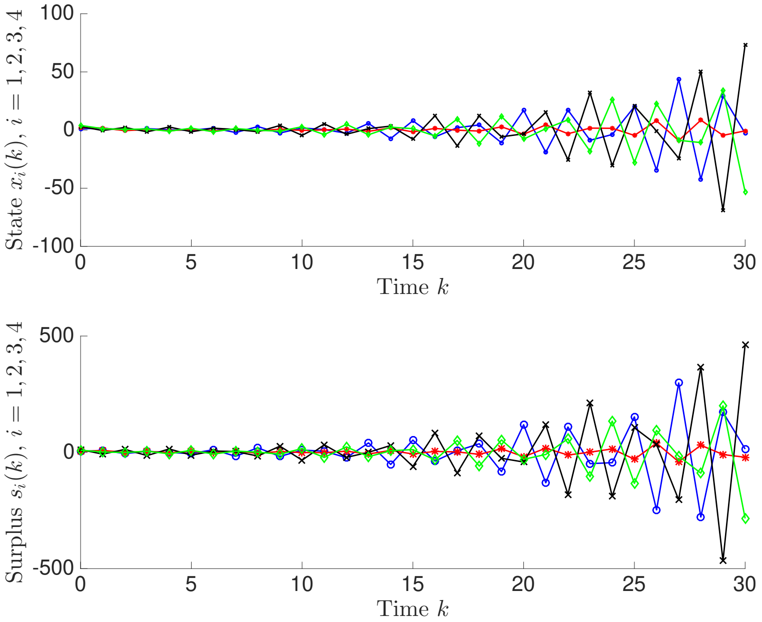


Figure 3.4: Failure to converge when $\varepsilon = 0.2$

$\pi_r^\top \mathbf{1} = 1$. Write $\Pi_l := \mathbf{1}\pi_l^\top$ and $\Pi_r := \pi_r \mathbf{1}^\top$. The proof of Theorem 3.1 is structured into the following three steps. First, we construct two special vector norms $\|\cdot\|_{\Pi_l}, \|\cdot\|_{\Pi_r}$ with which $I - L$ and $I - L^o$ have a special contraction property. Second, when the parameter $\varepsilon > 0$ satisfies a certain bound, we bound several relevant norms to derive the following inequality:

$$\begin{bmatrix} \|x(k+1) - \Pi_l x(k+1)\|_{\Pi_l} \\ \|\Pi_l x(k+1) - \xi^* \mathbf{1}\|_2 \\ \|s(k+1) - \Pi_r s(k+1)\|_{\Pi_r} \end{bmatrix} \leq C \begin{bmatrix} \|x(k) - \Pi_l x(k)\|_{\Pi_l} \\ \|\Pi_l x(k) - \xi^* \mathbf{1}\|_2 \\ \|s(k) - \Pi_r s(k)\|_{\Pi_r} \end{bmatrix} \tag{3.9}$$

where C is a nonnegative matrix. Finally, we prove for small $\varepsilon > 0$ that the spectral radius of C satisfies $\rho(C) < 1$. Hence all three eigenvalues of C lie within the unit circle; thereby

$$\begin{bmatrix} \|x(k) - \Pi_l x(k)\|_{\Pi_l} \\ \|\Pi_l x(k) - \xi^* \mathbf{1}\|_2 \\ \|s(k) - \Pi_r s(k)\|_{\Pi_r} \end{bmatrix} \to 0.$$

In particular $x(k) \to \xi^* \mathbf{1}$, meaning all the states converge to the optimal solution ξ^* of the global cost function.

In the sequel, we will introduce several lemmas corresponding to the three steps outlined above. The following lemma is for step 1.

Lemma 3.1 *Suppose that Assumption 3.2 holds. Then there exist vector norms $\|\cdot\|_{\Pi_l}$ and $\|\cdot\|_{\Pi_r}$ such that*

$$(\exists \sigma_l \in (0,1))(\forall v \in \mathbb{R}^n)\|(I - L)v - \Pi_l v\|_{\Pi_l} \leq \sigma_l \|v - \Pi_l v\|_{\Pi_l} \tag{3.10}$$

$$(\exists \sigma_r \in (0,1))(\forall v \in \mathbb{R}^n)\|(I - L^o)v - \Pi_r v\|_{\Pi_l} \leq \sigma_r \|v - \Pi_r v\|_{\Pi_r}. \tag{3.11}$$

Proof. The proof is by construction of such vector norms. We will do so for (3.10), and (3.11) follows similarly. Under Assumption 3.2 and by Lemma 2.1, we have $\rho((I - L) - \Pi_l) < 1$. Let $\delta \in (0, 1 - \rho((I - L) - \Pi_l))$; we are going to construct a matrix norm such that $\|(I - L) - \Pi_l\|_{\Pi_l} \leq \rho((I - L) - \Pi_l) + \delta < 1$.

By Schur triangularization, write $(I - L) - \Pi_l = U \Delta U^H$, where U is a unitary matrix, U^H the conjugate transpose of U, and Δ an upper triangular matrix:

$$U = \begin{bmatrix} \lambda_1 & d_{12} & d_{13} & \cdots & d_{1n} \\ 0 & \lambda_2 & d_{23} & \cdots & d_{2n} \\ 0 & 0 & \lambda_3 & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of $(I - L) - \Pi_l$. Let $T := \operatorname{diag}(t, t^2, \dots, t^n)$, where $t > 0$, and define $\|(I - L) - \Pi_l\|_{\Pi_l} := \|(TU^H)((I - L) - \Pi_l)(TU^H)^{-1}\|_1$. First, it is verified that $\|\cdot\|_{\Pi_l}$ is indeed a matrix norm (i.e. satisfying homogeneity, positive definiteness, triangle inequality, submultiplicativity). Moreover since

$$\begin{aligned} \|(TU^H)((I - L) - \Pi_l)(TU^H)^{-1}\|_1 &= \|TU^H U \Delta U^H U T^{-1}\|_1 \\ &= \|T \Delta T^{-1}\|_1 \\ &= \left\| \begin{bmatrix} \lambda_1 & t^{-1}d_{12} & t^{-2}d_{13} & \cdots & t^{-n+1}d_{1n} \\ 0 & \lambda_2 & t^{-1}d_{23} & \cdots & t^{-n+2}d_{2n} \\ 0 & 0 & \lambda_3 & \cdots & t^{-n+3}d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_n \end{bmatrix} \right\|_1 \end{aligned}$$

if t is large enough then the sum of all absolute values of off-diagonal entries is smaller than δ. Specifically, let t be such that

$$\begin{aligned} |t^{-1}d_{12}| + |t^{-2}d_{13}| + \cdots + |t^{-n+1}d_{1n}| &\le \delta \\ |t^{-1}d_{23}| + \cdots + |t^{-n+1}d_{2n}| &\le \delta \\ &\vdots \\ |t^{-1}d_{(n-1)n}| &\le \delta. \end{aligned}$$

Then it follows from the definition of 1-norm that $\|(I - L) - \Pi_l\|_{\Pi_l} \le \rho((I - L) - \Pi_l) + \delta < 1$. Let $\sigma_l := \|(I - L) - \Pi_l\|_{\Pi_l}$; thus $\sigma_l \in (0, 1)$.

Next, for the defined matrix norm $\|\cdot\|_{\Pi_l}$ we can always find a compatible vector norm. Note that for an arbitrary vector $v \in \mathbb{R}^n$, there holds

$$\begin{aligned}((I-L)-\Pi_l)(v-\Pi_l v) &= (I-L)v - \Pi_l v - (I-L)\Pi_l v + \Pi_l\Pi_l v \\ &= (I-L)v - \Pi_l v - (I-L)\mathbf{1}\pi_l^\top v + \mathbf{1}\pi_l^\top \mathbf{1}\pi_l^\top v \\ &= (I-L)v - \Pi_l v - \mathbf{1}\pi_l^\top v + \mathbf{1}\pi_l^\top v \\ &= (I-L)v - \Pi_l v.\end{aligned}$$

Therefore

$$\begin{aligned}\|(I-L)v - \Pi_l v\|_{\Pi_l} &= \|((I-L)-\Pi_l)(v-\Pi_l v)\|_{\Pi_l} \\ &\le \|(I-L)-\Pi_l\|_{\Pi_l}\|v-\Pi_l v\|_{\Pi_l} \\ &= \sigma_l\|v-\Pi_l v\|_{\Pi_l}.\end{aligned}$$

This establishes (3.10). □

The next five lemmas are for step 2. The first two are preliminaries for the latter three; and the latter three each derive a bound for a relevant norm in (3.9).

The first preliminary lemma below states that a gradient descent step $(\xi - \varepsilon\nabla F(\xi))$ yields a reduced distance to the optimal solution (ξ^*) by at least a fixed ratio.

Lemma 3.2 *Suppose that Assumption 3.1 holds. Then*

$$(\forall \xi \in \mathbb{R})(\forall \varepsilon \in (0, \frac{1}{l}])\|\xi - \varepsilon\nabla F(\xi) - \xi^*\|_2 \le (1-m\varepsilon)\|\xi-\xi^*\|_2.$$

Proof. Let $\xi \in \mathbb{R}$ and $\varepsilon \in (0, \frac{1}{l}]$. Since $l \ge m$ (Assumption 3.1), $\varepsilon \le \frac{2}{l+m}$ and thus $l \le \frac{2}{\varepsilon} - m$. Writing $l' := \frac{2}{\varepsilon} - m$, we have from Assumption 3.1 that F is l'-smooth and m-strongly convex. Then

$$\begin{aligned}\|\xi - \varepsilon\nabla F(\xi) - \xi^*\|_2^2 &= \|\xi-\xi^*\|_2^2 - 2\varepsilon\nabla F(\xi)(\xi-\xi^*) + \varepsilon^2\|\nabla F(\xi)\|_2^2 \\ &\le (1-\frac{2\varepsilon m l'}{m+l'})\|\xi-\xi^*\|_2^2 + \varepsilon(\varepsilon - \frac{2}{m+l'})\|\nabla F(\xi)\|_2^2\end{aligned}$$

where the inequality is due to the properties of smoothness and strong convexity (see Lemma 3.10 in Appendix) as well as $\nabla F(\xi^*) = 0$. Substituting $l' := \frac{2}{\varepsilon} - m$ into the above inequality yields $\|\xi - \varepsilon\nabla F(\xi) - \xi^*\|_2^2 = (1-\varepsilon m)^2\|\xi-\xi^*\|_2^2$. Since $1 - \varepsilon m \ge 1 - \frac{m}{l} \ge 0$, we finally derive $\|\xi - \varepsilon\nabla F(\xi) - \xi^*\|_2 \le (1-\varepsilon m)\|\xi-\xi^*\|_2$. □

The second preliminary lemma provides a bound for $\|s(k)\|_2$ in terms of the three relevant norms in (3.9). Here three different types of vector norms are involved: 2-norm, Π_l-norm, and Π_r-norm.

By norm-equivalence we have

$$\begin{aligned}(\exists c_1, c_2, c_3, c_4, c_5, c_6 > 0)\|\cdot\|_2 \le c_1\|\cdot\|_{\Pi_l}, \quad \|\cdot\|_2 \le c_2\|\cdot\|_{\Pi_r}, \quad \|\cdot\|_{\Pi_l} \le c_3\|\cdot\|_{\Pi_r} \\ \|\cdot\|_{\Pi_l} \le c_4\|\cdot\|_2, \quad \|\cdot\|_{\Pi_r} \le c_5\|\cdot\|_2, \quad \|\cdot\|_{\Pi_r} \le c_6\|\cdot\|_{\Pi_l}.\end{aligned}$$

Let $c := \max\{c_1, c_2, c_3, c_4, c_5, c_6\}$. Then for any two of the above three types of vector norms (say) $\|\cdot\|_{\text{type1}}$ and $\|\cdot\|_{\text{type2}}$, we have

$$\|\cdot\|_{\text{type1}} \le c\|\cdot\|_{\text{type2}} \tag{3.12}$$

Lemma 3.3 *Suppose that Assumption 3.1 holds. Then for all $k \ge 0$,*

$$\|s(k)\|_2 \le c\bar{l}\|\Pi_r\|_2\|x(k) - \Pi_l x(k)\|_{\Pi_l} + \bar{l}\|\Pi_r\|_2\|\Pi_l x(k) - \xi^*\mathbf{1}\|_2 + c\|s(k) - \Pi_r s(k)\|_{\Pi_r}$$

where c is in (3.12) and $\bar{l}$ in (3.4).

Proof. Writing $s(k) = s(k) - \Pi_r s(k) + \Pi_r s(k)$, where $\Pi_r = \pi_r\mathbf{1}^\top$, we have

$$\begin{aligned}\|s(k)\|_2 &\le \|s(k) - \Pi_r s(k)\|_2 + \|\Pi_r s(k)\|_2 \\ &\le c\|s(k) - \Pi_r s(k)\|_{\Pi_r} + \|\pi_r\mathbf{1}^\top s(k)\|_2.\end{aligned} \tag{3.13}$$

It follows from Remark 3.1 that $\mathbf{1}^\top s(k) = \mathbf{1}^\top \nabla f(x(k))$. Thus we next bound $\|\pi_r\mathbf{1}^\top\nabla f(x(k))\|_2$ as follows:

$$\begin{aligned}\|\pi_r\mathbf{1}^\top\nabla f(x(k))\|_2 &\le \|\pi_r\|_2\|\sum_i \nabla f_i(x_i(k)) - \sum_i \nabla f_i(\xi^*)\|_2 \\ &\overset{f_i \text{ are } l_i\text{-smooth}}{\le} \|\pi_r\|_2\sum_i l_i\|x_i(k) - \xi^*\|_2 \\ &\overset{\text{Jensen's inequality}}{\le} \bar{l}\|\pi_r\|_2\sqrt{n}\|x(k) - \xi^*\mathbf{1}\|_2 \\ &\overset{\|\Pi_r\|_2 = \|\pi_r\|_2\sqrt{n}}{\le} \bar{l}\|\Pi_r\|_2\|x(k) - \xi^*\mathbf{1} - \Pi_l x(k) + \Pi_l x(k)\|_2 \\ &\le c\bar{l}\|\Pi_r\|_2\|x(k) - \Pi_l x(k)\|_{\Pi_l} + \bar{l}\|\Pi_r\|_2\|\Pi_l x(k) - \xi^*\mathbf{1}\|_2.\end{aligned} \tag{3.14}$$

The lemma is proved by substituting (3.14) into (3.13). □

The next three lemmas each provide a bound for a relevant norm in (3.9).

Lemma 3.4 *Suppose that Assumptions 3.1 and 3.2 hold. Then for all $k \geq 0$,*

$$\begin{aligned}\|x(k+1) - \Pi_l x(k+1)\|_{\Pi_l} \leq & c_{11}\|x(k) - \Pi_l x(k)\|_{\Pi_l} + c_{12}\|\Pi_l x(k) - \xi^* \mathbf{1}\|_2 + \\ & c_{13}\|s(k) - \Pi_r s(k)\|_{\Pi_r}\end{aligned}$$

where the constants are (c in (3.12), σ_l in (3.10), and $\bar{l}$ in (3.4))

$$\begin{aligned}c_{11} &= \sigma_l + c^2 \varepsilon \bar{l} \|\Pi_r - \Pi_l\|_{\Pi_l} \|\Pi_r\|_2 \\ c_{12} &= c\varepsilon \bar{l} \|\Pi_r - \Pi_l\|_{\Pi_l} \|\Pi_r\|_2 \\ c_{13} &= c\varepsilon + c^2 \varepsilon \|\Pi_r - \Pi_l\|_{\Pi_l}.\end{aligned}$$

Proof. Since $x(k+1) = (I - L)x(k) - \varepsilon s(k)$ in (3.8), we have

$$\begin{aligned}\|x(k+1) - \Pi_l x(k+1)\|_{\Pi_l} &= \|((I-L)x(k) + \varepsilon s(k)) - \Pi_l((I-L)x(k) + \varepsilon s(k))\|_{\Pi_l} \\ &\overset{\Pi_l(I-L)=\Pi_l}{\leq} \|(I-L)x(k) - \Pi_l x(k)\|_{\Pi_l} + \varepsilon\|s(k) - \Pi_l s(k)\|_{\Pi_l} \\ &\overset{\text{Lemma 3.1}}{\leq} \sigma_l \|x(k) - \Pi_l x(k)\|_{\Pi_l} + \varepsilon\|s(k) - \Pi_l s(k) - \Pi_r s(k) + \Pi_r s(k)\|_{\Pi_l} \\ &\leq \sigma_l \|x(k) - \Pi_l x(k)\|_{\Pi_l} + c\varepsilon\|s(k) - \Pi_r s(k)\|_{\Pi_r} + c\varepsilon\|\Pi_r - \Pi_l\|_{\Pi_l}\|s(k)\|_2.\end{aligned} \tag{3.15}$$

The lemma is proved by substituting $\|s(k)\|_2$ from Lemma 3.3 into (3.15). (Note that Assumption 3.1 is needed to apply Lemma 3.3 and Assumption 3.2 to apply Lemma 3.1.) □

Lemma 3.5 *Suppose that Assumption 3.1 holds. If $\varepsilon < \frac{1}{l\pi_r^\top \pi_l}$ (l in (3.4)) then for all $k \geq 0$,*

$$\|\Pi_l x(k+1) - \xi^* \mathbf{1}\|_2 \leq c_{21}\|x(k) - \Pi_l x(k)\|_{\Pi_l} + c_{22}\|\Pi_l x(k) - \xi^* \mathbf{1}\|_2 + c_{23}\|s(k) - \Pi_r s(k)\|_{\Pi_r}$$

where the constants are (c in (3.12), $\bar{l}$ and m in (3.4))

$$\begin{aligned}c_{21} &= c\varepsilon \bar{l} n \pi_l^\top \pi_r \\ c_{22} &= (1 - \varepsilon m \pi_l^\top \pi_r) \\ c_{23} &= c\varepsilon \|\Pi_l\|_2.\end{aligned}$$

Proof. Since $x(k+1) = (I-L)x(k) - \varepsilon s(k)$ in (3.8), we have

$$\begin{aligned}\|\Pi_l x(k+1) - \xi^*\mathbf{1}\|_2 &= \|\Pi_l((I-L)x(k) + \varepsilon s(k) + \Pi_r s(k)(-\varepsilon + \varepsilon)) - \xi^*\mathbf{1}\|_2 \\ &\overset{\Pi_l(I-L)=\Pi_l=\mathbf{1}\pi_l^\top}{\le} \|\mathbf{1}\pi_l^\top x(k) - \xi^*\mathbf{1} - \varepsilon\Pi_l\Pi_r s(k)\|_2 + c\varepsilon\|\Pi_l\|_2\|s(k) - \Pi_r s(k)\|_{\Pi_r}.\end{aligned} \tag{3.16}$$

Noting that $\Pi_r = \pi_r\mathbf{1}^\top$, we bound $\|\mathbf{1}\pi_l^\top x(k) - \xi^*\mathbf{1} - \varepsilon\Pi_l\Pi_r s(k)\|_2$ as follows:

$$\begin{aligned}&\|\mathbf{1}\pi_l^\top x(k) - \xi^*\mathbf{1} - \varepsilon\Pi_l\Pi_r s(k)\|_2 \\ &= \|\pi_l^\top x(k)\mathbf{1} - \xi^*\mathbf{1} - \varepsilon\pi_l^\top\pi_r\nabla F(\pi_l^\top x(k))\mathbf{1} + \varepsilon\pi_l^\top\pi_r\nabla F(\pi_l^\top x(k))\mathbf{1} - \varepsilon\mathbf{1}\pi_l^\top\pi_r\mathbf{1}^\top s(k)\|_2 \\ &\le \|(\pi_l^\top x(k) - \varepsilon\pi_l^\top\pi_r\nabla F(\pi_l^\top x(k)) - \xi^*)\mathbf{1}\|_2 + \varepsilon\pi_l^\top\pi_r\|(\nabla F(\pi_l^\top x(k)) - \mathbf{1}^\top s(k))\mathbf{1}\|_2.\end{aligned} \tag{3.17}$$

Since Assumption 3.1 holds and $\varepsilon < \frac{1}{l\pi_r^\top\pi_l}$, it follows from Lemma 3.2 that the first term in (3.17)

$$\begin{aligned}\|(\pi_l^\top x(k) - \varepsilon\pi_l^\top\pi_r\nabla F(\pi_l^\top x(k)) - \xi^*)\mathbf{1}\|_2 &\le (1 - \varepsilon m\pi_l^\top\pi_r)\|(\pi_l^\top x(k) - \xi^*)\mathbf{1}\|_2 \\ &\overset{\Pi_l=\mathbf{1}\pi_l^\top}{=} (1 - \varepsilon m\pi_l^\top\pi_r)\|\Pi_l x(k) - \xi^*\mathbf{1}\|_2.\end{aligned} \tag{3.18}$$

It is left to bound the second term in (3.17):

$$\begin{aligned}\varepsilon\pi_l^\top\pi_r\|(\nabla F(\pi_l^\top x(k)) - \mathbf{1}^\top s(k))\mathbf{1}\|_2 &\overset{\mathbf{1}^\top s(k)=\mathbf{1}^\top\nabla f(x(k))}{=} \varepsilon\pi_l^\top\pi_r\|(\mathbf{1}^\top\nabla f(\pi_l^\top x(k)\mathbf{1}) - \mathbf{1}^\top\nabla f(k))\mathbf{1}\|_2 \\ &\le \varepsilon\pi_l^\top\pi_r\|\mathbf{1}^\top\|_2\|f(\pi_l^\top x(k)\mathbf{1}) - \nabla f(x(k))\|_2\|\mathbf{1}\|_2 \\ &\overset{f_i\text{ are }l_i\text{-smooth}}{\le} \varepsilon\bar{l}n\pi_l^\top\pi_r\|\pi_l^\top x(k)\mathbf{1} - x(k)\|_2 \\ &\le c\varepsilon\bar{l}n\pi_l^\top\pi_r\|x(k) - \Pi_l x(k)\|_{\Pi_l}.\end{aligned} \tag{3.19}$$

Finally substituting (3.18) and (3.19) into (3.16) establishes the lemma. □

Lemma 3.6 *Suppose that Assumptions 3.1 and 3.2 hold. Then for all $k \ge 0$,*

$$\begin{aligned}\|s(k+1) - \Pi_r s(k+1)\|_{\Pi_r} \le &c_{31}\|x(k) - \Pi_l x(k)\|_{\Pi_l} + c_{32}\|\Pi_l x(k) - \xi^*\mathbf{1}\|_2 + \\ &c_{33}\|s(k) - \Pi_r s(k)\|_{\Pi_r}\end{aligned}$$

where the constants are (c in (3.12), σ_r in (3.11), and $\bar{l}$ in (3.4))

$$\begin{aligned}
c_{31} &= c^2\bar{l}\|I-\Pi_r\|_2\|L\|_2 + c^2\varepsilon\bar{l}^2\|I-\Pi_r\|_2\|\Pi_r\|_2\\
c_{32} &= c\varepsilon\bar{l}^2\|I-\Pi_r\|_2\|\Pi_r\|_2\\
c_{33} &= \sigma_r + c^2\varepsilon\bar{l}\|I-\Pi_r\|_2.
\end{aligned}$$

Proof. Since $s(k+1) = (I-L^o)s(k) + \nabla f(x(k+1)) - \nabla f(x(k))$ in (3.8), we have

$$\begin{aligned}
&\|s(k+1) - \Pi_r s(k+1)\|_{\Pi_r}\\
&\le \|(I-L^o)s(k) + \nabla f(x(k+1)) - \nabla f(x(k)) - \Pi_r((I-L^o)s(k) + \nabla f(x(k+1)) - \nabla f(x(k)))\|_{\Pi_r}\\
&\le \|(I-L^o)s(k) - \Pi_r s(k)\|_{\Pi_r} + c\|(I-\Pi_r)(\nabla f(x(k+1)) - \nabla f(x(k)))\|_2\\
&\overset{\text{Lemma 3.1}}{\le} \sigma_r\|s(k) - \Pi_r s(k)\|_{\Pi_r} + c\|I-\Pi_r\|_2\|\nabla f(x(k+1)) - \nabla f(x(k))\|_2\\
&\overset{f_i \text{ are } l_i\text{-smooth}}{\le} \sigma_r\|s(k) - \Pi_r s(k)\|_{\Pi_r} + c\bar{l}\|I-\Pi_r\|_2\|x(k+1) - x(k)\|_2.
\end{aligned} \tag{3.20}$$

Since $x(k+1) = (I-L)x(k) - \varepsilon s(k)$ in (3.8), we next bound $\|x(k+1) - x(k)\|_2$ as follows:

$$\begin{aligned}
&c\bar{l}\|I-\Pi_r\|_2\|x(k+1) - x(k)\|_2\\
&\overset{(I-L)\Pi_l = \Pi_l}{=} c\bar{l}\|I-\Pi_r\|_2\| - Lx(k) - \varepsilon s(k) - (I-L)\Pi_l x(k) + \Pi_l x(k)\|_2\\
&\le c\bar{l}\|I-\Pi_r\|_2\| - L(x(k) - \Pi_l x(k))\|_2 + c\varepsilon\bar{l}\|I-\Pi_r\|_2\|s(k)\|_2\\
&\le c^2\bar{l}\|I-\Pi_r\|_2\|L\|_2\|x(k) - \Pi_l x(k)\|_{\Pi_l} + c\varepsilon\bar{l}\|I-\Pi_r\|_2\|s(k)\|_2.
\end{aligned} \tag{3.21}$$

The lemma is proved by substituting $\|s(k)\|_2$ from Lemma 3.3 into (3.21) and then into (3.20). (Note that Assumption 3.1 is needed to apply Lemma 3.3 and Assumption 3.2 to apply Lemma 3.1.) □

The last lemma below is for step 3.

Lemma 3.7 *Let $C \ge 0$ be a nonnegative matrix, $v > 0$ a positive vector, and $\lambda > 0$ a positive real number. If $Cv < \lambda v$, then $\rho(C) < \lambda$.*

Proof. Write $v := [v_1 \cdots v_n]^\top$ and let $D := \mathrm{diag}(v_1, \ldots, v_n)$. Since $v > 0$, D^{-1} exists and define the similarity transformation $\tilde{C} = D^{-1}CD$. Then

$$\tilde{C}\mathbf{1} = D^{-1}CD\mathbf{1} = D^{-1}Cv < D^{-1}\lambda v = \lambda D^{-1}v = \lambda\mathbf{1}.$$

This means that every row sum of $\tilde{C}$ is smaller than λ, i.e. $\|\tilde{C}\|_\infty < \lambda$. Since the spectral radius of every nonnegative matrix is upper bounded by its infinite norm, $\rho(\tilde{C}) \le \|\tilde{C}\|_\infty < \lambda$. Therefore we conclude that $\rho(C) = \rho(\tilde{C}) < \lambda$. □

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1: Suppose that Assumptions 3.1 and 3.2 hold. First, we construct by Lemma 3.1 two special norms $\|\cdot\|_{\Pi_l}$ and $\|\cdot\|_{\Pi_r}$ with constants $\sigma_l, \sigma_r \in (0,1)$, respectively.

Second, according to Lemmas 3.4–3.6, if $\varepsilon < \frac{1}{l\pi_r^\top \pi_l}$ (l in (3.4)) then for all $k \geq 0$,

$$\begin{bmatrix} \|x(k+1) - \Pi_l x(k+1)\|_{\Pi_l} \\ \|\Pi_l x(k+1) - \xi^* \mathbf{1}\|_2 \\ \|s(k+1) - \Pi_r s(k+1)\|_{\Pi_r} \end{bmatrix} \leq C \begin{bmatrix} \|x(k) - \Pi_l x(k)\|_{\Pi_l} \\ \|\Pi_l x(k) - \xi^* \mathbf{1}\|_2 \\ \|s(k) - \Pi_r s(k)\|_{\Pi_r} \end{bmatrix}$$

where the nonnegative matrix C is as follows (c in (3.12), σ_l in (3.10), σ_r in (3.11), m and $\bar{l}$ in (3.4)):

$$C = \begin{bmatrix} \sigma_l + c^2\varepsilon\bar{l}\|\Pi_r - \Pi_l\|_{\Pi_l}\|\Pi_r\|_2 & c\varepsilon\bar{l}\|\Pi_r - \Pi_l\|_{\Pi_l}\|\Pi_r\|_2 & c\varepsilon + c^2\varepsilon\|\Pi_r - \Pi_l\|_{\Pi_l} \\ c\varepsilon\bar{l}n\pi_l^\top\pi_r & (1 - \varepsilon m\pi_l^\top\pi_r) & c\varepsilon\|\Pi_l\|_2 \\ c^2\bar{l}\|I - \Pi_r\|_2\|L\|_2 + c^2\varepsilon\bar{l}^2\|I - \Pi_r\|_2\|\Pi_r\|_2 & c\varepsilon\bar{l}^2\|I - \Pi_r\|_2\|\Pi_r\|_2 & \sigma_r + c^2\varepsilon\bar{l}\|I - \Pi_r\|_2 \end{bmatrix}.$$

It is left to find a bound on ε such that $\rho(C) < 1$. According to Lemma 3.7, it suffices to find a positive vector $v = [v_1\ v_2\ v_3]^\top$ such that $Cv < v$. This inequality yields

$$\varepsilon < \frac{(1-\sigma_l)v_1}{c^2\bar{l}\|\Pi_r - \Pi_l\|_{\Pi_l}\|\Pi_r\|_2 v_1 + c\bar{l}\|\Pi_r - \Pi_l\|_{\Pi_l}\|\Pi_r\|_2 v_2 + c(1+c)\|\Pi_r - \Pi_l\|_{\Pi_l} v_3} \tag{3.22}$$

$$v_2 > \frac{c\bar{l}n\pi_l^\top\pi_r v_1 + c\|\Pi_l\|_2 v_3}{m\pi_l^\top\pi_r} \tag{3.23}$$

$$\varepsilon < \frac{(1-\sigma_r)v_3 - c^2\bar{l}\|I - \Pi_r\|_2\|L\|_2 v_1}{c^2\bar{l}^2\|I - \Pi_r\|_2\|\Pi_r\|_2 v_1 + c\bar{l}^2\|I - \Pi_r\|_2\|\Pi_r\|_2 v_2 + c^2\bar{l}\|I - \Pi_r\|_2 v_3}. \tag{3.24}$$

Since $\varepsilon > 0$, the numerator on the right of (3.24) must be positive, which yields

$$v_1 < \frac{(1-\sigma_r)v_3}{c^2\bar{l}\|I - \Pi_r\|_2\|L\|_2}.$$

This inequality may be satisfied by setting

$$v_3 = c^2\bar{l}\|I - \Pi_r\|_2\|L\|_2 > 0 \tag{3.25}$$

$$v_1 = \frac{1-\sigma_r}{2} > 0. \tag{3.26}$$

Substituting v_1, v_3 into (3.23) yields

$$v_2 > \frac{c\bar{l}n\pi_l^\top \pi_r(1-\sigma_r) + 2c^3\bar{l}\|\Pi_l\|_2\|I-\Pi_r\|_2\|L\|_2}{2m\pi_l^\top \pi_r}$$

which may be satisfied by setting

$$v_2 = \frac{c\bar{l}n\pi_l^\top \pi_r(1-\sigma_r) + 2c^3\bar{l}\|\Pi_l\|_2\|I-\Pi_r\|_2\|L\|_2}{m\pi_l^\top \pi_r} > 0. \tag{3.27}$$

Thus we have found $v = [v_1 \ v_2 \ v_3]^\top > 0$ such that if ε satisfies (3.22) and (3.24), where v_1, v_2, v_3 are in (3.26), (3.27), (3.25), then $Cv < v$, i.e. $\rho(C) < 1$.

Therefore, if $\varepsilon > 0$ is sufficiently small, specifically

$$\varepsilon < \bar{\varepsilon} := \min\{\frac{1}{l\pi_r^\top \pi_l}, \gamma_1, \gamma_2\} \tag{3.28}$$

where γ_1, γ_2 are the right-hand sides of (3.22), (3.24) respectively, then

$$\begin{bmatrix} \|x(k) - \Pi_l x(k)\|_{\Pi_l} \\ \|\Pi_l x(k) - \xi^* \mathbf{1}\|_2 \\ \|s(k) - \Pi_r s(k)\|_{\Pi_r} \end{bmatrix} \to 0 \text{ as } k \to \infty.$$

This implies that $\lim_{k\to\infty} x(k) = \xi^* \mathbf{1}$, i.e. SOA solves the optimization problem. □

Remark 3.4 *(Convergence Speed) In the above proof of Theorem 3.1, if the parameter $\varepsilon \in (0, \bar{\varepsilon})$ with $\bar{\varepsilon}$ in (3.28), then SOA converges to the optimal solution ξ^* of the global cost function. The speed of convergence is governed by the spectrum radius of the 3×3 matrix C, i.e. $\rho(C)$. We refer to $\rho(C)$ as the* convergence factor *of SOA; that is, SOA converges linearly at the rate of $O(\rho(C)^k)$. Note that $\rho(C) < 1$ is equivalent to achieving optimization; and the value of $\rho(C)$ depends on a number of factors related to certain norms, parameter ε, graph topology, and condition number of cost functions. We will demonstrate this latter point in Section 3.5 using simulation examples.*

3.4 Distributed Resource Allocation

In this section we introduce a widely studied distributed constrained optimization problem, and show that it is dual with the optimization problem we have formulated and solved. Hence the distributed algorithm SOA may be adapted as a solution here as well.

Consider a network of n (> 1) agents that cooperatively allocate their local resources to meet a global demand. Each agent i $(\in [1, n])$ has a *state* variable $x_i \in \mathbb{R}$, representing the amount of resource agent i needs to allocate, and has a local cost function $g_i : \mathbb{R} \to \mathbb{R}$. Since it is typical in practice that resource is bounded, each x_i satisfies $x_i \in [\underline{x}_i, \bar{x}_i]$. Let D_i be the resource demand received by agent i; then $D := \sum_{i=1}^n D_i$ is the total demand of resource that the network must allocate. The goal of distributed resource allocation is that the agents cooperatively solves the following problem:

$$\begin{aligned} &\min_{x_1,\dots,x_n \in \mathbb{R}} \sum_{i=1}^n g_i(x_i) \\ &\text{subject to } (\forall i \in [1,n]) x_i \in [\underline{x}_i, \bar{x}_i] \ \& \ \sum_{i=1}^n x_i = D. \end{aligned} \tag{3.29}$$

Let $G(\xi) := \sum_{i=1}^n g_i(\xi_i)$ be the global cost function, where $\xi := [\xi_1 \cdots \xi_n]^\top \in \mathbb{R}^n$. We shall restrict our attention to the case where G has a unique optimal solution $\xi^* = [\xi_1^* \cdots \xi_n^*]^\top$. To ensure this, we again need Assumption 3.1 (on g_i); and in addition, due to boundedness of states x_i, we also need the following assumption.

Assumption 3.3 *The total amount D of resource satisfies $D \in [\sum_{i=1}^n \underline{x}_i, \sum_{i=1}^n \bar{x}_i]$.*

Denote the optimal value of the global cost function G by $G^* = G(\xi^*)$.

Resource Allocation Problem:

Consider a network of n agents interconnected through a digraph $\mathcal{G}$. Suppose that Assumptions 3.1 (on g_i), 3.2, and 3.3 hold and $\xi^* = [\xi_1^* \cdots \xi_n^*]^\top$ is the (unique) optimal solution to the constrained optimization problem in (3.29). Design a distributed algorithm such that

$$(\forall i \in [1,n])(\forall x_i(0) \in \mathbb{R}) \lim_{k\to\infty} x_i(k) = \xi_i^*.$$

In the following, we consider the dual problem of (3.29) and transform it to the form of the optimization problem (3.1). Then the distributed algorithm SOA that solves the optimization problem can be adapted to solve the resource allocation problem.

Define the Lagrange function of (3.29) as

$$L(x,\lambda) = \sum_{i=1}^n g_i(x_i) + \lambda(\sum_{i=1}^n x_i - D) \tag{3.30}$$

where $x := [x_1 \cdots x_n]^\top \in [\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_n, \bar{x}_n] =: \mathcal{X}$ and $\lambda \in \mathbb{R}$ is the Lagrange multiplier. Then the dual problem of (3.29) is

$$\max_{\lambda \in \mathbb{R}} \inf_{x \in \mathcal{X}} L(x,\lambda). \tag{3.31}$$

Note that

$$\begin{aligned}\inf_{x\in\mathcal{X}} L(x,\lambda) &= \inf_{x\in\mathcal{X}} \sum_{i=1}^{n}(g_i(x_i)+\lambda x_i)-\lambda D\\ &= \sum_{i=1}^{n} \inf_{x_i\in[\underline{x}_i,\bar{x}_i]}(g_i(x_i)+\lambda x_i)-\lambda D\\ &= \sum_{i=1}^{n} -\sup_{x_i\in[\underline{x}_i,\bar{x}_i]} -(g_i(x_i)+\lambda x_i)-\lambda D\\ &= \sum_{i=1}^{n} -g_i^*(-\lambda)-\lambda D\end{aligned}$$

where $g_i^*(\lambda)=\sup_{x_i\in[\underline{x}_i,\bar{x}_i]}(\lambda x_i - g_i(x_i))$ is the conjugate function of $g_i(x_i)$. Since g_i is strongly convex and has a Lipschitz-continuous gradient (Assumption 3.1), $g_i^*(\lambda)$ exists (i.e. the supremum is attainable) and also enjoys strong convexity and Lipschitz-continuous gradient. Now let

$$f_i(\lambda) := g_i^*(-\lambda)+\lambda D_i. \tag{3.32}$$

This f_i satisfies Assumption 3.1. Then the dual problem (3.31) is transformed into:

$$\max_{\lambda\in\mathbb{R}} \sum_{i=1}^{n}(-f_i(\lambda)) = -\min_{\lambda\in\mathbb{R}} \sum_{i=1}^{n}(f_i(\lambda)).$$

The latter without the minus sign is in the same form as (3.1):

$$\begin{aligned}&\min_{\lambda_1,\ldots,\lambda_n\in\mathbb{R}} \sum_{i=1}^{n} f_i(\lambda_i)\\ &\text{subject to } \lambda_1=\cdots=\lambda_n.\end{aligned} \tag{3.33}$$

Remark 3.5 *Owing to Assumptions 3.1 (on g_i) and 3.3, strong duality holds between (3.33) and (3.29). This means that the optimal solutions $[\lambda^*\cdots\lambda^*]^\top$ of (3.33) and $[\xi_1^*\cdots\xi_n^*]^\top$ of (3.29) are related by*

$$(\forall i\in[1,n])g_i(\xi_i^*)+g_i^*(-\lambda^*) = -\xi_i^*\lambda^*$$

and the optimal values F^* of (3.33) and G^* of (3.29) are related by $F^*=-G^*$. Hence an optimal solution to (3.33) provides an optimal solution to (3.29).

To solve (3.33) by SOA, we need to compute the gradient of f_i. From (3.32) we derive

$$\nabla f_i(\lambda) = -\nabla g_i^*(-\lambda) + D_i.$$

Since the gradient of the conjugate function g_i^* is given by $\nabla g_i^*(\lambda) = \text{argmax}_{x_i \in [\underline{x}_i, \bar{x}_i]}\{\lambda x_i - g_i(x_i)\}$, we derive

$$\begin{aligned}\nabla f_i(\lambda) &= -\text{argmin}_{x_i \in [\underline{x}_i, \bar{x}_i]}\{\lambda x_i + g_i(x_i)\} + D_i \\ &= \begin{cases} \nabla^{-1} g_i(\lambda) + D_i, & \text{if } \underline{x}_i \le \nabla^{-1} g_i(\lambda) \le \bar{x}_i \\ \underline{x}_i + D_i, & \text{if } \nabla^{-1} g_i(\lambda) < \underline{x}_i \\ \bar{x}_i + D_i, & \text{if } \nabla^{-1} g_i(\lambda) > \bar{x}_i \end{cases}\end{aligned}$$

Substituting $\nabla f_i(\lambda)$ into (3.7), we obtain from SOA the following (specialized) algorithm to solve (3.33):

$$\lambda_i(k+1) = \lambda_i(k) + \sum_{j \in \mathcal{N}_i} a_{ij}(\lambda_j(k) - \lambda_i(k)) - \varepsilon s_i(k) \tag{3.34}$$

$$x_i(k+1) = \text{argmin}_{x_i \in [\underline{x}_i, \bar{x}_i]}\{\lambda_i(k+1) x_i + g_i(x_i)\} \tag{3.35}$$

$$s_i(k+1) = (1 - \sum_{j \in \mathcal{N}_i^o} a_{ji}) s_i(k) + \sum_{j \in \mathcal{N}_i} a_{ij} s_j(k) + \Big(x_i(k) - x_i(k+1)\Big). \tag{3.36}$$

The parameter ε is a positive real number. We call this algorithm *Surplus-based Resource Allocation Algorithm (SRAA).*

Following the initialization of SOA, $\lambda_i(0)$ can be arbitrary real numbers, whereas

$$\begin{aligned} x_i(0) &= \text{argmin}_{x_i \in [\underline{x}_i, \bar{x}_i]}\{\lambda_i(0) x_i + g_i(x_i)\} \\ s_i(0) &= D_i - \text{argmin}_{x_i \in [\underline{x}_i, \bar{x}_i]}\{\lambda_i(0) x_i + g_i(x_i)\}. \end{aligned}$$

In fact, the initialization of SRAA can be simpler: namely $x_i(0) = 0$ and $s_i(0) = D_i$. The updates with or without computing $\text{argmin}_{x_i \in [\underline{x}_i, \bar{x}_i]}\{\lambda_i(0) x_i + g_i(x_i)\}$ become the same after the first iteration due to the special form of $\nabla f_i(\lambda)$. In any case, note from (3.36) that $\mathbf{1}^\top(x(k) + s(k))$ is a constant. Hence if $s(k) \to 0$ then $\mathbf{1}^\top x(k) \to \mathbf{1}^\top(x(0) + s(0)) = D$. That is, $x_i(k)$ jointly satisfy the total demanded resource in an asymptotic fashion.

The main result of this section is the following.

Theorem 3.2 *Suppose that Assumptions 3.1 (on g_i), 3.2, and 3.3 hold. If the parameter $\varepsilon > 0$ is sufficiently small, then SRAA solves the resource allocation problem.*

Proof. Let Assumptions 3.1 (on g_i), 3.2, 3.3 hold, and assume that $\varepsilon > 0$ is sufficiently small. Then it follows from strong duality and Theorem 3.1 that $\|\Pi_l \lambda(k) - \lambda^* \mathbf{1}\|_2 \to 0$. This implies $\pi_l^\top \lambda(k) \to \lambda^*$, and hence $F(\pi_l^\top \lambda(k)) \to F^*$. Note again by strong duality that $F^* = -G^* = -L(\xi^*, \lambda)$ for every $\lambda \in \mathbb{R}$, where $L(\cdot, \cdot)$ is the Lagrangian function given in (3.30). Consequently

$$\begin{aligned}
F(\pi_l^\top \lambda(k)) - F^* &= L(\xi^*, \pi_l^\top \lambda(k)) - \inf_{x \in \mathcal{X}} L(x, \pi_l^\top \lambda(k)) \\
&= L(\xi^*, \pi_l^\top \lambda(k)) - L(x(k), \pi_l^\top \lambda(k)) \\
&\geq \nabla L(x(k), \pi_l^\top \lambda(k))(\xi^* - x(k)) + \frac{m}{2}\|\xi^* - x(k)\|_2^2 \\
&\geq \frac{m}{2}\|x(k) - \xi^*\|_2^2.
\end{aligned}$$

The first inequality above is due to m-strong convexity of G following Assumption 3.1 (on g_i); and the second inequality uses the first-order necessary condition for constrained minimization problems. By the above inequality and the fact that $F(\pi_l^\top \lambda(k)) \to F^*$, we derive $x(k) \to \xi^*$. This proves that the resource allocation problem is solved. □

3.5 Simulation Examples

In this section we illustrate by simulation the convergence properties of SOA for the optimization problem, as well as SRAA for the resource allocation problem.

Example 3.4 *We demonstrate the influences of graph topologies and condition number of cost functions on the convergence speed of SOA. First, we investigate the influence of graph topologies, especially for different densities of edges. Consider a digraph of* $n = 100$ *nodes; we choose uniformly at random 10%, 30%, and 50% of directed edges from all possible* $n(n-1)$ *edges. We take only those digraphs that are strongly connected, and set uniform weights* $\frac{1}{100}$*. For cost functions we consider*

$$f_i(\xi) = a_i \xi^2 + b_i \xi + c_i + d_i \log(1 + \mathrm{e}^{-\xi})$$

where a_i, b_i, c_i, d_i *are chosen uniformly at random from the open interval* $(0, 1)$*. Such* f_i *is* $(2a_i + \delta)$*-strongly convex (*$\delta > 0$ *is a small number) and* $(2a_i + 0.25d_i)$*-smooth. Then the global cost function* $F(\xi) = \sum_{i=1}^n f_i(\xi)$ *is also strongly convex and smooth, and let* ξ^* *be the (unique) optimal solution.*

Fig. 3.5 displays the curves of the error $\frac{1}{n}\|x(k) - \xi^* \mathbf{1}\|_2$ *with respect to the above chosen three different densities of edges; each plotted point is the mean value of the error over* 100 *random digraphs of the respective densities, and each component of the initial state vector*

$x(0)$ is chosen uniformly at random from the closed interval $[-10, 10]$. It is observed that the denser the digraph, the faster SOA converges to the optimal solution ξ^*.
Next, we investigate the influence of the condition number of cost functions on the convergence speed of SOA. For this, we consider cost functions

$$f_i(\xi) = a\xi^2 + b_i\xi + c_i + d\log(1 + \mathrm{e}^{-\xi})$$

where b_i, c_i are again chosen uniformly at random from the open interval $(0, 1)$, but a, d are the same for all f_i. Thus f_i, as well as the global cost function F, all have the condition number $Q = \frac{2a+0.25d}{2a+\delta}$ ($\delta > 0$ is a small number). Fix $\delta = a = 0.01$ and choose three values $0.72, 7.92, 79.92$ for d; then the condition numbers Q are $10, 100, 1000$.
To reduce the influence of digraph topology, we apply SOA for different cost functions on the same digraphs of 100 nodes and 10% of directed edges chosen uniformly at random. Fig. 3.6 displays the curves of the error $\frac{1}{n}\|x(k) - \xi^*\mathbf{1}\|_2$ with respect to three different condition numbers of the cost functions; each plotted point is the mean value of the error over 100 random digraphs. It is observed that the smaller the condition number (i.e. better conditioned), the faster SOA converges to the optimal solution ξ^*.

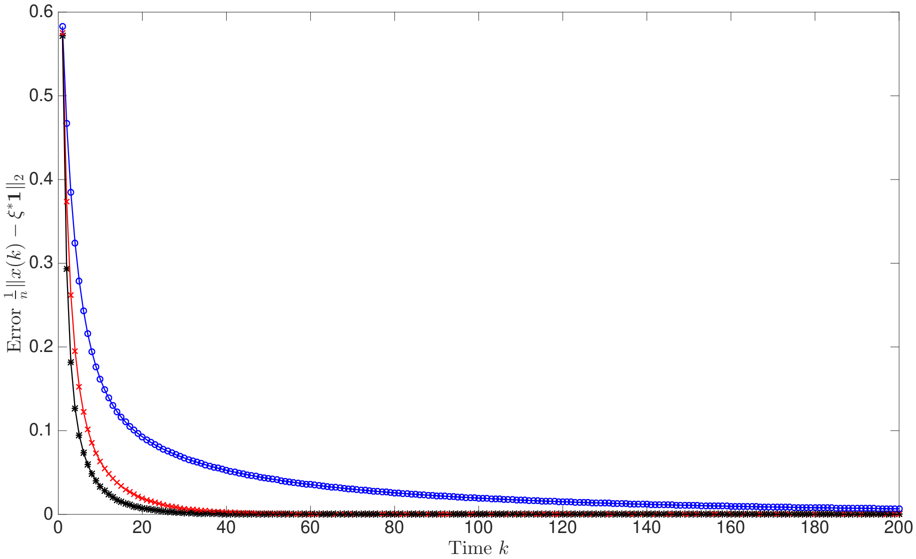


Figure 3.5: Convergence speed with respect to 10% (blue ∘), 30% (red ×), and 50% (black ∗) of directed edges

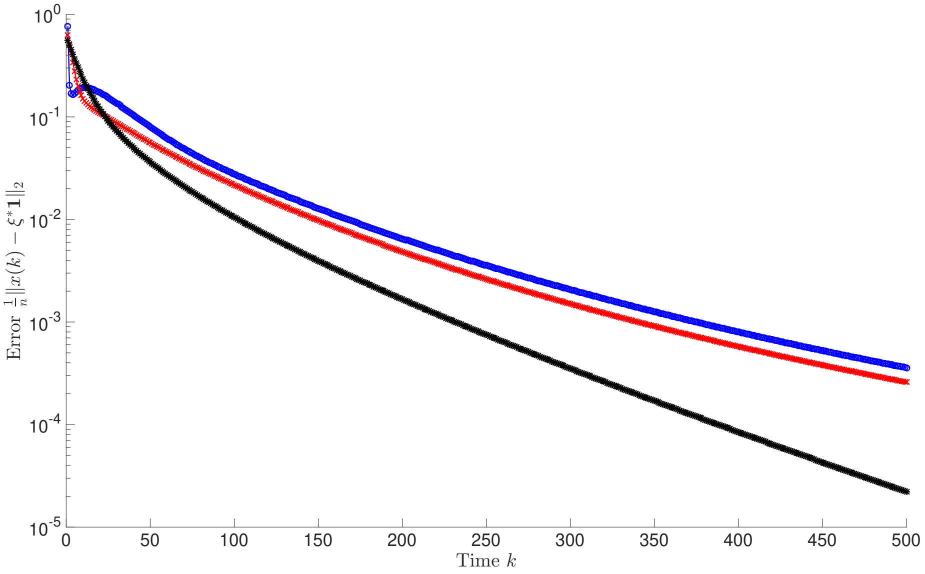


Figure 3.6: Convergence speed with respect to condition numbers of cost functions: 10 (black), 100 (red), and 1000 (blue). Vertical axis is in logarithmic scale for clear comparison.

Table 3.1: Generator parameters of IEEE 14-bus test system

Generator	α_i (\$/MW^2h)	β_i (\$/MWh)	γ_i (\$/h)	$[\underline{x}_i, \bar{x}_i]$ (MW)
1 (bus 1)	0.04	2.0	12	$[0, 80]$
2 (bus 2)	0.03	3.0	20	$[0, 90]$
3 (bus 3)	0.035	4.0	15	$[0, 70]$
4 (bus 6)	0.03	4.0	23	$[0, 70]$
5 (bus 8)	0.04	2.5	16	$[0, 80]$

Example 3.5 *In this example, we apply SRAA to solve a distributed resource allocation problem in power networks. Specifically, consider the IEEE 14-bus test system as displayed in Fig. 3.7; the power demands at individual buses are (unit: MW)*

$$0,\ 21.7,\ 66.2,\ 47.8,\ 7.6,\ 11.2,\ 0,\ 0,\ 29.5,\ 9,\ 3.5,\ 6.1,\ 13.5,\ 14.9.$$

Thus the total demand is $D = 231$. To satisfy the demand, there are 5 generators at buses 1,2,3,6,8; the associated cost functions are quadratic: $g_i(x_i) = \alpha_i x_i^2 + \beta_i x_i + \gamma_i$, where x_i is the power (MW) generated by generator i. These quadratic functions satisfy Assumption 3.1.

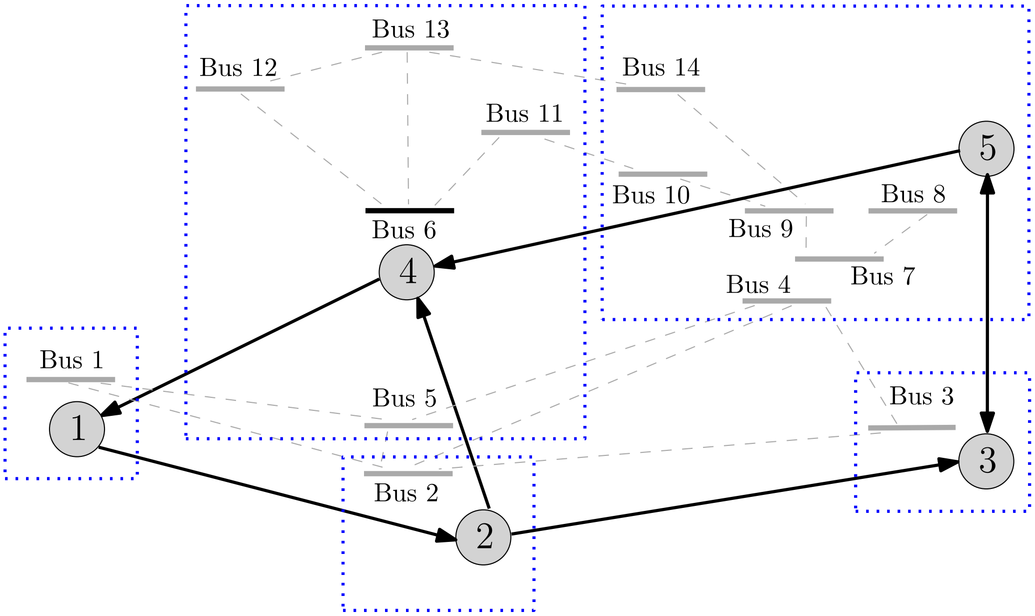


Figure 3.7: IEEE 14-bus test system with 5 generators (denoted by circles) and 14 demands (imposed at Buses)

The parameters $\alpha_i, \beta_i, \gamma_i$ and the ranges $[\underline{x}_i, \bar{x}_i]$ of x_i are given in Table 3.1.
Since the total demand $D \in [\sum_{i=1}^{5} \underline{x}_i, \sum_{i=1}^{5} \bar{x}_i] = [0, 390]$, Assumption 3.3 holds. The communication digraph among the 5 generators is displayed in Fig. 3.7; this digraph is strongly connected, and hence Assumption 3.2 holds. Thus the resource allocation problem (aka. economic dispatching problem in this context) is to solve $\min_{x_1,\ldots,x_5 \in \mathbb{R}} \sum_{i=1}^{5} g_i(x_i)$ such that each x_i is in the respective range and the total generated power $\sum_{i=1}^{5} x_i$ meets the total demand $231MW$.
We apply SRAA to solve this problem. Let the weights $a_{ij} = \frac{1}{|\mathcal{N}_i|+1}$, the parameter $\varepsilon = 0.01$, the initial $\lambda_i(0)$ drawn uniformly at random from $[-10, 10]$, and the inital $x_i(0) = 0$. Finally to initialize $s_i(0)$, suppose that each generator is in charge of a certain area (areas are

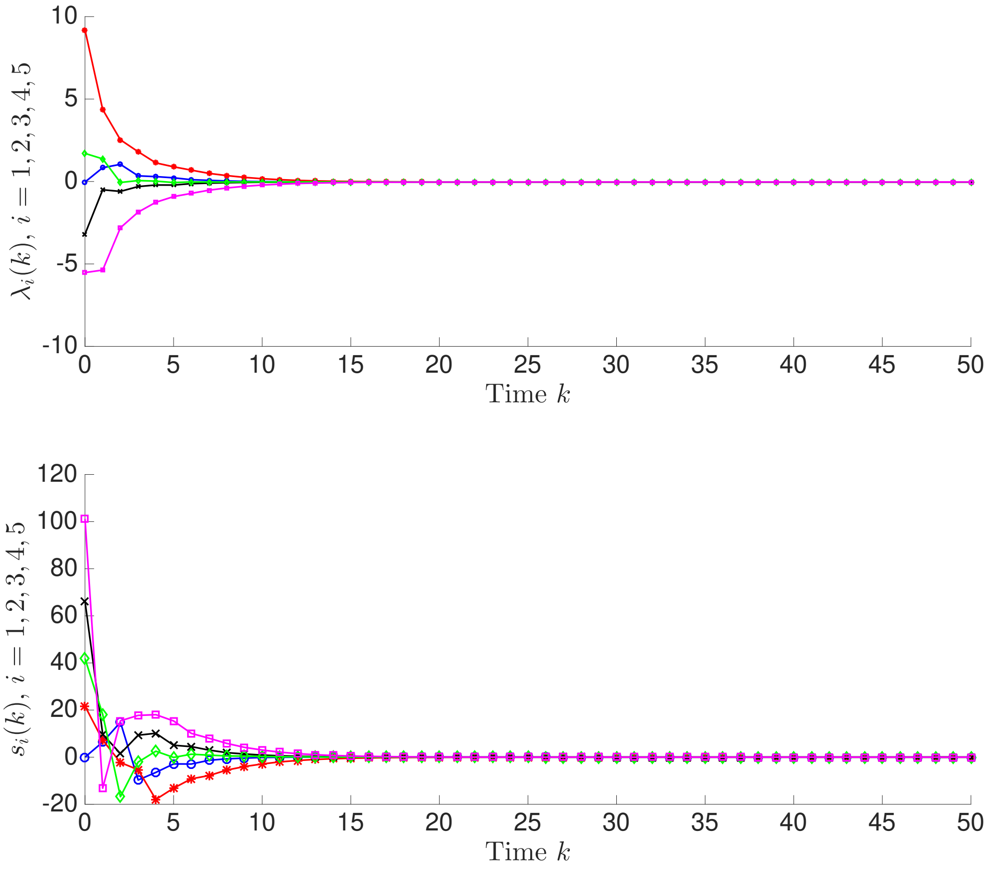


Figure 3.8: IEEE 14-bus test system: convergence of λ_i and s_i

displayed as dotted boxes in Fig. 3.7); thereby naturally:

$$
\begin{aligned}
s_1(0) &= D_1 = 0\\
s_2(0) &= D_2 = 21.7\\
s_3(0) &= D_3 = 66.2\\
s_4(0) &= D_4 = 7.6 + 11.2 + 3.5 + 6.1 + 13.5 = 41.9\\
s_5(0) &= D_5 = 47.8 + 0 + 0 + 29.5 + 9 + 14.9 = 101.2.
\end{aligned}
$$

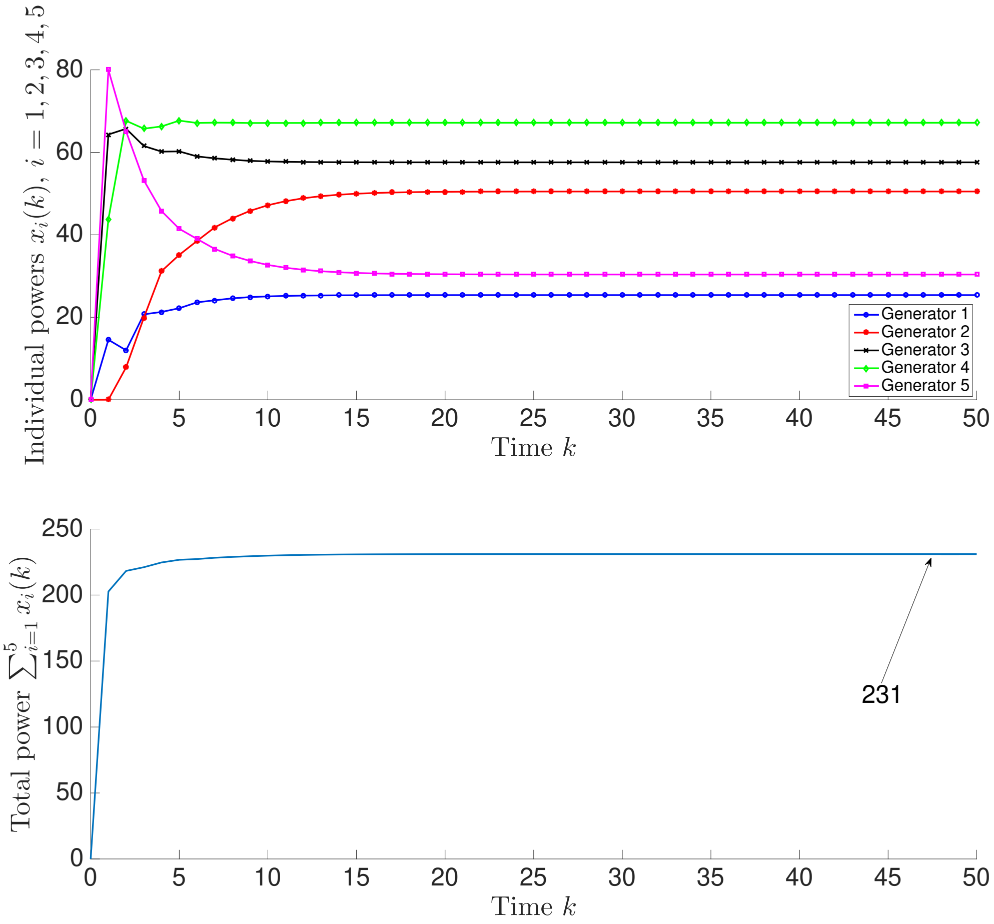


Figure 3.9: IEEE 14-bus test system: convergence of x_i $(i = 1, \dots, 5)$ without range violation and the total generated power meeting the total demand

The simulation results are displayed in Figs. 3.8 and 3.9. Observe that surplus variables $s_i(k)$ diminish from the initialized values (demands) to zero, while states $x_i(k)$ converge from zero initial values to the optimal solution of the resource allocation problem. Moreover, all $x_i(k)$ stay in their respective ranges, and the sum of $x_i(k)$, namely the total generated power converges (rapidly and smoothly) to the required total demand $231MW$.

3.6 Notes and References

The surplus-based optimization algorithm (SOA) is originated in

- R. Xin and U. Khan, A linear algorithm for optimization over directed graphs with geometric convergence, IEEE Control Systems Letters, vol.2, pp.315–320, 2018
- S. Pu, W. Shi, J. Xu, and A. Nedic, Push-pull gradient methods for distributed optimization in networks, IEEE Transactions on Automatic Control, vol.66, pp.1–16, 2021

Extension to time-varying digraphs is reported in

- F. Saadatniaki, R. Xin, and U. Khan, Decentralized optimization over time-varying directed graphs with row and column stochastic matrices, IEEE Transactions on Automatic Control, vol.65, pp.4769–4780, 2020

The Surplus-based Resource Allocation Algorithm (SRAA) is from

- J. Zhang, K. You, and K. Cai, Distributed conjugate gradient tracking for resource allocation in unbalanced networks, IEEE Transactions on Signal Processing, vol.68, pp.2186–2198, 2020

A variant that addresses time-varying networks is in

- Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, and M. Fu, A distributed algorithm for resource allocation over dynamic digraphs, IEEE Transactions on Signal Processing, vol.65, pp.2600–2612, 2017

The proof techniques for Lemmas 3.1 and 3.7 are from

- R.A. Horn and C.R. Johnson, Matrix Analysis, 2nd ed., Cambridge University Press, 2013

The properties of smooth and strongly convex functions used in Section 3.3, dual functions and strong duality in Section 3.4, and the material on convex optimization in Appendix below are standard, and can be found in textbooks e.g.

- Y. Nesterov, Lecture on Convex Optimization, 2nd ed., Springer, 2018
- S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004

3.7 Appendix: Convex Optimization

In this appendix we present a brief introduction of basic convexity definitions, as well as a useful result that was used in proving the convergence of SOA in Section 3.3.

Throughout this appendix we consider a continuously differentiable function $F : \mathbb{R} \to \mathbb{R}$. We say that F is *convex* if

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) F(\xi_2) \geq F(\xi_1) + \nabla F(\xi_1)(\xi_2 - \xi_1); \tag{3.37}$$

F is *strictly convex* if

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) \xi_1 \neq \xi_2 \Rightarrow F(\xi_2) > F(\xi_1) + \nabla F(\xi_1)(\xi_2 - \xi_1); \tag{3.38}$$

and (recall that) F is *m-strongly convex* for some $m > 0$ if

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) F(\xi_2) \geq F(\xi_1) + \nabla F(\xi_1)(\xi_2 - \xi_1) + \frac{m}{2}\|\xi_2 - \xi_1\|^2.$$

In this appendix, $\|\cdot\|$ denotes an arbitrary vector norm. By definition the relation among these three convexity concepts is: strong convexity ⇒ strict convexity ⇒ convexity.

Lemma 3.8 *Consider an optimization problem*

$$\min_{\xi \in \mathbb{R}} \; F(\xi). \tag{3.39}$$

(i) *If F is convex and $\nabla F(\xi^*) = 0$, then ξ^* is a global optimal solution.*

(ii) *If F is strictly convex and $\nabla F(\xi^*) = 0$, then ξ^* is the unique global optimal solution.*

(iii) *If F is strongly convex, then the global optimal solution ξ^* exists and is unique.*

Proof. For (i), it follows from the definition of convexity (3.37) that for an arbitrary $\xi \in \mathbb{R}$ we have

$$F(\xi) \geq F(\xi^*) + \nabla F(\xi^*)(\xi - \xi^*) = F(\xi^*).$$

This proves that ξ^* is a global optimal solution of (3.39).

For (ii), since strict convexity implies convexity, we know from (i) that ξ^* is a global optimal solution. Suppose that $\tilde{\xi}(\neq \xi^*)$ is another global optimal solution, i.e. $F(\tilde{\xi}) = F(\xi^*)$. By the definition of strict convexity (3.38), however

$$F(\tilde{\xi}) > F(\xi^*) + \nabla F(\xi^*)(\tilde{\xi} - \xi^*) = F(\xi^*).$$

Hence $\tilde{\xi}$ cannot be a global solution, and the uniqueness of ξ^* ensues.

For (iii), let $\bar{\xi} \in \mathbb{R}$ and consider the set $\mathcal{S} := \{\xi \in \mathbb{R} \mid f(\xi) \leq f(\bar{\xi})\}$. Note that the optimization problem (3.39) is equivalent to the following:

$$\min_{\xi \in \mathcal{S}} \ F(\xi). \tag{3.40}$$

Since F is strongly convex with a parameter $m > 0$, for an arbitrary $\xi \in \mathcal{S}$ we have

$$\begin{aligned} & F(\bar{\xi}) \geq F(\xi) \geq F(\bar{\xi}) + \nabla F(\bar{\xi})(\xi - \bar{\xi}) + \frac{m}{2}\|\xi - \bar{\xi}\|^2 \\ \Rightarrow & \frac{m}{2}\|\xi - \bar{\xi}\|^2 \leq \nabla F(\bar{\xi})(\bar{\xi} - \xi) \\ \Rightarrow & \|\xi - \bar{\xi}\| \leq \frac{2}{m}\|\nabla F(\bar{\xi})\|. \end{aligned}$$

Thus the set $\mathcal{S}$ is a closed and bounded interval, i.e. a compact set. Moreover since F is continuously differentiable (thus continuous), it follows from the Weierstrass extreme value theorem that an optimal solution ξ^* of (3.40) (and of (3.39)) exists.

Being an optimal solution of (3.39), ξ^* satisfies $\nabla F(\xi^*) = 0$. Since strong convexity implies strict convexity, we derive from (ii) that ξ^* is the unique global optimal solution. □

Recall that a convex function $F : \mathbb{R} \to \mathbb{R}$ is *l-smooth* for some $l > 0$ if

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) \|\nabla F(\xi_1) - \nabla F(\xi_2)\| \leq l\|\xi_1 - \xi_2\|.$$

Lemma 3.9 *The following are equivalent:*

F is l-smooth

$$(\forall \xi_1, \xi_2 \in \mathbb{R}) 0 \leq F(\xi_2) - F(\xi_1) - \nabla F(\xi_1)(\xi_2 - \xi_1) \leq \frac{l}{2}\|\xi_1 - \xi_2\|^2 \tag{3.41}$$

$$(\forall \xi_1, \xi_2 \in \mathbb{R})(\nabla F(\xi_1) - \nabla F(\xi_2))(\xi_1 - \xi_2) \geq \frac{1}{l}\|\nabla F(\xi_1) - \nabla F(\xi_2)\|^2 \tag{3.42}$$

Proof. Let $\xi_1, \xi_2 \in \mathbb{R}$. We will prove: l-smoothness ⇒ (3.41) ⇒ (3.42) ⇒ l-smoothness.

First assume that F is l-smooth. To prove (3.41), note that the left inequality is directly from the definition of convexity (3.37). To see the inequality on the right, note that

$$F(\xi_2) - F(\xi_1) - \nabla F(\xi_1)(\xi_2 - \xi_1) = \int_0^1 (\nabla F(\xi_1 + \tau(\xi_2 - \xi_1)) - \nabla F(\xi_1))(\xi_2 - \xi_1) d\tau.$$

By Cauchy-Schwarz inequality and the definition of l-smoothness,

$$F(\xi_2) - F(\xi_1) - \nabla F(\xi_1)(\xi_2 - \xi_1) \leq \int_0^1 l\tau\|\xi_2 - \xi_1\|^2 d\tau = \frac{l}{2}\|\xi_2 - \xi_1\|^2.$$

Next assume that (3.41) holds. To prove (3.42), let $\xi_0 \in \mathbb{R}$ and define $\phi(\xi) := F(\xi) - \nabla F(\xi_0)\xi$. Thus $\phi(\cdot)$ is also l-smooth and its optimal solution is $\xi^* = \xi_0$. Hence

$$\phi(\xi^*) = \min_{\xi_1 \in \mathbb{R}} \phi(\xi_1) \overset{(3.41)}{\leq} \min_{\xi_1 \in \mathbb{R}} \left(\phi(\xi_2) + \nabla\phi(\xi_2)(\xi_1 - \xi_2) + \frac{l}{2}\|\xi_1 - \xi_2\|^2 \right).$$

Again by Cauchy-Schwarz inequality we obtain

$$\phi(\xi^*) \leq \min_{r \geq 0} \left(\phi(\xi_2) - r\|\nabla\phi(\xi_2)\| + \frac{l}{2}r^2 \right) = \phi(\xi_2) - \frac{1}{2l}\|\nabla\phi(\xi_2)\|^2.$$

Substituting $\phi(\xi_2) = F(\xi_2) - \nabla F(\xi_2)\xi_0$ and $\nabla\phi(\xi_2) = \nabla F(\xi_2) - \nabla F(\xi_0)$ into the above inequality yields

$$F(\xi_1) + \nabla F(\xi_1)(\xi_2 - \xi_1) + \frac{1}{2l}\|\nabla F(\xi_1) - \nabla F(\xi_2)\|^2 \leq F(\xi_2).$$

Adding two copies of the above inequality and exchanging ξ_1, ξ_2 lead to (3.42).

Finally assume that (3.42) holds. Applying Cauchy-Schwarz inequality yields $\|\nabla F(\xi_1) - \nabla F(\xi_2)\| \leq l\|\xi_1 - \xi_2\|$, namely F is l-smooth. □

When F is both m-strongly convex and l-smooth (thus necessarily $m \leq l$), the following result was used in the proof of Lemma 3.2 (in part to show the convergence of SOA in Section 3.3).

Lemma 3.10 *If F is m-strongly convex and l-smooth, then*

$$(\forall \xi_1, \xi_2 \in \mathbb{R})(\nabla F(\xi_1) - \nabla F(\xi_2))(\xi_1 - \xi_2) \leq \frac{ml}{m+l}\|\xi_1 - \xi_2\|^2 + \frac{1}{m+l}\|\nabla F(\xi_1) - \nabla F(\xi_2)\|^2 \tag{3.43}$$

Proof. Suppose that F is m-strongly convex and l-smooth. Let $\phi(x) := F(x) - \frac{1}{2}m\|x\|^2$. Then $\nabla\phi(x) = \nabla F(x) - mx$ and it is verified that $\phi(x)$ is convex. Moreover, for arbitrary $\xi_1, \xi_2 \in \mathbb{R}$, since

$$\begin{aligned} \phi(\xi_2) &= F(\xi_2) - \frac{1}{2}m\|\xi_2\|^2 \\ &\overset{(3.41)}{\leq} F(\xi_1) + \nabla F(\xi_1)(\xi_2 - \xi_1) + \frac{l}{2}\|\xi_1 - \xi_2\|^2 - \frac{1}{2}m\|\xi_2\|^2 \\ &= \phi(\xi_1) + \nabla\phi(\xi_1)(\xi_2 - \xi_1) + \frac{l-m}{2}\|\xi_1 - \xi_2\|^2 \end{aligned}$$

it follows again from (3.41) that $\phi(x)$ is $(l-m)$-smooth. Note that $m \leq l$ holds always. If $m = l$

then (3.43) holds. If $m < l$, then by (3.42) we derive

$$(\nabla\phi(\xi_1) - \nabla\phi(\xi_2))(\xi_1 - \xi_2) \geq \frac{1}{l-m}\|\nabla\phi(\xi_1) - \nabla\phi(\xi_2)\|^2$$

Substituting $\nabla\phi(\cdot)$ into the above inequality yields (3.43). □

Part III
Spanning Tree Digraphs: Consensus and Synchronization

This part introduces distributed consensus and synchronization over digraphs. The necessary graphical condition for solving these two problems is that digraphs contain a spanning tree. The type of Laplacian matrices involved in these two problems is again the standard Laplacian matrices. For agent dynamics, continuous-time linear time-invariant systems are considered.

Chapter 4

Consensus

In this chapter we introduce the problem of distributed consensus. This problem can be viewed as a generalized version of averaging in Chapter 2, in that as long as the networked agents reach an agreement, the agreed value can be arbitrary and need not be the initial average.

Consensus has been studied in a variety of disciplines, including social behaviors, political science, biology, computer animation, and robotics. For example, reaching consensus among a group of people is one of the central investigation in social/political opinion dynamics. In natural/animated group behaviors such as bird flocking and fish schooling, consensus on heading angles and velocities among group members is key. As a final example, rendezvous of a team of mobile robots means that these robots reach consensus on their meeting locations.

Modeling the interacting agents by digraphs, we show that a necessary graphical condition to achieve consensus is that the digraph contains a *spanning tree*, namely there exists (at least) one agent that can reach all the other agents. This is intuitively evident, as for all agents to reach consensus, at least some agent's information need to be spread across the whole network. Under this graphical condition, we present a distributed algorithm that achieves consensus.

4.1 Problem Statement

Consider a network of n (> 1) agents. Each agent i $(\in [1, n])$ has a *state* variable $x_i(t) \in \mathbb{R}$, where $t \geq 0$ is a nonnegative real number and denotes the *continuous* time. Each agent i is modeled as a single integrator:

$$\dot{x}_i(t) := \frac{dx_i(t)}{dt} = u_i(t) \tag{4.1}$$

where $u_i(t) \in \mathbb{R}$ is a real-valued control input. For simplicity we often write (4.1) as $\dot{x}_i = u_i$ (omitting the time).

For agents modeled by (4.1), we say that an algorithm is *distributed* if every agent i's control input $u_i(t)$ is based only on the information received from $\mathcal{N}_i$.

Consensus Problem:

Consider a network of n agents (4.1) interconnected through a digraph $\mathcal{G}$. Design a distributed algorithm such that

$$(\forall i \in [1, n])(\forall x_i(0) \in \mathbb{R})(\exists c \in \mathbb{R}) \lim_{t\to\infty} x_i(t) = c.$$

We say that c is the *consensus value*. As we shall see, this c depends on the initial states $x_i(0)$ as well as the graph topology.

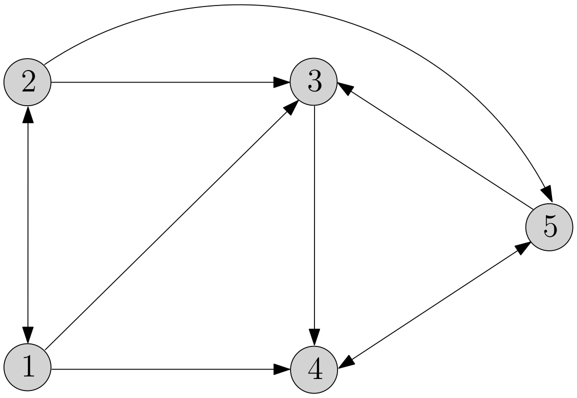


Figure 4.1: Illustrating example of consensus problem with five agents

Example 4.1 *We provide an example to illustrate the consensus problem. As displayed in Fig. 4.1, five agents are interconnected through a digraph. The neighbor sets of the agents are* $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_2 = \{1\}$, $\mathcal{N}_3 = \{1, 2, 5\}$, $\mathcal{N}_4 = \{1, 3, 5\}$, *and* $\mathcal{N}_5 = \{2, 4\}$.
Suppose that the initial states of the agents are $x_1(0) = 1$, $x_2(0) = 2$, $x_3(0) = 3$, $x_4(0) = 4$, $x_5(0) = 5$. *The consensus problem is to design a distributed algorithm such that each agent's state asymptotically converges to the same value. This consensus value by no means needs to be the initial average (which is* 3*); hence consensus problem includes averaging as a special case.*

A necessary graphical condition for solving the consensus problem is given below.

Proposition 4.1 *Suppose that there exists a distributed algorithm that solves the consensus problem. Then the digraph contains a spanning tree.*

Proof. The proof is by contradiction. Suppose that the digraph $\mathcal{G}$ does *not* contain a spanning tree. Then it follows from Theorem 1.1 that $\mathcal{G}$ has at least two (distinct) closed strong components (say) $\mathcal{G}_1, \mathcal{G}_2$. In this case, consider an initial condition such that the agents in $\mathcal{G}_1$ have initial state $c_1 \in \mathbb{R}$, those in $\mathcal{G}_2$ have $c_2 \in \mathbb{R}$, and $c_1 \neq c_2$. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ are closed, information cannot be communicated from one to the other. Consequently, there exists no distributed algorithm that can solve the consensus problem. □

Owing to Proposition 4.1, we shall henceforth assume that the digraph contains a spanning tree.

Assumption 4.1 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning tree.*

4.2 Distributed Algorithm

Example 4.2 *Consider again Example 4.1. To achieve consensus, a natural idea is that each agent 'pursuits' the state values received from neighbors. Namely, for $i \in [1,5]$*

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} (x_j - x_i).$$

Concretely, based on the neighbor sets of the agents (see Fig. 4.1):

$$\begin{aligned}
\dot{x}_1 &= (x_2 - x_1) \\
\dot{x}_2 &= (x_1 - x_2) \\
\dot{x}_3 &= (x_1 - x_3) + (x_2 - x_3) + (x_5 - x_3) \\
\dot{x}_4 &= (x_1 - x_4) + (x_3 - x_4) + (x_5 - x_4) \\
\dot{x}_5 &= (x_2 - x_5) + (x_4 - x_5).
\end{aligned}$$

Write the above in vector form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & -3 & 0 & 1 \\ 1 & 0 & 1 & -3 & 1 \\ 0 & 1 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}.$$

Observe that the matrix above has zero row sums, and is indeed the minus of the standard Laplacian matrix (i.e. $-L$) with weights $a_{ij} = 1$ for all existing edges (v_j, v_i).
With the initial condition in Example 4.1 (i.e. $x_i(0) = i$ for $i = 1, \ldots, 5$), Fig. 5.3 displays that all states converge to the same value, namely consensus. Note that the consensus value 1.5 *is different from the initial average* 3.

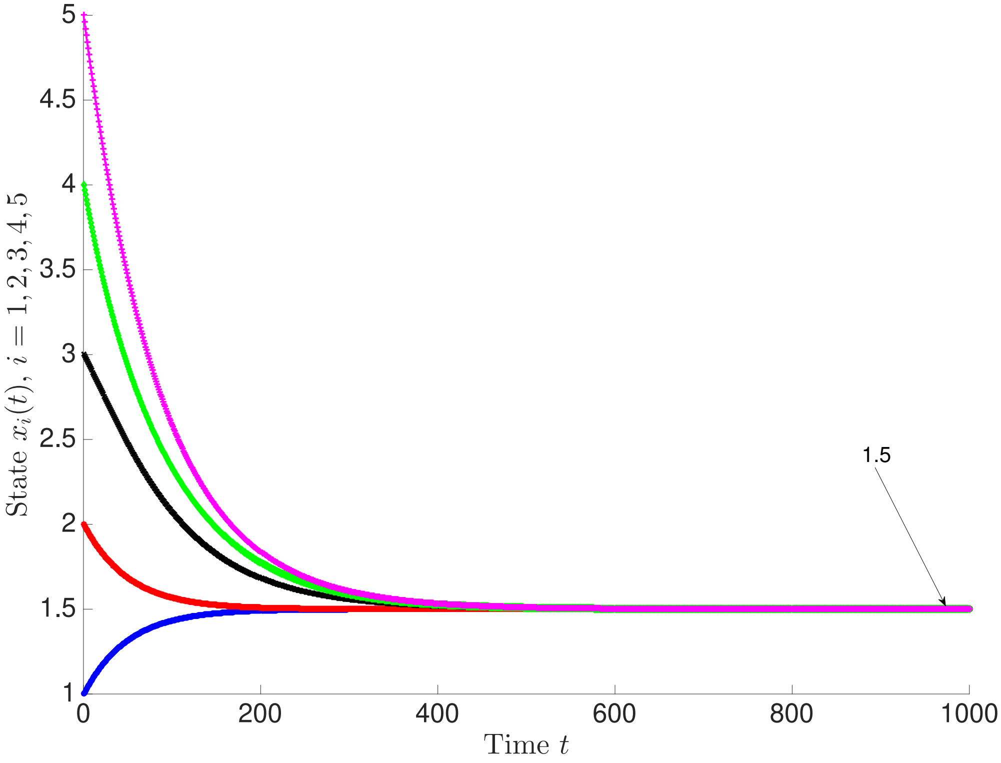


Figure 4.2: Success of achieving consensus

Given the effectiveness of 'pursuing neighbors' states', we describe the following distributed algorithm that updates the state $x_i(t)$ such that the agents achieve consensus.

Consensus Algorithm (CA):

Every agent i has a state variable $x_i(t)$ whose initial value is an arbitrary real number. At time $t \geq 0$, every agent i updates its state $x_i(t)$ as follows:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i). \tag{4.2}$$

Here the *updating weights* $a_{ij} > 0$ are the weights of the existing edges (i.e. the entries of the adjacency matrix). For this update, agent i needs to receive the state $x_j(t)$ or relative state $x_j(t) - x_i(t)$ from each neighbor $j \in \mathcal{N}_i$.

In words, (4.2) updates each state $x_i(t)$ towards the direction of pursuing a weighted average of the relative state differences with the neighbors. Regarding the updating weights a_{ij}, there are different choices. A simple valid choice is $a_{ij} = 1$ whenever $j \in \mathcal{N}_i$ (as in Example 4.2). Let $x := [x_1 \cdots x_n]^\top$ be the aggregated state of the networked agents. Then the n equations (4.2) become

$$\dot{x} = -Lx. \tag{4.3}$$

4.3 Convergence Result

The following is the main result of this section.

Theorem 4.1 *Suppose that Assumption 4.1 holds. Then CA solves the consensus problem.*

To prove Theorem 4.1, we will analyze the locations of eigenvalues of the matrix $-L$ in (4.3). For this, the following tool is convenient.

Theorem 4.2 (Gershgorin Discs Theorem) *Consider an arbitrary real square matrix $M = (m_{ij}) \in \mathbb{R}^{n\times n}$, and for every $i \in [1, n]$ let*

$$D_i := \left\{ z \in \mathbb{C} \;\middle|\; |z - m_{ii}| \leq \sum_{j\neq i} |m_{ij}| \right\} \tag{4.4}$$

be the disc centered at the diagonal entry m_{ii} with radius equal to the sum of absolute values of ith row's off-diagonal entries. Then the spectrum $\sigma(M)$, i.e. the set of n eigenvalues of M, satisfies

$$\sigma(M) \subseteq \bigcup_i D_i.$$

Theorem 4.2 provides an easy estimation of the locations of eigenvalues; namely every eigenvalue lies in the union of the Gershgorin discs in (4.4). This estimation is particularly useful for the spectrum of standard Laplacian matrices owing to the way they are defined (i.e. degree matrix minus adjacency matrix).

In addition to the Gershgorin Discs Theorem, we also need the following facts on solution and stability of linear ordinary differential equations. Let $A \in \mathbb{R}^{n\times n}$. Then the *matrix exponential* e^A is as follows:

$$\mathrm{e}^A := I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!}A^k.$$

Lemma 4.1 *Consider an ordinary differential equation* $\dot{x} = Ax$ *with an initial condition* $x(0)$ *and* $A \in \mathbb{R}^{n \times n}$.

- *The solution to* $\dot{x} = Ax$ *is* $x(t) = \mathrm{e}^{At}x(0)$
- *If all the eigenvalues of* A *have negative real parts, then* $\lim_{t\to\infty} \mathrm{e}^{At} = 0$.

Proof. First, it is a basic fact from the theory of differential equations that $\dot{x} = Ax$ with an initial condition $x(0)$ has a unique solution. Thus we only need to verify that $x(t) = \mathrm{e}^{At}x(0)$ satisfies $\dot{x} = Ax$ with $x(0)$. Substituting $x(t) = \mathrm{e}^{At}x(0)$ into $\dot{x} = Ax$ yields:

$$
\begin{aligned}
\dot{x} &= \frac{d}{dt}\mathrm{e}^{At}x(0) \\
&= \frac{d}{dt}(I + At + \frac{1}{2!}(At)^2 + \frac{1}{3!}(At)^3 + \cdots)x(0) \\
&= A + A^2t + \frac{1}{2!}A^3t^2 + \cdots)x(0) \\
&= A(I + At + \frac{1}{2!}(At)^2 + \cdots)x(0) \\
&= A\mathrm{e}^{At}x(0) \\
&= Ax.
\end{aligned}
$$

This verifies that $x(t) = \mathrm{e}^{At}x(0)$ is the unique solution of $\dot{x} = Ax$ with the initial condition $x(0)$.

Second, let J be the Jordan canonical form of the matrix A, i.e.

$$
\begin{aligned}
A &= VJV^{-1} \\
&= \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}
\begin{bmatrix} J_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_l \end{bmatrix}
\begin{bmatrix} z_1^\top \\ \vdots \\ z_n^\top \end{bmatrix}
\end{aligned}
$$

where y_i, z_i ($i \in [1, n]$) are respectively the (generalized) right and left eigenvectors of A, and J_i ($i \in [1, l]$) are the Jordan blocks of the l distinct eigenvalues $\lambda_1, \ldots, \lambda_l$ of A. These Jordan blocks J_i have the following special form:

$$
J_i = \begin{bmatrix}
\lambda_i & * & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & 0 & \ddots & * \\
0 & 0 & \cdots & \lambda_i
\end{bmatrix}
$$

where $* \in \{0, 1\}$. Owing to the above special form, J_i may be written as

$$J_i = \lambda_i I + N_i$$

where N_i is a nilpotent matrix whose eigenvalues are all zeros. As a result, there exists a positive integer k_i such that $N_i^{k_i} = 0$. Now let us consider $x(t)$:

$$\begin{aligned}
x(t) &= \mathrm{e}^{At}x(0) \\
&= \mathrm{e}^{VJV^{-1}t}x(0) \\
&= (I + VJV^{-1}t + \frac{1}{2!}(VJV^{-1}t)^2 + \frac{1}{3!}(VJV^{-1}t)^3 + \cdots)x(0) \\
&= (VV^{-1} + VJV^{-1}t + \frac{1}{2!}VJ^2V^{-1}t^2 + \frac{1}{3!}VJ^3V^{-1}t^3 + \cdots)x(0) \\
&= V(I + Jt + \frac{1}{2!}J^2t^2 + \frac{1}{3!}J^3t^3 + \cdots)V^{-1}x(0) \\
&= V\mathrm{e}^{Jt}V^{-1}x(0).
\end{aligned}$$

Hence the asymptotical behavior of $x(t)$ depends on that of e^{Jt}. According to the special forms of the Jordan canonical form J and the component Jordan blocks J_i, we derive

$$\begin{aligned}
\mathrm{e}^{Jt} &= \begin{bmatrix} \mathrm{e}^{J_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{e}^{J_l t} \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{e}^{(\lambda_1 I + N_1)t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{e}^{(\lambda_l I + N_l)t} \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{e}^{\lambda_1 t}\mathrm{e}^{N_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{e}^{\lambda_l t}\mathrm{e}^{N_l t} \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{e}^{\lambda_1 t}(I + N_1 t + \frac{1}{2!}N_1^2 t^2 + \cdots) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{e}^{\lambda_l t}(I + N_l t + \frac{1}{2!}N_l^2 t^2 + \cdots) \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{e}^{\lambda_1 t}(I + N_1 t + \frac{1}{2!}N_1^2 t^2 + \cdots + \frac{1}{k_1!}N_1^{k_1} t^{k_1}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathrm{e}^{\lambda_l t}(I + N_l t + \frac{1}{2!}N_l^2 t^2 + \cdots + \frac{1}{k_l!}N_k^{k_l} t^{k_l}) \end{bmatrix}.
\end{aligned}$$

Since all the eigenvalues $\lambda_1, \dots, \lambda_l$ have negative real parts, we have

$$(\forall i \in [1, l]) \mathrm{e}^{\lambda_i t} \to 0$$

exponentially fast as $t \to \infty$. Hence

$$(\forall i \in [1, l]) \mathrm{e}^{\lambda_i t}(I + N_i t + \frac{1}{2!} N_i^2 t^2 + \cdots + \frac{1}{k_i!} N_i^{k_i} t^{k_i}) \to 0$$

as $t \to \infty$. This means that

$$\lim_{t\to\infty} \mathrm{e}^{Jt} = 0.$$

Therefore

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} \mathrm{e}^{At} x(0) = \lim_{t\to\infty} V \mathrm{e}^{Jt} V^{-1} x(0) = 0.$$

□

Now we are ready to prove Theorem 4.1.

Proof of Theorem 4.1: Suppose that Assumption 4.1 holds. Since L in (4.3) is a standard Laplacian matrix, by definition L has an eigenvalue 0 with an associated eigenvector $\mathbf{1}$ (the vector of all ones). Moreover, it follows from Theorem 1.7 and Assumption 4.1 that the eigenvalue 0 is simple. For later use let w be a left eigenvector of L associated with the eigenvalue 0 (i.e. $w^\top L = 0$), which is normalized such that $w^\top \mathbf{1} = 1$.

Now we invoke the Gershgorin Discs Theorem (Theorem 4.2) to estimate the locations of the rest $n-1$ nonzero eigenvalues of L. Since $L = D - A$, $A \geq 0$, and $D = \mathrm{diag}(A\mathbf{1})$, by Theorem 4.2 all the eigenvalues of L lie on the right-hand side of the complex plane including the origin. We have shown that the eigenvalue 0 of L is simple; hence the rest $n-1$ nonzero eigenvalues have positive real parts. It follows that $-L$ has a simple eigenvalue 0 and all the other eigenvalues have negative real parts.

Write $-L$ in Jordan canonical form as

$$-L = VJV^{-1} = \begin{bmatrix} \mathbf{1} & y_2 & \cdots & y_n \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & J' \end{bmatrix} \begin{bmatrix} w^\top \\ z_2^\top \\ \vdots \\ z_n^\top \end{bmatrix}$$

where $y_i, z_i \in \mathbb{C}^n$ $(i \in [2, n])$ are respectively the (generalized) right and left eigenvectors of $-L$; and $J' \in \mathbb{C}^{(n-1)\times(n-1)}$ is a block diagonal matrix consisting of the Jordan blocks corresponding to those nonzero eigenvalues with negative real parts. It follows from Lemma 4.1 that the matrix exponential e^{-Lt} is

$$\begin{aligned} \mathrm{e}^{-Lt} &= \mathrm{e}^{VJV^{-1}t} = V\mathrm{e}^{Jt}V^{-1} \\ &= V\begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{J't} \end{bmatrix} V^{-1} \\ &\to \mathbf{1}w^\top, \quad \text{as } t \to \infty. \end{aligned}$$

Therefore based on the CA in (4.3):

$$\begin{aligned} x(t) &= \mathrm{e}^{-Lt}x(0) \\ &\to \mathbf{1}w^\top x(0), \quad \text{as } t \to \infty. \end{aligned}$$

That is,

$$(\forall i \in [1, n]) \lim_{t\to\infty} x_i(t) = w^\top x(0)$$

i.e. CA solves the consensus problem. □

Remark 4.1 *(Convergence Speed) Theorem 4.1 asserts that as long as the digraph contains a spanning tree, CA described as $\dot{x} = -Lx$ in (4.3) converges to the one-dimensional kernel spanned by the vector* **1** *(aka. consensus vector). The speed of convergence is governed by the non-zero eigenvalue with the largest real part (or the smallest absolute real part since all nonzero eigenvalues have negative real parts) of the standard Laplacian matrix L. Denote the largest real part by $\mathrm{Re}(\lambda_2(L))$, and refer to $\mathrm{Re}(\lambda_2(L))$ as the* convergence factor *of CA; that is, CA converges exponentially with the exponent $-\mathrm{Re}(\lambda_2(L))$. The value of $\mathrm{Re}(\lambda_2(L))$ depends on the topology of digraph $\mathcal{G}$, which we will illustrate in Section 4.4 using simulation examples.*

As stated in the proof of Theorem 4.1, the consensus value is $w^\top x(0)$, where w is the normalized left eigenvector of L associated with the eigenvalue 0 and $x(0)$ is the initial condition. Thus the consensus value is a weighted average of the agents' initial states. The weight distribution across the network is determined by the digraph topology, and reflects different roles of individual nodes. The following proposition provides a precise relation between the weight vector w and the graph topology.

Proposition 4.2 *Suppose that Assumption 4.1 holds, and let w be the normalized left eigenvector of L associated with the eigenvalue 0 satisfying $w^\top\mathbf{1} = 1$. Then the following statements hold.*

(i) *$w \geq 0$, and $w_i > 0$ if and only if node i is a root (i.e. only roots are weighted).*

(ii) *If digraph $\mathcal{G}$ is strongly connected, then $w > 0$.*

(iii) *If digraph $\mathcal{G}$ is strongly connected and weight-balanced, then $w = \frac{1}{n}\mathbf{1}$ (namely averaging is achieved).*

Proof. We prove these statements in the order (iii), (ii), and (i). First for (iii), since $\mathcal{G}$ is strongly connected and weight-balanced, every column of L also sums up to zero. Namely $\mathbf{1}^\top L = 0$, which means that $\mathbf{1}$ is (also) a left eigenvector of L associated with eigenvalue 0. Hence the normalized left eigenvector is $w = \frac{1}{n}\mathbf{1}$.

Next for (ii), we follow the proof of Lemma 1.6. Since $\mathcal{G}$ is strongly connected, by Theorem 1.3 the nonnegative adjacency matrix A of $\mathcal{G}$ is irreducible and the degree matrix D is invertible. As a result, the Laplacian matrix $L = D - A$ can be written as $L = D(I - D^{-1}A)$. Let $\tilde{A} := D^{-1}A$ and $\tilde{L} := D^{-1}L = I - \tilde{A}$. Then $\tilde{A}$ is row-stochastic and has zero entries at the same locations as A does; the latter means that $\tilde{A}$ is irreducible too. By the Perron-Frobenius Theorem (Theorem 1.5), the spectral radius 1 is a simple eigenvalue of $\tilde{A}$ and has a positive left eigenvector w, i.e. $w^\top \tilde{A} = w^\top$. Normalize w if necessary to satisfy $w^\top \mathbf{1} = 1$, which does not change its positivity. Since

$$\begin{aligned} w^\top L &= w^\top D(I - \tilde{A}) \\ &= Dw^\top - Dw^\top \tilde{A} \\ &= 0 \end{aligned}$$

we conclude that $w > 0$ is a left eigenvector of L associated with eigenvalue 0.

Finally for (i), we follow the proof of Theorem 1.7. Since $\mathcal{G}$ contains a spanning tree, by Theorem 1.1 the set of roots $\mathcal{V}_r$ induces a subdigraph $\mathcal{G}_r$ which is the unique closed strong component of $\mathcal{G}$. Consider without loss of generality the case that the nodes are ordered according to the partition $\mathcal{V}_r \cup (\mathcal{V} \setminus \mathcal{V}_r)$. Then the nonnegative adjacency matrix A and degree matrix D have the following forms:

$$A = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_3 \end{bmatrix}$$

Define an invertible $\tilde{D}$ such that $\tilde{D} := D$ if $\mathcal{V}_r$ contains more than one node, and

$$\tilde{D} := \begin{bmatrix} 1 & 0 \\ 0 & D_3 \end{bmatrix}$$

if $\mathcal{V}_r$ contains exactly one node. Thus $\tilde{D}$ is invertible. Use $\tilde{D}^{-1}$ to define

$$\tilde{A} := \tilde{D}^{-1}A = \begin{bmatrix} \tilde{A}_1 & 0 \\ \tilde{A}_2 & \tilde{A}_3 \end{bmatrix}, \quad \tilde{L} := \tilde{D}^{-1}L = I - \tilde{A}.$$

Then $\tilde{A}$ is row-stochastic. Consider an artificial discrete-time system $\tilde{x}(k+1) = \tilde{A}\tilde{x}(k)$, and partition the vector $\tilde{x}(k)$ according to the sizes of $\tilde{A}_1$ and $\tilde{A}_3$, respectively. Thus we derive

$$\tilde{x}_1(k+1) = \tilde{A}_1\tilde{x}_1(k) \tag{4.5}$$

$$\tilde{x}_2(k+1) = \tilde{A}_2\tilde{x}_1(k) + \tilde{A}_3\tilde{x}_2(k). \tag{4.6}$$

For (4.5), since $\tilde{A}_1$ corresponds to $\mathcal{G}_r$ which is strongly connected, similar to (ii) above $\tilde{A}_1$ has a simple eigenvalue 1 with a positive normalized left eigenvector $w_1 > 0$ and $\lim_{k\to\infty} \tilde{A}_1^k = \mathbf{1}w_1^\top$. For (4.6), since $\rho(\tilde{A}_3) < 1$ (in the proof of Theorem 1.7), taking the limit as $k \to \infty$ yields

$$\begin{aligned}\lim_{k\to\infty} \tilde{x}_2(k) &= (I - \tilde{A}_3)^{-1}\tilde{A}_2 \lim_{k\to\infty} \tilde{x}_1(k) \\ &= (I - \tilde{A}_3)^{-1}\tilde{A}_2\mathbf{1}w_1^\top\tilde{x}_1(0).\end{aligned}$$

Note that $(I - \tilde{A}_3)^{-1}\tilde{A}_2\mathbf{1} = \mathbf{1}$ because $\tilde{A}_2\mathbf{1} + \tilde{A}_3\mathbf{1} = \mathbf{1}$ implied by the row-stochasticity of $\tilde{A}$. Hence

$$\lim_{k\to\infty} \tilde{x}(k) = \lim_{k\to\infty} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} = \begin{bmatrix} \mathbf{1}w_1^\top\tilde{x}_1(0) \\ \mathbf{1}w_1^\top\tilde{x}_1(0) \end{bmatrix}.$$

On the other hand

$$\lim_{k\to\infty} \tilde{x}(k) = \lim_{k\to\infty} \tilde{A}^k\tilde{x}(0) = \lim_{k\to\infty} \begin{bmatrix} \tilde{A}_1^k & 0 \\ X & \tilde{A}_3^k \end{bmatrix} \begin{bmatrix} \tilde{x}_1(0) \\ \tilde{x}_2(0) \end{bmatrix} = \begin{bmatrix} \mathbf{1}w_1^\top & 0 \\ X & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1(0) \\ \tilde{x}_2(0) \end{bmatrix}.$$

From the above we have $X = \mathbf{1}w_1^\top$ and

$$\lim_{k\to\infty} \tilde{A}^k = \begin{bmatrix} \mathbf{1}w_1^\top & 0 \\ \mathbf{1}w_1^\top & 0 \end{bmatrix} = \mathbf{1} \begin{bmatrix} w_1^\top \\ 0 \end{bmatrix} =: \mathbf{1}w^\top$$

Note that $w \geq 0$ is a nonnegative normalized left eigenvector of $\tilde{A}$ associated with eigenvalue 1, and

$w_i > 0$ if and only if node i is a root. Since

$$\begin{aligned} w^\top L &= w^\top \tilde{D}(I - \tilde{A}) \\ &= \tilde{D} w^\top - \tilde{D} w^\top \tilde{A} \\ &= 0 \end{aligned}$$

we conclude that $w \geq 0$ is a left eigenvector of L associated with eigenvalue 0. □

4.4 Simulation Examples

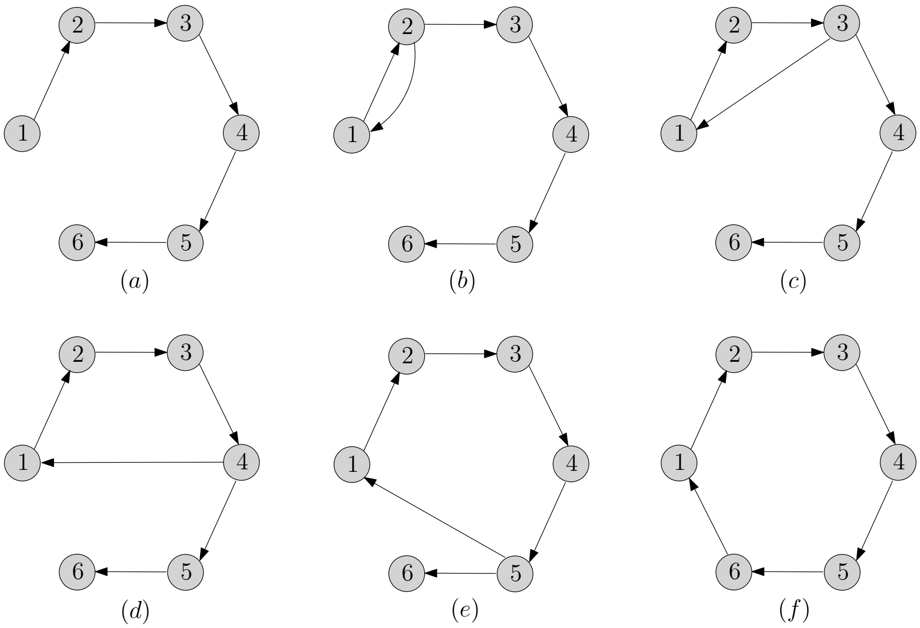


Figure 4.3: Six digraph topologies of 6 agents

Example 4.3 *We consider 6 agents interconnected through digraphs of 6 different topologies (Fig. 4.3). Every digraph contains a spanning tree; hence by Theorem 4.1, CA achieves consensus on all the* 6 *digraphs. For simplicity consider uniform, unit weight for all edges. Then the standard Laplacian matrices, (normalized) left eigenvectors of eigenvalue* 0*, and convergence factors are as follows.*

Digraph in Fig. 4.3(a): one root (agent 1)

$$L_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad w_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathrm{Re}(\lambda_2(L_1)) = 1.$$

Digraph in Fig. 4.3(b): two roots (agents 1, 2)

$$L_2 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad w_2 = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathrm{Re}(\lambda_2(L_2)) = 2.$$

Digraph in Fig. 4.3(c): three roots (agents 1, 2, 3)

$$L_3 = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad w_3 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathrm{Re}(\lambda_2(L_3)) = 1.5.$$

Digraph in Fig. 4.3(d): four roots (agents 1, 2, 3, 4)

$$L_4 = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad w_4 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ 0 \\ 0 \end{bmatrix}, \quad \mathrm{Re}(\lambda_2(L_4)) = 2.$$

Digraph in Fig. 4.3(e): five roots (agents $1, 2, 3, 4, 5$*)*

$$L_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad w_5 = \begin{bmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ 0 \end{bmatrix}, \quad \mathrm{Re}(\lambda_2(L_5)) = 1.8.$$

Digraph in Fig. 4.3(f): six roots (agents $1, 2, 3, 4, 5, 6$*)*

$$L_6 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad w_6 = \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \end{bmatrix}, \quad \mathrm{Re}(\lambda_2(L_6)) = 2.$$

From the above it is observed:

- *All the normalized eigenvectors* w_i $(i \in [1, 6])$ *are nonnegative; only roots are positively and uniformly weighted; in the particular case of Fig. 4.3(f), whose topology is strongly connected and weight-balanced, average consensus is achieved. Therefore the statements of Proposition 4.2 are demonstrated.*

- *Convergence factor is topology dependent; however, it is not the case that the more roots the larger the convergence factor; and even numbers of leaders tend to yield larger convergence factor than odd number of leaders.*

Finally as an illustration, CA is run on the 6 digraphs with the same initial condition $x(0) = [1\ 2\ 3\ 4\ 5\ 6]^\top$*; the result is displayed in Fig. 4.4. Observe that the consensus value changes as the number of roots increases: when only agent 1 is the root, the consensus value is agent 1's initial state* 1*; whereas when all the agents are roots, the consensus value is the average of all agents' initial states, namely 3.5. Also observe that the more roots, the more oscillatory trajectories exist; this is intuitively due to more 'negotiation' takes place when more roots participate in determining the final consensus value.*

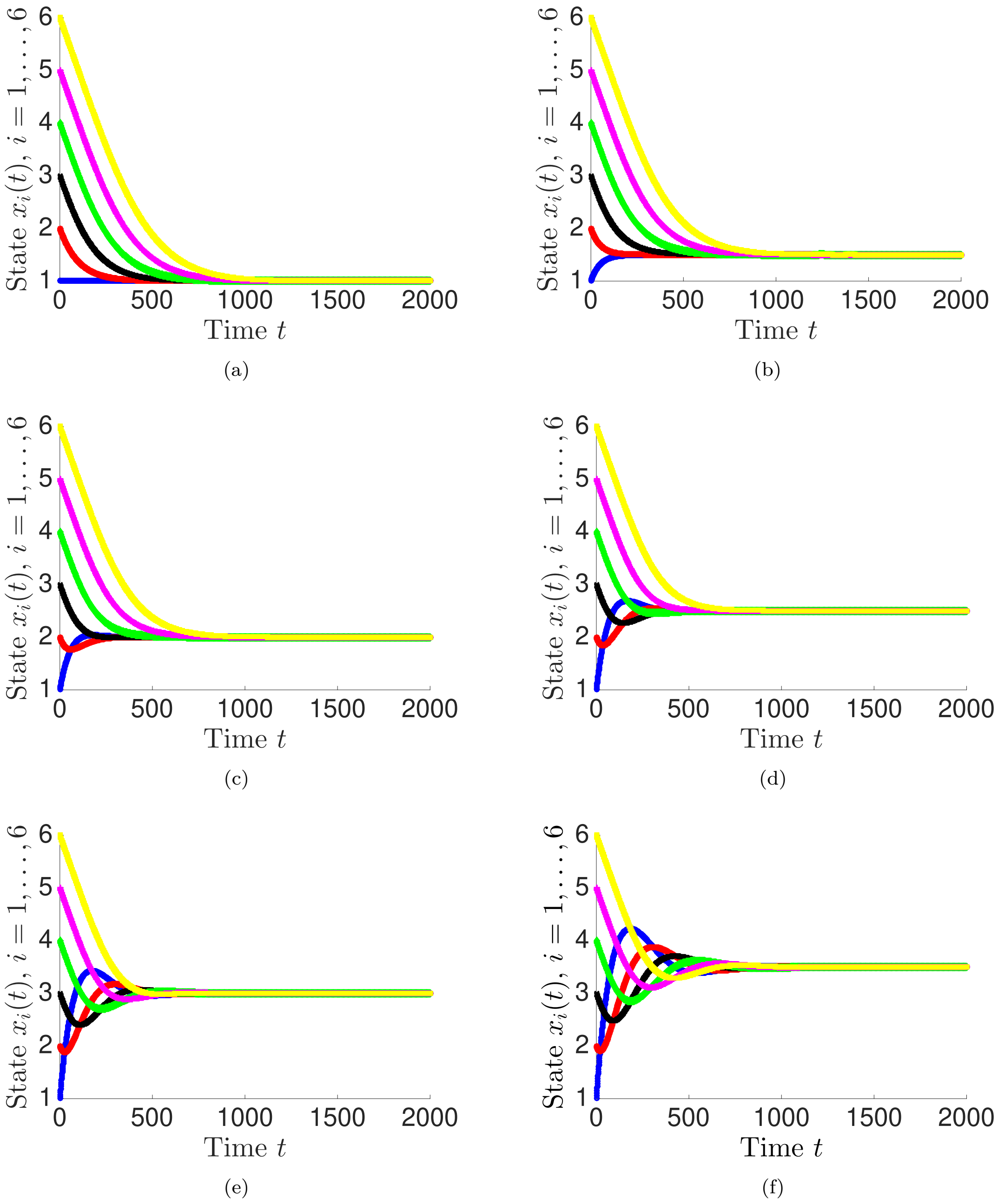


Figure 4.4: Convergence patterns (consensus values and convergence factors) of six agents. Colors of agents $1, \ldots, 6$ are sequentially blue, red, black, green, pink, and yellow

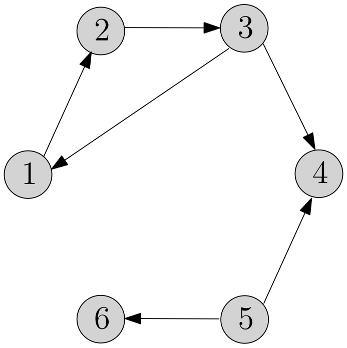


Figure 4.5: Six networked agents whose interconnection digraph does not contain a spanning tree

Example 4.4 *We consider again 6 agents interconnected through the digraph in Fig. 4.5. This digraph is Fig. 4.3(c) with one edge flipped direction:* $(4,5)$ *becomes* $(5,4)$*. As a result, this digraph no longer contains a spanning tree. Hence by Theorem 4.1, CA fails to achieve consensus. Indeed, consider uniform, unit weight for all edges and run CA with the initial condition* $x(0) = [1\ 2\ 3\ 4\ 5\ 6]^\top$*; the result is displayed in Fig. 4.6. Evidently consensus is not achieved. More specifically, while agents* $1, 2, 3$ *and agents* $5, 6$ *reach consensus respectively on different values, these two groups have no path for mutual communication. Consequently no global consensus can be reached in general. Observe also that agent 4 is equally influenced by the above-mentioned two groups, and therefore agent 4 converges to the average of the two distinct consensus values of the two groups.*

Example 4.5 *We demonstrate the influence of graph topologies on the convergence speed of CA. Specially, we investigate the influence in terms of different densities of edges. Consider a digraph of* $n = 100$ *nodes; we choose uniformly at random 10%, 50%, and 90% of directed edges from all possible* $n(n-1)$ *edges. We take only those digraphs that contain spanning trees, and set uniform weights* 1*.*
Fig. 4.7 displays the curves of the error $\sum_{i=1}^{n} \|x_i(k) - x^*\mathbf{1}\|_2$*, where* x^* *is the consensus value, with respect to the above chosen three different densities of edges. Here* x^* *is computed from the initial condition* $x(0)$*, each component of* $x(0)$ *being chosen uniformly at random from the closed interval* $[-10, 10]$*. In Fig. 4.7, each plotted point is the mean value of the error over* 100 *random digraphs of the respective densities. It is observed that the denser the digraph, the faster CA converges to the consensus value* x^**.*

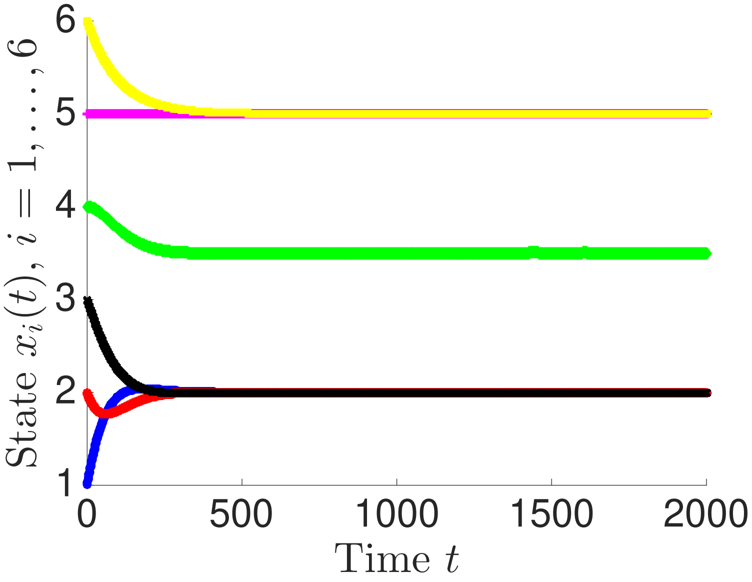


Figure 4.6: CA fails to achieve consensus for digraph in Fig. 4.5 that does not contain a spanning tree

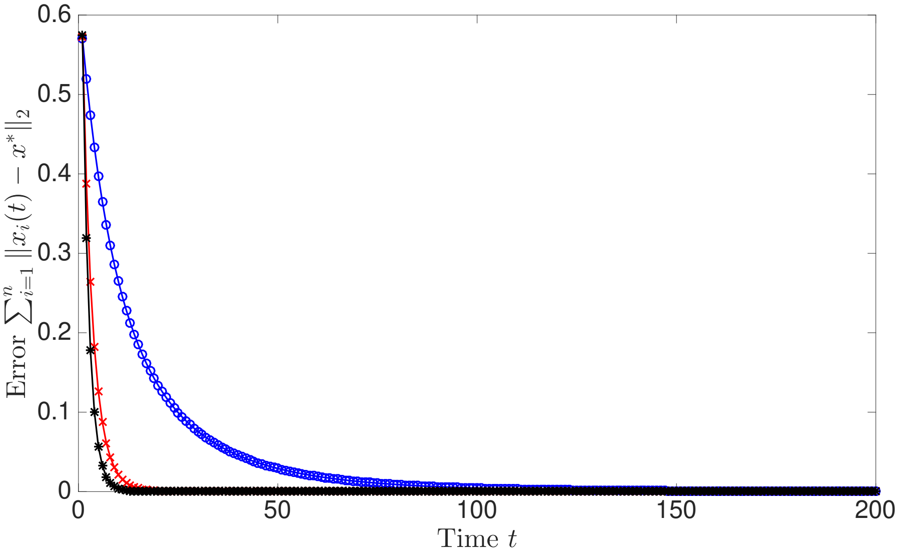


Figure 4.7: Convergence speed with respect to 10% (blue ∘), 50% (red ×), and 90% (black ∗) of directed edges

4.5 Notes and References

The consensus algorithm (CA) in the context of distributed control of multi-agent systems is first studied in

- A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Transactions on Automatic Control, vol.48, pp.988–1001, 2003

An important source of inspiration for this study is from computer animation of animal group behaviors and the related physics models:

- C. Reynolds, Flocks, birds, and schools: a distributed behavioral model, Computer Graphics, vol.21, pp.25–34, 1987
- T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen, O. Schochet, Novel type of phase transitions in a system of self-driven particles, Physical Review Letters, vol.75, pp.1226–1229, 1995.

Extension of CA to time-varying networks is reported in

- Z. Lin, M. Broucke, B. Francis, Local control strategies for groups of mobile autonomous agents, IEEE Transactions on Automatic Control, vol.49, pp.622–629, 2004
- L. Moreau, Stability of multiagent systems with time-dependent communication links, IEEE Transactions on Automatic Control, vol.50, pp.169–182, 2005.
- W. Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Transactions on automatic control, vol.50, pp.655–661, 2005

The Gershgorin Discs Theorem (Theorem 4.2) can be found in e.g.

- R.A. Horn and C.R. Johnson, Matrix Analysis, 2nd ed., Cambridge University Press, 2013

Chapter 5

Synchronization

The problem of consensus in the preceding chapter requires all the agents to converge to the same value, which is static in steady state. A generalized notion is the requirement that all the agents converge to the same but dynamic values. This is the problem of synchronization.

A familiar example is a network of harmonic oscillators to synchronize their phases and angular velocities. Another example is a group of autonomous vehicles to flock with the same velocities. A physiology example is a network of neurons to fire with the same frequencies. Indeed the synchronization problem typically involves higher-order dynamic models of the agents.

In this chapter we study the synchronization problem of (homogeneous) linear time-invariant dynamic agents. We show that a necessary graphical condition to achieve synchronization is that the digraph contains a spanning tree (the same as that to achieve consensus). Under this condition, we present a distributed algorithm that achieves synchronization.

5.1 Problem Statement

Consider a network of n (> 1) agents. Each agent i $(\in [1, n])$ is modeled by a general linear time-invariant (LTI) dynamic system:

$$\begin{aligned} \dot{x}_i &= Ax_i + Bu_i \\ y_i &= Cx_i + Du_i \end{aligned} \tag{5.1}$$

where $x_i \in \mathbb{R}^p$ is the state vector, $u_i \in \mathbb{R}^q$ the (control) input vector, and $y_i \in \mathbb{R}^r$ the (observation) output vector. A compact graphical notation of LTI is displayed in Fig. 5.1.

The matrices A, B, C, D in (5.1) are of the following sizes:

$$A \in \mathbb{R}^{p\times p}, \quad B \in \mathbb{R}^{p\times q}, \quad C \in \mathbb{R}^{r\times p}, \quad D \in \mathbb{R}^{r\times q}.$$

These matrices are the same for all agents; thus the multi-agent system is called *homogeneous*. Several assumptions are made concerning these matrices.

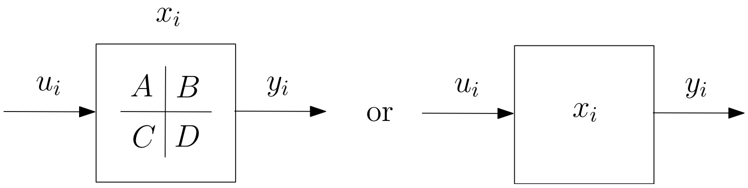


Figure 5.1: LTI system

Assumption 5.1 *The matrices A, B, C satisfy the following conditions.*

- (A, B) *is* stabilizable, *i.e. there exists a matrix F such that all the eigenvalues of $A + BF$ have negative real parts.*
- (C, A) *is* detectable, *i.e. there exists a matrix G such that all the eigenvalues of $A+GC$ have negative real parts.*
- *All the eigenvalues of matrix A have nonpositive real parts.*

The first two assumptions are standard for the feasibility of feedback control design (see Appendix). The third condition means that the uncontrolled agent dynamics does not contain exponentially unstable modes. The reason why this last condition is needed is because we need to ensure that the rate of convergence to synchronization (determined by graph Laplacian) is able to dominate the possibly divergence of uncontrolled system dynamics.

Synchronization Problem:

Consider a network of agents modeled by (5.1) interconnected through a digraph $\mathcal{G}$. Suppose that Assumption 5.1 holds. Design a distributed algorithm such that

$$(\forall x_1(0), \ldots, x_n(0) \in \mathbb{R}^p)(\forall i, j \in [1, n]) \lim_{t \to \infty} (x_i(t) - x_j(t)) = 0.$$

Example 5.1 *We provide an example to illustrate the synchronization problem. Consider a network of five harmonic oscillators:*

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= -x_{i1} + u_i \\ y_i &= x_{i1}, \quad i \in [1, 5]. \end{aligned}$$

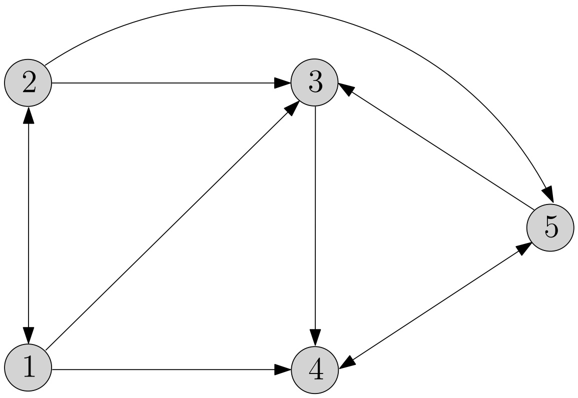


Figure 5.2: Illustrating example of synchronization problem with five agents

This corresponds to (5.1) with

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0.$$

Here x_{i1}, x_{i2} are respectively the phase angle and angular velocity of oscillator i. Since

$$\operatorname{rank}([B \ AB]) = 2$$
$$\operatorname{rank}(\begin{bmatrix} C \\ CA \end{bmatrix}) = 2$$

the pair (A, B) is controllable and thus stabilizable, and the pair (C, A) is observable and thus detectable.[a] *Moreover, the eigenvalues of A are $\pm j$ whose real parts are zero. Hence Assumption 5.1 holds.*

The interconnection of the five oscillators is modeled by the digraph in Fig. 5.2. The neighbor sets of the agents are $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_2 = \{1\}$, $\mathcal{N}_3 = \{1, 2, 5\}$, $\mathcal{N}_4 = \{1, 3, 5\}$, and $\mathcal{N}_5 = \{2, 4\}$. Given arbitrary initial conditions $x_1(0), \ldots, x_5(0) \in \mathbb{R}^2$, the synchronization problem is to design a distributed algorithm such that each oscillator's phase angle (resp. angular velocity) asymptotically converges to the same dynamic phases (resp. dynamic velocities).

[a] A review of these basic concepts of linear systems is provided in Appendix.

A necessary graphical condition for solving the synchronization problem is given below.

Proposition 5.1 *Suppose that there exists a distributed algorithm that solves the synchronization problem. Then the digraph contains a spanning tree.*

Proof. The proof is by contradiction. Suppose that the digraph $\mathcal{G}$ does *not* contain a spanning tree. Then it follows from Theorem 1.1 that $\mathcal{G}$ has at least two (distinct) closed strong components (say) $\mathcal{G}_1, \mathcal{G}_2$. In this case, consider an initial condition such that the agents in $\mathcal{G}_1$ have initial state $c_1 \in \mathbb{R}^p$, those in $\mathcal{G}_2$ have $c_2 \in \mathbb{R}^p$, and $c_1 \neq c_2$. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ are closed, information cannot be communicated from one to the other. Consequently, there exists no distributed algorithm that can solve the synchronization problem. □

Owing to Proposition 5.1, we shall henceforth assume that the digraph contains a spanning tree.

Assumption 5.2 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning tree.*

5.2 Distributed Algorithm

Example 5.2 *Consider again Example 5.1. To achieve synchronization, a natural idea is to use the consensus algorithm in Chapter 4 on the output y_i ($i \in [1, 5]$):*

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij}(y_j(k) - y_i(k)).$$

For simplicity consider unit weight for all edges (i.e. $a_{ij} = 1$). Then substitute u_i into (5.1) and write in vector form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} A-BC & BC & 0 & 0 & 0 \\ BC & A-BC & 0 & 0 & 0 \\ BC & BC & A-3BC & 0 & BC \\ BC & 0 & BC & A-3BC & BC \\ 0 & BC & 0 & BC & A-2BC \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}.$$

More compactly

$$\dot{x} = (I \otimes A - L \otimes BC)x$$

where $x = [x_1^\top \ \cdots \ x_5^\top]^\top$ is the aggregated state, L is the graph Laplacian matrix, and $\otimes$ denotes Kronecker product. With a random initial condition $x(0) \in \mathbb{R}^{10}$, a simulation result

of the above system is displayed in Fig. 5.3.
Evidently, synchronization did not occur. Thus the simple idea of achieving consensus fails to work for synchronization.

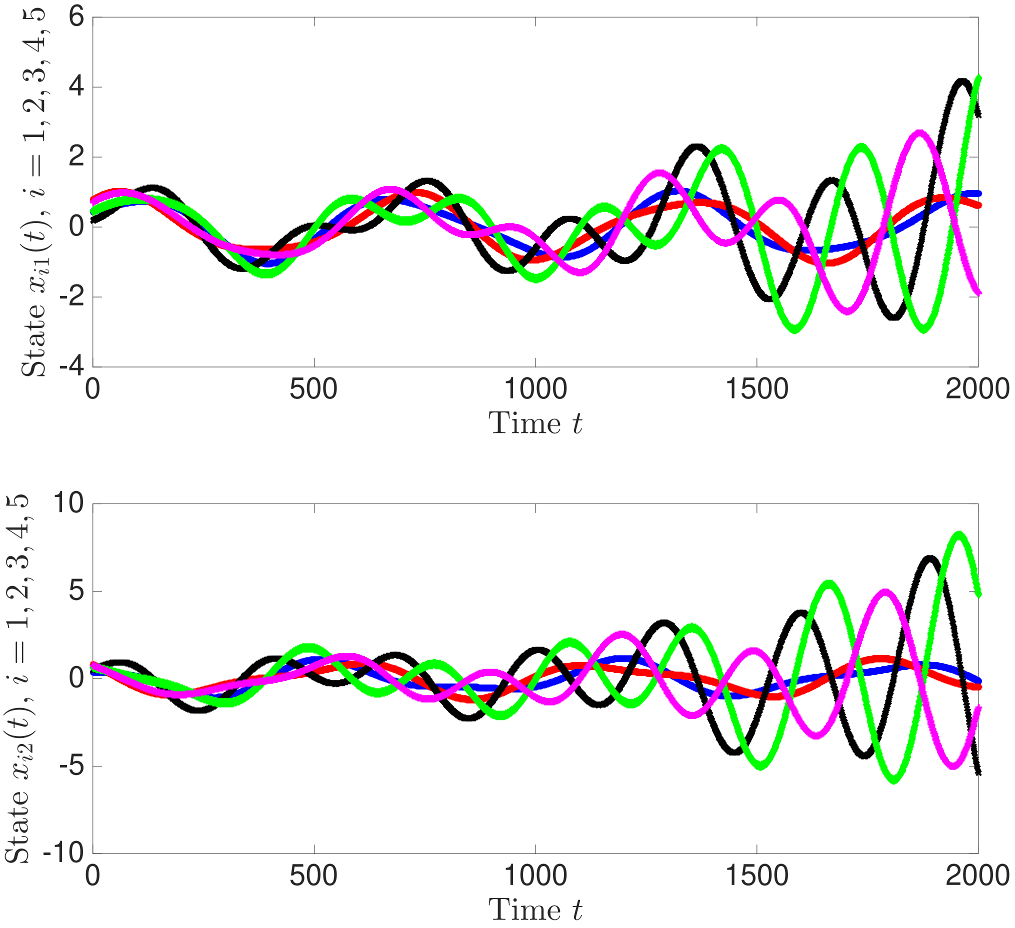


Figure 5.3: Failure to achieve synchronization using consensus algorithm

In the following, we describe a distributed algorithm that employs an *observer* that estimates the state x_i based on the output y_i, as well as a *generator* that applies the consensus algorithm based on stable dynamics.

Synchronization Algorithm (SA):

Every agent i has a dynamic model in (5.1) with an arbitrary initial state $x_i(0) \in \mathbb{R}^p$. Let F, G be matrices such that all the eigenvalues of $A+BF$ and $A+GC$ have negative real parts. At each time $t \geq 0$, every agent i performs the following updates:

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i + G(C\hat{x}_i + Du_i - y_i) \tag{5.2}$$

$$\dot{\xi}_i = (A+BF)\xi + \sum_{j\in\mathcal{N}_i} a_{ij}(\xi_j - \xi_i) - \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_j - \hat{x}_i) \tag{5.3}$$

$$u_i = F\xi_i. \tag{5.4}$$

Here the *updating weights* $a_{ij} > 0$ are the weights of the existing edges (i.e. the entries of the adjacency matrix); the initial conditions $\hat{x}_i(0) \in \mathbb{R}^p$ and $\xi_i(0) \in \mathbb{R}^p$ are arbitrary.

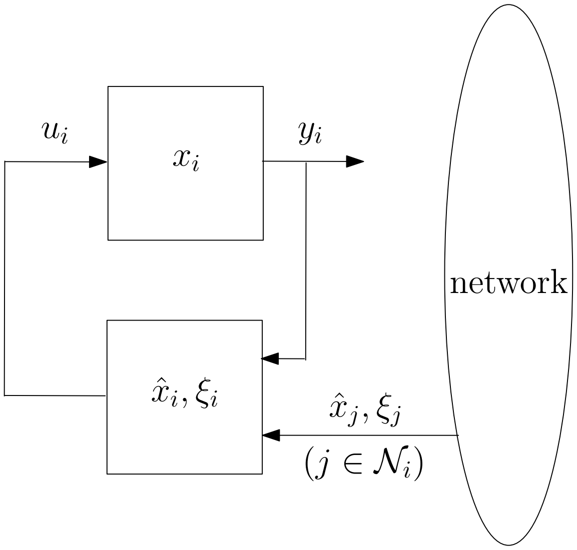


Figure 5.4: Distributed distributed controller

Remark 5.1 *In words, (5.2) is a local observer that estimates the state x_i based on output y_i and input u_i. The observer has stable dynamics (since $A+GC$ is stable), so that the estimate $\hat{x}_i$ (exponentially) converges to the true state x_i. Next, (5.3) is a local generator also with stable dynamics (since $A+BF$ is stable). This generator executes two consensus algorithms on the generators' states and on the observers' states, for which agent i needs to receive information $\xi_j(t), \hat{x}_j(t)$ or relative information $\xi_j(t) - \xi_i(t), \hat{x}_j(t) - \hat{x}_i(t)$ from each (in-)neighbor $j \in \mathcal{N}_i$. The purpose of this generator is to achieve consensus on the generator states on one hand, and on the other hand drive the difference in generator states $\xi_j(t) - \xi_i(t)$ to the difference in estimated states*

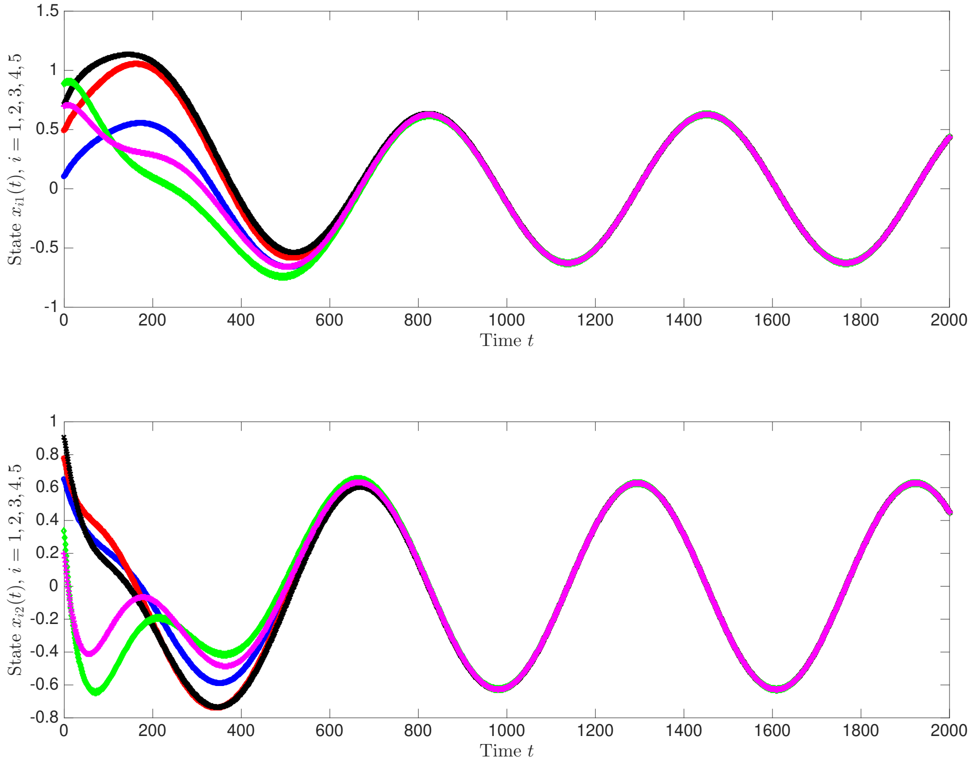


Figure 5.5: Synchronization of true states

$\hat{x}_j(t) - \hat{x}_i(t)$. *Since the estimated states converge to the true states, the difference in any pair of true states will diminish, and desired synchronization occurs. Finally, (5.4) computes the control input* u_i. *Overall, this is a dynamic distributed controller for agent* i, *whose inputs are* y_i *(from itself) and* $\hat{x}_j, \xi_j$ *(from its neighbors) while the output is* u_i. *A graphical illustration of this dynamic distributed controller is provided in Fig. 5.4.*

Remark 5.2 *If* $C = I$, *i.e.* $y_i = x_i$, *then the observer in (5.2) is not needed. Namely in this special case, SA becomes*

$$\dot{\xi}_i = (A + BF)\xi + \sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) - \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i)$$

$$u_i = F\xi_i$$

Let $x := [x_1^\top \cdots x_n^\top]^\top$, $\hat{x} := [\hat{x}_1^\top \cdots \hat{x}_n^\top]^\top$, and $\xi := [\xi_1^\top \cdots \xi_n^\top]^\top$ be the aggregated true state, estimated state, and generator state of the networked agents. Then the equations (5.1), (5.2), and (5.3) become

$$\begin{aligned}\dot{x} &= (I_n \otimes A)x + (I_n \otimes BF)\xi \\ \dot{\hat{x}} &= (I_n \otimes (A + GC))\hat{x} + (I_n \otimes BF)\xi - (I_n \otimes GC)x \\ \dot{\xi} &= (I_n \otimes (A + BF) - L \otimes I_p)\xi + (L \otimes I_p)\hat{x}\end{aligned} \tag{5.5}$$

Note that the Laplacian L appears only in the last equation of the generator dynamics.

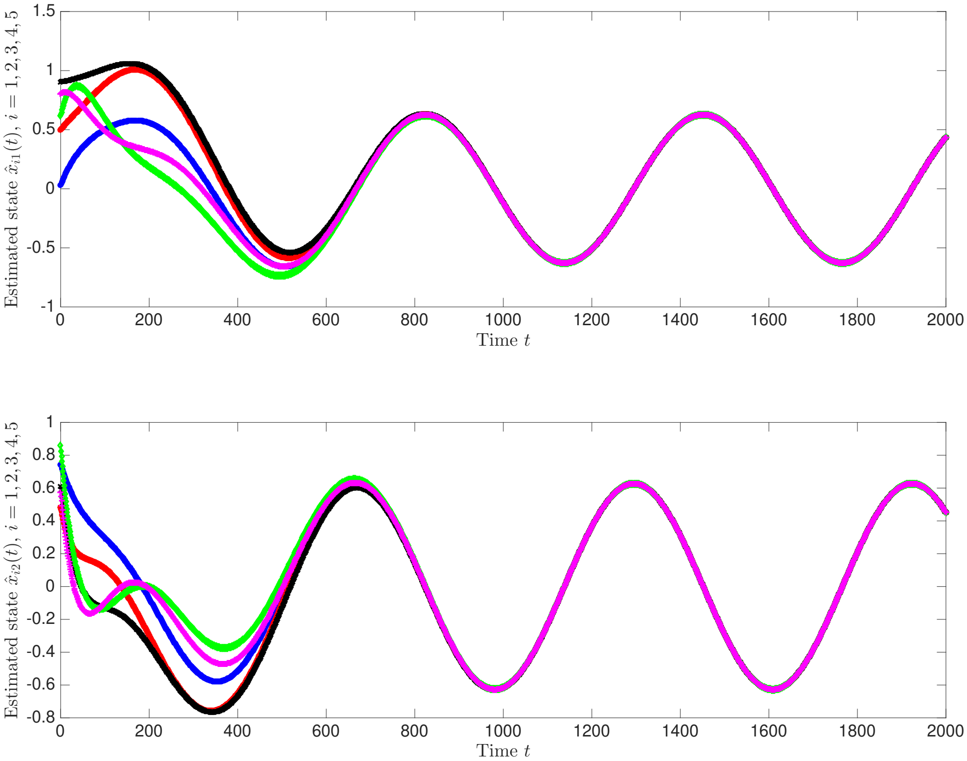


Figure 5.6: Synchronization of estimated states

Example 5.3 *Let us revisit Example 5.2. First, we assign desired eigenvalues for $A + BF$ and $A + GC$. Say for both matrices, let the desired eigenvalues be $-1, -2$. Then by pole*

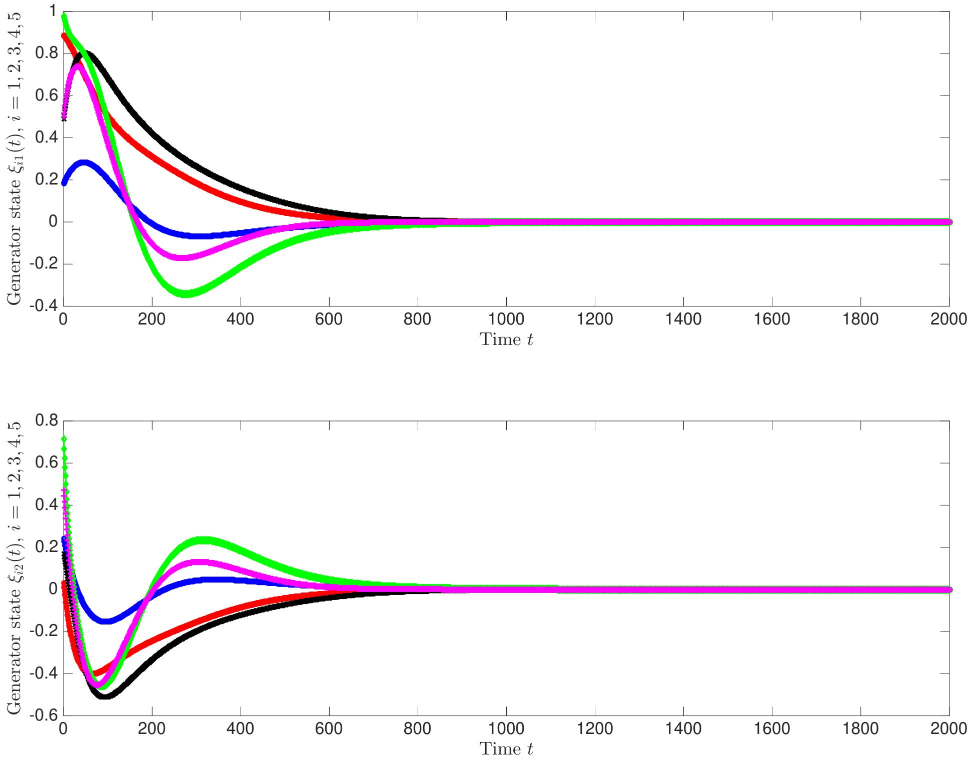


Figure 5.7: Convergence of generator states

assignment (see appendix), we obtain

$$F = \begin{bmatrix} -1 & -3 \end{bmatrix}, \quad G = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.$$

Substituting A, B, C, F, G, L into (5.5) and performing simulation with a set of randomized initial conditions $x(0), \hat{x}(0), \xi(0)$, we obtain the synchronized states of the oscillators as displayed in Fig. 5.5. Observe that both phase angles and angular velocities of the five oscillators converge to the same dynamic values. The estimated states also synchronize (Fig. 5.6), as they converge to the true states that are synchronized. Finally, the generator states converge to 0 (Fig. 5.7), for these generators are so designed that the difference in pairwise generator states converge to the difference in pairwise estimated states (the latter *converges to* 0*).*

5.3 Convergence Result

The following is the main result of this section.

Theorem 5.1 *Suppose that Assumptions 5.1 and 5.2 hold. The SA solves the synchronization problem.*

To proceed, let us first consider the third equation in (5.5):

$$\begin{aligned}\dot{\xi} &= (I_n \otimes (A+BF) - L \otimes I_p)\xi + (L \otimes I_p)\hat{x} \\ &= (I_n \otimes (A+BF))\xi + (L \otimes I_p)(\hat{x} - \xi).\end{aligned}$$

Since the eigenvalues of $A + BF$ have negative real parts, the convergence of $\xi(t)$ depends on that of $(\hat{x}(t) - \xi(t))$. Let

$$\epsilon := \hat{x} - \xi.$$

Then $\dot{\epsilon} = \dot{\hat{x}} - \dot{\xi}$. Substituting $\dot{\hat{x}}, \dot{\xi}$ by the second and third equations in (5.5) and arranging the terms yield

$$\dot{\epsilon} = (I_n \otimes A - L \otimes I_p)\epsilon - (I_n \otimes GC)(x - \hat{x}).$$

Ignoring for now the second term (i.e. the state estimation error which exponentially diminishes):

$$\dot{\epsilon} = (I_n \otimes A - L \otimes I_p)\epsilon; \tag{5.6}$$

thus corresponding to each ϵ_i $(i \in [1, n])$ is a consensus-like algorithm:

$$\dot{\epsilon}_i = A\epsilon_i + \sum_{j \in \mathcal{N}_i} a_{ij}(\epsilon_j - \epsilon_i) \tag{5.7}$$

The following lemma states that for every $i \in [1, n]$, $\epsilon_i(t)$ converges to $\epsilon_0(t)$ which is a solution of $\dot{\epsilon}_0 = A\epsilon_0$. This means that $\epsilon_1(t), \ldots, \epsilon_n(t)$ synchronize as $t \to \infty$.

Lemma 5.1 *Consider (5.7) and suppose that Assumptions 5.1 and 5.2 hold. Then*

$$(\forall i \in [1,n])(\forall \epsilon_i(0) \in \mathbb{R}^p)(\exists c \in \mathbb{R}^p) \lim_{t\to\infty} \|\epsilon_i(t) - c\mathrm{e}^{At}\| = 0. \tag{5.8}$$

To prove Lemma 5.1, we need the following property of matrix exponential :

$$(\forall A \in \mathbb{R}^{n\times n}) A\mathrm{e}^A = \mathrm{e}^A A. \tag{5.9}$$

That is, a matrix and its exponential commute. To see this, employ the definition of matrix exponential to derive

$$\begin{aligned} A\mathrm{e}^A &= A(I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots) \\ &= (I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots)A \\ &= \mathrm{e}^A A. \end{aligned}$$

Proof of Lemma 5.1. Let $i \in [1,n]$ and $\delta_i := \mathrm{e}^{-At}\epsilon_i$. Then

$$\begin{aligned} \dot{\delta}_i &= -A\mathrm{e}^{-At}\epsilon_i + \mathrm{e}^{-At}\dot{\epsilon}_i \\ &\overset{(5.6)}{=} -A\mathrm{e}^{-At}\epsilon_i + \mathrm{e}^{-At}(A\epsilon_i + \sum_{j\in\mathcal{N}_i} a_{ij}(\epsilon_j - \epsilon_i)) \\ &\overset{(5.9)}{=} \mathrm{e}^{-At} \sum_{j\in\mathcal{N}_i} a_{ij}(\delta_j - \delta_i) \\ &= \sum_{j\in\mathcal{N}_i} a_{ij}(\delta_j - \delta_i). \end{aligned}$$

Let $\delta := [\delta_1^\top \ \cdots \ \delta_n^\top]^\top$. Hence in compact form we have

$$\dot{\delta} = -(L \otimes I_p)\delta.$$

This is the standard consensus algorithm (CA) in p dimensions. Since Assumption 5.2 holds, it follows from Theorem 4.1 that

$$(\forall i \in [1,n])(\forall \delta_i(0) \in \mathbb{R}^p)(\exists c \in \mathbb{R}^p) \lim_{t\to\infty} \delta_i(t) = c.$$

In fact the above convergence is of exponential rate. Namely there exist constants $c_1, c_2 \in \mathbb{R}$ such that

$$\|\delta_i(t) - c\| \leq c_1 \mathrm{e}^{-c_2 t} \|\delta_i(0) - c\|.$$

The constant $c_2 = \mathrm{Re}(\lambda_2(L))$, the convergence factor of consensus algorithm. It then follows that

$$\begin{aligned} \|\epsilon_i(t) - c\mathrm{e}^{At}\| &= \|\mathrm{e}^{At}\delta_i(t) - c\mathrm{e}^{At}\| \\ &\leq \|\mathrm{e}^{At}\|\|\delta_i(t) - c\| \\ &\leq \|\mathrm{e}^{At}\| c_1 \mathrm{e}^{-c_2 t}\|\delta_i(0) - c\| \\ &= c_1 \mathrm{e}^{-c_2 t}\|\mathrm{e}^{At}\|\|\epsilon_i(0) - c\|. \end{aligned} \tag{5.10}$$

Since Assumption 5.1 holds (in particular the eigenvalues of A have nonpositive real parts), there exist a constant $c_3 \in \mathbb{R}$ such that

$$\|\epsilon_i(t) - c\mathrm{e}^{At}\| \leq c_1 \mathrm{e}^{-c_3 t}\|\epsilon_i(0) - c\|.$$

This implies that $\lim_{t\to\infty} \|\epsilon_i(t) - c\mathrm{e}^{At}\| = 0$, i.e. (5.8). □

Remark 5.3 *In the proof above, Assumption 5.1 on nonpositive real parts of A's eigenvalues is used to ensure exponential convergence of (5.10). It is worth pointing out that even when A has eigenvalues with positive real parts (so $\|\mathrm{e}^{At}\|$ exponentially diverges), if $c_2 = \mathrm{Re}(\lambda_2(L))$ (the convergence factor) can dominate the divergence rate of $\|\mathrm{e}^{At}\|$, then the exponential convergence of (5.10) can still be achieved. An illustration of this point is provided in Section 5.4 below using simulation.*

Remark 5.4 *An essential implication of Lemma 5.1 is that the spectrum (i.e. set of eigenvalues) of $(I_n \otimes A - L \otimes I_p)$ in (5.6) consists of those of A and the stable ones with negative real parts. To see this, consider the Jordan canonical form of the graph Laplacian L:*

$$V^{-1}LV = \begin{bmatrix} 0 & 0 \\ 0 & J \end{bmatrix}.$$

Here V be a nonsingular matrix whose columns are (generalized) eigenvectors of L, and $J \in \mathbb{C}^{(n-1)\times(n-1)}$ consists of Jordan blocks corresponding to the $n-1$ nonzero eigenvalues of L with *positive real parts (under Assumption 5.2). Then*

$$\begin{aligned} V^{-1}(I_n \otimes A - L \otimes I_p)V &= (V^{-1}I_n) \otimes (AV) - (V^{-1}L) \otimes (I_p V) \otimes I_p \\ &= I_n \otimes A - (V^{-1}LV) \otimes I_p \\ &= \begin{bmatrix} A & 0 \\ 0 & J' \end{bmatrix} \end{aligned}$$

Hence the spectrum of $(I_n \otimes A - L \otimes I_p)$ in (5.6) is the union of the spectrum of A and the spectrum of J'. Since Lemma 5.1 implies that $\epsilon(t) \to \mathrm{e}^{At}\mathbf{1}$, the eigenvalues of J' must all be stable.

With Lemma 5.1 we are ready to prove Theorem 5.1.

Proof of Theorem 5.1: Suppose that Assumptions 5.1 and 5.2 hold. Define the state estimation error $e := x - \hat{x}$. Then from the first and second equations in (5.5) we obtain

$$\begin{aligned} \dot{e} &= \dot{\hat{x}} - \dot{x} \\ &= (I_n \otimes (A + GC))e. \end{aligned} \tag{5.11}$$

Since G is such that the eigenvalues of $A + GC$ have negative real parts, $e(t) \to 0$ as $t \to \infty$.

Next define $\epsilon := \hat{x} - \xi$ and derive from the second and third equations in (5.5) as well as (5.11) the following:

$$\begin{aligned} \dot{\epsilon} &= \dot{\hat{x}} - \dot{\xi} \\ &= (I_n \otimes A - L \otimes I_p)\epsilon - (I_n \otimes GC)e. \end{aligned} \tag{5.12}$$

Since Assumptions 5.1 and 5.2 hold, by Lemma 5.1 we know that if e was constantly zero, then for every $i \in [1, n]$, $\epsilon_i(t)$ converges to $\epsilon_0(t)$ which is a solution of $\dot{\epsilon}_0 = A\epsilon_0$. Now from (5.11) and (5.12) we have

$$\begin{aligned} \begin{bmatrix} \dot{e} \\ \dot{\epsilon} \end{bmatrix} &= \begin{bmatrix} I_n \otimes (A + GC) & 0 \\ -I_n \otimes GC & I_n \otimes A - L \otimes I_p \end{bmatrix} \begin{bmatrix} e \\ \epsilon \end{bmatrix} \\ &=: M \begin{bmatrix} e \\ \epsilon \end{bmatrix}. \end{aligned}$$

The spectrum of the above matrix M is the union of the spectrum of $A + GC$ and the spectrum of $I_n \otimes A - L \otimes I_p$. For $A + GC$, all of its eigenvalues are stable. For $I_n \otimes A - L \otimes I_p$, it follows from Lemma 5.1 and Remark 5.4 that its spectrum includes the eigenvalues of A and stable ones. Hence overall, the spectrum of M consists of the eigenvalues of A and stable ones. Since $\lim_{t\to\infty} e(t) = 0$ and A's eigenvalues all having nonpositive real parts (Assumption 5.1), there exists $c \in \mathbb{R}^p$ such that $\epsilon_i(t) \to ce^{At}$ as $t \to \infty$ for all $i \in [1, n]$. That is, $\epsilon_1(t), \ldots, \epsilon_n(t)$ synchronize as $t \to \infty$.

With the above convergence result of $\epsilon(t)$, we analyze the generator state $\xi(t)$ based on the third equation in (5.5):

$$\begin{aligned}\dot{\xi} &= (I_n \otimes (A + BF) - L \otimes I_p)\xi + (L \otimes I_p)\hat{x} \\ &= (I_n \otimes (A + BF))\xi + (L \otimes I_p)\epsilon.\end{aligned}$$

Since $\epsilon_1(t), \ldots, \epsilon_n(t)$ synchronize as $t \to \infty$, we have

$$(L \otimes I_p)\epsilon(t) \to 0 \text{ as } t \to \infty.$$

In addition, since F is such that the eigenvalues of $A + BF$ have negative real parts, we derive that

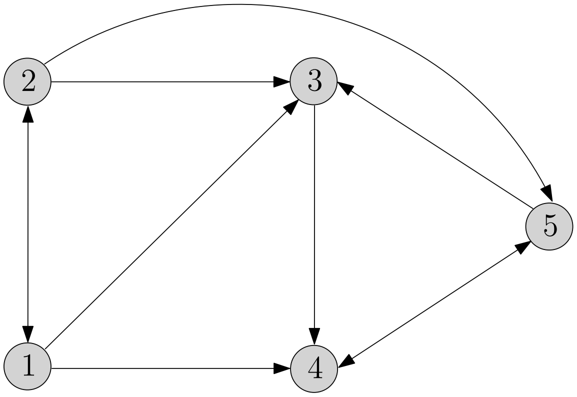
$$\xi(t) \to 0 \text{ as } t \to \infty.$$

Finally, since

$$\begin{aligned}x &= \hat{x} + e \\ &= \epsilon + \xi + e\end{aligned}$$

and $\epsilon_i(t) \to ce^{At}, \xi(t) \to 0, e(t) \to 0$ as $t \to \infty$, we conclude that $x_1(t), \ldots, x_n(t)$ synchronize as $t \to \infty$. Namely for every $i \in [1, n]$ and every $x_i(0) \in \mathbb{R}^p$, $x_i(t)$ converges to $x_0(t)$ which is a solution of $\dot{x}_0 = Ax_0$. □

5.4 Simulation Examples



Example 5.4 *Consider again the network in Fig. 5.2 (re-displayed here for convenience) and five double integrators:*

$$\begin{aligned}\dot{x}_{i1} &= x_{i2}\\ \dot{x}_{i2} &= u_i\\ y_i &= x_{i1} + x_{i2}, \quad i \in [1,5].\end{aligned}$$

This corresponds to (5.1) with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D = 0.$$

Here x_{i1}, x_{i2} are respectively the position and velocity of agent i. Since

$$\begin{aligned}\operatorname{rank}([B \;\; AB]) &= 2\\ \operatorname{rank}([C^\top \;\; A^\top C^\top]) &= 2\end{aligned}$$

the pair (A, B) is controllable and thus stabilizable, and the pair (C, A) is observable and thus detectable. Moreover, the two eigenvalues of A are both 0. Hence Assumption 5.1 holds. First, we assign desired eigenvalues for $A + BF$ and $A + GC$. Say for both matrices, let the desired eigenvalues be $-1, -2$. Then by pole assignment, we obtain

$$F = \begin{bmatrix} -2 & -3 \end{bmatrix}, \quad G = \begin{bmatrix} -1 \\ -2 \end{bmatrix}.$$

Substituting A, B, C, F, G, L into (5.5) and performing simulation with a set of randomized initial conditions $x(0), \hat{x}(0), \xi(0)$, we obtain the synchronized states of the agents as displayed in Fig. 5.8. Observe that all the agents converge to the same dynamic positions, as well as move with by the same velocity. The estimated states also synchronize (Fig. 5.9), and the generator states converge to 0 (Fig. 5.10).

Example 5.5 *While Assumption 5.2 allows A to have eigenvalues on the imaginary axis (possibly repeated such eigenvalues which can cause polynomially unstable dynamics), it rules out exponentially unstable dynamics for individual agents (when A has eigenvalues with positive real parts). However, synchronization may still be possible for exponentially unstable dynamics if the network interconnectivity is 'strong' enough to counterbalance the*

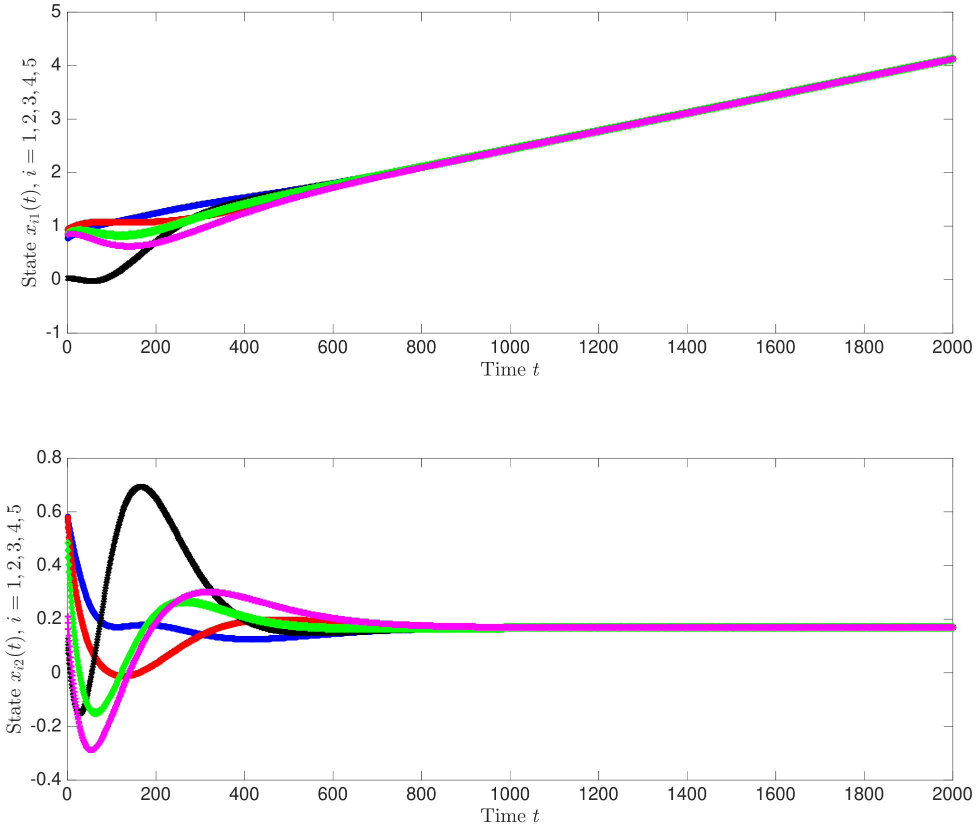


Figure 5.8: Synchronization of true states

unstable modes (refer to Remark 5.3).
For an illustration, consider a network of six inverted pendula:

$$\begin{aligned} \dot{x}_i &= Ax_i + Bu_i \\ y_i &= Cx_i + Du_i \quad x_i \in \mathbb{R}^4, u_i \in \mathbb{R}, y_i \in \mathbb{R}, i \in [1, 6] \end{aligned}$$

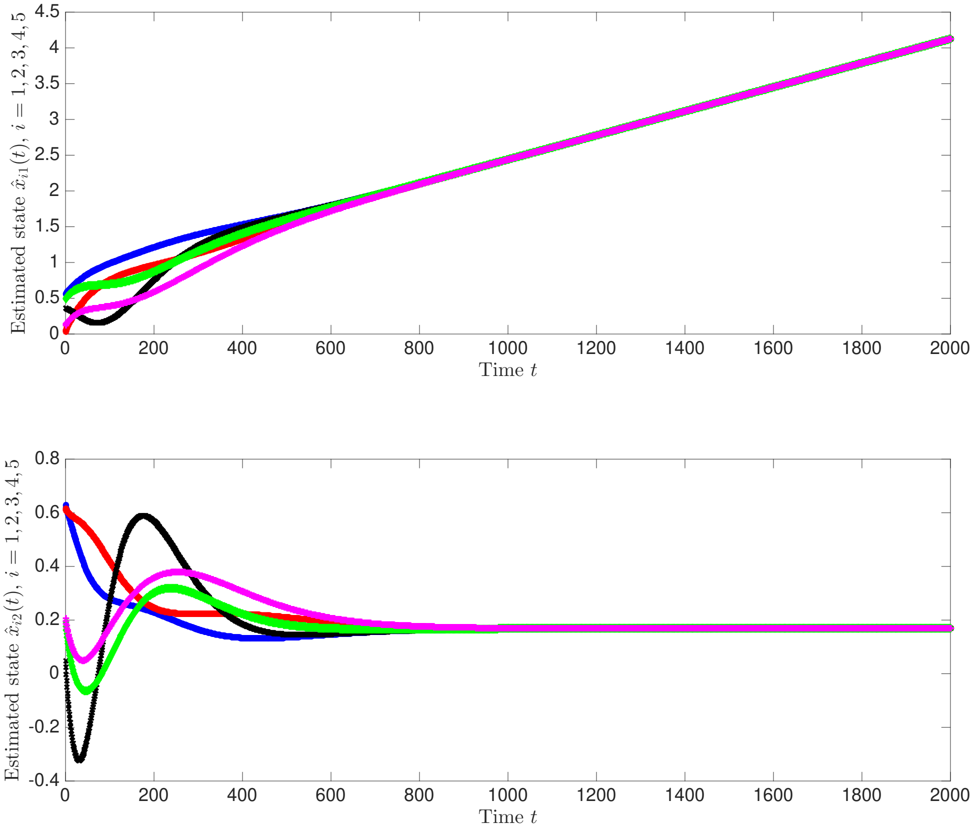


Figure 5.9: Synchronization of estimated states

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -0.098 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0.196 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}, \quad D = 0.$$

Note that the four eigenvalues of A are $0, 0, 0.4427, -0.4427$. The existence of the positive eigenvalue 0.4427 is not permitted by Assumption 5.2, which causes exponential divergence. On the other hand, it is verified that the pair (A, B) is controllable thus stabilizable, and the

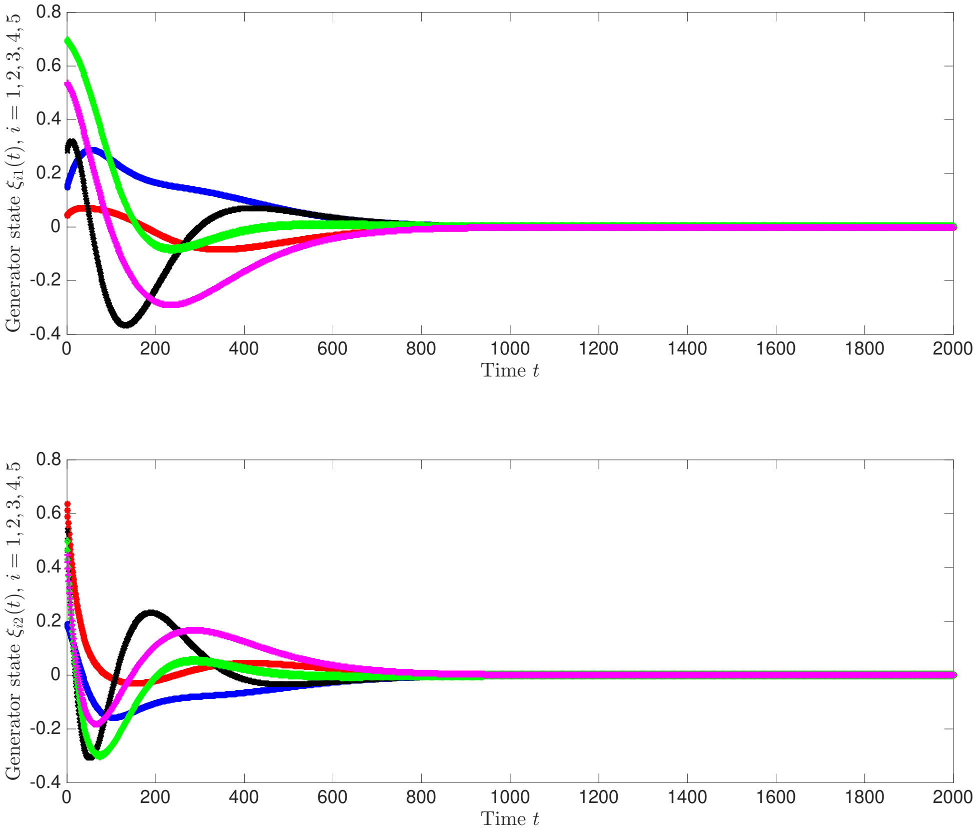


Figure 5.10: Convergence of generator states

pair (C, A) is observable thus detectable. Hence we design the following two matrices F, G to assign the desired eigenvalues

$$-1, -2, -1 + j, -1 - j$$

for both $A + BF$ and $A + GC$:

$$F = \begin{bmatrix} 40.8163 & 102.0408 & 51.0123 & 107.0408 \end{bmatrix}, \quad G = \begin{bmatrix} 107.0408 \\ 51.0123 \\ -112.0408 \\ -61.2083 \end{bmatrix}.$$

Consider the following interconnections of these six inverted pendula (starting from cyclic digraph, and adding one edge at a time), and perform the corresponding simulation of SA in (5.5). Observe from Figs. 5.12–5.17 that with the increasing number of edges, state trajectories are from divergence to convergence (indeed, synchronization of each of the state components among the six pendula). This illustrates a phase transition at which exponentially unstable dynamics are counterbalanced by tight interconnection.

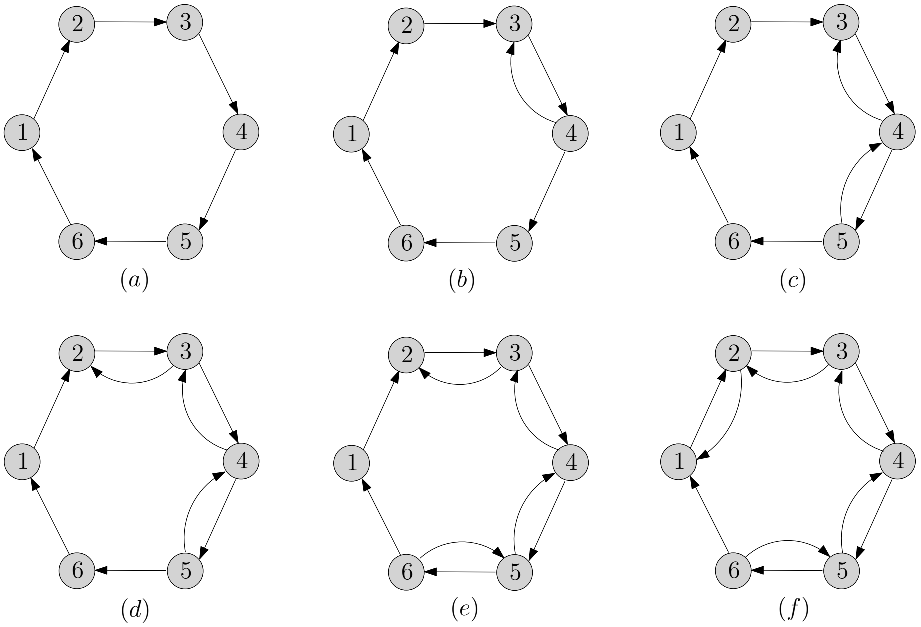


Figure 5.11: Six digraph topologies of 6 inverted pendula

5.5 Notes and References

The synchronization algorithm (SA) is first reported in

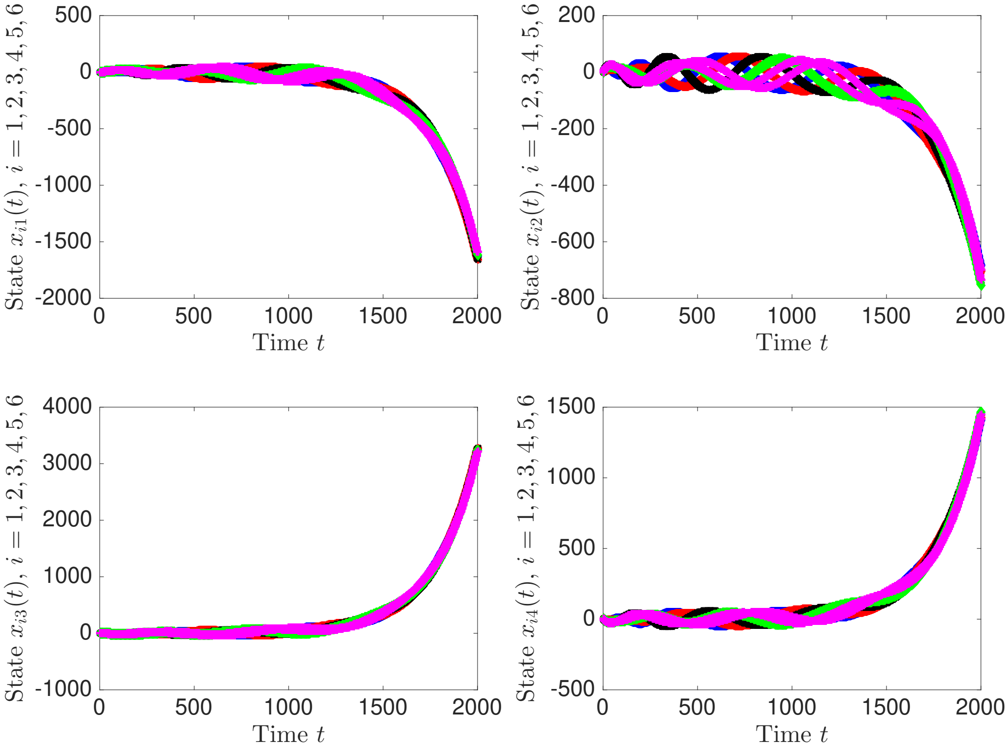


Figure 5.12: Trajectories of state components for Fig. 5.11(a)

- L. Scardovi, R. Sepulchre, Synchronization in networks of identical linear systems, Automatica, vol.45, pp.2557–2562, 2009

Extensions of SA to address time-varying networks, heterogeneous and nonlinear agent dynamics are investigated in

- P. Wieland, R. Sepulchre, F. Allgower, An internal model principle is necessary and sufficient for linear output synchronization, Automatica, vol.47, pp.1068–1074, 2011
- W. Liu, J. Huang, Adaptive leader-following consensus for a class of higher-order nonlinear multi-agent systems with directed switching networks. Automatica, vol.79, pp.84–92, 2017
- S. Kawamura, K. Cai, M. Kishida, Distributed output regulation of heterogeneous uncertain linear agents, Automatica, vol.119, 2020

Pole Assignment Theorem (Lemma 5.2) is from

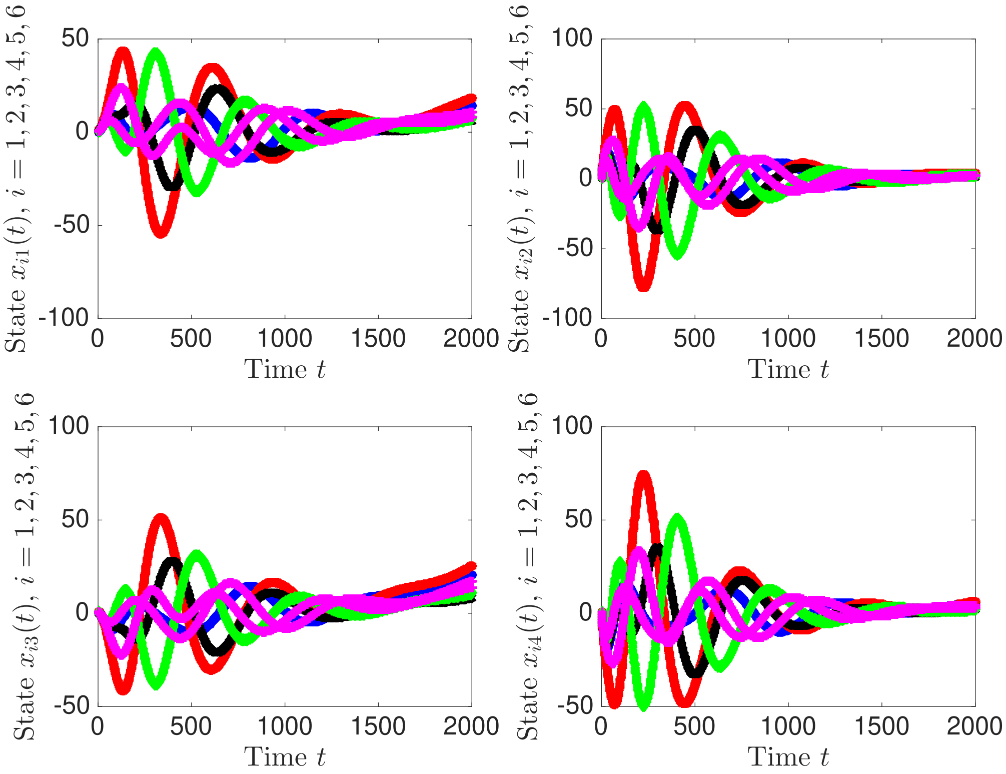


Figure 5.13: Trajectories of state components for Fig. 5.11(b)

- W.M. Wonham, On pole assignment in multi-input controllable linear systems, IEEE Transactions on Automatic Control, vol.12, pp.660–665, 1967

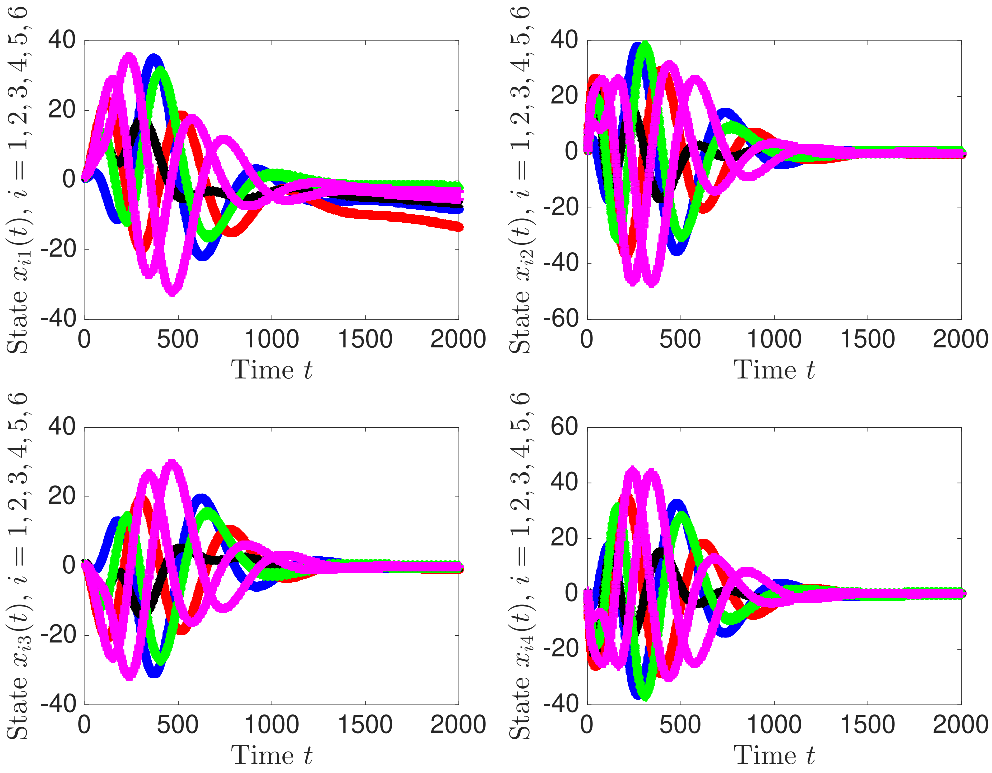


Figure 5.14: Trajectories of state components for Fig. 5.11(c)

5.6 Appendix: Linear Systems and Feedback Control

In this appendix we present fundamental concepts of linear systems and basic designs of feedback control.

Consider a linear time-invariant (LTI) dynamic system:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{5.13}$$

where $x \in \mathbb{R}^p$ is the state vector, $u \in \mathbb{R}^q$ the control input vector, and $y \in \mathbb{R}^r$ the observation output vector. The matrices A, B, C, D are of appropriate sizes.

We say that the pair (A, B) is

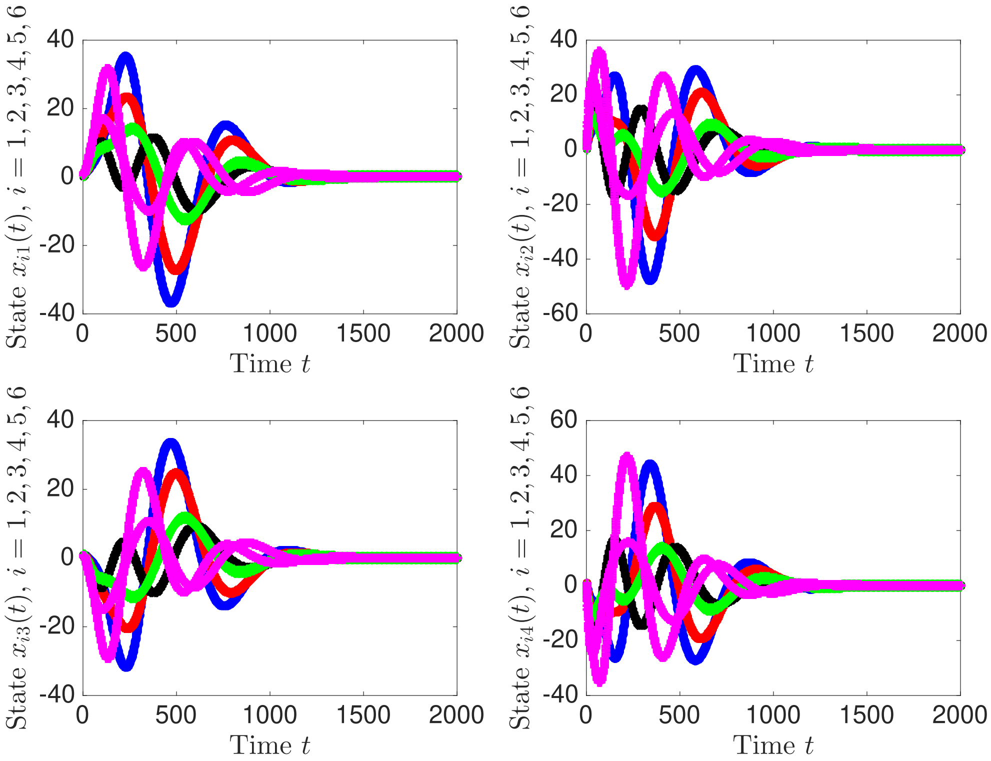


Figure 5.15: Trajectories of state components for Fig. 5.11(d)

- *controllable* if

$$\operatorname{rank}([B\ AB\ \cdots\ A^{p-1}B]) = p;$$

- *stabilizable* if there exists a control input $u = Fx$ such that

$$\text{all the eigenvalues of } A + BF \text{ have negative real parts.}$$

The control $u = Fx$ is called a *state feedback control*, because u is a linear function of the state vector x. State feedback control assumes that all the state components are available (i.e. can be measured/observed) for control, which is equivalent to assuming $C = I$, $D = 0$, and $y = x$ (see Fig. 5.18).

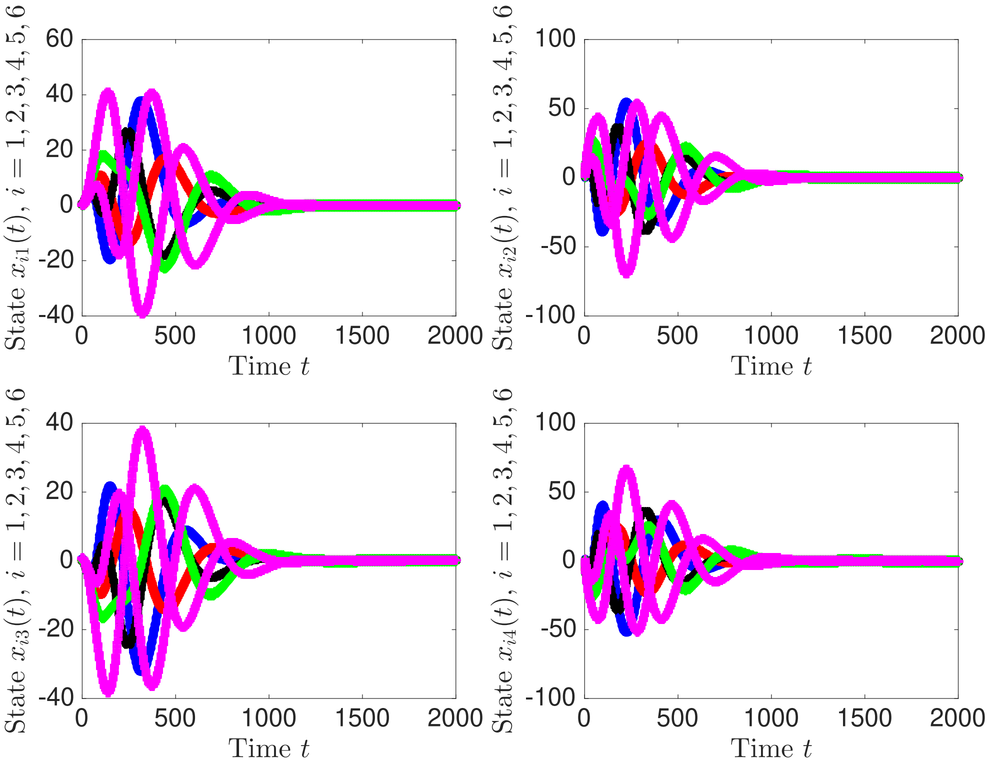


Figure 5.16: Trajectories of state components for Fig. 5.11(e)

Substituting $u = Fx$ into the first equation in (5.13) yields

$$\dot{x} = (A + BF)x. \tag{5.14}$$

This is called the *closed-loop* system (under state feedback control). We say that the closed-loop system is stable if its state $x(t) \to 0$ as $t \to \infty$. According to (5.14), the closed-loop system is stable if and only if all the eigenvalues of $A + BF$ have negative real parts, i.e. (A, B) is stabilizable. Hence stabilizability of the pair (A, B) is a necessary and sufficient condition for the stability of the closed-loop system under state feedback control.

It is also important to point out that if (A, B) is controllable, then (A, B) is stabilizable (the reverse need not hold). Thus the stabilizability of (A, B) may be verified by the rank condition of controllability. One explanation of this relation between controllability and stabilizability is the following.

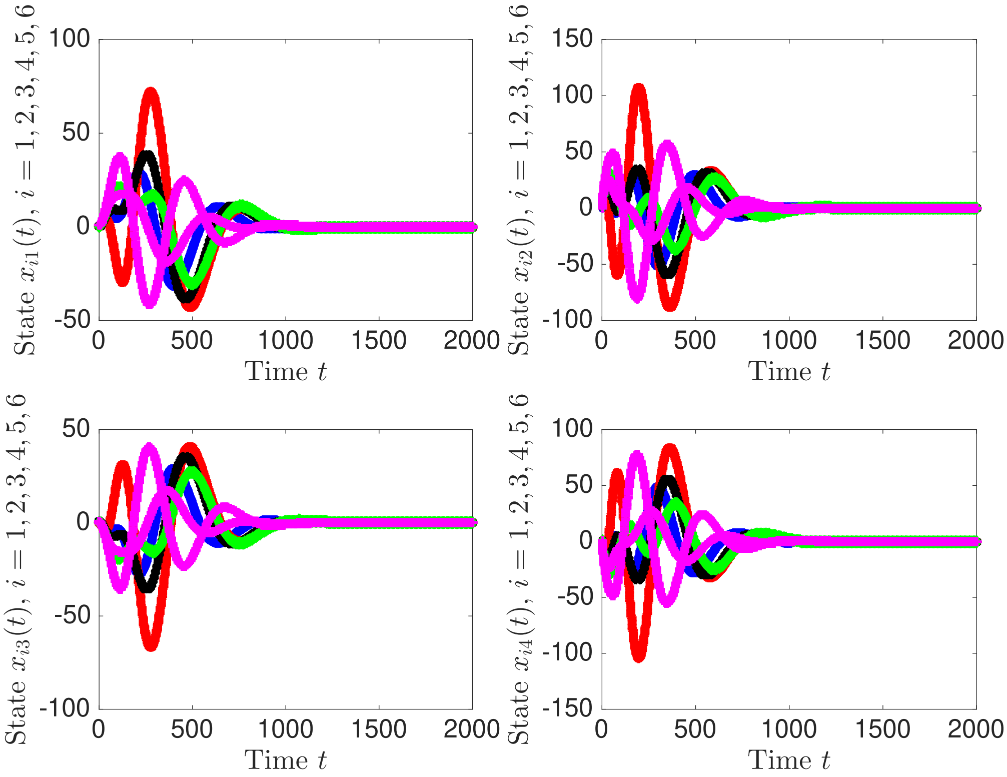


Figure 5.17: Trajectories of state components for Fig. 5.11(f)

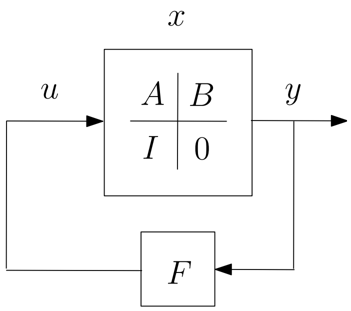


Figure 5.18: State feedback control

Lemma 5.2 (Pole Assignment Theorem) *Consider an LTI system in (5.13). The pair (A, B) is controllable if and only if for an arbitrary set of complex numbers $\{\lambda_1, \ldots, \lambda_p\}$ which are symmetric with respect to the real axis, there exists F such that the eigenvalues of*

$A + BF$ are $\lambda_1, \ldots, \lambda_p$.

If the entire state vector x is not available, then feedback control design has to be based on the observation output y. We say that the pair (C, A) is

- *observable* if

$$\operatorname{rank}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix}\right) = p;$$

- *detectable* if there exists G such that

$$\text{all the eigenvalues of } A + GC \text{ have negative real parts.}$$

It is observed that observability and detectability are dual respectively with controllability and stabilizability:

- (C, A) is observable if and only if $(A^\top, C^\top)$ is controllable;
- (C, A) is detectable if and only if $(A^\top, C^\top)$ is stabilizable.

As a result, if (C, A) is observable then (C, A) is detectable, while the reverse is false in general.

If the pair (C, A) is detectable, an *observer* can be constructed to estimate the true x:

$$\dot{\hat{x}} = A\hat{x} + Bu + G(C\hat{x} + Du - y) \tag{5.15}$$

where $\hat{x}$ is the estimated state vector. To see this, consider the error between the estimated state $\hat{x}$ and the true state x, i.e. $e := \hat{x} - x$. Take the time derivative of e to obtain

$$\begin{aligned} \dot{e} &= \dot{\hat{x}} - \dot{x} \\ &= (A\hat{x} + Bu + G(C\hat{x} + Du - y)) - (Ax + Bu) \\ &= (A + GC)(\hat{x} - x) \\ &= (A + GC)e. \end{aligned}$$

Since (C, A) is detectable, there exists G such that all the eigenvalues of $A+GC$ have negative real parts. This means that the error $e(t) \to 0$ as $t \to \infty$, namely the estimated state $\hat{x}$ converges to the true state x.

Now that an observer can be designed to estimate the true state, we may consider feeding back the estimate as was done in state feedback control (namely pretending that the estimated state was the true state). This leads to the following *output feedback control*:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + G(C\hat{x} + Du - y) \\ u &= F\hat{x}. \end{aligned} \tag{5.16}$$

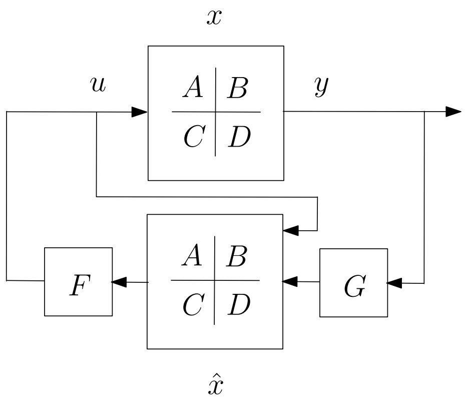


Figure 5.19: Output feedback control

Under the above output feedback control, the *closed-loop system* is displayed in Fig. 5.19. The overall state of the closed-loop system is

$$\begin{bmatrix} x \\ \hat{x} \end{bmatrix}.$$

Combining (5.13) and (5.16) yields the dynamics of the closed-loop system as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BF \\ -GC & A+BF+GC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}.$$

We say that the closed-loop system under output feedback control is stable if its state $[x(t)^\top \;\; \hat{x}(t)^\top]^\top \to 0$ as $t \to \infty$. According to (5.14), the closed-loop system is stable if and only if all the eigenvalues of the matrix

$$\begin{bmatrix} A & BF \\ -GC & A+BF+GC \end{bmatrix} =: M$$

have negative real parts. For this to holds, a necessary and sufficient condition is that (C, A) is detectable and (A, B) is stabilizable. To see this, consider the following similarity transformation of M:

$$\begin{aligned} T^{-1}MT &= \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} A & BF \\ -GC & A+BF+GC \end{bmatrix} \begin{bmatrix} I & 0 \\ I & I \end{bmatrix} \\ &= \begin{bmatrix} A+BF & BF \\ 0 & A+GC \end{bmatrix}. \end{aligned}$$

Hence the spectrum (set of eigenvalues) of M is the union of the spectra of $A+BF$ and $A+GC$. Therefore all the eigenvalues of M have negative real parts if and only if (A, B) is stabilizable and (C, A) is detectable.

We close this appendix by recapitulating the following facts:

- Under state feedback control (5.14), the closed-loop system is stable if and only if (A, B) is stabilizable.

- Under output feedback control (5.16), the closed-loop system is stable if and only if (A, B) is stabilizable and (C, A) is detectable.

Part IV
Spanning Two-Tree Digraphs: Similar Formation and Localization

This part introduces distributed similar formation control and localization in two-dimensional space. The necessary graphical condition for solving these two problems is that digraphs contain a spanning 2-tree. The type of Laplacian matrices involved in these two problems is the complex Laplacian matrices. For agent dynamics, linear time-invariant first-order systems are considered, with continuous-time for similar formation control while discrete-time for localization.

Chapter 6

Similar Formation in Two-Dimensional Space

In this chapter, we introduce a formation control problem of multi-agent systems in two-dimensional (2D) space. The consensus problem studied in Chapter 4 can be viewed as to achieve a special point formation (all the agents reach consensus on their positions in both dimensions respectively). In this sense, the formation control problem in this chapter includes consensus and generalize it to a rich set of geometric shapes in 2D.

Formation control is an interesting fundamental topic in teams of autonomous robots, mobile sensors, unmanned aerial vehicles, and autonomous underwater vehicles. Important applications of formation control include source seeking and exploration, map construction, formation flying, and ocean data retrieval. This chapter focuses on formation control in 2D, while 3D formation control is covered in Chapter 8.

Specifically, the problem studied in this chapter is called *similar formation control*: a network of agents is required to form a geometric shape, which can be obtained from a prescribed desired shape via planar translation, rotation, and scaling. To solve this 2D similar formation control problem, we introduce the second type of graph Laplacian: *complex Laplacian*. Modeling the interacting agents by digraphs, we show that a necessary graphical condition to achieve similar formation is that the digraph contains a *spanning* 2-*tree*, namely there exists (at least) two agents that can reach all the other agents through independent paths. These two root agents play the role of *leaders*, which determine the translation, rotation, and scaling offsets from the prescribed shape. Under this graphical condition, we present a distributed algorithm for the agents to achieve similar formations.

6.1 Problem Statement

Consider a network of n (> 1) agents in a plane (2D space). Each agent i $(\in [1, n])$ has a *state* variable $x_i(t) \in \mathbb{C}$, which is complex and denotes the position of agent i in the plane at time t. Thus $\mathrm{Re}(x_i(\cdot))$ and $\mathrm{Im}(x_i(\cdot))$ are the positions of agents i on the real and imaginary axes, respectively. The time $t \geq 0$ is a (nonnegative) real number and denotes the *continuous* time. The motion of each agent is governed by the following:

$$\dot{x}_i = u_i, \quad i \in [1, n] \tag{6.1}$$

where $u_i(t) \in \mathbb{C}$ is the (complex) control input at time t. Thus $\mathrm{Re}(x_i(\cdot))$ (resp. $\mathrm{Im}(x_i(\cdot))$) is the control input along the real axis (resp. imaginary axis).

Let digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ model the interconnection structure of the n agents. Each *node* in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each directed *edge* (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent i can measure the *relative position* of agent j (namely $x_j - x_i$ in agent i's coordinate frame). The *neighbor set* of agent i is $\mathcal{N}_i := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$.

Moreover, consider that digraph $\mathcal{G}$ is weighted: each edge $(j, i) \in \mathcal{V}$ is associated with a complex weight $a_{ij} \in \mathbb{C}$. Hence the adjacency matrix $A = (a_{ij})$, degree matrix $D = \mathrm{diag}(A\mathbf{1})$, and Laplacian matrix $L = D - A$ are all complex.

Define a *target configuration* $\xi = [\xi_1 \cdots \xi_n]^\top \in \mathbb{C}^n$ to be the assignment of the n agents to points in the plane, which specifies the formation *shape* that the agents are tasked to achieve. Given a target configuration ξ, we say that another configuration ξ' is *similar* to ξ if

$$(\exists \omega_1, \omega_2 \in \mathbb{C})\xi' = \omega_1 \mathbf{1} + \omega_2 \xi.$$

Write $\omega_2 = \rho \mathrm{e}^{\theta}$, $\rho \geq 0$ and $\theta \in [0, 2\pi)$. Then ξ' can be obtained from ξ via (two-dimensional) translation ω_1, rotation θ, and scaling ρ.

For example, Fig. 6.1 displays a target configuration

$$\xi = [1 \;\; \mathrm{e}^{\frac{\pi}{3}\mathrm{j}} \;\; \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}} \;\; \mathrm{e}^{\pi\mathrm{j}} \;\; \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}} \;\; \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$$

which is a regular hexagon. Also displayed is another configuration ξ' similar to ξ, as it can be obtained from ξ via translation ω_1, rotation θ, and scaling ρ.

For a given target configuration ξ, let

$$\mathcal{S}(\xi) := \{\xi' \in \mathbb{C}^n \mid (\exists \omega_1, \omega_2 \in \mathbb{C})\xi' = \omega_1 \mathbf{1} + \omega_2 \xi\} \tag{6.2}$$

be the family of all configurations similar to ξ. Thus $\mathcal{S}(\xi)$ is the (complex) span of the two vectors $\mathbf{1}$ and ξ. If $\xi = c\mathbf{1}$ for some $c \in \mathbb{C}$, then $\mathcal{S}(\xi)$ is degenerated and we are back to consensus in the plane. To consider more general planar formations, we henceforth assume in this chapter that ξ is *linearly independent* from $\mathbf{1}$. Towards the end of this section, we will see that another condition (called 'generic') needs to be imposed on ξ. We say that the n agents with the aggregated state vector $x = [x_1 \cdots x_n]^\top$ form a *similar formation* with respect to ξ if $x \in \mathcal{S}(\xi)$.

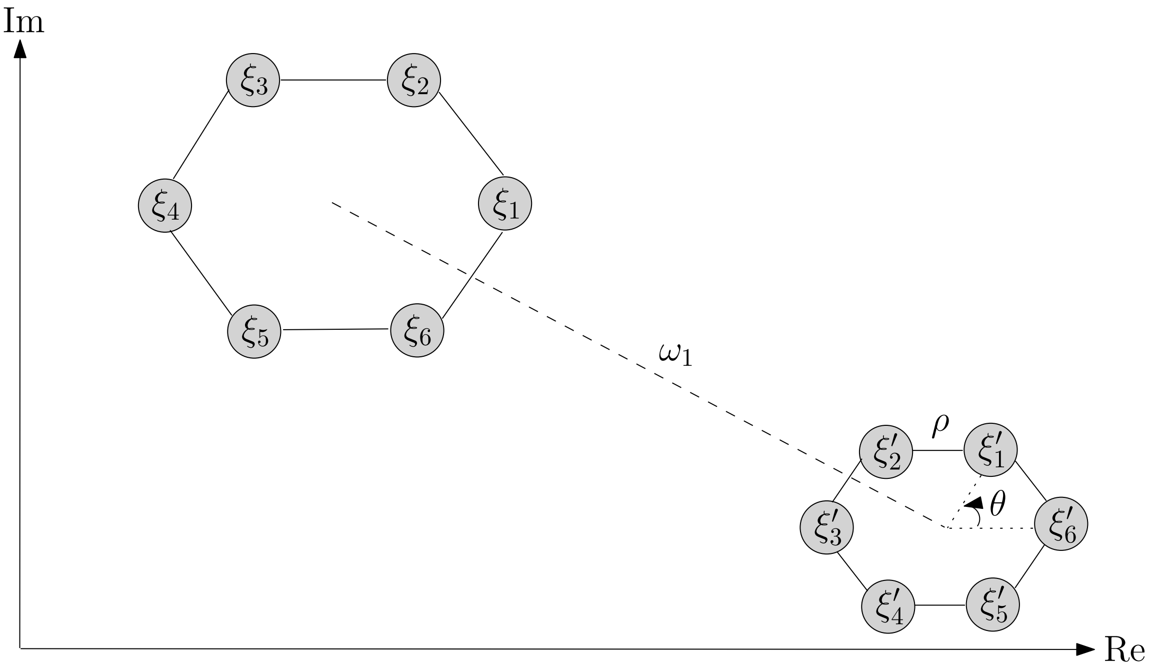


Figure 6.1: Illustration of target configuration and similar configuration

To achieve a similar formation, consider the distributed control

$$u_i = \sum_{j \in \mathcal{N}_i} w_{ij}(x_j - x_i) \tag{6.3}$$

where the control gain $w_{ij} \in \mathbb{C}$ satisfies

$$\text{(i)} \sum_{j \in \mathcal{N}_i} w_{ij}(\xi_j - \xi_i) = 0 \tag{6.4}$$

$$\text{(ii)}\ w_{ij} = \epsilon_i a_{ij}, \quad \epsilon_i \in \mathbb{C}, \epsilon_i \neq 0. \tag{6.5}$$

This control (6.3) is in the same form as that for consensus, but the gains w_{ij} are not simply the edge weights a_{ij}. Indeed, w_{ij} is a complex multiple of a_{ij} (6.5), and moreover satisfies a linear constraint with respect to the target configuration ξ (6.4).

Substituting (6.5) into (6.4) and removing the common multiple ϵ_i yield

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0. \tag{6.6}$$

This in matrix form is $L\xi = 0$; namely the target configuration lies in the null space of the complex

Laplacian of the (complex-)weighted digraph. Since we also have $L\mathbf{1} = 0$, it follows that

$$\ker L \supseteq \mathcal{S}(\xi). \tag{6.7}$$

Thus if the control in (6.3) satisfying (6.4) and (6.5) can be found, the kernel of the complex Laplacian at least contains the family of all configurations similar to the target ξ.

Similar Formation Control Problem:

Consider a network of agents modeled by (6.1) interconnected through a digraph, and let $\xi \in \mathbb{C}^n$ be a target configuration (linearly independently of $\mathbf{1}$). Design a distributed control $u_i(t)$ in (6.3) such that

$$\begin{aligned}&\text{(i)}\ \ \ker L = \mathcal{S}(\xi)\\ &\text{(ii)}\ \ (\forall x(0) \in \mathbb{C}^n)(\exists \xi' \in \mathcal{S}(\xi)) \lim_{t\to\infty} x(t) = \xi'.\end{aligned}$$

The first requirement (i) strengthens (6.7) to equality; namely the kernel of the complex Laplacian is *exactly* the family of all configurations similar to ξ. The second requirement (ii) means that every trajectory of the networked agents converges to a similar formation in $\mathcal{S}(\xi)$.

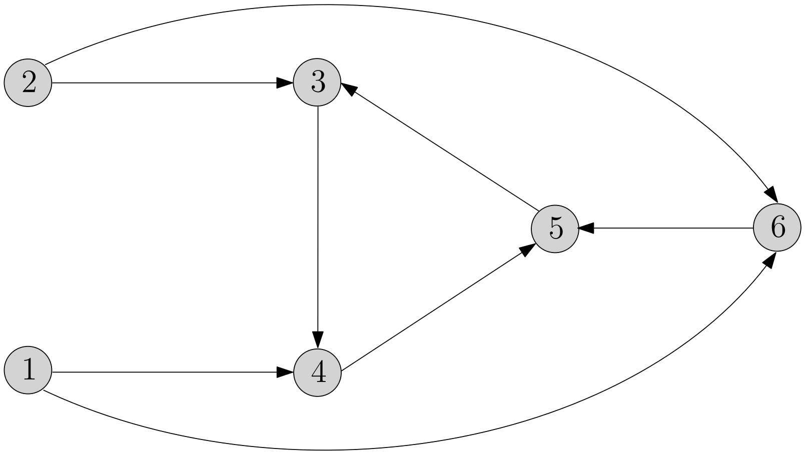


Figure 6.2: Illustrating example of six agents

Example 6.1 *We provide an example to illustrate the similar formation control problem. As displayed in Fig. 6.2, six agents are interconnected through a digraph. The neighbor sets of the agents are* $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$, $\mathcal{N}_3 = \{2,5\}$, $\mathcal{N}_4 = \{1,3\}$, $\mathcal{N}_5 = \{4,6\}$, *and* $\mathcal{N}_6 = \{1,2\}$. *Let the target configuration be* $\xi = [1\ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}}\ \mathrm{e}^{\pi\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$, *i.e. the desired formation*

shape is a regular hexagon (see Fig. 6.1). Thus the family $\mathcal{S}(\xi)$ contains all hexagons that can be obtained from ξ by translation, rotation, and scaling.
The similar formation control problem is to design a distributed control $u_i(t)$ in (6.3) such that the kernel of the complex graph Laplacian coincides with $\mathcal{S}(\xi)$, and moreover the agents' aggregated state vector asymptotically converges to a similar formation in $\mathcal{S}(\xi)$.

A necessary graphical condition for solving the similar formation control problem is given below.

Proposition 6.1 *Suppose that there exists a distributed control $u_i(t)$ in (6.3) that solves the similar formation control problem. Then the digraph contains a spanning 2-tree.*

Proof. Let ξ be a target configuration. Suppose that there exists a distributed control in (6.3) that solves the similar formation control problem with respect to ξ, but that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ does *not* contain a spanning 2-tree. We will derive a contradiction that $\ker L \supsetneqq \mathcal{S}(\xi)$, thereby proving that $\mathcal{G}$ must contain a spanning 2-tree.

First, by definition $\mathcal{G}$ containing no spanning 2-tree means the following. Let $\mathcal{R} = \{v_i, v_j\}$ be a set of arbitrary two nodes. Then after removing a node $v_k \in \mathcal{V} \setminus \mathcal{R}$ and all its incoming and outgoing edges, a subset $\mathcal{V}_k \subsetneqq \mathcal{V} \setminus \{v_k\}$ is unreachable from $\mathcal{R}$ in the new subdigraph $\mathcal{G}'$. We write this as $\mathcal{R} \not\to \mathcal{V}_k$ in $\mathcal{G}'$.

Now let $\bar{\mathcal{V}}_k := \mathcal{V} \setminus (\mathcal{V}_k \cup \{v_k\})$. This set $\bar{\mathcal{V}}_k$ is nonempty because $\mathcal{R} \subseteq \bar{\mathcal{V}}_k$ (trivially). In addition, even after removing v_k, the nodes in $\bar{\mathcal{V}}_k$ can still be reached from $\mathcal{R}$, i.e. $\mathcal{R} \to \bar{\mathcal{V}}_k$ in $\mathcal{G}'$; but $\bar{\mathcal{V}}_k \not\to \mathcal{V}_k$ in $\mathcal{G}'$.

Let $m := |\mathcal{V}_k|$ (≥ 1), and relabel

- nodes $\mathcal{V}_k$ from v_1 to v_m;
- node v_k as v_{m+1};
- nodes in $\bar{\mathcal{V}}_k$ from v_{m+2} to v_n.

Then the complex graph Laplacian L of $\mathcal{G}'$ after relabeling (denoted by L') has the following structure:

$$L' = \begin{bmatrix} L'_{11} & L'_{12} & 0 \\ L'_{21} & L'_{22} & L'_{23} \end{bmatrix}.$$

The 0 matrix in the $(1,3)$-block is due to $\bar{\mathcal{V}}_k \not\to \mathcal{V}_k$ in $\mathcal{G}'$.

Also reorder the components of the target configuration ξ according to the above relabeling, and denote the result by

$$\xi' = \begin{bmatrix} \xi_1' \\ \xi_2' \\ \xi_3' \end{bmatrix}.$$

By the assumption that there exists a distributed control in (6.3), we have $L\xi = 0$ and $L\mathbf{1} = 0$. Substituting the relabeled L' and ξ' into the two equations yields

$$\begin{bmatrix} L_{11}' & L_{12}' \end{bmatrix} \begin{bmatrix} \xi_1' \\ \xi_2' \end{bmatrix} = 0, \quad \begin{bmatrix} L_{11}' & L_{12}' \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} = 0.$$

Since ξ' and $\mathbf{1}$ are linearly independent (linear independence of ξ and $\mathbf{1}$ is assumed at the outset), so are

$$\begin{bmatrix} \xi_1' \\ \xi_2' \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}.$$

Hence the rows of $[L_{11}' \ L_{12}']$ are linearly dependent.

Now remove from L' the two rows corresponding to $\mathcal{R} = \{v_i, v_j\}$ and two arbitrary columns. We still use indices i, j after the above relabeling, but since $\mathcal{R} \subseteq \bar{\mathcal{V}}_k$, it holds that $i, j \in [m+2, n]$. Then the resulting matrix $L_{\mathcal{R}}' \in \mathbb{C}^{(n-2)\times(n-2)}$ is

$$L_{\mathcal{R}}' = \begin{bmatrix} L_{\mathcal{R},11}' & L_{\mathcal{R},12}' & 0 \\ L_{\mathcal{R},21}' & L_{\mathcal{R},22}' & L_{\mathcal{R},23}' \end{bmatrix}.$$

It follows from $i, j \in [m+2, n]$ that $[L_{\mathcal{R},11}' \ L_{\mathcal{R},12}']$ have m rows. Since the m rows of $[L_{11}' \ L_{12}']$ are linearly dependent, so are the m rows of $[L_{\mathcal{R},11}' \ L_{\mathcal{R},12}']$. Thus $L_{\mathcal{R}}'$ has fewer than $n-2$ linearly independent rows, and $\det(L_{\mathcal{R}}') = 0$.

Finally since the set $\mathcal{R}$ of two nodes is arbitrary, the original complex graph Laplacian L of $\mathcal{G}'$ does not have any minor with size $n-2$ that has nonzero determinant. This means that $\operatorname{rank}(L) \leq n-3$, and therefore $\ker L \supsetneqq \mathcal{S}(\xi)$. This is a contradiction to the solvability of the similar formation control problem. The proof is now complete. □

Owing to Proposition 6.1, we shall henceforth assume that the digraph contains a spanning 2-tree.

Assumption 6.1 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning 2-tree.*

Even if Assumption 6.1 holds, not every configuration ξ (linearly independent with $\mathbf{1}$) whose similar configurations may be achieved by a distributed control $u_i(t)$ in (6.3). The following is such an example.

Example 6.2 *Consider again the six-agent digraph in Fig. 6.2. This digraph $\mathcal{G}$ contains a spanning 2-tree, with the root set $\mathcal{R} = \{1, 2\}$. Now consider the following target configuration:*

$$\xi = \begin{bmatrix} 0 \\ -3-3j \\ -1-j \\ -0.8-1.6j \\ 1+j \\ -6j \end{bmatrix}.$$

While ξ is (evidently) linearly independent from $\mathbf{1}$, for every complex Laplacian L of $\mathcal{G}$ with $L\xi = 0$, it is verified that rank$(L) \leq 3$. To see this, write $L\xi$ explicitly as

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 & l_{35} & 0 \\ l_{41} & 0 & l_{43} & l_{44} & 0 & 0 \\ 0 & 0 & 0 & l_{54} & l_{55} & l_{56} \\ l_{61} & l_{62} & 0 & 0 & 0 & l_{66} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix}.$$

For the third row (other rows are similar), it follows from $L\mathbf{1} = 0$ and $L\xi = 0$ that

$$l_{32} + l_{33} + l_{35} = 0$$
$$l_{32}\xi_2 + l_{33}\xi_3 + l_{35}\xi_5 = 0.$$

To satisfy these two equations, the entries l_{32}, l_{33}, l_{35} are such that

$$\begin{bmatrix} l_{32} \\ l_{33} \\ l_{35} \end{bmatrix} = c_3 \begin{bmatrix} \xi_5 - \xi_3 \\ \xi_2 - \xi_5 \\ \xi_3 - \xi_2 \end{bmatrix} = c_3 \begin{bmatrix} 2+2\mathrm{j} \\ -4-4\mathrm{j} \\ 2+2\mathrm{j} \end{bmatrix}$$

for some nonzero complex number c_3. Similarly the (three) entries of rows 4,5,6 may be determined up to a nonzero complex multiples c_4, c_5, c_6 (respectively). For simplicity, letting

$c_3 = c_4 = c_5 = c_6 = 1$ we have one instance of L as follows:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2+2\mathrm{j} & -4-4\mathrm{j} & 0 & 2+2\mathrm{j} & 0 \\ 0.2-0.6\mathrm{j} & 0 & 0.8+1.6\mathrm{j} & -1-\mathrm{j} & 0 & 0 \\ 0 & 0 & 0 & -1-7\mathrm{j} & -0.8+4.4\mathrm{j} & 1.8+2.6\mathrm{j} \\ 3-3\mathrm{j} & 6\mathrm{j} & 0 & 0 & 0 & -3-3\mathrm{j} \end{bmatrix}.$$

This L has rank 3, meaning that the last four rows are linearly dependent. Then for arbitrary values of c_3, c_4, c_5, c_6, these four rows cannot become linearly independent. Hence rank$(L) \leq 3$ for every L with $L\xi = 0$. This means that $\ker L \supsetneqq \mathcal{S}(\xi)$, and consequently there does not exist a distributed control in (6.3) that solves the similar formation control problem with the chosen target configuration ξ.

The target configuration ξ in the above example satisfies a linear algebraic equation with integer coefficients:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 4 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -3-3j \\ -1-j \\ -0.8-1.6j \\ 1+j \\ -6j \end{bmatrix} = 0.$$

Such a configuration ξ is called *non-generic*. Geometrically, in the plane there are four components of ξ (1st, 2nd, 3rd, and 5th) on the same line.

Since Example 6.2 shows a case where similar formations of a non-generic configuration may not be achievable on a digraph containing a spanning 2-tree, we henceforth require that the target configuration be generic. A configuration $\xi = [\xi_1 \cdots \xi_n]^\top \in \mathbb{C}^n$ is said to be *generic* if ξ_i's do not satisfy any nontrivial algebraic equation with integer coefficients. Intuitively speaking, a generic configuration has no degeneracy: in 2D, no three points on the same line and no three lines go through the same point. As a consequence, any generic configuration ξ is linearly independent with $\mathbf{1}$.

It is noted, however, that not all non-generic configurations whose similar configurations cannot be achieved. In fact, if the digraph considered in Example 6.2 had one more edge $(1, 3)$, the non-generic configuration ξ's similar configurations could be achievable. Indeed, following the same procedure described in Example 6.2, with a new edge $(1, 3)$ we derive an instance of the new

Laplacian below:

$$L' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2+2\mathrm{j} & -4-4\mathrm{j} & 0 & 2+2\mathrm{j} & 0 \\ 0.2-0.6\mathrm{j} & 0 & 0.8+1.6\mathrm{j} & -1-\mathrm{j} & 0 & 0 \\ 0 & 0 & 0 & -1-7\mathrm{j} & -0.8+4.4\mathrm{j} & 1.8+2.6\mathrm{j} \\ 3-3\mathrm{j} & 6\mathrm{j} & 0 & 0 & 0 & -3-3\mathrm{j} \end{bmatrix}.$$

The only change is the $(3,1)$-entry from 0 to 1, owing to the added edge $(1,3)$. This L' has rank 4; therefore $\ker L' = \mathcal{S}(\xi)$. Thus one may consider imposing further digraph connectivity to deal with non-generic configurations.

On the other hand, the set of all non-generic configurations has Lebesgue measure zero, because random perturbations destroy integer-coefficient algebraic equations. This means that for a given non-generic configuration ξ (e.g. the one in Example 6.2), randomly perturbing its components generates a generic configuration. For this reason, we assume that the target configuration ξ is generic.

Assumption 6.2 *The target configuration $\xi = [\xi_1 \cdots \xi_n]^\top \in \mathbb{C}^n$ is generic.*

Remark 6.1 (Global versus local coordinate frames) *We end this section with a discussion on the local coordinate frames of the agents with respect to the global coordinate frame. So far the state x_i and control u_i of agent i that we have discussed are in the global coordinate frame Σ. In formation control, the agents are usually robots with onboard sensors, thus having their own local coordinate frames that are not necessarily aligned with the global Σ and time-varying. For distributed control, knowledge of Σ is often not available and thus should not be assumed. Let the local frame of agent i at time t be $\Sigma_i(t)$, whose orientation is $\theta_i(t)$ counterclockwise from the orientation of Σ. Also let $x_{i,\mathrm{loc}}(t)$ and $u_{i,\mathrm{loc}}(t)$ be (respectively) the state and control at time t of agent i in $\Sigma_i(t)$. Then*

$$\begin{aligned} x_i(t) &= x_{i,\mathrm{loc}}(t)e^{-j\theta_i(t)} \\ u_i(t) &= u_{i,\mathrm{loc}}(t)e^{-j\theta_i(t)}. \end{aligned}$$

Recall from (6.3) that

$$u_i(t) = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j(t) - x_i(t)).$$

Substituting the above equation of $x_i(t)$ into the right-hand-side yields

$$\begin{aligned} u_i(t) &= \sum_{j\in\mathcal{N}_i} w_{ij}(x_{j,\mathrm{loc}}(t)e^{-j\theta_i(t)} - x_{i,\mathrm{loc}}(t)e^{-j\theta_i(t)}) \\ &= \sum_{j\in\mathcal{N}_i} w_{ij}(x_{j,\mathrm{loc}}(t) - x_{i,\mathrm{loc}}(t))e^{-j\theta_i(t)}. \end{aligned}$$

Now equating the right-hand-sides of the above two $u_i(t)$-equations, we derive

$$u_{i,\mathrm{loc}}(t) = \sum_{j\in\mathcal{N}_i} w_{ij}(x_{j,\mathrm{loc}}(t) - x_{i,\mathrm{loc}}(t)).$$

This shows that the control $u_{i,\mathrm{loc}}(t)$ in the local $\Sigma_i(t)$ is unaffected by the time-varying orientation difference from the global Σ. Hence the control u_i in (6.3), though with respect to the global frame Σ, may be implemented in agent i's local frame $\Sigma_i(t)$ (as $u_{i,\mathrm{loc}}$) based on the state difference $x_{j,\mathrm{loc}} - x_{i,\mathrm{loc}}$ in $\Sigma_i(t)$ as well. With this justification and for simplicity, we will write u_i, x_i (instead of $u_{i,\mathrm{loc}}$, $x_{i,\mathrm{loc}}$)

6.2 Distributed Algorithm

Example 6.3 *Consider again Example 6.1, where the target configuration is the regular hexagon $\xi = [1\ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}}\ \mathrm{e}^{\pi\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$. This ξ is generic.*

To achieve a similar formation of ξ, we consider using the simplest form of the distributed control (6.3) by setting all $\epsilon_i = 1$:

$$\dot{x}_i = \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [1,6] \tag{6.8}$$

where $a_{ij} \in \mathbb{C}$ are complex weights of edges to be designed to satisfy (6.6):

$$\sum_{j\in\mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0, \quad i \in [1,6].$$

In Fig. 6.3, we illustrate how such complex weights may be designed. For agent 3, it has two neighbors 2, 5. Thus we need to find weights a_{32}, a_{52} such that

$$a_{32}(\xi_2 - \xi_3) + a_{35}(\xi_5 - \xi_3) = 0.$$

Writing a_{32}, a_{52} in polar coordinates, the above equation may be satisfied through making *proper rotations and scalings (dashed arrows in Fig. 6.3), i.e.*

$$\rho_{32}\mathrm{e}^{\theta_{32}\mathrm{j}}(\xi_2-\xi_3)+\rho_{35}\mathrm{e}^{\theta_{35}\mathrm{j}}(\xi_5-\xi_3)=0.$$

There are infinitely many choices; a simple one is $\rho_{32}=\sqrt{3}, \theta_{32}=0$ *and* $\rho_{35}=1, \theta_{35}=-\frac{\pi}{2}$*. Hence* $w_{32}=\sqrt{3}$*,* $w_{35}=-\mathrm{j}$*. Note that this weight design can be done locally by individual agents if relative information* $\xi_j-\xi_i$ *(*$j\in\mathcal{N}_i$*) is available.*
Similarly we design other complex weights to satisfy (6.6), and write (6.8) in vector form:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\\ \dot{x}_6\end{bmatrix}=\begin{bmatrix}0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&\sqrt{3}&-\sqrt{3}+\mathrm{j}&0&-\mathrm{j}&0\\ \frac{1}{2}&0&-\frac{1}{2}+\frac{\sqrt{3}}{2}\mathrm{j}&-\frac{\sqrt{3}}{2}\mathrm{j}&0&0\\ 0&0&0&\frac{1}{2}+\frac{\sqrt{3}}{2}\mathrm{j}&-\frac{3}{2}-\frac{\sqrt{3}}{2}\mathrm{j}&1\\ -\frac{3}{2}-\frac{\sqrt{3}}{2}\mathrm{j}&1&0&0&0&\frac{1}{2}+\frac{\sqrt{3}}{2}\mathrm{j}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\end{bmatrix}.$$

Inspect that the matrix above has zero row sums, and is indeed the minus of the complex Laplacian matrix L *of the (complex) weighted digraph. It is also checked that* $L\xi=0$*, namely the target configuration lies in the kernel of* L*. Moreover, there are exactly two eigenvalues* 0 *of* L*, and hence* $\ker L=\mathcal{S}(\xi)$ *(the first requirement of the similar formation control problem is satisfied).*
However, the nonzero eigenvalues of matrix $-L$ *are*

$$-1.917+0.8963\mathrm{j}, -1.1283-1.042\mathrm{j}, -0.1867-0.5863\mathrm{j}, 0.5+0.866\mathrm{j}$$

and hence $-L$ *is not stable (the last eigenvalue has positive real part). Therefore to stabilize* $x(t)$ *to the kernel of* L *(to satisfy the second requirement of the similar formation control problem), the unstable eigenvalues of* $-L$ *must be moved to the open left-half plane. This shows that simply setting all* $\epsilon_i=1$ *in (6.3) does not work in general. In fact,* ϵ_i *need to be properly chosen in order to stabilize* $-L$*.*

In the following we re-describe the distributed control (6.3) in vector form, and will analyze its stability in relation to the values of ϵ_i in the next section.

Similar Formation Control Algorithm (SFCA):

Every agent i has a state variable $x_i(t)\in\mathbb{C}$ representing its position in 2D at time $t\geq 0$; the initial state $x_i(0)$ is an arbitrary complex number. Offline, each agent i computes weights

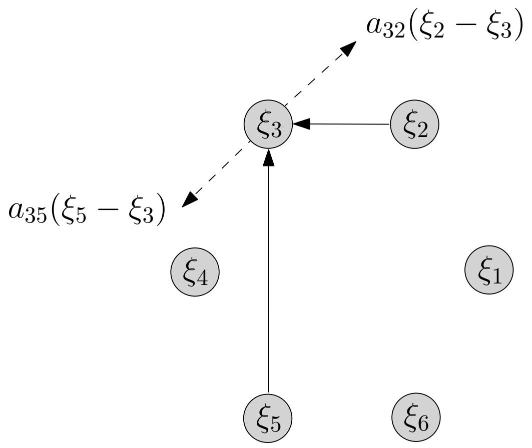


Figure 6.3: Illustration of design of complex weights

$a_{ij} = \rho_{ij}\mathrm{e}^{\theta_{ij}}$ by solving

$$\sum_{j\in\mathcal{N}_i} \rho_{ij}\mathrm{e}^{\theta_{ij}}(\xi_j - \xi_i) = 0 \tag{6.9}$$

such that (6.6) holds. Then online, at each time $t \geq 0$, every agent i updates its state $x_i(t)$ using the following distributed control:

$$u_i = \epsilon_i \sum_{j\in\mathcal{N}_i} a_{ij}(x_j - x_i) \tag{6.10}$$

where $\epsilon_i \in \mathbb{C} \setminus \{0\}$ is a (nonzero) complex control gain.

Let $x := [x_1 \cdots x_n]^\top$ be the aggregated state of the networked agents, and $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ the (diagonal) control gain matrix. Then the n equations (6.10) become

$$\dot{x} = (-EL)x. \tag{6.11}$$

Remark 6.2 *The above AFCA requires that the following information is available for each individual agent i:*

- $\xi_j - \xi_i$ *for all* $j \in \mathcal{N}_i$ *(offline computation of weights)*
- $x_j - x_i$ *for all* $j \in \mathcal{N}_i$ *(online computation of control inputs).*

6.3 Convergence Result

The following is the main result of this section.

Theorem 6.1 *Suppose that Assumptions 6.1 and 6.2 hold. There exists a (diagonal and invertible) control gain matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ *such that the SFCA solves the similar formation control problem.*

To prove Theorem 6.1, we will analyze the eigenvalues of the matrix $-EL$ in (6.11). For this, the following fact is useful.

Lemma 6.1 *Consider an arbitrary square complex matrix* $M \in \mathbb{C}^{n\times n}$*. If all the principal minors of* M *are nonzero, then there exists an invertible diagonal matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{C}^{n\times n}$ *such that all the eigenvalues of* EM *have positive real parts.*

Proof: The proof is based on induction on n. For the base case $n = 1$, $M = m_{11}$ is a nonzero scalar (as the principal minor of M is nonzero). Write $m_{11} = \rho_1 \mathrm{e}^{\mathrm{j}\theta_1}$, and let $\epsilon_1 := \gamma_1 \mathrm{e}^{\mathrm{j}\phi_1}$ where $\gamma_1 \neq 0$ and ϕ_1 is such that $(\phi_1 + \theta_1)(\mathrm{mod}\ 2\pi) \in (-\frac{\pi}{2}, \frac{\pi}{2})$. Then $EM = \epsilon_1 m_{11} = \rho_1\gamma_1 \mathrm{e}^{\mathrm{j}(\phi_1+\theta_1)}$, which has positive real part.

For the induction step, suppose that the conclusion holds for $M \in \mathbb{C}^{(n-1)\times(n-1)}$. Now consider $M \in \mathbb{C}^{n\times n}$, with all of its principal minors nonzero. Let M_1 be the submatrix of M with the last row and last column removed. Then all the principal minors of M_1 are nonzero, and by the hypothesis there exists an invertible diagonal matrix $E_1 = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_{n-1})$ such that all the eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ of E_1M_1 have positive real parts. Now write

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix}$$

where m_{nn} is a nonzero scalar (since all the principal minors of M are nonzero). Also let

$$E = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix}$$

for some complex ϵ_n. Thus

$$EM = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix} = \begin{bmatrix} E_1M_1 & E_1M_2 \\ \epsilon_n M_3 & \epsilon_n m_{nn} \end{bmatrix}$$

If $\epsilon_n = 0$, then

$$EM = \begin{bmatrix} E_1 M_1 & E_1 M_2 \\ 0 & 0 \end{bmatrix}$$

which means that EM has a (simple) eigenvalue $\lambda_n = 0$ and all the rest $n-1$ eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ have positive real parts. Since eigenvalues are continuous functions of matrix entries, for $\epsilon_n := \gamma_n \mathrm{e}^{\mathrm{j}\phi_n}$ with sufficiently small $\gamma_n > 0$, EM still has $n-1$ eigenvalues $\lambda'_1, \ldots, \lambda'_{n-1}$ with positive real parts. This in turn implies that the difference between the angles of λ_i and λ'_i is small for all $i \in [1, n-1]$. Let

$$\delta = |\angle \prod_{i=1}^{n-1} \lambda_i - \angle \prod_{i=1}^{n-1} \lambda'_i|. \tag{6.12}$$

Then δ can be made arbitrarily small by choosing sufficiently small $\gamma_n > 0$.

Now we consider the last eigenvalue λ'_n. Since $\det(E) \neq 0$, $\det(M) \neq 0$, and $\det(EM) = \lambda'_1 \cdots \lambda'_n$, we have $\lambda'_n \neq 0$. It is thus left to show that the angle of λ'_n is in $(-\frac{\pi}{2}, \frac{\pi}{2})$. Noting that

$$\det(EM) = \epsilon_n \det(E_1) \det(M) = \lambda'_1 \cdots \lambda'_{n-1} \lambda'_n$$

we derive

$$\begin{aligned} \angle \lambda'_n &= \angle \epsilon_n + \angle \det(E_1) + \angle \det(M) - \angle \prod_{i=1}^{n-1} \lambda'_i \\ &= \phi_n + \angle \det(E_1) + \angle \det(M) - (\angle \prod_{i=1}^{n-1} \lambda_i \pm \delta). \end{aligned}$$

Choosing

$$\phi_n \in (\angle \prod_{i=1}^{n-1} \lambda_i - \angle \det(E_1) - \angle \det(M) - \frac{\pi}{2} + \delta', \angle \prod_{i=1}^{n-1} \lambda_i - \angle \det(E_1) - \angle \det(M) + \frac{\pi}{2} - \delta')$$

for some positive δ', we have

$$\phi_n \in (-\frac{\pi}{2} + \delta' \mp \delta, \frac{\pi}{2} - \delta' \mp \delta).$$

Since δ can be made arbitrarily small (by choosing sufficiently small $\gamma_n > 0$), in particular δ can be made such that $\delta < \delta'$, thereby we derive $\angle \lambda'_n \in (-\frac{\pi}{2}, \frac{\pi}{2})$. Hence λ'_n also has positive real part. This proves the induction step, and thereby completes the proof. □

The above proof suggests an algorithm (Algorithm 6.1 below) to compute an invertible diagonal matrix $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ such that all the eigenvalues EM have positive real parts. In the algorithm when computing ϵ_i ($i \in [1, n]$) in lines 2 and 6, a specific choice of angles is adopted to render the resulting eigenvalues of EM to positive real numbers. By the proof of Lemma 6.1, one can always choose appropriate (small) $\delta_1, \ldots, \delta_n$ in line 1 so that Algorithm 6.1 outputs an invertible diagonal matrix E that renders all the eigenvalues EM with positive real parts. In the algorithm, the notation $M(1 : i, 1 : i)$ used in lines 7 and 9 denotes the submatrix of M with the first i rows and columns (i.e. the ith leading principal submatrix of M).

Algorithm 6.1 Diagonal Stabilization Algorithm (case of complex matrix, right-half plane)

```
Input: square complex matrix $M \in \mathbb{C}^{n\times n}$ with nonzero principal minors
Output: invertible diagonal matrix $E \in \mathbb{C}^{n\times n}$
 1: set $\delta_1, \ldots, \delta_n$ to be small positive real numbers
 2: $\epsilon_1 = \delta_1 \mathrm{e}^{-\mathrm{j}\angle \det(M(1,1))}$
 3: $E_1 = \mathrm{diag}(\epsilon_1)$
 4: $\{\lambda_1\}$ = spectrum of $E_1 M(1, 1)$
 5: for $i = 2, \ldots, n$ do
 6:     $\Lambda = \lambda_1 \cdots \lambda_{i-1}$
 7:     $\epsilon_i = \delta_i \mathrm{e}^{-\mathrm{j}\angle \frac{\det(E_{i-1})\det(M(1:i,1:i))}{\Lambda}}$
 8:     $E_i = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_i)$
 9:     $\{\lambda_1, \ldots, \lambda_i\}$ = spectrum of $E_i M(1 : i, 1 : i)$
10: end for
11: $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$
```

Lemma 6.1 provides a sufficient condition under which the eigenvalues of a complex matrix may be moved to the open right-half plane using an invertible diagonal complex matrix. The following proposition asserts that this condition holds for the submatrix of complex Laplacian of a digraph containing a spanning 2-tree, with the two rows and two columns corresponding to the two roots removed. More formally, consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let L be a complex Laplacian matrix of $\mathcal{G}$ (corresponding to a specific choice of edge weights). Let $\mathcal{R} \subseteq \mathcal{V}$, and denote by $L_{\mathcal{R}}$ the submatrix of L by removing the rows and columns corresponding to $\mathcal{R}$.

Proposition 6.2 *Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a configuration ξ. Suppose that Assumptions 6.1 and 6.2 hold. Let $\mathcal{R}$ be a set of two roots. Then for almost all complex Laplacian L of $\mathcal{G}$ satisfying $L\xi = 0$, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

To prove Proposition 6.2, we first establish two lemmas.

Lemma 6.2 *Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.*

(i) *Suppose that $\mathcal{G}$ contains a spanning tree. Let $v_1 \in \mathcal{V}$ be a root (renumbering if necessary)* *and $\mathcal{R} := \{v_1\}$. Then for almost all complex Laplacian L of $\mathcal{G}$, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

(ii) *Suppose that $\mathcal{G}$ contains a spanning 2-tree (Assumption 6.1). Let $v_1, v_2 \in \mathcal{V}$ be two roots (renumbering if necessary) and $\mathcal{R} := \{v_1, v_2\}$. Then for almost all complex Laplacian L of $\mathcal{G}$, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

Proof. (i) Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$. Here $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$. Without loss of generality let $v_1 \in \mathcal{V}$ be the root of $\mathcal{T}$ and $\mathcal{R} := \{v_1\}$. Then a standard Laplacian matrix T of $\mathcal{T}$ has the following form:

$$T := \begin{bmatrix} 0 & 0 \\ * & T_{\mathcal{R}} \end{bmatrix}.$$

Since $\mathcal{T}$ is a spanning tree, by Theorem 1.7 we have $\operatorname{rank}(T) = n-1$, and hence $\det(T_{\mathcal{R}}) \neq 0$.

Next let $\mathcal{V}' \subseteq \mathcal{V} \backslash \mathcal{R}$ be an arbitrary nonempty subset of m $(\in [1, n-2])$ nodes, and renumber these nodes from $v_2, \ldots, v_{m+1}$. Also let $\mathcal{R}' := \mathcal{R} \cup \mathcal{V}' = \{v_1, \ldots, v_{m+1}\}$, and remove all the incoming edges from nodes $v_{m+2}, \ldots, v_n$ to $\mathcal{R}'$. Denote the corresponding subgraph by $\mathcal{T}'$. Then a nonnegative adjacency matrix A' and degree matrix D' of $\mathcal{T}'$ have the following forms:

$$A' = \begin{bmatrix} A_1' & 0 \\ A_2' & A_3' \end{bmatrix}, \quad D' = \begin{bmatrix} D_1' & 0 \\ 0 & D_2' \end{bmatrix}.$$

Accordingly a standard Laplacian matrix T' of $\mathcal{T}'$ is

$$T' = D' - A' = \begin{bmatrix} D_1' & 0 \\ 0 & D_2' \end{bmatrix} - \begin{bmatrix} A_1' & 0 \\ A_2' & A_3' \end{bmatrix} =: \begin{bmatrix} T_1' & 0 \\ T_2' & T_{\mathcal{R}'}' \end{bmatrix}.$$

It will be shown that $\det(T_{\mathcal{R}'}') \neq 0$ by proving that $T_{\mathcal{R}'}'$ does not have an eigenvalue 0. To that end, let $\tilde{D}' = \operatorname{diag}(\tilde{d}_1', \ldots, \tilde{d}_1')$ be such that

$$\tilde{d}_1' := \begin{cases} d_i', & \text{if } d_i' \neq 0; \\ 1, & \text{if } d_i' = 0. \end{cases}$$

Thus $\tilde{D}'$ is invertible and use $(\tilde{D}')^{-1}$ to define

$$\tilde{A}' := (\tilde{D}')^{-1} A' = \begin{bmatrix} \tilde{A}_1' & 0 \\ \tilde{A}_2' & \tilde{A}_3' \end{bmatrix}, \quad \tilde{T}' := (\tilde{D}')^{-1} T = I - \tilde{A}' = \begin{bmatrix} \tilde{T}_1' & 0 \\ \tilde{T}_2' & \tilde{T}_{\mathcal{R}'}' \end{bmatrix}.$$

Note that $\tilde{A}'$ is nonnegative and every row sums up to 1. Hence for every integer $k \geq 1$, it holds that $(\tilde{A}')^k$ is nonnegative and every row sums up to 1. Let us focus on $(\tilde{A}')^n$ (i.e. $k = n$), which has the form

$$(\tilde{A}')^n := \begin{bmatrix} (\tilde{A}'_1)^n & 0 \\ X & (\tilde{A}'_3)^n \end{bmatrix}.$$

Since every node in $\mathcal{V} \setminus \mathcal{R}'$ can be reached from some node in $\mathcal{R}'$, it follows from Lemma 1.1 that every row of the $(2,1)$-block X contain positive entries. Hence

$$\begin{aligned} \|(\tilde{A}'_3)^n\|_\infty < 1 &\Rightarrow \rho((\tilde{A}'_3)^n) \leq \|(\tilde{A}'_3)^n\|_\infty < 1 \\ &\Rightarrow \rho(\tilde{A}'_3) < 1 \\ &\Rightarrow \tilde{T}'_{\mathcal{R}'} = I - \tilde{A}'_3 \text{ has no eigenvalue } 0. \end{aligned}$$

It follows that $\tilde{T}'_{\mathcal{R}'}$ has full rank, and so does $T'_{\mathcal{R}'} = D'_3 \tilde{T}'_{\mathcal{R}'}$. The latter means that $T'_{\mathcal{R}'}$ has no eigenvalue 0. Hence $\det(T'_{\mathcal{R}'}) \neq 0$. Compared with T', T has more nonzero entries. According to the fact that a polynomial is either constantly zero or nonzero almost everywhere, it follows from $\det(T'_{\mathcal{R}'}) \neq 0$ that $\det(T_{\mathcal{R}'}) \neq 0$ for almost all T. Therefore for almost all standard Laplacian T, all principal minors of $T_{\mathcal{R}}$ are nonzero.

Finally consider a complex Laplacian L of the digraph $\mathcal{G}$. Compared with T, L has more nonzero complex entries. Again according to the fact that a polynomial is either constantly zero or nonzero almost everywhere, we conclude that for almost all complex Laplacian L, all principal minors of $L_{\mathcal{R}}$ are nonzero.

(ii) Suppose that $\mathcal{G}$ contains a spanning 2-tree with a root set $\mathcal{R} := \{v_1, v_2\}$ (without loss of generality). Remove either node, say v_1, and all its incoming and outgoing edges, and denote the resulting subgraph $\mathcal{G}'$. Then $\mathcal{G}'$ contains a spanning tree (v_2 being a root). It then follows from (i) above that for almost all complex Laplacian L' of $\mathcal{G}'$, all the principal minors of $L'_{\{v_2\}}$ are nonzero. Since the principal minors of $L'_{\{v_2\}}$ are identical with those of $L_{\mathcal{R}}$, where L is a complex Laplacian of $\mathcal{G}$, the conclusion is established. □

For the second lemma, we introduce the following notation. Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let L be a complex Laplacian matrix of $\mathcal{G}$. Let $\mathcal{R} \subseteq \mathcal{V}$, and denote by $L^{\mathcal{R}}$ a submatrix of L by removing the rows corresponding to $\mathcal{R}$ and arbitrary $|\mathcal{R}|$ columns.

Lemma 6.3 *Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.*

(i) *Suppose that $\mathcal{G}$ contains a spanning tree. Let $v_1 \in \mathcal{V}$ be a root (renumbering if necessary) and $\mathcal{R} := \{v_1\}$. Then for almost all complex Laplacian L of $\mathcal{G}$, det$(L^{\mathcal{R}}) \neq 0$.*

(ii) *Suppose that $\mathcal{G}$ contains a spanning 2-tree (Assumption 6.1). Let $v_1, v_2 \in \mathcal{V}$ be two roots* *(renumbering if necessary) and $\mathcal{R} := \{v_1, v_2\}$. Then for almost all complex Laplacian L of $\mathcal{G}$, $det(L^{\mathcal{R}}) \neq 0$.*

Proof. (i) Suppose that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$. Here $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$. Without loss of generality let $v_1 \in \mathcal{V}$ be the root of $\mathcal{T}$ and $\mathcal{R} := \{v_1\}$. Also let T be a complex Laplacian of $\mathcal{T}$, and $T^{\mathcal{R}}$ be a submatrix of T with the row $p_1 (= 0)$ corresponding to root v_1 and an arbitrary column q_i removed. If $i = 1$, it follows from Lemma 6.2(i) that $\det(T^{\mathcal{R}}) = \det(T_{\mathcal{R}}) \neq 0$ for almost all T. If $i \neq 1$, let p_i be the ith row of T and consider the following elementary row transformation:

$$T = \begin{bmatrix} p_1 \\ \vdots \\ p_i \\ \vdots \end{bmatrix} \Longrightarrow \tilde{T} := \begin{bmatrix} p_1 + p_i \\ \vdots \\ p_i \\ \vdots \end{bmatrix} = \begin{bmatrix} p_i \\ \vdots \\ p_i \\ \vdots \end{bmatrix}.$$

Denote by $\tilde{\mathcal{T}}$ the digraph corresponding to $\tilde{T}$. Compared with $\mathcal{T}$, some incoming edges are added to node v_1 in $\tilde{\mathcal{T}}$. Hence v_1 is still a root of $\tilde{\mathcal{T}}$. Moreover, since $\tilde{T}(1, i) = T(i, i) \neq 0$, there is an edge from v_i to v_1 in $\tilde{\mathcal{T}}$, and thus v_i is also a root. Let $\tilde{\mathcal{R}} := \{v_i\}$. Then it follows from Lemma 6.2(i) that $\det(\tilde{T}_{\tilde{\mathcal{R}}}) \neq 0$ for almost all $\tilde{T}$. Since $T^{\mathcal{R}}$ is $\tilde{T}_{\tilde{\mathcal{R}}}$ by reordering the 1st row to the ith position (i.e. via elementary row transformations), we derive $\det(T^{\mathcal{R}}) = \det(\tilde{T}_{\tilde{\mathcal{R}}}) \neq 0$ for almost all T.

Finally consider a complex Laplacian L of the digraph $\mathcal{G}$ and a submatrix $L^{\mathcal{R}}$. Compared with T and $T^{\mathcal{R}}$, L and $L^{\mathcal{R}}$ (respectively) have more nonzero complex entries. According to the fact that a polynomial is either constantly zero or nonzero almost everywhere, we conclude that for almost all complex Laplacian L of $\mathcal{G}$, $\det(L^{\mathcal{R}}) \neq 0$.

(ii) Suppose that $\mathcal{G}$ contains a spanning 2-tree with a root set $\mathcal{R} := \{v_1, v_2\}$ (without loss of generality). Consider a complex Laplacian L of $\mathcal{G}$, and a submatrix $L^{\mathcal{R}}$ obtained from L by removing the two rows p_1, p_2 corresponding to the two roots v_1, v_2 and arbitrary two columns q_i, q_j. If $i = 1$ (similarly for $i = 2$), remove v_1 and all its incoming and outgoing edges, and denote the resulting subgraph $\mathcal{G}'$. Then $\mathcal{G}'$ contains a spanning tree (v_2 being a root), and it follows from (i) above that for almost all complex Laplacian L' of $\mathcal{G}'$, $\det((L')^{\{v_2\}}) \neq 0$. This implies $\det(L^{\mathcal{R}}) \neq 0$ for almost all complex Laplacian L of $\mathcal{G}$.

It remains to consider the case where $i, j \neq 1, 2$. For this, let $v_i \in \mathcal{V} \setminus \mathcal{R}$ and p_i ($i \in [3, n]$) be the ith row of L. Consider the following elementary row transformations:

$$L = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_i \\ \vdots \end{bmatrix} \Longrightarrow \tilde{L} := \begin{bmatrix} k_1 p_1 + \cdots + k_n p_n \\ p_2 \\ \vdots \\ p_i \\ \vdots \end{bmatrix}$$

where $k_1, \ldots, k_n$ are proper coefficients such that the three entries $\tilde{L}(1,1), \tilde{L}(1,2), \tilde{L}(1,i)$ on the first row of $\tilde{L}$ are nonzero. Such coefficients always exist because each of the two roots has at least one outgoing edge. Denote by $\tilde{\mathcal{T}}$ the digraph corresponding to $\tilde{T}$. We claim that $\tilde{\mathcal{T}}$ contains a spanning 2-tree with a root set $\tilde{\mathcal{R}} := \{v_2, v_i\}$. To see this, first note that v_1 is 2-reachable from $\tilde{\mathcal{R}}$ because $\tilde{L}(1,2), \tilde{L}(1,i)$ are nonzero and there are two edges $(v_2, v_1), (v_i, v_1)$. Now consider a node v_j $(j \neq 1, 2, i)$; there are three cases:

- Two disjoint paths from $\mathcal{R}$ to v_j do not go through v_i. Then v_j is 2-reachable from $\tilde{\mathcal{R}}$: $v_2 \to v_j$ and $v_i \to v_1 \to v_j$.
- The path from v_1 to v_j does not go through v_i, but $v_2 \to v_i \to v_j$. Then v_j is 2-reachable from $\tilde{\mathcal{R}}$: $v_2 \to v_1 \to v_j$ and $v_i \to v_j$.
- The path from v_2 to v_j does not go through v_i, but $v_1 \to v_i \to v_j$. Then v_j is 2-reachable from $\tilde{\mathcal{R}}$: $v_2 \to v_j$ and $v_i \to v_j$.

Note that it is not possible that both paths from $\mathcal{R}$ to v_j go through v_i in virtual of the definition of spanning 2-tree. Hence our claim is established.

Now remove node v_i and all its incoming and outgoing edges, and denote the resulting subgraph $\tilde{\mathcal{G}}'$. Then $\tilde{\mathcal{G}}'$ contains a spanning tree (v_2 being a root), and it follows from (i) above that for almost all complex Laplacian $\tilde{L}'$ of $\tilde{\mathcal{G}}'$, $\det((\tilde{L}')^{\{v_2\}}) \neq 0$. Since $L^{\mathcal{R}}$ may be obtained from $(\tilde{L}')^{\{v_2\}}$ via elementary row transformations (reordering the first row to the ith position and recovering p_i), we conclude that $\det(L^{\mathcal{R}}) = \det((\tilde{L}')^{\{v_2\}}) \neq 0$ for almost all complex Laplacian L of $\mathcal{G}$. The proof is now complete. □

With the above two lemmas, we provide the proof of Proposition 6.2.

Proof of Proposition 6.2: By Assumption 6.1, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning 2-tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$, where $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$ and the set of two roots $\mathcal{R} = \{v_1, v_2\}$ (renumbering if necessary). Consider a complex Laplacian T of $\mathcal{T}$ such that all principal minors of $T_{\mathcal{R}}$ are nonzero ($T_{\mathcal{R}}$ is the submatrix of T by deleting the two rows and columns corresponding to v_1, v_2). Such T always exists by Lemma 6.2(ii). For the rank of T, on one hand $\text{rank}(T) \geq n-2$ since $\det(T_{\mathcal{R}}) \neq 0$; on the other hand $\text{rank}(T) \leq n-2$ since the first two rows of T are zero row vectors. Hence $\text{rank}(T) = n-2$, and the kernel of T is two dimensional. One basis of this kernel is $\mathbf{1}$ since T is a complex Laplacian. Denote the other basis by η which is linearly independent of $\mathbf{1}$.

We claim that all the entries of η are distinct. To see this, suppose on the contrary that there are two entries η_i, η_j $(i, j \in [1, n])$ are equal. Scale η such that $\eta_i = \eta_j = 1$, and denote by $\tilde{\eta}$ the $n-2$ dimensional subvector of η with the entries other than η_i, η_j. Let $T^{\mathcal{R}}$ be the submatrix of T by deleting the two rows corresponding to v_1, v_2 and the two columns corresponding to v_i, v_j; while $\tilde{T}$ be the submatrix of T by deleting the two rows corresponding to v_1, v_2 and the $n-2$ columns corresponding to the nodes in $\mathcal{V} \setminus \{v_i, v_j\}$. Then it follows from $T\mathbf{1} = 0$ and $T\eta = 0$ that

$$\begin{aligned} T^{\mathcal{R}}\mathbf{1}_{n-2} + \tilde{T}\mathbf{1}_2 &= 0 \\ T^{\mathcal{R}}\tilde{\eta} + \tilde{T}\mathbf{1}_2 &= 0. \end{aligned}$$

Equating the left-hand sides of the above two equations yields

$$T^{\mathcal{R}}(\tilde{\eta} - \mathbf{1}_{n-2}) = 0.$$

Since $T^{\mathcal{R}}$ is of full rank by Lemma 6.3(ii), we derive $\tilde{\eta} = \mathbf{1}_{n-2}$. Therefore $\eta = \mathbf{1}$, which contradicts that η and $\mathbf{1}$ are linearly independent. Hence, all the entries of η are distinct after all.

Moreover, since each node $v_i \in \mathcal{V} \setminus \mathcal{R}$ has exactly two neighbors, each corresponding row of T has at most three nonzero entries. Thus equations $T\mathbf{1} = 0$ and $T\eta = 0$ yield

$$\begin{bmatrix} 1 & 1 & 1 \\ \eta_i & \eta_{i_1} & \eta_{i_2} \end{bmatrix} \begin{bmatrix} T_{ii} \\ T_{ii_1} \\ T_{ii_2} \end{bmatrix} = 0$$

where v_{i_1}, v_{i_2} are the two neighbors of v_i. More explicitly

$$\begin{aligned} T_{ii} + T_{ii_1} + T_{ii_2} &= 0 \\ \eta_i T_{ii} + \eta_{i_1} T_{ii_1} + \eta_{i_2} T_{ii_2} &= 0. \end{aligned}$$

Hence

$$\begin{bmatrix} T_{ii} \\ T_{ii_1} \\ T_{ii_2} \end{bmatrix} = c_i \begin{bmatrix} \eta_{i_2} - \eta_{i_1} \\ \eta_i - \eta_{i_2} \\ \eta_{i_1} - \eta_i \end{bmatrix}$$

for some nonzero complex number c_i. Since all the entries of η are distinct, each row of T corresponding to a non-root node has exactly three nonzero entries.

Now consider a generic configuration ξ and another complex Laplacian T' of $\mathcal{T}$ such that $T'\xi = 0$.

Since ξ is generic, all the entries of ξ are distinct. Hence T' has the same zero/nonzero pattern as T. Since all principal minors of $T_{\mathcal{R}}$ are nonzero, it follows from the fact that a polynomial is either constantly zero or nonzero almost everywhere that all principal minors of $T'_{\mathcal{R}}$ are also nonzero.

Finally, returning to the digraph $\mathcal{G}$ and let L be a complex Laplacian of $\mathcal{G}$ satisfying $L\xi = 0$. Compared with T', L has more nonzero complex entries. Again according to the fact that a polynomial is either constantly zero or nonzero almost everywhere, we conclude that all principal minors of $L_{\mathcal{R}}$ are nonzero. The proof is now complete. □

Finally we are ready to prove Theorem 6.1.

Proof of Theorem 6.1: Let Assumptions 6.1 and 6.2 hold. On one hand, it follows from Proposition 6.2 that for almost all complex Laplacian L of $\mathcal{G}$ satisfying $L\xi = 0$ (where ξ is generic), $\operatorname{rank}(L) \geq n-2$, i.e. $\dim(\ker L) \leq 2$. On the other hand, by using the distributed control in SFCA, we derive $\ker L \supseteq \mathcal{S}(\xi)$ as in (6.7), and thus $\dim(\ker L) \geq 2$. Therefore for almost all complex Laplacian L satisfying $L\xi = 0$, we have $\ker L = \mathcal{S}(\xi)$, which establishes the first condition in the similar formation control problem.

For the second condition, let $\mathcal{R} = \{v_1, v_2\}$ (renumbering if necessary) be the set of two roots and $L_{\mathcal{R}}$ the submatrix of L with the first two rows and columns corresponding to $\mathcal{R}$ removed. Then by Proposition 6.2, for almost all complex Laplacian L satisfying $L\xi = 0$, all principal minors of $L_{\mathcal{R}}$ are nonzero. It then follows from Lemma 6.1 that there exists an invertible diagonal matrix $E_{\mathcal{R}} = \operatorname{diag}(\epsilon_3, \ldots, \epsilon_n)$ such that all the eigenvalues of $-E_{\mathcal{R}}L_{\mathcal{R}}$ have negative real parts. Let

$$E' := \begin{bmatrix} 0 & 0 \\ 0 & E_{\mathcal{R}} \end{bmatrix}, \quad L = \begin{bmatrix} L_1 & L_2 \\ L_3 & L_{\mathcal{R}} \end{bmatrix}.$$

Then

$$-E'L = -\begin{bmatrix} 0 & 0 \\ E_{\mathcal{R}}L_3 & E_{\mathcal{R}}L_{\mathcal{R}} \end{bmatrix}.$$

Hence the spectrum (i.e. set of eigenvalues) of $-E'L$ is the union of the spectrum of $-E_{\mathcal{R}}L_{\mathcal{R}}$ and $\{0, 0\}$ (set of two zeros). Let ϵ_1, ϵ_2 have sufficiently small magnitudes (i.e. $|\epsilon_1|, |\epsilon_2|$ sufficiently small) and

$$E := \begin{bmatrix} \epsilon_1 & 0 & 0 \\ 0 & \epsilon_2 & 0 \\ 0 & 0 & E_{\mathcal{R}} \end{bmatrix}.$$

Then all the diagonal entries of E are nonzero, and E is invertible. Thus $\operatorname{rank}(EL) = \operatorname{rank}(L) = 2$ (i.e. $\ker EL = \ker L$), and there are two eigenvalues 0 of $-EL$. Moreover, since eigenvalues are continuous functions of matrix entries and $|\epsilon_1|, |\epsilon_2|$ are sufficiently small, the rest $n-2$ eigenvalues of $-EL$ still have negative real parts.

Write $-EL$ in Jordan canonical form as

$$-EL = VJV^{-1} = \begin{bmatrix} \mathbf{1} & \xi & y_3 & \cdots & y_n \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J' \end{bmatrix} \begin{bmatrix} z_1^\top \\ z_2^\top \\ z_3^\top \\ \vdots \\ z_n^\top \end{bmatrix}$$

where $y_i, z_i \in \mathbb{C}^n$ are respectively the (generalized) right and left eigenvectors of $-EL$, and $J' \in \mathbb{C}^{(n-2)\times(n-2)}$ is a block diagonal matrix consisting of the Jordan blocks corresponding to those eigenvalues with negative real parts. Hence the matrix exponential e^{-ELt} is

$$\begin{aligned} \mathrm{e}^{-ELt} &= \mathrm{e}^{VJV^{-1}t} = V\mathrm{e}^{Jt}V^{-1} \\ &= V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathrm{e}^{J't} \end{bmatrix} V^{-1} \\ &\to \mathbf{1}z_1^\top + \xi z_2^\top, \quad \text{as } t \to \infty. \end{aligned}$$

Therefore based on the SFCA in (6.11):

$$\begin{aligned} x(t) &= \mathrm{e}^{-ELt}x(0) \\ &\to \mathbf{1}z_1^\top x(0) + \xi z_2^\top x(0), \quad \text{as } t \to \infty. \end{aligned}$$

Let $\xi' := \mathbf{1}z_1^\top x(0) + \xi z_2^\top x(0)$. Then $\xi' \in \mathcal{S}(\xi)$, and therefore

$$\lim_{t\to\infty} x(t) \in \mathcal{S}(\xi)$$

i.e. the second condition in the similar formation control problem is established. This completes the proof. □

6.4 Simulation Examples

Example 6.4 *Let us consider again Example 6.3, where the (generic) target configuration is the regular hexagon* $\xi = [1\ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}}\ \mathrm{e}^{\pi\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}}\ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$*. We have designed a complex Laplacian* L *of the digraph modeling the interconnection of the six agents (copied below for convenience):*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{3} & -\sqrt{3}+\mathrm{j} & 0 & -\mathrm{j} & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2}+\frac{\sqrt{3}}{2}\mathrm{j} & -\frac{\sqrt{3}}{2}\mathrm{j} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}+\frac{\sqrt{3}}{2}\mathrm{j} & -\frac{3}{2}-\frac{\sqrt{3}}{2}\mathrm{j} & 1 \\ -\frac{3}{2}-\frac{\sqrt{3}}{2}\mathrm{j} & 1 & 0 & 0 & 0 & \frac{1}{2}+\frac{\sqrt{3}}{2}\mathrm{j} \end{bmatrix}.$$

While it is satisfied that $\ker L = \mathcal{S}(\xi)$*, one of the nonzero eigenvalues of* $-L$ *is unstable (i.e. with positive real part). Thus we need to design an invertible diagonal matrix* E *such that all the nonzero eigenvalues of* $-EL$ *are stable.*

Since the target configuration ξ *is generic and the digraph* $\mathcal{G}$ *contains a spanning* 2*-tree with the root set* $\mathcal{R} = \{1, 2\}$*, all the principal minors of the submatrix* $L_{\mathcal{R}}$ *(with the two rows and columns corresponding to* $\mathcal{R}$ *removed) are nonzero. Therefore by Lemma 6.1, there exists an invertible diagonal matrix* $E_{\mathcal{R}}$ *such that all the eigenvalues of* $-E_{\mathcal{R}}L_{\mathcal{R}}$ *are stable. For computing such* $E_{\mathcal{R}}$*, we apply Algorithm 6.1 and obtain*

$$E_{\mathcal{R}} = \mathrm{diag}(0.433 + 0.25\mathrm{j}, -0.1\mathrm{j}, 0.0866 - 0.05\mathrm{j}, -0.05 + 0.0866\mathrm{j}).$$

It is verified that all the eigenvalues of $-E_{\mathcal{R}}L_{\mathcal{R}}$ *are stable:*

$$-0.0456, -0.1, -0.221, -0.9933.$$

Then an invertible diagonal matrix E *such that all the nonzero eigenvalues of* $-EL$ *are stable is:*

$$E = \mathrm{diag}(1, 1, 0.433 + 0.25\mathrm{j}, -0.1\mathrm{j}, 0.0866 - 0.05\mathrm{j}, -0.05 + 0.0866\mathrm{j}).$$

Indeed, the eigenvalues of $-EL$ *are:*

$$0, 0, -0.0456, -0.1, -0.221, -0.9933.$$

With a random initial condition $x(0) \in \mathbb{C}^6$ (whose entries represent six random positions of the agents in a 2D space), a simulation of the SFCA (i.e. $\dot{x} = (-EL)x$) yields the trajectories displayed in Fig. 6.4. It is observed that a similar formation of regular hexagon is formed. In the figure, × denotes the initial positions of the agents, while ∘ the final positions. Observe that the two root agents (left middle and left top) have stayed put as their initial and final positions coincide; this is because they have no neighbors and thus have never updated their positions.

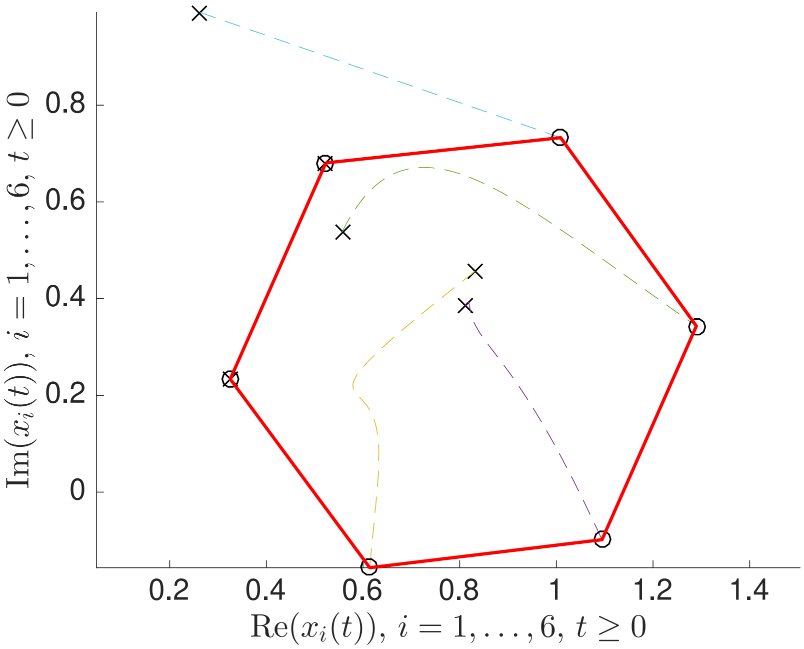


Figure 6.4: Six agents converging to a similar formation of regular hexagon (×: initial position; ∘: final position)

Example 6.5 *Consider a network of 15 agents as displayed in Fig. 6.5. This digraph contains a spanning 2-tree, and any two of the set $\{6, 7, 9, 10\}$ of agents are two roots. Different from the digraph in Fig. 6.2 where the two roots have no neighbors, every node including the roots has two or three neighbors.*

First, we consider a regular polygon to be the target configuration:

$$\xi = [1\ \mathrm{e}^{\frac{2\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{6\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{8\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{10\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{12\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{14\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{16\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{18\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{20\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{22\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{24\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{26\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{28\pi}{15}\mathrm{j}}\ ]^\top.$$

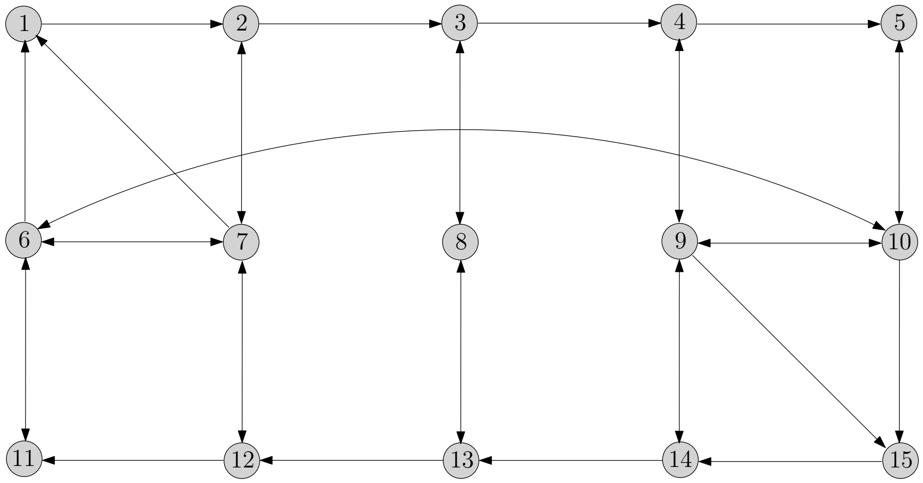


Figure 6.5: Fifteen networked agents

Thus ξ is generic. We then design a complex Laplacian L of the digraph in Fig. 6.5 such that rank$(L) = 13$, and apply Algorithm 6.1 to compute an invertible diagonal matrix E such that all the eigenvalues of $-EL$ are stable. With a random initial condition $x(0) \in \mathbb{C}^{15}$, a simulation of the SFCA (i.e. $\dot{x} = (-EL)x$) yields the trajectories displayed in Fig. 6.6. Observe that a regular polygon similar to ξ is formed. Also observe that no agent stays put, as everyone has neighbors and thus updates its state correspondingly.

Second, we consider a triangle shape to be the target configuration:

$$\xi = [4\mathrm{j}\ \ -1+3\mathrm{j}\ \ 1+3\mathrm{j}\ \ -2+2\mathrm{j}\ \ 2\mathrm{j}\ \ 2+2\mathrm{j}\ \ -3+\mathrm{j}\ \ -1+\mathrm{j}\ \ 1+\mathrm{j}\ \ 3+\mathrm{j}\ \ -4\ \ -2\ \ 0\ \ 2\ \ 4]^\top.$$

Note that this ξ is not *generic, because there are multiple cases of three points on the same line: e.g. the last three entries $0, 2, 4$ of ξ.*

For this example, nevertheless, a complex Laplacian L of the digraph in Fig. 6.5 may still be designed such that rank$(L) = 13$, and an invertible diagonal matrix E is obtained by Algorithm 6.1 such that all the nonzero eigenvalues of $-EL$ are stable. With a random initial condition $x(0) \in \mathbb{C}^{15}$, a simulation of the SFCA (i.e. $\dot{x} = (-EL)x$) yields the trajectories displayed in Fig. 6.7. Observe that a triangle similar to ξ is formed, and all agents have moved in the transient (before they converge to a similar formation of ξ in the steady state).

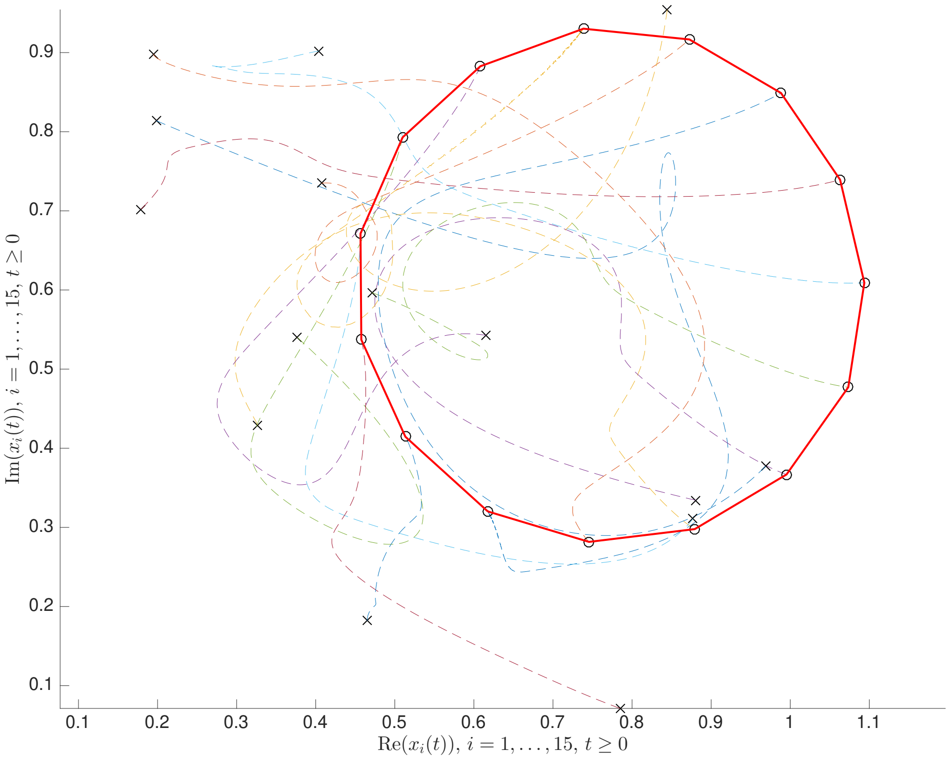


Figure 6.6: Fifteen agents converging to a similar formation of regular polygon (×: initial position; ○: final position)

6.5 Notes and References

The concept of complex Laplacian and similar formation control algorithm (SFCA) are originated in the following series of work:

- Z. Lin, W. Ding, G. Yan, C. Yu, A. Giua, Leader-follower formation via complex Laplacian, Automatica, vol.49, pp.1900–1906, 2013
- Z. Lin, L. Wang, Z. Han, M. Fu, Distributed formation control of multi-agent systems using complex laplacian, IEEE Transactions on Automatic Control, vol.59, pp.1765–1777, 2014
- Z. Lin, L. Wang, Z. Han, M. Fu, A graph laplacian approach to coordinate-free formation stabilization for directed networks, IEEE Transactions on Automatic Control, vol.61, pp.1269–1280, 2016

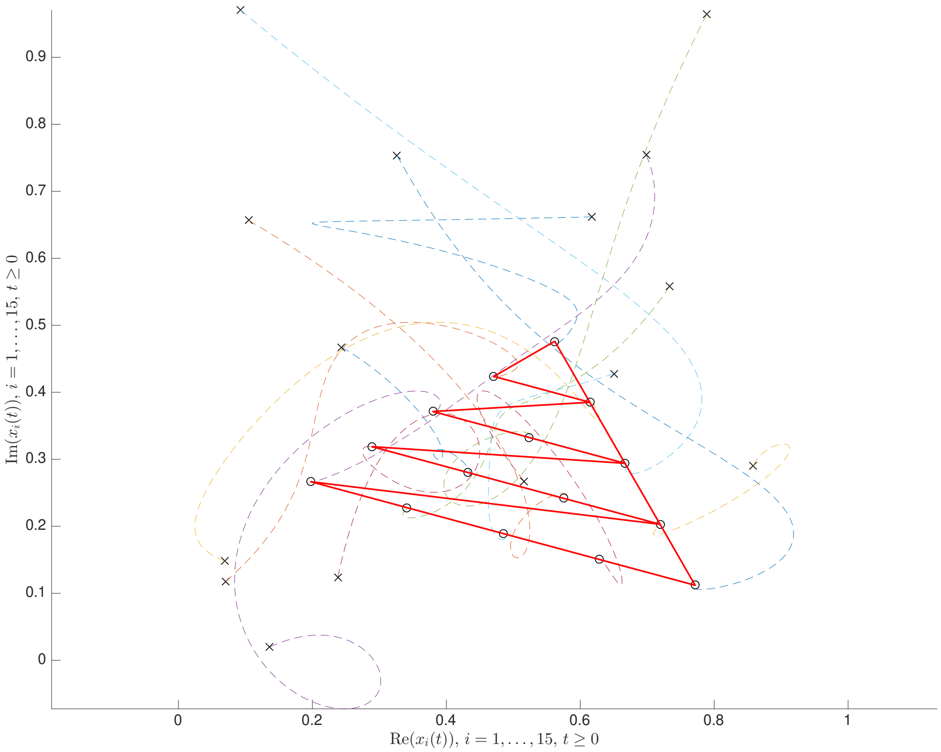


Figure 6.7: Fifteen agents converging to a similar formation of triangle (×: initial position; ○: final position)

Stabilization by diagonal matrices (Lemma 6.1) are studied in

- C.S. Ballantine, Stabilization by a diagonal matrix, Proceedings of the American Mathematical Society, vol.25, pp.728–734, 1970
- S. Friedland, On inverse multiplicative eigenvalue problems for matrices, Linear Algebra and Its Applications, vol.12, pp.127–137, 1975

Chapter 7

Localization in Two-Dimensional Space

In this chapter, we introduce a distributed localization problem of multi-agent systems in two-dimensional (2D) space. This problem has found numerous important applications in (wireless) sensor networks, including environment information collection, wildlife monitoring, target tracking, and intrusion detection. In these applications, it is essential that the individual sensor nodes know their positions in a common (global) reference frame. For example, it would be ideal to have a GPS onboard each sensor. In practical sensor networks, however, there are typically a large number of sensor nodes each with limited hardware/software capacities. Thus it is costly and implementationally difficult to install a device like GPS on every sensor, not to mention that there are situations where GPS is at best very inaccurate and at worst denied.

Therefore it is desirable to have a distributed scheme to determine the global positions of individual sensor nodes based on low-cost, easily implementable onboard devices. A typical such scheme is to compose a sensor network with a minority of *anchor* nodes that do know their positions in the global reference frame (e.g. using a GPS), and the rest majority of *free* nodes that need to determine their global positions based on their local frames and locally sensed information (e.g. distances and bearing angles with respect to neighboring nodes). Those anchor nodes play the role of leaders or landmarks, while the free nodes are followers. We adopt this distributed scheme, and focus in this chapter on solving a localization problem in 2D, while 3D localization is covered in Chapter 9.

To solve the 2D distributed localization problem, we present an approach based on complex Laplacian matrices. Modeling the interacting sensor nodes by digraphs, we show that a necessary graphical condition to achieve 2D localization is that the digraph contains a *spanning* 2-*tree* whose two roots are anchor nodes. This condition is similar to that for achieving 2D similar formations in the preceding chapter. However, the two anchor nodes (i.e. two roots) who already know their global positions should not, and will not, change their positions; hence they do not have, nor do they need, any neighbors (i.e. incoming edges). In this way, the exact global positions of the free nodes may be determined; that is without the flexibility of translation, rotation, and scaling as in the similar formation problem). Under the above graphical condition, we present a distributed algorithm for the free nodes to achieve localization in 2D.

7.1 Problem Statement

Consider a network of n (> 1) agents that are stationary in a plane (i.e. their two-dimensional positions are fixed), and a global reference frame Σ which is unknown to the agents. The agents labeled $1, 2$ (renumbering if necessary) are the *anchor agents*, whose positions $\xi_1, \xi_2 \in \mathbb{C}$ in Σ are known. Here $\mathrm{Re}(\xi_i)$ and $\mathrm{Im}(\xi_i)$ are the positions of agents $i \in [1, 2]$ on the real and imaginary axes, respectively. The rest agents labeled $3, \ldots, n$ are the *free agents*, whose positions $\xi_3, \ldots, \xi_n \in \mathbb{C}$ in Σ are unknown and need to be determined by these individual free agents. Let

$$\xi_a := \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \in \mathbb{C}^2, \quad \xi_f := \begin{bmatrix} \xi_3 \\ \vdots \\ \xi_n \end{bmatrix} \in \mathbb{C}^{n-2}$$

be the aggregated positions of the anchor and free agents, respectively. Write

$$\xi := \begin{bmatrix} \xi_a \\ \xi_f \end{bmatrix} \in \mathbb{C}^n$$

and call ξ the *configuration* of the agents.

To determine its own position, each free agent i $(\in [3, n])$ is equipped with a *state* variable $x_i(k) \in \mathbb{C}$, which denotes the *estimate* of agent i's position ξ_i under the global frame Σ. The time $k \geq 0$ is a nonnegative integer and denotes the *discrete* time. Let

$$x_f(k) := \begin{bmatrix} x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} \in \mathbb{C}^{n-2}$$

be the aggregated state of the free agents at time k. It is desired that

$$x_f(k) \to \xi_f \text{ as } k \to \infty.$$

For convenience, also let $x_a(k) := [x_1(k) \; x_2(k)]^\top \in \mathbb{C}^2$ be the aggregated state vector of the two anchor agents, such that $x_a(k) = \xi_a$ for all $k \geq 0$ (i.e. the anchor agents know their positions in the global frame Σ from the initial time $k = 0$ and never update their estimates). Write

$x(k) := [x_a(k)^\top \; x_f(k)^\top]^\top \in \mathbb{C}^n$. Hence the purpose of localization is to achieve

$$\lim_{k \to \infty} x(k) = \xi.$$

We model the interconnection structure of the networked agents by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Each *node* in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each directed *edge* (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent i can obtain the relative state information from agent j. The *neighbor set* of agent i is $\mathcal{N}_i := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. For the two anchor nodes (numbered 1 and 2), since they do not update their states, even if they had neighbors, the corresponding incoming edges would be associated with weight 0. This is equivalent to considering that the anchor nodes do not have neighbors. For this reason, henceforth in this chapter we consider that $\mathcal{N}_i = \emptyset$ $(i = 1, 2)$.

Moreover, consider that digraph $\mathcal{G}$ is weighted: each edge $(j, i) \in \mathcal{V}$ is associated with a complex weight $a_{ij} \in \mathbb{C}$. Hence the adjacency matrix $A = (a_{ij})$, degree matrix $D = \mathrm{diag}(A\mathbf{1})$, and Laplacian matrix $L = D - A$ are all complex. Since $\mathcal{N}_i = \emptyset$ for the anchor nodes $i = 1, 2$, the Laplacian matrix L has the following structure:

$$L = \begin{bmatrix} L_{aa} & L_{af} \\ L_{fa} & L_{ff} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix}. \tag{7.1}$$

Here $L_{fa} \in \mathbb{C}^{(n-2) \times 2}$ and $L_{ff} \in \mathbb{C}^{(n-2) \times (n-2)}$.

To achieve localization, consider the distributed control

$$u_i(k) = \sum_{j \in \mathcal{N}_i} w_{ij}(x_j(k) - x_i(k)), \quad i \in [1, n]. \tag{7.2}$$

Here the control gain w_{ij} satisfies

$$\text{(i)} \sum_{j \in \mathcal{N}_i} w_{ij}(\xi_j - \xi_i) = 0 \tag{7.3}$$

$$\text{(ii)} \; w_{ij} = \epsilon_i a_{ij}, \quad \epsilon_i \in \mathbb{C}, \epsilon_i \neq 0. \tag{7.4}$$

This control (7.2) is in the same form as that for similar formation control: the gains w_{ij} are not simply the edge weights a_{ij}, but are complex multiples of a_{ij} (7.4) and satisfy linear constraints with respect to the target configuration ξ (7.3).

Substituting (7.4) into (7.3) and removing the common multiple ϵ_i yield

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0. \tag{7.5}$$

This in matrix form is $L\xi = 0$. In view of (7.1) we have

$$\begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \xi_f \end{bmatrix} = 0$$

Hence the following equation ensues:

$$L_{ff}\xi_f = -L_{fa}\xi_a \tag{7.6}$$

which relates the configuration of the free agents to that of the anchor agents through appropriate multiplications of submatrices of the complex Laplacian.

Two-Dimensional Localization Problem:

Consider a network of agents (stationary on a plane) interconnected through a digraph and a configuration $\xi := [\xi_a^\top \ \xi_f^\top]^\top \in \mathbb{C}^n$, which represents the fixed positions of the agents under the global reference frame Σ. Here $\xi_a \in \mathbb{C}^2$ is known but $\xi_f \in \mathbb{C}^{n-2}$ is unknown. Design a distributed algorithm using the control in (7.2) such that

$$\begin{aligned} &\text{(i) } \operatorname{rank}(L) = n - 2 \\ &\text{(ii) } (\forall x_f(0) \in \mathbb{C}^{n-2}) \lim_{k\to\infty} x_f(k) = \xi_f. \end{aligned}$$

The first requirement (i) implies $\operatorname{rank}(L_{ff}) = n - 2$; namely L_{ff} is invertible. Then it follows from (7.6) that $\xi_f = -L_{ff}^{-1} L_{fa}\xi_a$. Hence the second requirement (ii) becomes:

$$(\forall x_f(0) \in \mathbb{C}^{n-2}) \lim_{k\to\infty} x_f(k) = -L_{ff}^{-1} L_{fa}\xi_a.$$

Example 7.1 *We provide an example to illustrate the localization problem in 2D. As displayed in Fig. 7.1, six agents are interconnected through a digraph; agents 1 and 2 are anchor agents while the rest four are free agents. The neighbor sets of the agents are* $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$, $\mathcal{N}_3 = \{2,5\}$, $\mathcal{N}_4 = \{1,3\}$, $\mathcal{N}_5 = \{4,6\}$, *and* $\mathcal{N}_6 = \{1,2\}$.
Let the configuration of the agents be $\xi = [1 \ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}} \ \mathrm{e}^{\pi\mathrm{j}} \ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$, *i.e. a regular hexagon. The position vector of the anchor agents* $\xi_a = [1 \ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}]^\top$ *is known, and that of the free nodes* $\xi_f = [\mathrm{e}^{\frac{2\pi}{3}\mathrm{j}} \ \mathrm{e}^{\pi\mathrm{j}} \ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$ *is unknown and needs to be determined.*
The localization problem is to design a distributed algorithm using the control in (7.2) such that the rank of the complex Laplacian L *is* $n-2$, *and moreover the free agents' state vector asymptotically converges to* ξ_f.

A necessary graphical condition for solving the two-dimensional localization problem is given

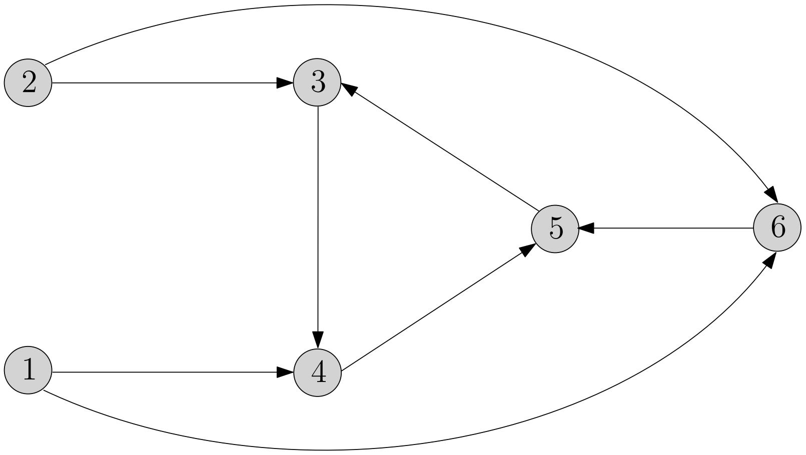


Figure 7.1: Illustrating example of six agents

below.

Proposition 7.1 *Suppose that there exists a distributed control in (7.2) that solves the two-dimensional localization problem. Then the digraph contains a spanning* 2*-tree whose two roots are the two anchor agents.*

Proof. Suppose that there exists a distributed control in (7.2) that solves the two-dimensional localization problem, but that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ does *not* contain a spanning 2-tree whose two roots are the two anchor agents. We will derive a contradiction that $\text{rank}(L) < n - 2$, thereby proving that after all $\mathcal{G}$ must contain a spanning 2-tree whose two roots are the two anchor agents.

There are two cases that need to be considered separately. First, the digraph contains a spanning 2-tree but at least one of the two roots is a free agent. In this case, the subdigraph of free agents contains either a spanning tree or a spanning 2-tree. Hence $\text{rank}(L_{ff}) < n - 2$. Since the anchor agents do not have neighbors, $\text{rank}(L) < n - 2$.

The second case is that the digraph does not contain a spanning 2-tree. Then it follows similarly to the proof of Proposition 6.1 that $\text{rank}(L) < n - 2$.

Therefore in both cases above, a contradiction is derived to the solvability of the two-dimensional localization problem. The proof is now complete. □

Owing to Proposition 7.1, we shall henceforth assume the following graphical condition.

Assumption 7.1 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning* 2*-tree whose two roots are the two anchor agents.*

Even if Assumption 7.1 holds, not every configuration ξ may be determined by a distributed control in (7.2). Similar to Example 6.2, if ξ is not generic, it is possible that $\text{rank}(L) < n - 2$

for all complex Laplacian matrices L satisfying $L\xi = 0$. This means that the two-dimensional localization problem is not solvable. For this reason, and also the fact that the set of all non-generic configurations has Lebesgue measure zero after all, we assume that the configuration ξ is generic.

Assumption 7.2 *The configuration* $\xi := [\xi_a^\top \ \xi_f^\top]^\top \in \mathbb{C}^n$ *is generic.*

7.2 Distributed Algorithm

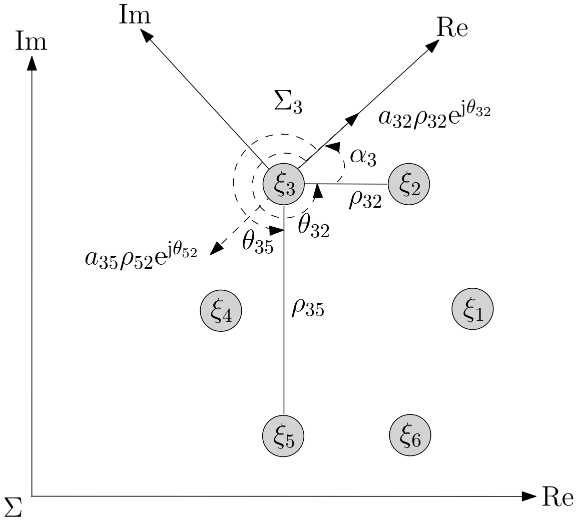


Figure 7.2: Illustration of design of complex weights

Example 7.2 *Consider again Example 7.1, where the configuration is the regular hexagon* $\xi = [1 \ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}} \ \mathrm{e}^{\pi\mathrm{j}} \ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$. *This* ξ *is generic.*

The anchor agents' configuration $\xi_a = [1 \ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}]^\top$ *is known, and the free agents' configuration* $\xi_f = [\mathrm{e}^{\frac{2\pi}{3}\mathrm{j}} \ \mathrm{e}^{\pi\mathrm{j}} \ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$ *is to be determined. To this end, we consider using the simplest form of distributed control (7.2) by setting all* $\epsilon_i = 1$*:*

$$x_i(k+1) = x_i(k) + \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [1,6] \tag{7.7}$$

where $a_{ij} \in \mathbb{C}$ *are complex weights to be designed to satisfy (7.5):*

$$\sum_{j\in\mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0, \quad i \in [1,6].$$

In the following we illustrate how the complex weights may be designed locally to satisfy the above linear constraints. Each free agent $i \in [3,6]$ has a local reference frame Σ_i, whose origin is the (stationary) position of agent i. The orientation of Σ_i is fixed, but the offset angle θ_i with respect to the global reference frame Σ is unknown. For each neighbor (free or anchor) $j \in \mathcal{N}_i$, we assume that agent i can sense the relative position by measuring the relative distance and relative bearing angle in Σ_i. That is, if agent j is a neighbor of agent i, then the distance ρ_{ij} between j and i, as well as the bearing angle θ_{ij} of j in Σ_i are measured by i. Thus the relative position in Σ_i is

$$y_{ij} := \rho_{ij}\mathrm{e}^{\mathrm{j}\theta_{ij}}. \tag{7.8}$$

Note that $y_{ij}\mathrm{e}^{\mathrm{j}\theta_i} = \xi_j - \xi_i$; since θ_i is unknown, even though the relative position y_{ij} in Σ_i is known, $\xi_j - \xi_i$ in Σ is unknown. Substituting $\xi_j - \xi_i = y_{ij}\mathrm{e}^{\mathrm{j}\theta_i}$ into (7.5) and removing the common factor $\mathrm{e}^{\mathrm{j}\theta_i}$, we derive

$$\sum_{j \in \mathcal{N}_i} a_{ij} y_{ij} = 0. \tag{7.9}$$

Hence the weights a_{ij} may be designed based on the relative position y_{ij} in (7.8) under the local reference frame Σ_i.

For example, Fig. 7.2 provides an illustrative example. For agent 3, it has two neighbors $2, 5$. Thus we must find weights a_{32}, a_{52} such that $a_{32}y_{32} + a_{35}y_{35} = 0$. In the local reference frame Σ_3, $y_{32} = \rho_{32}\mathrm{e}^{\mathrm{j}\theta_{32}}$ and $y_{35} = \rho_{35}\mathrm{e}^{\mathrm{j}\theta_{35}}$. Thus we want to find a_{32}, a_{35} such that

$$a_{32}\rho_{32}\mathrm{e}^{\mathrm{j}\theta_{32}} + a_{35}\rho_{35}\mathrm{e}^{\mathrm{j}\theta_{35}} = 0.$$

There are infinitely many choices; a simple one is $a_{32} = \frac{\mathrm{e}^{-\mathrm{j}\theta_{32}}}{\rho_{32}}$ and $a_{35} = -\frac{\mathrm{e}^{-\mathrm{j}\theta_{35}}}{\rho_{35}}$. Concretely, $\rho_{32} = 1$, $\rho_{35} = \sqrt{3}$, and let $\theta_{32} = \frac{7\pi}{4}$, $\theta_{35} = \frac{5\pi}{4}$; then $a_{32} = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}\mathrm{j}$, $a_{35} = \frac{\sqrt{6}}{6} - \frac{\sqrt{6}}{6}\mathrm{j}$. Similarly we design other complex weights to satisfy (7.9), and write (7.7) in vector form: $x(k+1) = (I - L)x(k)$ where

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\mathrm{j} & \frac{3\sqrt{2}+\sqrt{6}}{6} + \frac{3\sqrt{2}-\sqrt{6}}{6}\mathrm{j} & 0 & -\frac{\sqrt{6}}{6} + \frac{\sqrt{6}}{6}\mathrm{j} & 0 \\ -\frac{\sqrt{3}}{4} - \frac{1}{4}\mathrm{j} & 0 & \frac{\sqrt{3}}{2} - \frac{1}{2}\mathrm{j} & -\frac{\sqrt{3}}{4} + \frac{3}{4}\mathrm{j} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{j} & -\sqrt{3}\mathrm{j} & \frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{j} \\ -\frac{\sqrt{3}}{2} + \frac{1}{2}\mathrm{j} & \frac{\sqrt{3}}{6} - \frac{1}{2}\mathrm{j} & 0 & 0 & 0 & \frac{\sqrt{3}}{3} \end{bmatrix}.$$

It is verified that the complex Laplacian matrix L has zero row sums and satisfies $L\xi = 0$. Moreover, partition the matrix L according to anchor agents and free agents:

$$L = \begin{bmatrix} L_{aa} & L_{af} \\ L_{fa} & L_{ff} \end{bmatrix}.$$

Thus $L_{aa} = L_{af} = 0$; $L_{fa} \in \mathbb{C}^{4\times 2}$ and $L_{ff} \in \mathbb{C}^{4\times 4}$. It is checked that rank$(L_{ff}) = 4$, and thus L_{ff} is invertible. Therefore the first condition of the two-dimensional localization problem is satisfied.

It is left to verify the second condition that the state vector of the free agents $x_f(k)$ converges to $-L_{ff}^{-1}L_{fa}\xi_a$ (when $x_a(k) = \xi_a$ for all $k \geq 0$). Fix $\xi_a \in \mathbb{C}^2$. First note that

$$\bar{x} = \begin{bmatrix} \bar{x}_a \\ \bar{x}_f \end{bmatrix} = \begin{bmatrix} \xi_a \\ -L_{ff}^{-1}L_{fa}\xi_a \end{bmatrix}$$

is the unique fixed point of (7.7). To see this, substituting $\bar{x}$ into (7.7) yields $\bar{x}$, which means that $\bar{x}$ is a fixed point of (7.7). Moreover, let

$$\bar{x}' = \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}$$

be another fixed point of (7.7), namely

$$\begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix} \right) \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \begin{bmatrix} I & 0 \\ -L_{fa} & I - L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}.$$

From the above we derive

$$\bar{x}'_f = -L_{ff}^{-1}L_{fa}\xi_a = \bar{x}_f.$$

This shows that $\bar{x}$ is the unique fixed point of (7.7), which in turn implies that starting from an arbitrary initial condition $x(0) = [\xi_a^\top \ x_f^\top(0)]^\top \in \mathbb{C}^n$, $x_f(k)$ converges to $-L_{ff}^{-1}L_{fa}\xi_a$ if and only if all the eigenvalues of $I - L_{ff}$ lie inside the unit circle.

Unfortunately, the eigenvalues of matrix $I - L_{ff}$ are

$$-0.5774, 0.3041 - 0.6475\mathrm{j}, -0.9368 - 0.3062\mathrm{j}, -0.0497 + 1.637\mathrm{j}.$$

The last eigenvalue lies outside of the unit circle. Hence (7.7) is unstable and $x_f(k)$ diverges.

To stabilize $x_f(k)$ to the desired fixed point $-L_{ff}^{-1}L_{fa}\xi_a$ (to satisfy the second requirement of the two-dimensional localization problem), the unstable eigenvalues of $I - L_{ff}$ must be moved inside the unit circle. This shows that simply setting all $\epsilon_i = 1$ in (7.2) does not work in general. In fact, ϵ_i need to be properly chosen in order to stabilize $I - L_{ff}$.

In the following we describe a distributed algorithm using (7.2) in vector form, and will analyze its stability in relation to the values of ϵ_i in the next section.

Two-Dimensional Localization Algorithm (TDLA):

Each anchor agent $i \in [1, 2]$ has a state variable $x_i(k) \in \mathbb{C}$ whose initial value is set to be $x_i(0) = \xi_i$ (which is known). Each free agent $i \in [3, \ldots, n]$ also has a state variable $x_i(k) \in \mathbb{C}$ whose initial value is an arbitrary complex number. Offline, each free agent i computes weights $a_{ij} \in \mathbb{C}$ based on the measured relative positions $y_{ij} = \rho_{ij}\mathrm{e}^{\theta_{ij}}$ in (7.8) by solving

$$\sum_{j \in \mathcal{N}_i} a_{ij} y_{ij} = 0.$$

Then online, at each time $k \geq 0$, while each anchor agent stays put, i.e.

$$x_i(k+1) = x_i(k), \quad i \in [1, 2]$$

each free agent i updates its $x_i(k)$ using the following local update protocol:

$$x_i(k+1) = x_i(k) + \epsilon_i \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [3, n] \tag{7.10}$$

where $\epsilon_i \in \mathbb{C} \setminus \{0\}$ is a (nonzero) complex control gain.

Let $x := [x_1 \cdots x_n]^\top$ be the aggregated state of the networked agents, and

$$E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$$

the (invertible diagonal) control gain matrix. Then the n equations (7.10) become

$$x(k+1) = x(k) - ELx(k) = (I - EL)x(k). \tag{7.11}$$

7.3 Convergence Result

The following is the main result of this section.

Theorem 7.1 *Suppose that Assumptions 7.1 and 7.2 hold. There exists an invertible diagonal control gain matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ *such that the TDLA solves the two-dimensional localization problem.*

To prove Theorem 7.1, we will analyze the eigenvalues of the matrix $I - EL$ in (7.11). For this, the following fact is useful (which is the discrete counterpart of Lemma 6.1).

Lemma 7.1 *Consider an arbitrary square complex matrix* $M \in \mathbb{C}^{n\times n}$*. If all the principal minors of* M *are nonzero, then there exists an invertible diagonal matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{C}^{n\times n}$ *such that all the eigenvalues of* $I - EM$ *lie inside the unit circle.*

Proof: The proof is based on induction on n. For the base case $n = 1$, $M = m_{11}$ is a nonzero scalar (as the principal minor of M is nonzero). Write $m_{11} = \rho_1 \mathrm{e}^{\mathrm{j}\theta_1}$, and let $\epsilon_1 := \gamma_1 \mathrm{e}^{\mathrm{j}\phi_1}$ where $\gamma_1 \in (0, \frac{1}{\rho_1})$ and $\phi_1 = -\theta_1$. Then $EM = \epsilon_1 m_{11} = \rho_1\gamma_1 \in (0, 1)$. Hence $1 - EM \in (0, 1)$ which lies inside the unit circle.

For the induction step, suppose that the conclusion holds for $M \in \mathbb{C}^{(n-1)\times(n-1)}$. Now consider $M \in \mathbb{C}^{n\times n}$, with all of its principal minors nonzero. Let M_1 be the submatrix of M with the last row and last column removed. Then all the principal minors of M_1 are nonzero, and by the hypothesis there exists an invertible diagonal matrix $E_1 = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_{n-1})$ such that all the eigenvalues $1 - \lambda_1, \ldots, 1 - \lambda_{n-1}$ of $I - E_1M_1$ lie inside the unit circle. Now write

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix}$$

where m_{nn} is a nonzero scalar (since all the principal minors of M are nonzero). Also let

$$E = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix}$$

for some complex ϵ_n. Thus

$$I - EM = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix} = \begin{bmatrix} I - E_1M_1 & -E_1M_2 \\ -\epsilon_n M_3 & 1 - \epsilon_n m_{nn} \end{bmatrix}$$

If $\epsilon_n = 0$, then

$$I - EM = \begin{bmatrix} I - E_1M_1 & -E_1M_2 \\ 0 & 1 \end{bmatrix}$$

which means that all the eigenvalues of $I - EM$ lie inside the unit circle except for a simple eigenvalue 1. Since eigenvalues are continuous functions of matrix entries, for $\epsilon_n := \gamma_n \mathrm{e}^{\mathrm{j}\phi_n}$ with sufficiently small $\gamma_n > 0$, $I - EM$ still has $n-1$ eigenvalues $1-\lambda'_1, \dots, 1-\lambda'_{n-1}$ which are inside the unit circle.

Now we consider the last eigenvalue $1-\lambda'_n$. In Lemma 6.1 it is proved that ϵ_n may be chosen such that the magnitude of λ'_n is sufficiently small and its angle lie in $[-\bar{\theta}, \bar{\theta}]$ for an arbitrary $\bar{\theta} \in [0, \frac{\pi}{2})$. Hence for a small enough $\bar{\theta}$, the last eigenvalue $1 - \lambda'_n$ lies within the unit circle. This proves the induction step, and thereby completes the proof. □

The above proof suggests an algorithm (Algorithm 7.1 below) to compute an invertible diagonal matrix $E = \mathrm{diag}(\epsilon_1, \dots, \epsilon_n)$ such that all the eigenvalues $I - EM$ lie inside the unit circle. Compared with Algorithm 6.1, the only difference is adding scaling terms in lines 2 and 7 so as to render the resulting eigenvalues into the unit circle. This effect can also be achieved by choosing small enough δ_i $(i \in [1, n])$ in line 1. By the proof of Lemma 7.1, one can always choose appropriate (small) $\delta_1, \dots, \delta_n$ in line 1 so that Algorithm 7.1 outputs an invertible diagonal matrix E which ensures that all the eigenvalues $I - EM$ inside the unit circle.

Algorithm 7.1 Diagonal Stabilization Algorithm (case of complex matrix, inside unit circle)

```
Input: square complex matrix $M \in \mathbb{C}^{n \times n}$ with nonzero principal minors
Output: invertible diagonal matrix $E \in \mathbb{C}^{n \times n}$
 1: set $\delta_1, \dots, \delta_n$ to be small positive real numbers
 2: $\epsilon_1 = \delta_1 \frac{1}{|\det(M(1,1))|} \mathrm{e}^{-\mathrm{j}\angle \det(M(1,1))}$
 3: $E_1 = \mathrm{diag}(\epsilon_1)$
 4: $\{\lambda_1\}$ = spectrum of $E_1 M(1,1)$
 5: for $i = 2, \dots, n$ do
 6:     $\Lambda = \lambda_1 \cdots \lambda_{i-1}$
 7:     $\epsilon_i = \delta_i \frac{1}{\left|\frac{\det(E_{i-1})\det(M(1:i,1:i))}{\Lambda}\right|} \mathrm{e}^{-\mathrm{j}\angle \frac{\det(E_{i-1})\det(M(1:i,1:i))}{\Lambda}}$
 8:     $E_i = \mathrm{diag}(\epsilon_1, \dots, \epsilon_i)$
 9:     $\{\lambda_1, \dots, \lambda_i\}$ = spectrum of $E_i M(1:i, 1:i)$
10: end for
11: $E = \mathrm{diag}(\epsilon_1, \dots, \epsilon_n)$
```

Lemma 7.1 provides a sufficient condition under which the eigenvalues of a complex matrix may be moved inside the unit circle using an invertible diagonal complex matrix. It then follows from Proposition 6.2 (recalled below for convenience) that under Assumptions 7.1 and 7.2 (Assumption 7.1 implies Assumption 6.1 and Assumption 7.2 is the same as Assumption 6.1), the sufficient condition holds for the submatrix L_{ff} of the complex Laplacian L. Hence there exists an invertible diagonal matrix $E_f = \mathrm{diag}(\epsilon_3, \dots, \epsilon_n)$ such that all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle.

Proposition 6.2 *Suppose that Assumptions 7.1 and 7.2 hold. Let $\mathcal{R}$ be the set of two roots and $L_{\mathcal{R}}$ the submatrix of complex Laplacian L by removing the two rows and two columns corresponding to $\mathcal{R}$. Then for almost all complex Laplacian L satisfying $L\xi = 0$, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

With the above preparation, we are ready to prove Theorem 7.1.

Proof of Theorem 7.1: Let Assumptions 7.1 and 7.2 hold. On one hand, it follows from Proposition 6.2 that for almost all complex Laplacian L of $\mathcal{G}$ satisfying $L\xi = 0$ (where ξ is generic), $\mathrm{rank}(L) \geq n-2$. On the other hand, since the first two rows of L corresponding to the anchor agents are zero, we have $\mathrm{rank}(L) \leq n-2$. Therefore for almost all complex Laplacian L satisfying $L\xi = 0$, we have $\mathrm{rank}(L) = n-2$, which establishes the first condition in the two-dimensional localization problem.

For the second condition, first note again from Proposition 6.2 that for almost all complex Laplacian L satisfying $L\xi = 0$, all principal minors of L_{ff} are nonzero. It then follows from Lemma 7.1 that there exists an invertible diagonal matrix $E_f = \mathrm{diag}(\epsilon_3, \ldots, \epsilon_n)$ such that all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle. Let

$$E_a := \begin{bmatrix} \epsilon_1 & 0 \\ 0 & \epsilon_2 \end{bmatrix}, \quad E := \begin{bmatrix} E_a & 0 \\ 0 & E_f \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix}.$$

Here $\epsilon_1, \epsilon_2 \neq 0$. Thus E is invertible and

$$I - EL = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ E_f L_{fa} & E_f L_{ff} \end{bmatrix} = \begin{bmatrix} I & 0 \\ -E_f L_{fa} & I - E_f L_{ff} \end{bmatrix}.$$

Hence the spectrum (i.e. set of eigenvalues) of $I - EL$ is the union of the spectrum of $I - E_f L_{ff}$ (all inside the unit circle) and $\{1, 1\}$ (set of two ones).

It is left to verify that for arbitrary initial states of the free agents $x_f(0) \in \mathbb{C}^{n-2}$, $x_f(k)$ converges to $-L_{ff}^{-1} L_{fa} \xi_a (= \xi_f)$ when $x_a(k) = \xi_a$ for all $k \geq 0$. Fix $\xi_a \in \mathbb{C}^2$. First note that

$$\bar{x} = \begin{bmatrix} \bar{x}_a \\ \bar{x}_f \end{bmatrix} = \begin{bmatrix} \xi_a \\ -L_{ff}^{-1} L_{fa} \xi_a \end{bmatrix}$$

is the unique fixed point of (7.11). To see this, substituting $\bar{x}$ into (7.11) yields $\bar{x}$ (thanks to the fact that both E_f and L_{ff} are invertible), which means that $\bar{x}$ is a fixed point of (7.11). Moreover, let

$$\bar{x}' = \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}$$

be another fixed point of (7.11), namely

$$\begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \begin{bmatrix} I & 0 \\ -E_f L_{fa} & I - E_f L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}.$$

From the above we derive

$$\bar{x}'_f = -L_{ff}^{-1} L_{fa} \xi_a = \bar{x}_f.$$

This shows that $\bar{x}$ is the unique fixed point of (7.11). Moreover, since all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle, we derive

$$(\forall x_f(0) \in \mathbb{C}^{n-2}) \lim_{k \to \infty} x_f(k) = -L_{ff}^{-1} L_{fa} \xi_a (= \xi_f).$$

Namely, the second condition in the two-dimensional localization problem is established. This completes the proof. □

7.4 Simulation Examples

Example 7.3 *Let us consider again Example 7.2, where the (generic) configuration is the regular hexagon $\xi = [1 \ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{2\pi}{3}\mathrm{j}} \ \mathrm{e}^{\pi\mathrm{j}} \ \mathrm{e}^{\frac{4\pi}{3}\mathrm{j}} \ \mathrm{e}^{\frac{5\pi}{3}\mathrm{j}}]^\top$. We have designed a complex Laplacian L (copied below for convenience)*

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\mathrm{j} & \frac{3\sqrt{2}+\sqrt{6}}{6} + \frac{3\sqrt{2}-\sqrt{6}}{6}\mathrm{j} & 0 & -\frac{\sqrt{6}}{6} + \frac{\sqrt{6}}{6}\mathrm{j} & 0 \\ -\frac{\sqrt{3}}{4} - \frac{1}{4}\mathrm{j} & 0 & \frac{\sqrt{3}}{2} - \frac{1}{2}\mathrm{j} & -\frac{\sqrt{3}}{4} + \frac{3}{4}\mathrm{j} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{j} & -\sqrt{3}\mathrm{j} & \frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{j} \\ -\frac{\sqrt{3}}{2} + \frac{1}{2}\mathrm{j} & \frac{\sqrt{3}}{6} - \frac{1}{2}\mathrm{j} & 0 & 0 & 0 & \frac{\sqrt{3}}{3} \end{bmatrix}.$$

While it is satisfied that rank$(L) = 4$, one of the eigenvalues of $I - L$ is unstable (i.e. outside the unit circle). Thus we need to design an invertible diagonal matrix E such that, except for the two eigenvalues 1, all the other four eigenvalues of $I - EL$ are stable (i.e. inside the

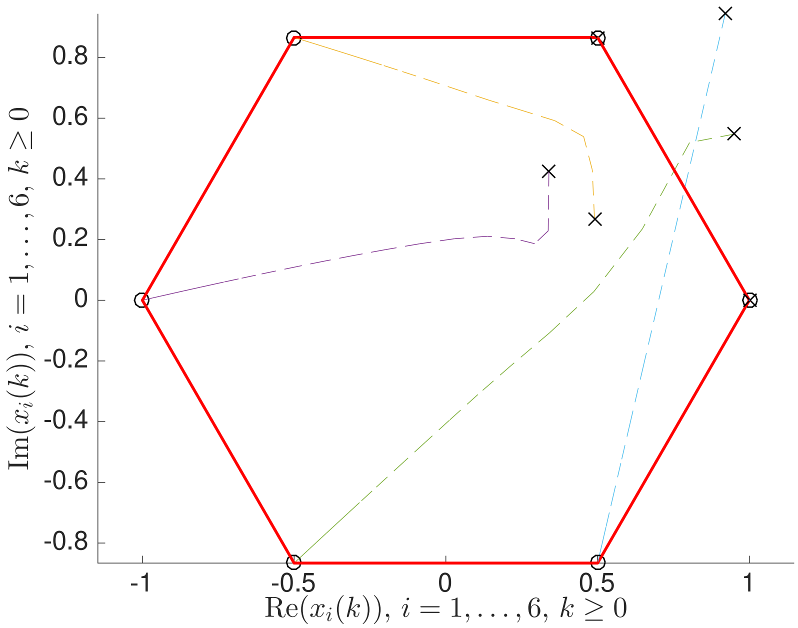


Figure 7.3: Estimations of four free agents converge to their true positions (×: initial estimation; ○: final estimation)

unit circle).

Since the configuration ξ is generic and the digraph $\mathcal{G}$ contains a spanning 2-tree whose two roots are the anchor agents 1 and 2, all the principal minors of the submatrix L_{ff} are nonzero. Therefore by Lemma 7.1, there exists an invertible diagonal matrix E_f such that all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle. For computing such E_f, we apply Algorithm 7.1 and obtain

$$E_f = \mathrm{diag}(-0.4183 + 0.1121\mathrm{j}, 0.25 + 0.433\mathrm{j}, -0.5\mathrm{j}, -0.5).$$

Then an invertible diagonal matrix E such that, except for two eigenvalues 1, all the other eigenvalues of $I - EL$ lying inside the unit circle is:

$$E = \mathrm{diag}(1, 1, -0.4183 + 0.1121\mathrm{j}, 0.25 + 0.433\mathrm{j}, -0.5\mathrm{j}, -0.5).$$

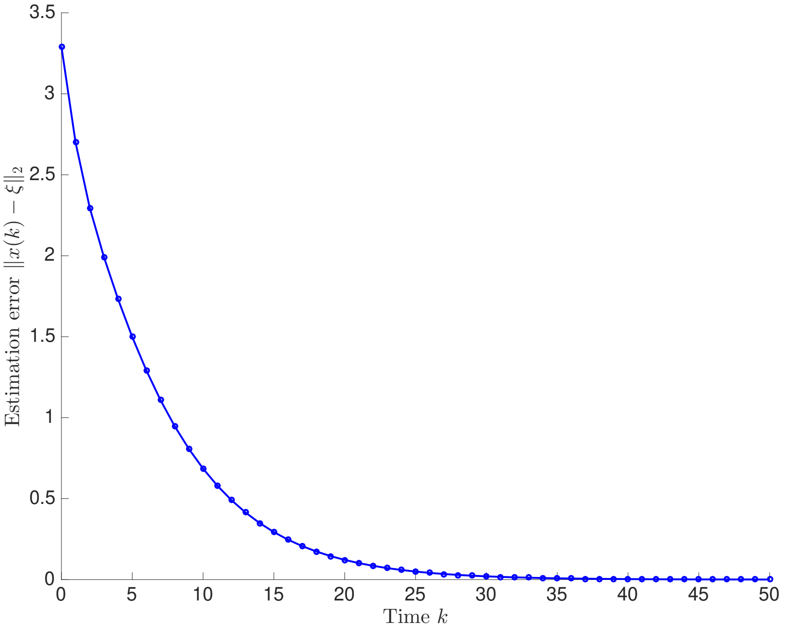


Figure 7.4: Estimation error of six networked agents asymptotically converges to zero

Indeed, the eigenvalues of $I - EL$ are:

$$1, 1, 0.8341, 0.7113, 0.1834 + 0.2947\mathrm{j}, 0.1834 - 0.2947\mathrm{j}.$$

With the initial condition $x_a(0) = [1\ \mathrm{e}^{\frac{\pi}{3}\mathrm{j}}]^\top$ of the two anchor agents and a random initial condition $x_f(0) \in \mathbb{C}^4$ of the 4 free agents, a simulation of the TDLA (i.e. $x(k+1) = (I - EL)x(k)$) yields the trajectories displayed in Fig. 7.3. In the figure, × denotes the initial estimated positions, while ∘ the final estimated positions. First observe that the two anchor agents never change their estimations of their positions (1 and $\mathrm{e}^{\frac{\pi}{3}\mathrm{j}}$ respectively), because these global positions are already known and never need to be updated. For the four free agents, they start from some random estimations of their positions, and it is observed that these estimations converge to their true positions.

Let $e(k) := \|x(k) - \xi\|_2$ be the total estimation error of the networked agents. Then Fig. 7.4 shows that $e(k)$ converges to zero asymptotically.

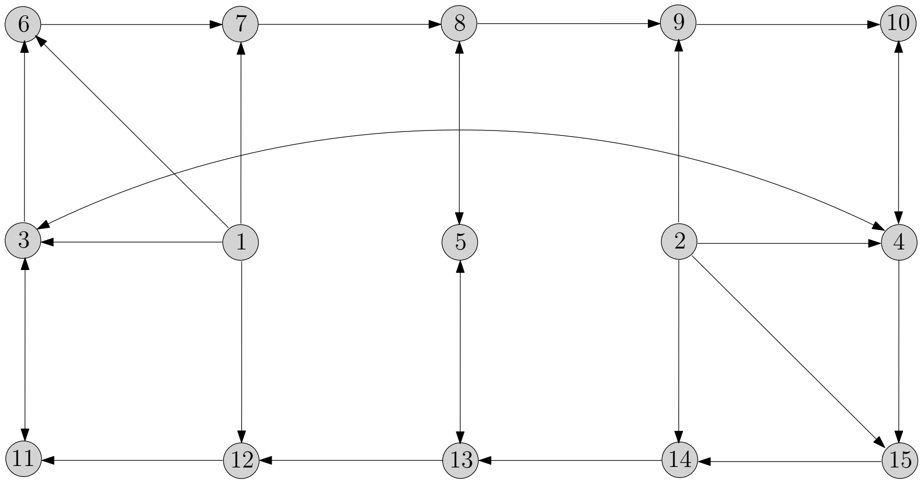


Figure 7.5: Fifteen networked agents

Example 7.4 *Consider a network of* 15 *agents as displayed in Fig. 7.5. Agents* 1 *and* 2 *are anchor agents, and the rest are free agents. This digraph contains a spanning* 2*-tree whose two roots are the two anchor agents.*

First, we consider a regular polygon to be the configuration (fixed positions of the 15 agents in a plane):

$$\xi = [1\ \mathrm{e}^{\frac{2\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{4\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{6\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{8\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{10\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{12\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{14\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{16\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{18\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{20\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{22\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{24\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{26\pi}{15}\mathrm{j}}\ \mathrm{e}^{\frac{28\pi}{15}\mathrm{j}}\ ]^\top.$$

Thus ξ is generic. We then design a complex graph Laplacian L such that rank$(L) = 13$, *and compute by Algorithm 7.1 an invertible diagonal matrix E such that all the eigenvalues (except for two eigenvalues* 1*) of $I - EL$ lie inside the unit circle. With the initial condition $x_a(0) = [1\ \mathrm{e}^{\frac{2\pi}{15}\mathrm{j}}]^\top$ of the two anchor agents and a random initial condition $x_f(0) \in \mathbb{C}^{13}$ of the thirteen free agents, a simulation of the TDLA yields the trajectories displayed in Fig. 7.6. Observe that the estimations of the free agents converge to their true positions. The estimation error $e(k) := \|x(k) - \xi\|_2$ is displayed in Fig. 7.7, which converges to zero asymptotically.*

Second, we consider a triangle shape to be the configuration (fixed positions of the agents in a plane):

$$\xi = [4\mathrm{j}\ \ -1+3\mathrm{j}\ \ 1+3\mathrm{j}\ \ -2+2\mathrm{j}\ \ 2\mathrm{j}\ \ 2+2\mathrm{j}\ \ -3+\mathrm{j}\ \ -1+\mathrm{j}\ \ 1+\mathrm{j}\ \ 3+\mathrm{j}\ \ -4\ \ -2\ \ 0\ \ 2\ \ 4]^\top.$$

Note that this ξ is not *generic, because there are multiple cases of three points on the same*

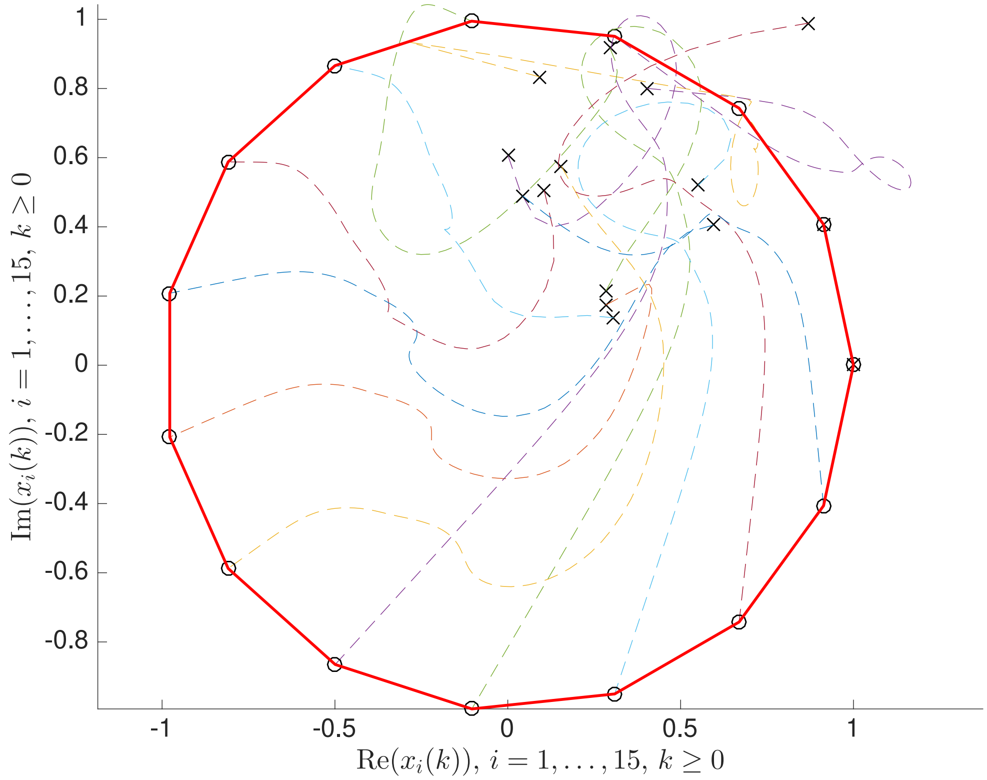


Figure 7.6: Generic configuration: estimations of thirteen free agents converge to their true positions (×: initial estimation; ○: final estimation)

line: e.g. the last three entries $0, 2, 4$ of ξ. For this example, nevertheless, a complex graph Laplacian L may still be designed such that $\mathrm{rank}(L) = 13$, and an invertible diagonal matrix E is obtained by Algorithm 7.1 such that all the eigenvalues (except for two eigenvalues 1) of $I - EL$ lie inside the unit circle. With the initial condition $x_a(0) = [4\mathrm{j} \quad -1+3\mathrm{j}]^\top$ of the two anchor agents and a random initial condition $x_f(0) \in \mathbb{C}^{13}$ of the thirteen free agents, a simulation of the TDLA yields the trajectories displayed in Fig. 7.8. Observe that the estimations of the free agents again converge to their true positions, and the estimation error asymptotically diminishes as displayed in Fig. 7.9.

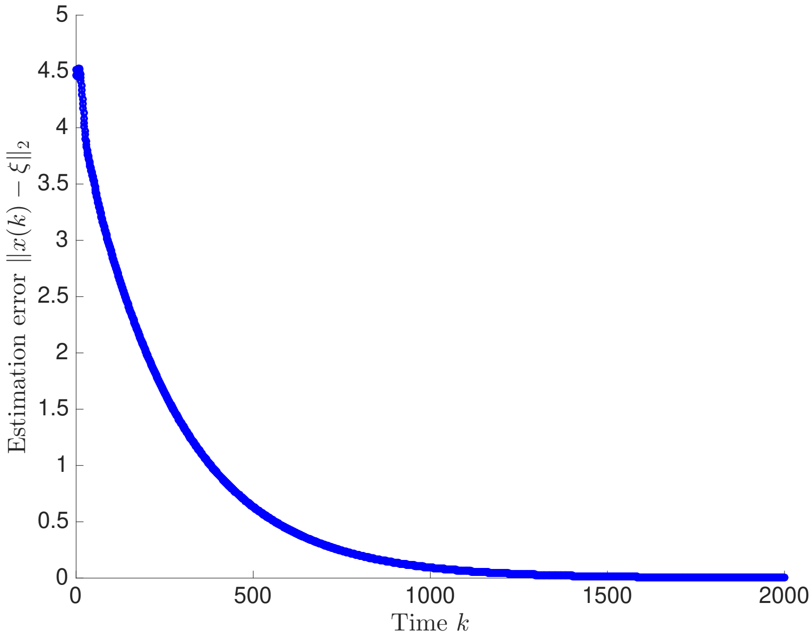


Figure 7.7: Generic configuration: estimation error of fifteen networked agents asymptotically converges to zero

7.5 Notes and References

The two-dimensional localization algorithm (TDLA) is adapted from

- Z. Lin, M. Fu, Y. Diao, Distributed self localization for relative position sensing networks in 2D space, IEEE Transactions on Signal Processing, vol.63, pp.3751–3761, 2015

Other variations of the distributed localization problem based on different assumptions on locally sensed information are reported in:

- Y. Diao, Z. Lin, M. Fu, A barycentric coordinate based distributed localization algorithm for sensor networks, IEEE Transactions on Signal Processing, vol.62, pp.4760–4771, 2014
- Z. Lin, T. Han, R. Zheng, M. Fu, Distributed localization for 2-D sensor networks with bearing-only measurements under switching topologies, IEEE Transactions on Signal Processing, vol.64, pp.6345–6359, 2016
- Z. Lin, T. Han, R. Zheng, C. Yu, Distributed localization with mixed measurements under switching topologies, Automatica, vol.76, pp.251–257, 2017

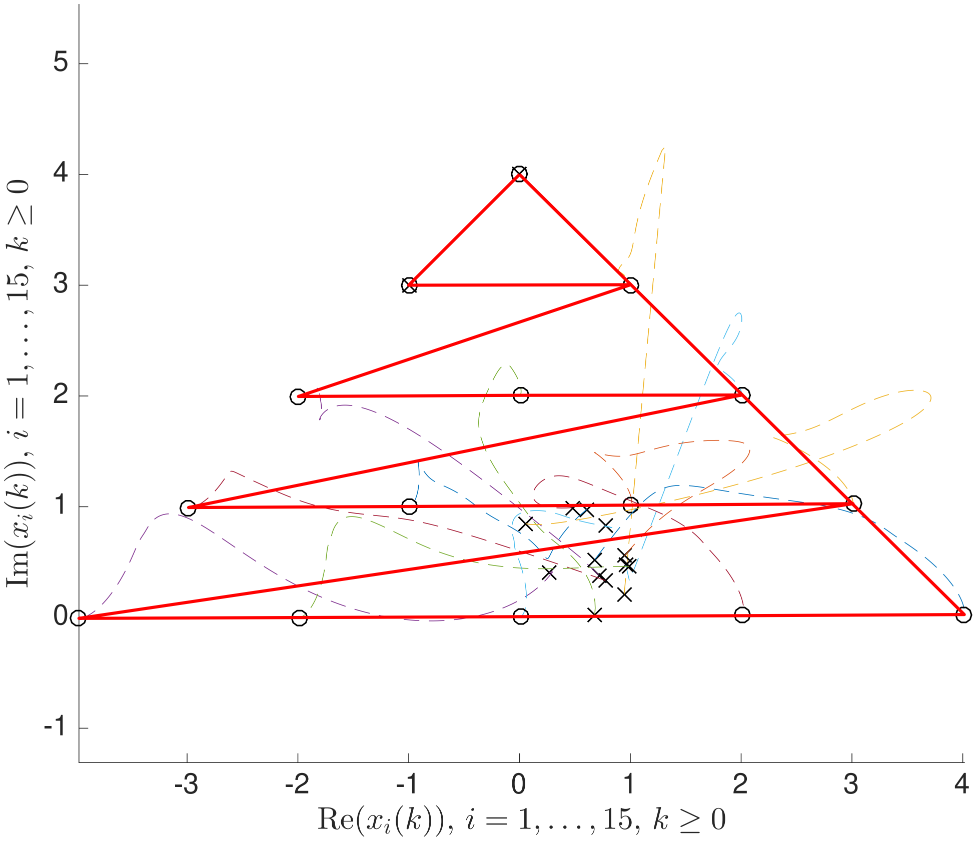


Figure 7.8: Nongeneric configuration: estimations of thirteen free agents converge to their true positions (×: initial estimation; ○: final estimation)

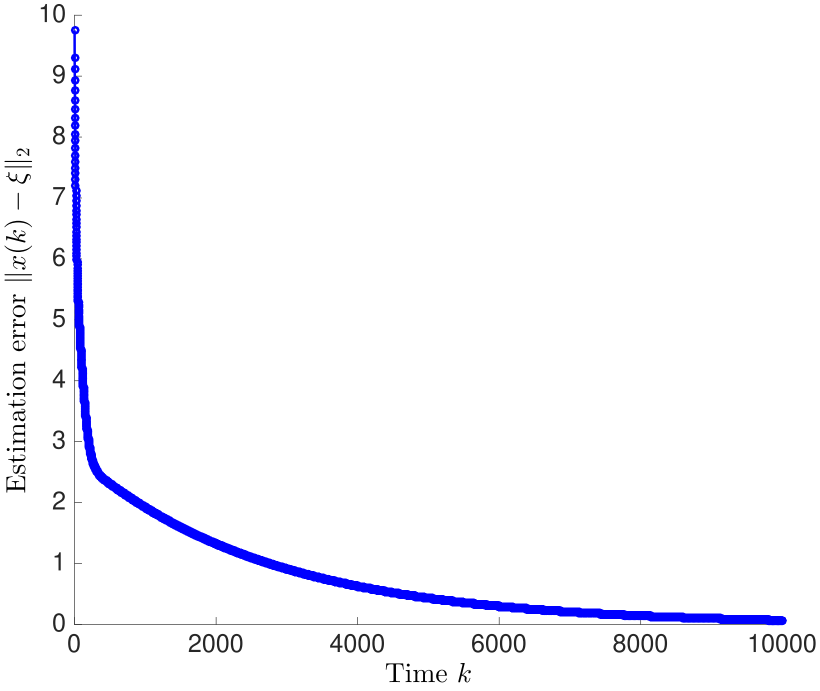


Figure 7.9: Nongeneric configuration: estimation error of fifteen networked agents asymptotically converges to zero

Part V
Spanning Multi-Tree Digraphs: Affine Formation and Localization

This part introduces distributed affine formation control and localization in arbitrary-dimensional space. The necessary graphical condition for solving these two problems in d-dimensions ($d \geq 2$) is that digraphs contain a spanning $(d+1)$-tree. The type of Laplacian matrices involved in these two problems is the signed Laplacian matrices. For agent dynamics, linear time-invariant first-order systems are considered, with continuous-time for affine formation control while discrete-time for localization.

Chapter 8

Affine Formation in Arbitrary Dimensional Space

In this chapter, we study a formation control problem of multi-agent systems in arbitrary dimensional space. In Chapter 6 we introduced a similar formation control problem in 2D, which is applicable to teams of autonomous robots and mobile sensors moving on a plane. However, applications such as formation flying of unmanned aerial vehicles and ocean data retrieval of autonomous underwater vehicles, 3D formation control methods are needed.

This chapter introduces a new formation control problem called *affine formation control*, which includes Chapter 6's 2D similar formation control as a special case. Specifically, in a d (≥ 2) dimensional space, a network of agents is required to form a geometric shape, which can be obtained from a prescribed desired shape via translation, rotation, and *dimension-wise scaling*. The dimension-wise scaling means that scaling factors along each dimension are possibly different. Precisely when all dimensional have identical scaling factors, affine formation control coincides with similar formation control.

The solution for similar formation control in Chapter 6 was based on complex Laplacian, which is however restricted to 2D only. To solve affine formation control in arbitrary dimensions, we introduce the third type of graph Laplacian: *signed Laplacian*. Modeling the interacting agents by digraphs, we show that a necessary graphical condition to achieve affine formation in a d (≥ 2) dimensional space is that the digraph contains a *spanning* $(d+1)$*-tree*, namely there exists (at least) $d+1$ agents that can reach all the other agents through independent paths. These $d+1$ root agents play the role of *leaders*, which determine the translation, rotation, and dimension-wise scaling offsets from the prescribed shape. Under this graphical condition, we present a distributed algorithm for the agents to achieve affine formations.

8.1 Problem Statement

Consider a network of n (> 1) agents in d (≥ 2) dimensional space. Each agent i ($\in [1, n]$) has a *state* variable $x_i(t) \in \mathbb{R}^d$, which is a d-dimensional real vector and denotes the position of agent i

in the d-dimensional space at time t. The time $t \geq 0$ is a (nonnegative) real number and denotes the *continuous* time. The motion of each agent is governed by the following:

$$\dot{x}_i = u_i, \quad i \in [1, n] \tag{8.1}$$

where $u_i(t) \in \mathbb{R}^d$ is the d-dimensional control input.

Let digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ model the interconnection structure of the n agents. Each *node* in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each directed *edge* (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent i can measure the *relative position* of agent j (namely $x_j - x_i$ in agent i's coordinate frame). The *neighbor set* of agent i is $\mathcal{N}_i := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$.

Moreover, consider that digraph $\mathcal{G}$ is weighted: each edge $(j, i) \in \mathcal{V}$ is associated with a real-valued weight $a_{ij} \in \mathbb{R}$. Hence the adjacency matrix $A = (a_{ij})$, degree matrix $D = \text{diag}(A\mathbf{1}_n)$, and Laplacian matrix $L = D - A$ are all real. Note that the adjacency matrix A is not a nonnegative matrix in general; thus L is a *signed Laplacian*.

Define a *target configuration*

$$\xi = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix} \in \mathbb{R}^{nd}, \quad \text{where } \xi_i \in \mathbb{R}^d \text{ and } i \in [1, n]$$

to be the assignment of the n agents to (d-dimensional) points in a global reference frame Σ. This configuration ξ specifies the d-dimensional formation *shape* that the agents are tasked to achieve. To consider not just the 'consensus formation', we henceforth assume in this chapter that ξ is linearly independent from $\mathbf{1}_{nd}$.

Given a target configuration $\xi \in \mathbb{R}^{nd}$, we say another configuration $\xi' \in \mathbb{R}^{nd}$ is *affine* to ξ if there exist a matrix $A \in \mathbb{R}^{d \times d}$ and a vector $a \in \mathbb{R}^d$ such that

$$(\forall i \in [1, n]) \xi'_i = A\xi_i + a.$$

Since an arbitrary real matrix A may be factorized by *singular value decomposition* as $A = U\Gamma V$, where U, V are *unitary matrices* (i.e. $UU^\top = U^\top U = I, VV^\top = V^\top V = I$) and Γ is a $d \times d$ diagonal matrix (diagonal entries being singular values), configuration ξ' can be obtained from ξ via a rotation by V, a scaling along every dimension by Γ, another rotation by U, and finally a translation by a. This is an *affine motion* from ξ.

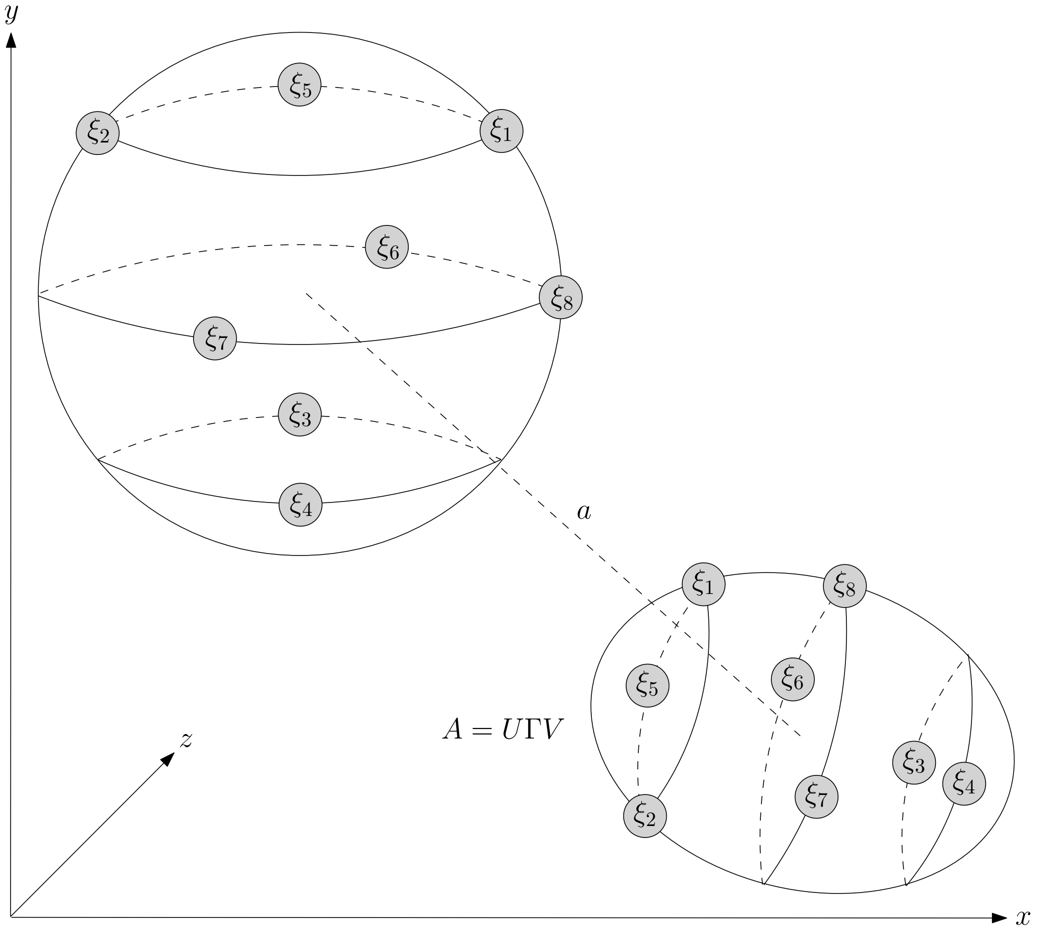


Figure 8.1: Illustration of target configuration and affine configuration

For example, Fig. 8.1 displays a target configuration $\xi = [\xi_1^\top \cdots \xi_8^\top]^\top$ where

$$\xi_1 = \begin{bmatrix} \cos\frac{\pi}{4} \\ 0 \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_2 = \begin{bmatrix} -\cos\frac{\pi}{4} \\ 0 \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_3 = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix}, \xi_4 = \begin{bmatrix} 0 \\ \cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix},$$
$$\xi_5 = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_6 = \begin{bmatrix} \cos\frac{\pi}{3} \\ -\sin\frac{\pi}{3} \\ 0 \end{bmatrix}, \xi_7 = \begin{bmatrix} -\cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} \\ 0 \end{bmatrix}, \xi_8 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

This target configuration consists of eight points on a unit sphere in 3D. Also displayed is another configuration ξ' affine to ξ, as it may be obtained from ξ via rotations and scalings via A and a translation via a.

For a given target configuration ξ, let

$$\begin{aligned}\mathcal{A}(\xi) &:= \{\xi' \in \mathbb{R}^{nd} \mid (\exists A \in \mathbb{R}^{d\times d}, \exists a \in \mathbb{R}^d)(\forall i \in [1,n])\xi_i' = A\xi_i + a\} \\ &= \{\xi' \in \mathbb{R}^{nd} \mid (\exists A \in \mathbb{R}^{d\times d}, \exists a \in \mathbb{R}^d)\xi' = (I_n \otimes A)\xi + \mathbf{1}_n \otimes a\} \qquad (8.2)\end{aligned}$$

be the family of all configurations affine to ξ. Here $\otimes$ is *Kronecker product*. We say that the n agents with the aggregated state vector $x = [x_1^\top \cdots x_n^\top]^\top$ form an *affine formation* with respect to ξ if $x \in \mathcal{A}(\xi)$.

To achieve an affine formation, consider the distributed control

$$u_i = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j - x_i) \qquad (8.3)$$

where the control gain $w_{ij} \in \mathbb{R}$ satisfies

$$\text{(i)} \ \sum_{j\in\mathcal{N}_i} w_{ij}(\xi_j - \xi_i) = 0 \qquad (8.4)$$

$$\text{(ii)} \ w_{ij} = \epsilon_i a_{ij}, \quad \epsilon_i \in \mathbb{R}, \epsilon_i \neq 0. \qquad (8.5)$$

This control (8.3) is in the same form as that for similar formation in Chapter 6: the gains w_{ij} are not simply the edge weights a_{ij}, but are real multiples of a_{ij} (8.5) and satisfy linear constraints with respect to the target configuration ξ (8.4). Different from the control for similar formations where edge weights and control gains are complex, here edge weights and control gains are real.

Moreover, substituting (8.5) into (8.4) and removing the common multiple ϵ_i yield

$$\sum_{j\in\mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0. \qquad (8.6)$$

This in matrix form is $(L \otimes I_d)\xi = 0$. Since $L\mathbf{1}_n = 0$, it follows that

$$\ker(L \otimes I_d) \supseteq \mathcal{A}(\xi). \qquad (8.7)$$

To see this, let $\xi' \in \mathcal{A}(\xi)$. Then there exist a matrix A and a vector a such that $\xi' = (I_n \otimes A)\xi + \mathbf{1}_n \otimes a$.

Hence

$$\begin{aligned}(L \otimes I_d)\xi' &= (L \otimes I_d)((I_n \otimes A)\xi + \mathbf{1}_n \otimes a) \\ &= (L \otimes I_d)(I_n \otimes A)\xi + (L \otimes I_d)(\mathbf{1}_n \otimes a) \\ &= (L \otimes A)\xi + (L\mathbf{1}_n) \otimes a \\ &= (I_n \otimes A)(L \otimes I_d)\xi \\ &= 0.\end{aligned}$$

The above derivation means $\xi' \in \ker(L \otimes I_d)$. From the above we know that if the control in (8.3) satisfying (8.4) and (8.5) can be found, the kernel of $L \otimes I_d$ at least contains the family of all configurations affine to the target ξ.

Affine Formation Control Problem:

Consider a network of agents modeled by (8.1) interconnected through a digraph, and let $\xi \in \mathbb{R}^{nd}$ be a target configuration (linearly independently of $\mathbf{1}_{nd}$). Design a distributed control $u_i(t)$ in (8.3) such that

$$\begin{aligned}&\text{(i)}\ \ \ker(L \otimes I_d) = \mathcal{A}(\xi) \\ &\text{(ii)}\ (\forall x(0) \in \mathbb{R}^{nd})(\exists \xi' \in \mathcal{A}(\xi)) \lim_{t\to\infty} x(t) = \xi'.\end{aligned}$$

The first requirement (i) strengthens (8.7) to equality; namely the kernel of $L \otimes I_d$ is *exactly* the family $\mathcal{A}(\xi)$ of all configurations affine to ξ. The second requirement (ii) means that every trajectory of the networked agents converges to an affine formation in $\mathcal{A}(\xi)$.

Example 8.1 *We provide an example to illustrate the affine formation control problem. As displayed in Fig. 8.2, eight agents are interconnected through a digraph. The neighbor sets of the agents are* $\mathcal{N}_1 = \mathcal{N}_2 = \mathcal{N}_3 = \mathcal{N}_4 = \emptyset$, $\mathcal{N}_5 = \{1, 2, 6, 7\}$, $\mathcal{N}_6 = \{3, 4, 7, 8\}$, $\mathcal{N}_7 = \{1, 5, 6, 8\}$, *and* $\mathcal{N}_8 = \{4, 5, 6, 7\}$.
Let the target configuration ξ *be eight (three-dimensional) points on a unit sphere (see Fig. 8.1). Thus the family* $\mathcal{A}(\xi)$ *contains all affine formations that can be obtained from* ξ *via affine motions.*
The affine formation control problem is to design a distributed control $u_i(t)$ *in (8.3) such that the kernel of* $L \otimes I_d$ *coincides with* $\mathcal{A}(\xi)$, *and moreover the agents' aggregated state vector asymptotically converges to an affine formation in* $\mathcal{A}(\xi)$.

A necessary graphical condition for solving the affine formation control problem is given below.

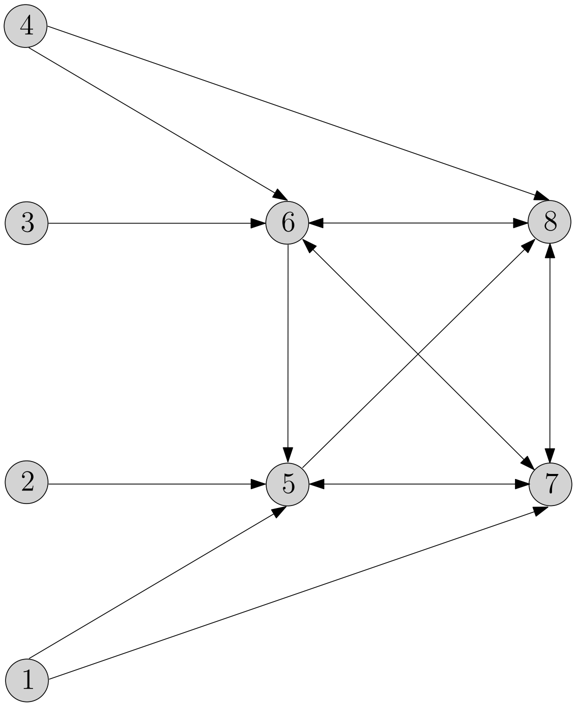


Figure 8.2: Illustrating example of eight agents

Proposition 8.1 *Suppose that there exists a distributed control $u_i(t)$ in (8.3) that solves the affine formation control problem in a d-dimensional space. Then the digraph contains a spanning $(d+1)$-tree.*

Proof. Let $\xi \in \mathbb{R}^{nd}$ be a target configuration. Suppose that there exists a distributed control in (8.3) that solves the d-dimensional affine formation control problem with respect to ξ, but that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ does *not* contain a spanning $(d+1)$-tree. We will derive a contradiction that $\ker(L \otimes I_d) \supsetneqq \mathcal{A}(\xi)$, thereby proving that $\mathcal{G}$ must contain a spanning $(d+1)$-tree.

First, by definition $\mathcal{G}$ containing no spanning $(d+1)$-tree means the following. Let $\mathcal{R}$ be an arbitrary set of $d+1$ nodes. Then removing a set $\mathcal{D}$ of d nodes in $\mathcal{V} \setminus \mathcal{R}$ and all their incoming and outgoing edges, a subset $\mathcal{V}_{\mathcal{D}} \subsetneqq \mathcal{V} \setminus \mathcal{D}$ is unreachable from $\mathcal{R}$ in the new digraph $\mathcal{G}'$. We write this as $\mathcal{R} \nrightarrow \mathcal{V}_{\mathcal{D}}$ in $\mathcal{G}'$.

Now let $\bar{\mathcal{V}}_{\mathcal{D}} := \mathcal{V} \setminus (\mathcal{V}_{\mathcal{D}} \cup \mathcal{D})$. This set $\bar{\mathcal{V}}_{\mathcal{D}}$ is nonempty because $\mathcal{R} \subseteq \bar{\mathcal{V}}_{\mathcal{D}}$ (trivially). In addition, even after removing $\mathcal{D}$, the nodes in $\bar{\mathcal{V}}_{\mathcal{D}}$ can still be reached from $\mathcal{R}$, i.e. $\mathcal{R} \to \bar{\mathcal{V}}_{\mathcal{D}}$; but $\bar{\mathcal{V}}_{\mathcal{D}} \nrightarrow \mathcal{V}_{\mathcal{D}}$.

Let $m := |\mathcal{V}_{\mathcal{D}}|$ (≥ 1), and relabel

- nodes in $\mathcal{V}_{\mathcal{D}}$ from v_1 to v_m;

- nodes in $\mathcal{D}$ from v_{m+1} to v_{m+d};
- nodes in $\bar{\mathcal{V}}_{\mathcal{D}}$ from v_{m+d+1} to v_n.

Then the signed graph Laplacian L of $\mathcal{G}'$ after relabeling (denoted by L') has the following structure:

$$L' = \begin{bmatrix} L'_{11} & L'_{12} & 0 \\ L'_{21} & L'_{22} & L'_{23} \end{bmatrix}.$$

The 0 matrix in the $(1,3)$-block is due to $\bar{\mathcal{V}}_{\mathcal{D}} \not\to \mathcal{V}_{\mathcal{D}}$ in $\mathcal{G}'$.

Also reorder the components ξ_i of the target formation ξ according to the above relabeling, and denote the result by ξ'. By the assumption that there exists a distributed control in (6.3), we have $(L \otimes I_d)\xi = 0$ and $L\mathbf{1}_n = 0$. Substituting the relabeled L' and ξ' into the two equations yields

$$\left(\begin{bmatrix} L'_{11} & L'_{12} & 0 \end{bmatrix} \otimes I_d\right)\xi' = 0, \quad \begin{bmatrix} L'_{11} & L'_{12} & 0 \end{bmatrix}\mathbf{1}_n = 0.$$

Since ξ' and $\mathbf{1}_{nd}$ are linearly independent (linear independence of ξ and $\mathbf{1}_{nd}$ is assumed at the outset), the rows of $[L'_{11}\ L'_{12}\ 0]$ are linearly dependent.

Now remove from L' the $d+1$ rows corresponding to $\mathcal{R}$ and $d+1$ arbitrary columns. Since $\mathcal{R} \subseteq \bar{\mathcal{V}}_{\mathcal{D}}$, it holds that the removed nodes have numbers in $[m+d+1, n]$. Then the resulting matrix $L'_{\mathcal{R}} \in \mathbb{R}^{(n-d-1)\times(n-d-1)}$ is

$$L'_{\mathcal{R}} = \begin{bmatrix} L'_{\mathcal{R},11} & L'_{\mathcal{R},12} & 0 \\ L'_{\mathcal{R},21} & L'_{\mathcal{R},22} & L'_{\mathcal{R},23} \end{bmatrix}.$$

Thus $[L'_{\mathcal{R},11}\ L'_{\mathcal{R},12}\ 0]$ still has m rows. Since the m rows of $[L'_{11}\ L'_{12}\ 0]$ are linearly dependent, so are the m rows of $[L'_{\mathcal{R},11}\ L'_{\mathcal{R},12}\ 0]$. Thus $L'_{\mathcal{R}}$ has less than $n-d-1$ linearly independent rows, and consequently $\det(L'_{\mathcal{R}}) = 0$.

Finally since the set $\mathcal{R}$ of $d+1$ nodes is arbitrary, the original signed graph Laplacian L of $\mathcal{G}'$ does not have any minor with size $n-d-1$ that has nonzero determinant. This means that $\text{rank}(L) \le n-d-2$, and therefore $\ker(L \otimes I_d) \supsetneqq \mathcal{A}(\xi)$. This is a contradiction to the solvability of the affine formation control problem. The proof is now complete. □

Owing to Proposition 8.1, we shall henceforth assume that the digraph contains a spanning $(d+1)$-tree.

Assumption 8.1 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning $(d+1)$-tree.*

Remark 8.1 (Affine formation versus similar formation in 2D) *Consider the special case $d = 2$, i.e. a 2D plane (with two axes labeled x, y). In this special case, both affine formations and* *similar formations may be defined, but there is a notable difference. Let $\xi \in \mathbb{C}^n$ or $\mathbb{R}^{2n}$. A similar formation $\xi' \in \mathbb{C}^n$ can be obtained from ξ via a translation, a rotation, and a scaling which is the* same *for both x and y axes. On the other hand, an affine formation $\xi' \in \mathbb{R}^{2n}$ can be obtained from ξ via a translation, a rotation, a scaling for x axis and a possibly* different *scaling for y axis. Hence an affine formation allows different scalings along different axes, and this is the reason why the necessary graphical condition for achieving affine formations requires a spanning 3-tree, in contrast with a spanning 2-tree required for similar formations.*

Even if Assumption 8.1 holds, not every configuration $\xi \in \mathbb{R}^{nd}$ (linearly independent with $\mathbf{1}_{nd}$) whose affine configurations may be achieved by a distributed control $u_i(t)$ in (8.3). An illustrative example is provided below.

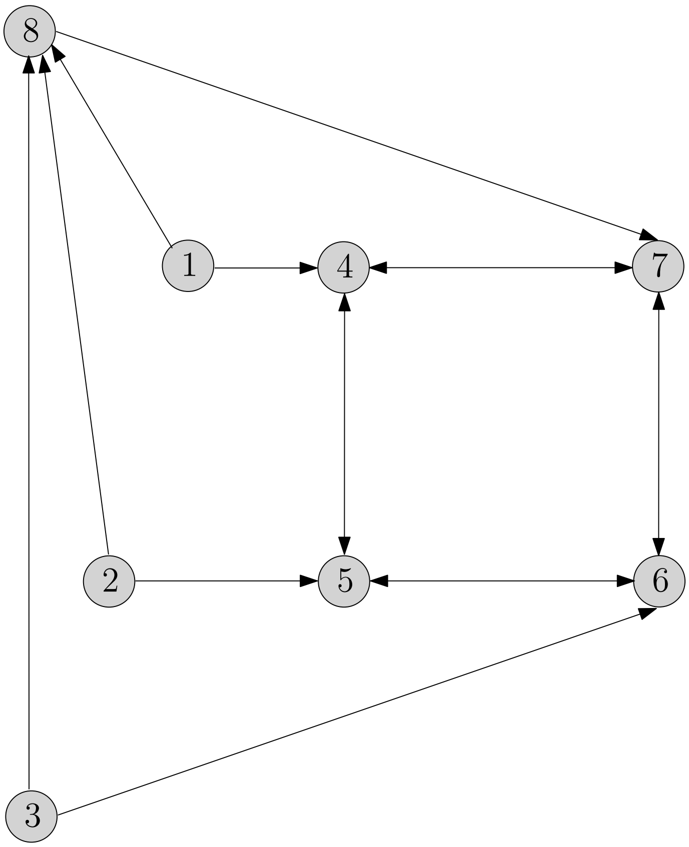


Figure 8.3: Eight-node digraph containing a spanning 3-tree

Example 8.2 *Consider a network of eight agents in a two-dimensional space (i.e. $d = 2$). Their interconnection is modeled by the digraph displayed in Fig. 8.3. This digraph $\mathcal{G}$ contains a spanning* 3*-tree, with the root set $\mathcal{R} = \{1, 2, 3\}$. Now consider the following target configuration $\xi = [\xi_1^\top \ \cdots \ \xi_8^\top]^\top$ where*

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \xi_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \xi_3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \xi_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \xi_5 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \xi_6 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \xi_7 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \xi_8 = \begin{bmatrix} 0 \\ -6 \end{bmatrix}.$$

This target configuration ξ has its first seven two-dimensional points on the same line. Thus ξ is not generic, though it is linearly independent from $\mathbf{1}_{16}$. For this non-generic ξ, for every signed Laplacian L of $\mathcal{G}$ with $(L \otimes I_2)\xi = 0$, it is verified that rank$(L) \leq 4$. To see this, write $(L \otimes I_2)\xi$ explicitly as

$$\left(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_{41} & 0 & 0 & l_{44} & l_{45} & 0 & l_{47} & 0 \\ 0 & l_{52} & 0 & l_{54} & l_{55} & l_{56} & 0 & 0 \\ 0 & 0 & l_{63} & 0 & l_{65} & l_{66} & l_{67} & 0 \\ 0 & 0 & 0 & l_{74} & 0 & l_{76} & l_{77} & l_{78} \\ l_{81} & l_{82} & l_{83} & 0 & 0 & 0 & 0 & l_{88} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \end{bmatrix}.$$

For the forth row of L (other rows are similar), it follows from $L\mathbf{1}_8 = 0$ and $(L \otimes I_2)\xi = 0$ that

$$\begin{aligned} l_{41} + l_{44} + l_{45} + l_{47} &= 0 \\ (l_{41} \otimes I_2)\xi_1 + (l_{44} \otimes I_2)\xi_4 + (l_{45} \otimes I_2)\xi_5 + (l_{47} \otimes I_2)\xi_7 &= 0. \end{aligned}$$

To satisfy these equations, the entries $l_{31}, l_{32}, l_{33}, l_{35}$ are such that

$$\begin{bmatrix} l_{41} \\ l_{44} \\ l_{45} \\ l_{47} \end{bmatrix} \otimes \mathbf{1}_2 = c_4 \begin{bmatrix} \xi_7 - \xi_4 \\ \xi_1 - \xi_5 \\ \xi_4 - \xi_7 \\ \xi_5 - \xi_1 \end{bmatrix} = c_4 \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \otimes \mathbf{1}_2$$

for some nonzero real number c_4. Similarly the (four) entries of rows 5,6,7,8 may be determined up to a nonzero real multiples c_5, c_6, c_7, c_8 (respectively). For simplicity, letting

$c = 4 = c_5 = c_6 = c_7 = c_8 = 1$ we have one instance of L as follows:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 2 & -1 & -2 & 0 & 0 \\ 0 & 0 & 3 & 0 & -3 & -3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ -2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This L has rank 4, meaning that the last five rows are linearly dependent. Then for arbitrary values of c_4, c_5, c_6, c_7, c_8, these five rows cannot become linearly independent. Hence rank$(L) \le 4$ for every L with $(L \otimes I_2)\xi = 0$. This means that $\ker(L \otimes I_2) \supsetneqq \mathcal{S}(\xi)$, and consequently there does not exist a distributed control in (8.3) that solves the affine formation control problem with the chosen target configuration ξ.

In virtue of Example 8.2, we henceforth require that the target formation ξ be generic. The requirement is mild, nevertheless, inasmuch as the set of all non-generic configurations has Lebesgue measure zero. This means that for a given non-generic configuration ξ, randomly perturbing its entries generates a generic configuration. It is also noted that every generic configuration ξ is linearly independent with $\mathbf{1}$.

Assumption 8.2 *The target configuration $\xi = [\xi_1^\top \cdots \xi_n^\top]^\top \in \mathbb{R}^{nd}$ is generic.*

8.2 Distributed Algorithm

Example 8.3 *Consider again Example 8.1, where the target configuration consists of eight (three-dimensional) points on a unit sphere (see Fig. 8.1). This ξ is generic (because no four points are on the same plane).*

To achieve an affine formation of ξ', we consider using the simplest form of the distributed control (8.3) by setting all $\epsilon_i = 1$:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [1, 8] \tag{8.8}$$

where $a_{ij} \in \mathbb{R}$ are real edge weights to be designed to satisfy (8.6):

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0, \quad i \in [1, 8].$$

Now we illustrate how such real weights may be designed. Take agent 6 for example: it has four neighbors $3, 4, 7, 8$. Thus we must find weights $a_{63}, a_{64}, a_{67}, a_{68}$ such that $a_{63}(\xi_3 - \xi_6) + a_{64}(\xi_4 - \xi_6) + a_{67}(\xi_7 - \xi_6) + a_{68}(\xi_8 - \xi_6) = 0$. Substituting vectors $\xi_3, \xi_4, \xi_6, \xi_7, \xi_8$ yields

$$a_{63}\begin{bmatrix} -\cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} - \cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix} + a_{64}\begin{bmatrix} -\cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} + \cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix} + a_{67}\begin{bmatrix} -2\cos\frac{\pi}{3} \\ 2\sin\frac{\pi}{3} \\ 0 \end{bmatrix} + a_{68}\begin{bmatrix} 1 - \cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} \\ 0 \end{bmatrix} = 0.$$

The above reduces to a system of linear equations, with four unknowns (the weights) and three equations. Thus there are infinitely many solutions (indeed the solution space is one dimensional). One solution is $a_{63} = -\sin\frac{\pi}{3}, a_{64} = \sin\frac{\pi}{3}, a_{67} = \cos\frac{\pi}{4}(\cos\frac{\pi}{3} - 1), a_{68} = -2\cos\frac{\pi}{3}\cos\frac{\pi}{4}$. Note that this weight design can be done locally by individual agents if relative information $\xi_j - \xi_i$ $(j \in \mathcal{N}_i)$ is available.

Similarly we design other weights to satisfy (8.6), and write (8.8) in vector form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \left(\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cos\frac{\pi}{3} - \sin\frac{\pi}{3} & -\cos\frac{\pi}{3} - \sin\frac{\pi}{3} & 0 & 0 & 2\sin\frac{\pi}{3} & -\cos\frac{\pi}{4} & \cos\frac{\pi}{4} & 0 \\ 0 & 0 & -\sin\frac{\pi}{3} & \sin\frac{\pi}{3} & 0 & \cos\frac{\pi}{4}(\cos\frac{\pi}{3} + 1) & \cos\frac{\pi}{4}(\cos\frac{\pi}{3} - 1) & -2\cos\frac{\pi}{3}\cos\frac{\pi}{4} \\ -\sin\frac{\pi}{3} & 0 & 0 & 0 & \sin\frac{\pi}{3} & -\frac{1}{2}\cos\frac{\pi}{4}(1 + \sin\frac{\pi}{3} + \cos\frac{\pi}{3}) & \frac{1}{2}\cos\frac{\pi}{4}(1 - \sin\frac{\pi}{3} - \cos\frac{\pi}{3}) & \cos\frac{\pi}{4}(\sin\frac{\pi}{3} + \cos\frac{\pi}{3}) \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}.$$

Inspect that the matrix above has zero row sums, and is indeed the minus of the signed Laplacian matrix L of the (real) weighted digraph. It is also checked that $(L \otimes I_3)\xi = 0$, namely the target configuration lies in the kernel of $L \otimes I_3$. Moreover, there are exactly four eigenvalues 0 of L, and hence $\ker(L \otimes I_3) = \mathcal{A}(\xi)$ (the first requirement of the affine formation control problem is satisfied).
However, the nonzero eigenvalues of matrix $-L$ are

$$-1.0578, -2.371, 0.3828 + 0.8926\mathrm{j}, 0.3828 - 0.8926\mathrm{j}$$

and hence $-L$ is not stable (the last two eigenvalue have positive real parts). Therefore to stabilize $x(t)$ to the kernel of $L \otimes I_3$ (to satisfy the second requirement of the affine formation control problem), the unstable eigenvalues of $-L$ must be moved to the open left-half plane. This shows that simply setting all $\epsilon_i = 1$ in (8.3) does not work in general. In fact, ϵ_i need to be properly chosen in order to stabilize $-L$.

In the following we re-describe the distributed control (8.3) in vector form, and will analyze its stability in relation to the values of ϵ_i in the next section.

Affine Formation Control Algorithm (AFCA):

Every agent i has a state variable $x_i(t) \in \mathbb{R}^d$ ($d \geq 1$) representing its position in a d-dimensional space at time t; the initial state $x_i(0)$ is an arbitrary d-dimensional real vector. Offline, each agent i computes weights a_{ij} by solving (8.6):

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0.$$

Then online, at each time $t \geq 0$, every agent i updates its state $x_i(t)$ using the following distributed control:

$$u_i = \epsilon_i \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i) \tag{8.9}$$

where $\epsilon_i \in \mathbb{R} \setminus \{0\}$ is a (nonzero) real control gain.

Let $x := [x_1^\top \cdots x_n^\top]^\top$ be the aggregated state of the networked agents, and $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ the (diagonal) control gain matrix. Then the n equations (8.9) become

$$\dot{x} = ((-EL) \otimes I_d)x. \tag{8.10}$$

Remark 8.2 *The above AFCA requires that the following information is available for each indi-*

vidual agent i:

- $\xi_j - \xi_i$ *for all* $j \in \mathcal{N}_i$ *(offline computation of weights)*
- $x_j - x_i$ *for all* $j \in \mathcal{N}_i$ *(online computation of control inputs).*

8.3 Convergence Result

The following is the main result of this section.

Theorem 8.1 *Suppose that Assumptions 8.1 and 8.2 hold. There exists a (diagonal and invertible) control gain matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ *such that the AFCA solves the affine formation control problem.*

To prove Theorem 8.1, we will analyze the eigenvalues of the matrix $(-EL) \otimes I_d$ in (8.10). For this, the following fact is useful (which is the real counterpart of Lemma 6.1).

Lemma 8.1 *Consider an arbitrary square real matrix* $M \in \mathbb{R}^{n \times n}$. *If all the principal minors of* M *are nonzero, then there exists an invertible diagonal matrix* $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{R}^{n \times n}$ *such that all the eigenvalues of* EM *have positive real parts.*

Proof: The proof is based on induction on n. For the base case $n = 1$, $M = m_{11}$ is a nonzero scalar (as the principal minor of M is nonzero). Let $\epsilon_1 := \frac{1}{m_{11}}$. Then $EM = \epsilon_1 m_{11} = 1 (= \det(E)\det(M))$.

For the induction step, suppose that the conclusion holds for $M \in \mathbb{R}^{(n-1)\times(n-1)}$. Since the $n-1$ eigenvalues are either positive real or conjugate pairs with positive real parts and $\det(E)\det(M) = \lambda_1 \cdots \lambda_{n-1}$, we have $\det(E)\det(M) > 0$. Now consider $M \in \mathbb{R}^{n \times n}$, with all of its principal minors nonzero. Let M_1 be the submatrix of M with the last row and last column removed. Then all the principal minors of M_1 are nonzero, and by the hypothesis there exists an invertible diagonal matrix $E_1 = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_{n-1})$ such that all the eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ of $E_1 M_1$ have positive real parts. Now write

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix}$$

where m_{nn} is a nonzero scalar (since all the principal minors of M is nonzero). Also let

$$E = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix}$$

for some real ϵ_n. Thus

$$EM = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix} = \begin{bmatrix} E_1 M_1 & E_1 M_2 \\ \epsilon_n M_3 & \epsilon_n m_{nn} \end{bmatrix}.$$

If $\epsilon_n = 0$, then

$$EM = \begin{bmatrix} E_1 M_1 & E_1 M_2 \\ 0 & 0 \end{bmatrix}$$

which means that EM has a (simple) eigenvalue $\lambda_n = 0$ and all the rest $n-1$ eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ have positive real parts. Since eigenvalues are continuous functions of matrix entries, for sufficiently small $|\epsilon_n| > 0$, EM still has $n-1$ eigenvalues $\lambda'_1, \ldots, \lambda'_{n-1}$ with positive real parts.

Now we consider the last eigenvalue λ'_n. Since $\det(E) \neq 0$, $\det(M) \neq 0$, and $\det(EM) = \lambda'_1 \cdots \lambda'_n$, we have $\lambda'_n \neq 0$. If λ'_n is complex, then it must be a conjugate to an existing eigenvalue whose real part is positive. Hence λ'_n also has positive real part. If λ'_n is real, then $\lambda'_1, \ldots, \lambda'_{n-1}$ are symmetric with respect to the real axis. As a result, the product of the first $n-1$ eigenvalues is positive, i.e. $\lambda'_1 \cdots \lambda'_{n-1} > 0$. Also note that

$$\det(EM) = \epsilon_n \det(E_1) \det(M) = \lambda'_1 \cdots \lambda'_{n-1} \lambda'_n.$$

Thus choosing (sufficiently small) ϵ_n such that $\epsilon_n \det(E_1) \det(M) > 0$, we derive $\lambda'_n > 0$. This proves the induction step, and thereby completes the proof. □

The above proof suggests an algorithm (Algorithm 8.1 below) to compute an invertible diagonal matrix $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ such that all the eigenvalues EM have positive real parts. This algorithm is simpler than Algorithm 6.1 in Chapter 6, since computing ϵ_i on line 5 does not involve product of eigenvalues. By the proof of Lemma 6.1, one can always choose appropriate (small) $\delta_1, \ldots, \delta_n$ in line 1 so that Algorithm 8.1 outputs an invertible diagonal matrix E that renders all the eigenvalues EM with positive real parts.

Lemma 8.1 provides a sufficient condition under which the eigenvalues of a real matrix may be moved to the open right-half plane using an invertible diagonal real matrix. The following proposition asserts that this condition holds for the submatrix of the signed Laplacian L of a digraph containing a spanning $(d+1)$-tree, with the $d+1$ rows and $d+1$ columns corresponding to the roots removed. More formally, consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let L be a signed Laplacian matrix of $\mathcal{G}$ (corresponding to a specific choice of edge weights). Let $\mathcal{R} \subseteq \mathcal{V}$, and denote by $L_{\mathcal{R}}$ the submatrix of L by removing the rows and columns corresponding to $\mathcal{R}$.

Algorithm 8.1 Diagonal Stabilization Algorithm (case of real matrix, right-half plane)

```
Input: square real matrix $M \in \mathbb{R}^{n\times n}$ with nonzero principal minors
Output: invertible diagonal matrix $E \in \mathbb{R}^{n\times n}$
set $\delta_1, \dots, \delta_n$ to be small positive real numbers
$\epsilon_1 = \frac{\delta_1}{M(1,1)}$
$E_1 = \text{diag}(\epsilon_1)$
for $i = 2, \dots, n$ do
    $\epsilon_i = \frac{\delta_i}{\det(E_{i-1})\det(M(1{:}i,1{:}i))}$
    $E_i = \text{diag}(\epsilon_1, \dots, \epsilon_i)$
end for
$E = \text{diag}(\epsilon_1, \dots, \epsilon_n)$
```

Proposition 8.2 *Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a configuration ξ. Suppose that Assumptions 8.1 and 8.2 hold. Let $\mathcal{R}$ be a set of $d+1$ roots. Then for almost all signed Laplacian L satisfying $(L \otimes I_d)\xi = 0$, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

To prove Proposition 8.2, we first establish two lemmas.

Lemma 8.2 *Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and suppose that $\mathcal{G}$ contains a spanning $(d+1)$-tree (Assumption 8.1). Let $v_1, \dots, v_{d+1} \in \mathcal{V}$ be $d+1$ roots (renumbering if necessary) and $\mathcal{R} := \{v_1, \dots, v_{d+1}\}$. Then for almost all signed Laplacian L, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

Proof: The proof is based on induction on k, where k is such that the digraph $\mathcal{G}$ contains a spanning k-tree. First consider the base case, namely $k = 1$ and $\mathcal{G}$ contains a spanning tree. Without loss of generality let $v_1 \in \mathcal{V}$ be a root and $\mathcal{R} := \{v_1\}$. For this case, in Lemma 6.2(i) we have shown that the conclusion holds for almost all complex Laplacians, which include signed Laplacians as a special case. Hence for almost all signed Laplacian L, all principal minors of $L_{\mathcal{R}}$ are nonzero.

Next consider the induction step, namely $k = d$ and $\mathcal{G}$ contains a spanning d-tree with a root set $\mathcal{R} = \{v_1, \dots, v_d\}$. Suppose that for almost all real Laplacian L of $\mathcal{G}$, all principal minors of $L_{\mathcal{R}}$ are nonzero. It will be shown that the same conclusion holds for $k = d+1$, in which $\mathcal{G}$ contains a spanning $(d+1)$-tree with a root set $\mathcal{R} = \{v_1, \dots, v_d, v_{d+1}\}$.

Remove an arbitrary node in $\mathcal{R}$ (say v_1) and all its incoming and outgoing edges, and denote the resulting subgraph $\mathcal{G}'$. Then $\mathcal{G}'$ contains a spanning d-tree ($\mathcal{R}' := \{v_2, \dots, v_{d+1}\}$ being a root set), and it follows from the induction hypothesis that for almost all signed Laplacian L' of $\mathcal{G}'$, all the principal minors of $L'_{\mathcal{R}'}$ are nonzero. Since the principal minors of $L'_{\mathcal{R}'}$ are identical with those of $L_{\mathcal{R}}$, where L is a signed Laplacian of $\mathcal{G}$, the conclusion is established. □

For the second lemma, we introduce the following notation. Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

and let L be a signed Laplacian matrix of $\mathcal{G}$. Let $\mathcal{R} \subseteq \mathcal{V}$, and denote by $L^{\mathcal{R}}$ a submatrix of L by removing the rows corresponding to $\mathcal{R}$ and arbitrary $|\mathcal{R}|$ columns.

Lemma 8.3 *Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and suppose that $\mathcal{G}$ contains a spanning $(d+1)$-tree (Assumption 8.1). Let $v_1, \ldots, v_{d+1} \in \mathcal{V}$ be $d+1$ roots (renumbering if necessary) and $\mathcal{R} := \{v_1, \ldots, v_{d+1}\}$. Then for almost all signed Laplacian L, det$(L^{\mathcal{R}}) \neq 0$.*

Proof: The proof is based on induction on k, where k is such that the digraph $\mathcal{G}$ contains a spanning k-tree. First consider the base case: namely $k = 1$ and $\mathcal{G}$ contains a spanning tree. Let v_1 be a root of $\mathcal{G}$ (without loss of generality), $\mathcal{R} := \{v_1\}$, and L a signed Laplacian of $\mathcal{G}$. For this case, in Lemma 6.3(i) we have shown that the conclusion holds for almost all complex Laplacian of $\mathcal{G}$, which include signed Laplacian of $\mathcal{G}$ as a special case. Hence for almost all signed Laplacian L of $\mathcal{G}$, $\det(L^{\mathcal{R}}) \neq 0$.

Next consider the induction step: namely $k = d$ and $\mathcal{G}$ contains a spanning d-tree with a root set $\mathcal{R} := \{v_1, \ldots, v_d\}$ (without loss of generality). Suppose that for almost all signed Laplacian L of $\mathcal{G}$, $\det(L^{\mathcal{R}}) \neq 0$. It will be shown that the same conclusion holds for $k = d+1$, where $\mathcal{G}$ contains a spanning $(d+1)$-tree with a root set $\mathcal{R} := \{v_1, \ldots, v_d, v_{d+1}\}$ (without loss of generality).

Consider $k = d+1$. Let $L^{\mathcal{R}}$ be a submatrix of L with $d+1$ rows corresponding to $\mathcal{R} = \{v_1, \ldots, v_d, v_{d+1}\}$ and arbitrary $d+1$ columns removed. Also let $\bar{\mathcal{V}}$ be the set of $d+1$ nodes that correspond to the removed columns. If $\bar{\mathcal{V}} \cap \mathcal{R} \neq \emptyset$; then let $v_i \in \bar{\mathcal{V}} \cap \mathcal{R}$. Remove v_i and all its incoming and outgoing edges, and denote the resulting subgraph $\mathcal{G}'$. Then $\mathcal{G}'$ contains a spanning d-tree ($\mathcal{R} \setminus \{v_i\}$ being a set of d roots), and it follows from the induction hypothesis that for almost all signed Laplacian L' of $\mathcal{G}'$, $\det((L')^{\mathcal{R}\setminus\{v_i\}}) \neq 0$. This implies $\det(L^{\mathcal{R}}) \neq 0$ for almost all signed Laplacian L of $\mathcal{G}$.

It remains to consider the case $\bar{\mathcal{V}} \cap \mathcal{R} = \emptyset$; namely the nodes corresponding to the removed columns are not in the root set $\mathcal{R}$. For this, let $v_i \in \mathcal{V} \setminus \mathcal{R}$, and denote by p_i ($i \in [1, n]$) the ith row of L. Consider the following elementary row transformations:

$$L = \begin{bmatrix} p_1 \\ \vdots \\ p_{d+1} \\ \vdots \\ p_i \\ \vdots \end{bmatrix} \Longrightarrow \tilde{L} := \begin{bmatrix} k_1 p_1 + \cdots + k_n p_n \\ \vdots \\ p_{d+1} \\ \vdots \\ p_i \\ \vdots \end{bmatrix}$$

where $k_1, \ldots, k_n$ are proper coefficients such that the $d+2$ entries $\tilde{L}(1,1), \ldots, \tilde{L}(1, d+1), \tilde{L}(1, i)$ on the first row of $\tilde{L}$ are nonzero. Such coefficients always exist because each of the $d+1$ roots has at least one outgoing edge. Denote by $\tilde{\mathcal{G}}$ the digraph corresponding to $\tilde{L}$. We claim that $\tilde{\mathcal{G}}$ contains a spanning $(d+1)$-tree with a root set $\tilde{\mathcal{R}} := \{v_2, \ldots, v_{d+1}, v_i\}$. To see this, first note that v_1 is $(d+1)$-reachable from $\tilde{\mathcal{R}}$ because $\tilde{L}(1,2), \ldots, \tilde{L}(1,d+1), \tilde{L}(1,i)$ are nonzero and there are $d+1$ edges $(v_2, v_1), \ldots, (v_{d+1}, v_1), (v_i, v_1)$. Now consider a node v_j $(j \neq 1, \ldots, d+1, i)$; there are two cases:

- All $d+1$ disjoint paths from $\mathcal{R}$ to v_j do not go through v_i. Then v_j is $(d+1)$-reachable from $\tilde{\mathcal{R}}$: $v_2 \to v_j, \ldots, v_{d+1} \to v_j$, and $v_i \to v_1 \to v_j$.
- Among $d+1$ disjoint paths from $\mathcal{R}$ to v_j, there exists a path from $v_m \in \mathcal{R}$ $(m \in [1, d+1])$ such that $v_m \to v_i \to v_j$. Then v_j is also $(d+1)$-reachable from $\tilde{\mathcal{R}}$: $v_m \to v_1 \to v_j$, $v_1 \to v_j, \ldots, v_{m-1} \to v_j, v_{m+1} \to v_j, \ldots, v_{d+1} \to v_j$, and $v_i \to v_j$.

Note that it is not possible that more than one path from $\mathcal{R}$ to v_j goes through v_i in virtual of the definition of spanning d-tree. Hence our claim is established.

Now remove node v_i and all its incoming and outgoing edges, and denote the resulting subgraph $\tilde{\mathcal{G}}'$. Then $\tilde{\mathcal{G}}'$ contains a spanning d-tree ($\tilde{\mathcal{R}} \setminus \{v_i\}$ being a root), and it follows from the induction hypothesis that for almost all signed Laplacian $\tilde{L}'$ of $\tilde{\mathcal{G}}'$, $\det((\tilde{L}')^{\tilde{\mathcal{R}}\setminus\{v_i\}}) \neq 0$. Since $L^{\mathcal{R}}$ may be obtained from $(\tilde{L}')^{\tilde{\mathcal{R}}\setminus\{v_i\}}$ via elementary row transformations (reordering the first row to the ith position and recovering p_i), we conclude that $\det(L^{\mathcal{R}}) \neq 0$ for almost all signed Laplacian L of $\mathcal{G}$. The proof is now complete. □

With the above two lemmas, we now provide the proof of Proposition 8.2.

Proof of Proposition 8.2: By Assumption 8.1, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning $(d+1)$-tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$, where $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$ and the set of $d+1$ roots $\mathcal{R} = \{v_1, \ldots, v_{d+1}\}$ (renumbering if necessary). Consider a signed Laplacian T of $\mathcal{T}$ such that all principal minors of $T_{\mathcal{R}}$ are nonzero. Such T always exists by Lemma 8.2. For the rank of T, on one hand $\mathrm{rank}(T) \geq n-d-1$ since $\det(T_{\mathcal{R}}) \neq 0$; on the other hand $\mathrm{rank}(T) \leq n-d-1$ since the first $d+1$ rows of T are zero row vectors. Hence $\mathrm{rank}(T) = n-d-1$, and the kernel of T is $d+1$ dimensional. One basis of this kernel is $\mathbf{1}_n$ since T is a signed Laplacian. Denote the other d bases by $\eta_1, \ldots, \eta_d$. Then $\mathbf{1}_n, \eta_1, \ldots, \eta_d$ are linearly independent.

Writing $H := [\mathbf{1}_n \ \eta_1 \ \cdots \ \eta_d] \in \mathbb{R}^{n \times (d+1)}$, we claim that by removing any $n-d-1$ rows of H, the remaining square matrix $H' \in \mathbb{R}^{(d+1)\times(d+1)}$ has full rank, i.e. $\mathrm{rank}(H') = d+1$. To see this, suppose on the contrary that by removing certain $n-d-1$ rows of H, the remaining matrix H' is such that $\mathrm{rank}(H') < d+1$. Renumbering the indices of the removed rows to be $\mathcal{I} := \{d+2, \ldots, n\}$ and accordingly reordering the rows of the matrix H transform H to

$$\tilde{H} = \begin{bmatrix} M \\ N \end{bmatrix}, \quad \text{where } M \in \mathbb{R}^{(d+1)\times(d+1)}, N \in \mathbb{R}^{(n-d-1)\times(d+1)}.$$

The above (contrapositive) assumption means $\operatorname{rank}(M) < d+1$. Namely, there exists a nonzero vector $\zeta \in \mathbb{R}^{d+1}$ such that $M\zeta = 0$. On the other hand, reordering the columns of the signed Laplacian matrix T according to $\mathcal{I}$ and then removing the $d+1$ rows corresponding to the $d+1$ roots transform T to

$$\tilde{T} = \begin{bmatrix} T_1 & T_2 \end{bmatrix}, \quad \text{where } T_1 \in \mathbb{R}^{(n-d-1)\times(d+1)}, T_2 \in \mathbb{R}^{(n-d-1)\times(n-d-1)}.$$

By Lemma 8.3, $\det(T_2) \neq 0$. It follows from $TH = 0$ that $\tilde{T}\tilde{H} = 0$; specifically:

$$\begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} M \\ N \end{bmatrix} = 0.$$

Since $M\zeta = 0$, we derive from above equation that $T_2 N\zeta = 0$. Further, since $\det(T_2) \neq 0$, we have $N\zeta = 0$. This implies $H\zeta = 0$, which means that its columns $\mathbf{1}_n, \eta_1, \ldots, \eta_d$ are not linearly independent. This is a contradiction, and hence by removing any $n-d-1$ rows of H, the remaining matrix H' has full rank after all.

Moreover, since each node $v_i \in \mathcal{V} \setminus \mathcal{R}$ has exactly $d+1$ neighbors (by the definition of spanning $(d+1)$-tree), each corresponding row of T has at most $d+2$ nonzero entries. Thus equation $TH = 0$ yields:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \eta_{1i} & \eta_{1i_1} & \cdots & \eta_{1i_{d+1}} \\ \vdots & \vdots & \vdots & \vdots \\ \eta_{di} & \eta_{di_1} & \cdots & \eta_{di_{d+1}} \end{bmatrix} \begin{bmatrix} T_{ii} \\ T_{ii_1} \\ \vdots \\ T_{ii_{d+1}} \end{bmatrix} = 0$$

where $v_{i_1}, \ldots, v_{i_{d+1}}$ are the $d+1$ neighbors of v_i. Write

$$H_i := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \eta_{1i} & \eta_{1i_1} & \cdots & \eta_{1i_{d+1}} \\ \vdots & \vdots & \vdots & \vdots \\ \eta_{di} & \eta_{di_1} & \cdots & \eta_{di_{d+1}} \end{bmatrix}, \quad T_i := \begin{bmatrix} T_{ii} & T_{ii_1} & \cdots & T_{ii_{d+1}} \end{bmatrix}.$$

Since by removing any $n-d-1$ rows of H, the remaining matrix $H' \in \mathbb{R}^{(d+1)\times(d+1)}$ has full rank, we have $\operatorname{rank}(H_i) = d+1$. Hence the kernel of H_i is one-dimensional, which means that T_i (the solution of the above system of linear equations) lies in a one-dimensional subspace.

Now consider a generic configuration $\xi = [\xi_1^\top \cdots \xi_n^\top]^\top \in \mathbb{R}^{nd}$ and another signed Laplacian T'

of $\mathcal{T}$ such that $(T' \otimes I_d)\xi = 0$. This equation leads to

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_{11} & \xi_{21} & \cdots & \xi_{n1} \\ \vdots & \vdots & \vdots & \vdots \\ \xi_{1d} & \xi_{2d} & \cdots & \xi_{nd} \end{bmatrix} \begin{bmatrix} T'_{i1} \\ T'_{i2} \\ \vdots \\ T'_{in} \end{bmatrix} = 0$$

for every ith row of T'. Similar to T above, each row of T' corresponding to a non-root node $v_i \in \mathcal{V} \setminus \mathcal{R}$ has at most $d+2$ nonzero entries. It follows from the above equation that

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_{i1} & \xi_{i_1 1} & \cdots & \xi_{i_{d+1}1} \\ \vdots & \vdots & \vdots & \vdots \\ \xi_{id} & \xi_{i_1 d} & \cdots & \xi_{i_{d+1}d} \end{bmatrix} \begin{bmatrix} T'_{ii} \\ T'_{ii_1} \\ \vdots \\ T'_{ii_{d+1}} \end{bmatrix} = 0$$

where $v_{i_1}, \ldots, v_{i_{d+1}}$ are the $d+1$ neighbors of v_i. Write

$$\Xi_i := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_{i1} & \xi_{i_1 1} & \cdots & \xi_{i_{d+1}1} \\ \vdots & \vdots & \vdots & \vdots \\ \xi_{id} & \xi_{i_1 d} & \cdots & \xi_{i_{d+1}d} \end{bmatrix}, \quad T'_i := \begin{bmatrix} T'_{ii} & T'_{ii_1} & \cdots & T'_{ii_{d+1}} \end{bmatrix}.$$

Since ξ is generic, $\mathrm{rank}(\Xi_i) = d+1$. Hence the kernel of Ξ_i is one-dimensional, which means that T'_i (the solution of the above system of linear equations) lies in a one-dimensional subspace.

We claim that T'_i and T_i have the same zero/nonzero patterns. To see this, suppose that $T'_{ij} \neq 0$ ($j \in \{i, i_1, \ldots, i_{d+1}\}$). Since T'_i is in a one-dimensional subspace, an arbitrary (nonzero) scaling of T'_i generates a new T''_i with (still) $T''_{ij} \neq 0$. This holds as long as $\mathrm{rank}(\Xi_i) = d+1$. In particular, as $\mathrm{rank}(H_i) = d+1$, we have $T_{ij} \neq 0$ (indeed T_{ij} is a nonzero real multiple of T'_{ij}). The other case where $T'_{ij} = 0$ implies $T_{ij} = 0$ is similar. Since all principal minors of $T_{\mathcal{R}}$ are nonzero, it follows from the fact that a polynomial is either constantly zero or nonzero almost everywhere that all principal minors of $T'_{\mathcal{R}}$ are also nonzero.

Finally, returning to the digraph $\mathcal{G}$ and let L be a signed Laplacian of $\mathcal{G}$ satisfying $(L \otimes I_d)\xi = 0$. Compared with T', L has more nonzero real entries. Again according to the fact that a polynomial is either constantly zero or nonzero almost everywhere, we conclude that all principal minors of $L_{\mathcal{R}}$ are also nonzero. The proof is now complete. □

Finally we are ready to prove Theorem 8.1.

Proof of Theorem 8.1: Let Assumptions 8.1 and 8.2 hold. On one hand, it follows from Proposition 8.2 that for almost all signed Laplacian L of $\mathcal{G}$ satisfying $(L \otimes I_d)\xi = 0$ (where ξ is generic),

$\mathrm{rank}(L) \geq n-d-1$, i.e. $\dim(\ker L) \leq d+1$. On the other hand, by using the distributed control in AFCA, we derive $\ker(L \otimes I_d) \supseteq \mathcal{A}(\xi)$ as in (8.7), and thus $\dim(\ker L) \geq d+1$. Therefore for almost all signed Laplacian L of $\mathcal{G}$ satisfying $(L \otimes I_d)\xi = 0$, we have $\ker(L \otimes I_d) = \mathcal{A}(\xi)$, which establishes the first condition in the affine formation control problem.

For the second condition, let $\mathcal{R} = \{v_1, \ldots, v_{d+1}\}$ (renumbering if necessary) be the set of $d+1$ roots and $L_{\mathcal{R}}$ the submatrix of L of $\mathcal{G}$ with the fist $d+1$ rows and columns corresponding to $\mathcal{R}$ removed. Then by Proposition 8.2, for almost all signed Laplacian L satisfying $(L \otimes I_d)\xi = 0$, all principal minors of $L_{\mathcal{R}}$ are nonzero. It then follows from Lemma 8.1 that there exists an invertible diagonal matrix $E_{\mathcal{R}} = \mathrm{diag}(\epsilon_{d+2}, \ldots, \epsilon_n)$ such that all the eigenvalues of $-E_{\mathcal{R}}L_{\mathcal{R}}$ have negative real parts. Let

$$E' := \begin{bmatrix} 0 & 0 \\ 0 & E_{\mathcal{R}} \end{bmatrix}, \quad L = \begin{bmatrix} L_1 & L_2 \\ L_3 & L_{\mathcal{R}} \end{bmatrix}.$$

It follows that

$$-E'L = -\begin{bmatrix} 0 & 0 \\ E_{\mathcal{R}}L_3 & E_{\mathcal{R}}L_{\mathcal{R}} \end{bmatrix}.$$

Hence the spectrum (i.e. set of eigenvalues) of $-E'L$ is the union of the spectrum of $-E_{\mathcal{R}}L_{\mathcal{R}}$ and $\{0, \ldots, 0\}$ (a set of $d+1$ zeros). Let $\epsilon_1, \ldots, \epsilon_{d+1}$ be sufficiently small positive real numbers and

$$E := \begin{bmatrix} \epsilon_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \epsilon_{d+1} & 0 \\ 0 & \cdots & 0 & E_{\mathcal{R}} \end{bmatrix}.$$

Then all the diagonal entries of E are nonzero, and E is invertible. Thus $\mathrm{rank}(EL) = \mathrm{rank}(L) = n-d-1$ (i.e. $\ker EL = \ker L$), and there are $d+1$ zero eigenvalues of $-EL$. Moreover, since eigenvalues are continuous functions of matrix entries and $\epsilon_1, \ldots, \epsilon_{d+1}$ are sufficiently small, the rest $n-d-1$ eigenvalues of $-EL$ still have negative real parts.

Finally consider the equation (8.10): $\dot{x} = ((-EL) \otimes I_d)x$. By the property of Kronecker product, the matrix $(-EL) \otimes I_d$ has $d(d+1)$ zero eigenvalues and $d(n-d-1)$ eigenvalues with negative real parts. Hence for an arbitrary initial condition $x(0)$,

$$x(t) \to \ker\,(-EL) \otimes I_d = \ker\,(-L) \otimes I_d = \mathcal{A}(\xi)$$

as $t \to \infty$. Namely the second condition of the affine formation control problem is established. This completes the proof. □

8.4 Simulation Examples

Example 8.4 *Let us consider again Example 8.3, where the (generic) target configuration consists of eight 3-dimensional points on the unit sphere (Example 8.1). We have designed a signed Laplacian L of the digraph modeling the interconnection of the eight agents in Example 8.3. While it is satisfied that* $\ker L \otimes I_3 = \mathcal{A}(\xi)$, *two of the nonzero eigenvalues of $-L$ are unstable (i.e. with positive real parts). Thus we need to design an invertible diagonal matrix E such that all the nonzero eigenvalues of $-EL$ are stable.*
Since the target configuration ξ is generic and the digraph $\mathcal{G}$ contains a spanning 4*-tree with the root set $\mathcal{R} = \{1, 2, 3, 4\}$, all the principal minors of the submatrix $L_{\mathcal{R}}$ (with the four rows and columns corresponding to $\mathcal{R}$ removed) are nonzero. Therefore by Lemma 8.1, there exists an invertible diagonal matrix $E_{\mathcal{R}}$ such that all the eigenvalues of $-E_{\mathcal{R}}L_{\mathcal{R}}$ are stable. For computing such $E_{\mathcal{R}}$, we apply Algorithm 8.1 and obtain*

$$E_{\mathcal{R}} = \mathrm{diag}(0.5774, 2.1213, -1.2879, -4).$$

Then an invertible diagonal matrix E such that all the nonzero eigenvalues of $-EL$ are stable is:

$$E = \mathrm{diag}(1, 1, 1, 1, 0.5774, 2.1213, -1.2879, -4).$$

Indeed, the eigenvalues of $-EL$ are:

$$0, 0, 0, 0, -0.7916 + 3.1798\mathrm{j}, -0.7916 - 3.1798\mathrm{j}, -0.9167 + 0.7416\mathrm{j}, -0.9167 - 0.7416\mathrm{j}.$$

With a random initial condition $x(0) \in \mathbb{R}^{24}$ (whose entries represent eight random positions of the agents in a 3D space), a simulation of the AFCA (i.e. $\dot{x} = ((-EL) \otimes I_3)x$) yields the trajectories displayed in Fig. 8.4. It is observed that an affine formation of sphere is formed. In the figure, $\times$ denotes the initial positions of the agents, while $\circ$ the final positions. Observe that the four root agents have stayed put as their initial and final positions coincide; this is because they have no neighbors and thus have never updated their positions.

Example 8.5 *Consider a network of* 12 *agents as displayed in Fig. 8.5. This digraph $\mathcal{G}$ contains a spanning* 4*-tree, with the root set $\mathcal{R} = \{1, 2, 3, 4\}$.*

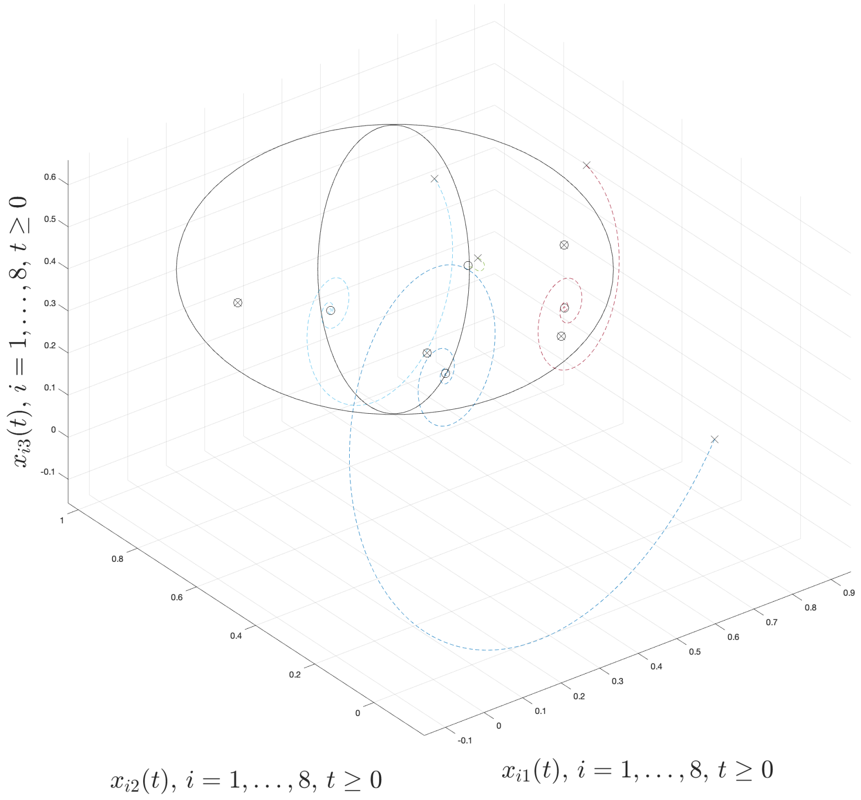


Figure 8.4: Eight agents converging to an affine formation of unit sphere (×: initial position; ∘: final position)

We consider a cuboid to be the target configuration $\xi = [\xi_1^\top \cdots \xi_{12}^\top]^\top$, *where* ξ_i *are*

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \xi_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \xi_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \xi_4 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \xi_5 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \xi_6 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

$$\xi_7 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \xi_8 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \xi_9 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \xi_{10} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \xi_{11} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \xi_{12} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}.$$

This ξ *is* not *generic, because there are multiple cases of four points on the same plane.*

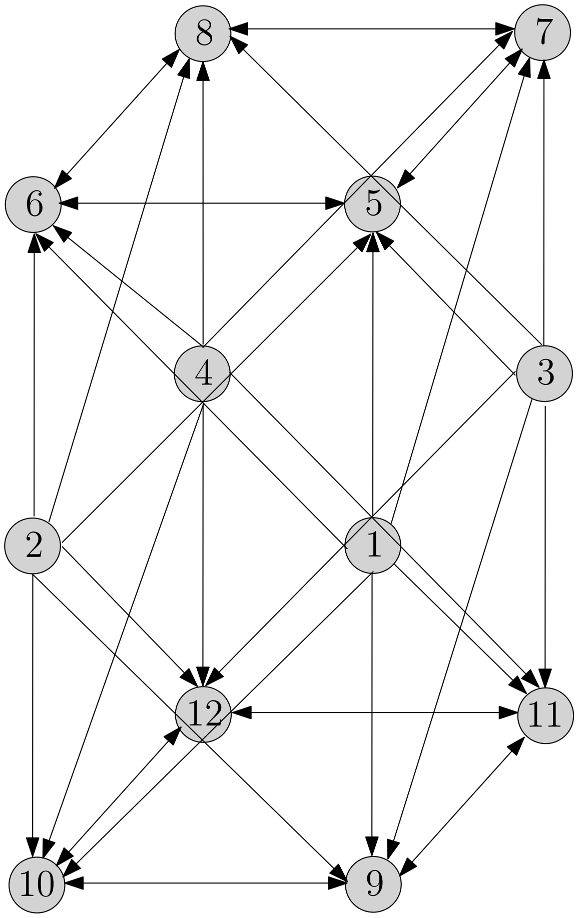


Figure 8.5: Twelve networked agents

Hence we add a random perturbation $[p_1\ \ p_2\ \ p_3]^\top$ *to each* ξ_i *(where* $p_1, p_2, p_3 \in (0, 0.1)$*. Denote the perturbed configuration by* ξ'*, which is verified to be generic.*

We then design a signed Laplacian L *of the digraph* $\mathcal{G}$ *in Fig. 8.7 such that* $\mathrm{rank}(L) = 8$*, and apply Algorithm 8.1 to compute an invertible diagonal matrix* E *such that all the nonzero eigenvalues of* $-EL$ *are stable. With a random initial condition* $x(0) \in \mathbb{R}^{36}$ *(whose entries represent twelve random positions of the agents in a 3D space), a simulation of the AFCA (i.e.* $\dot{x} = ((-EL) \otimes I_3)x$*) yields the trajectories displayed in Fig. 8.6. Observe that an (approximate) cuboid affine to the perturbed* ξ *is formed.*

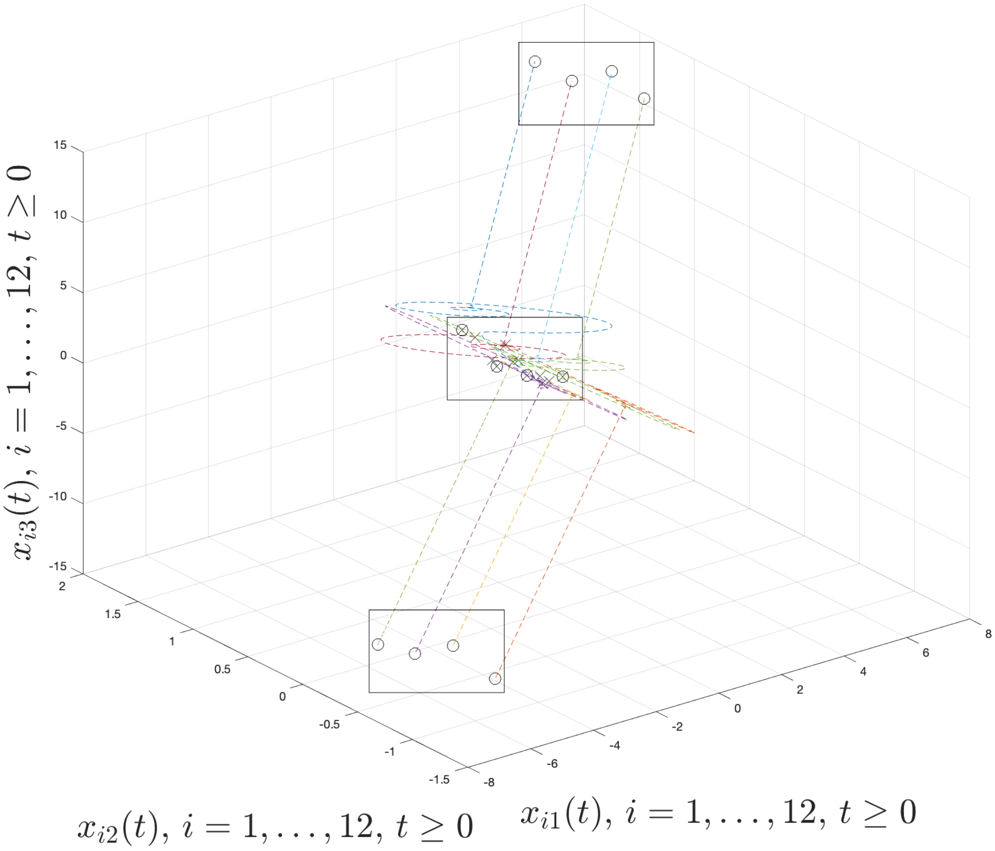


Figure 8.6: Twelve agents converging to an affine formation of cuboid (×: initial position; ○: final position)

Example 8.6 *Consider a network of* 27 *agents as displayed in Fig. 8.7. This digraph* $\mathcal{G}$ *contains a spanning* 3*-tree, with the root set* $\mathcal{R}=\{1,2,3\}$*. Note that every node has three neighbors, except for node* 2 *which has four neighbors.*

We consider a two-dimensional unit circle to be the target configuration $\xi=[\xi_1^\top\cdots\xi_{27}^\top]^\top$*, where* ξ_i *are*

$$\xi_i=\begin{bmatrix}\cos(\frac{2\pi \mathrm{j}(i-1)}{27})\\ \sin(\frac{2\pi \mathrm{j}(i-1)}{27})\end{bmatrix}\quad i\in[1,27].$$

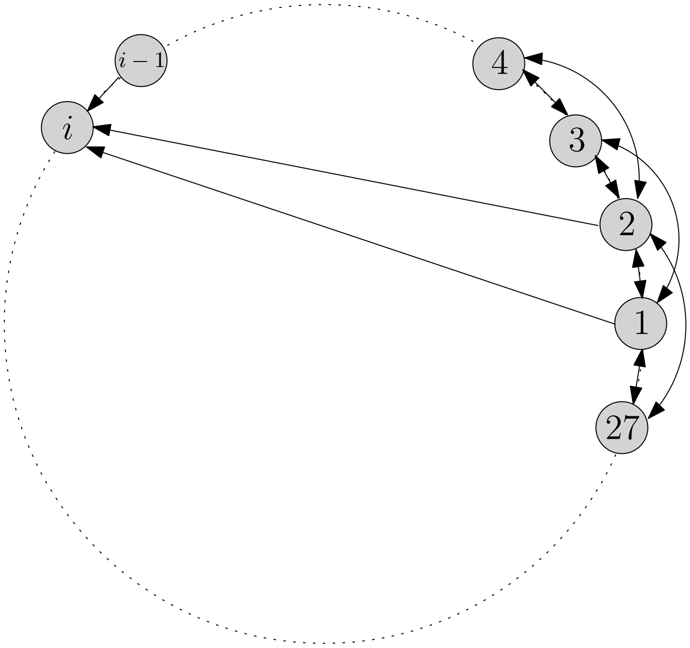


Figure 8.7: Twenty-seven networked agents (neighbor sets $\mathcal{N}_1 = \{2, 3, 27\}$, $\mathcal{N}_2 = \{1, 3, 4, 27\}$, $\mathcal{N}_3 = \{1, 2, 4\}$, $\mathcal{N}_i = \{1, 2, i-1\}, i \in [4, 27]$)

This ξ is generic. We then design a signed Laplacian L of the digraph $\mathcal{G}$ in Fig. 8.7 such that $\text{rank}(L) = 24$, and apply Algorithm 8.1 to compute an invertible diagonal matrix E such that all the nonzero eigenvalues of $-EL$ are stable. With a random initial condition $x(0) \in \mathbb{R}^{54}$ (whose entries represent twenty-seven random positions of the agents in a 2D space), a simulation of the AFCA (i.e. $\dot{x} = ((-EL) \otimes I_2)x$) yields the trajectories displayed in Fig. 8.8. Observe that an ellipsoid affine to the target circle ξ is formed. This is in contrast with the 2D similar formations in Chapter 6, because here generally different scalings are allowed along the two dimensions. Also observe that no agent stays put, as everyone has neighbors and thus updates its state correspondingly.

8.5 Notes and References

The concept of signed Laplacian and affine formation control algorithm (AFCA) are first studied in:

- Z. Lin, L. Wang, Z. Chen, M. Fu, Z. Han, Necessary and sufficient graphical conditions for

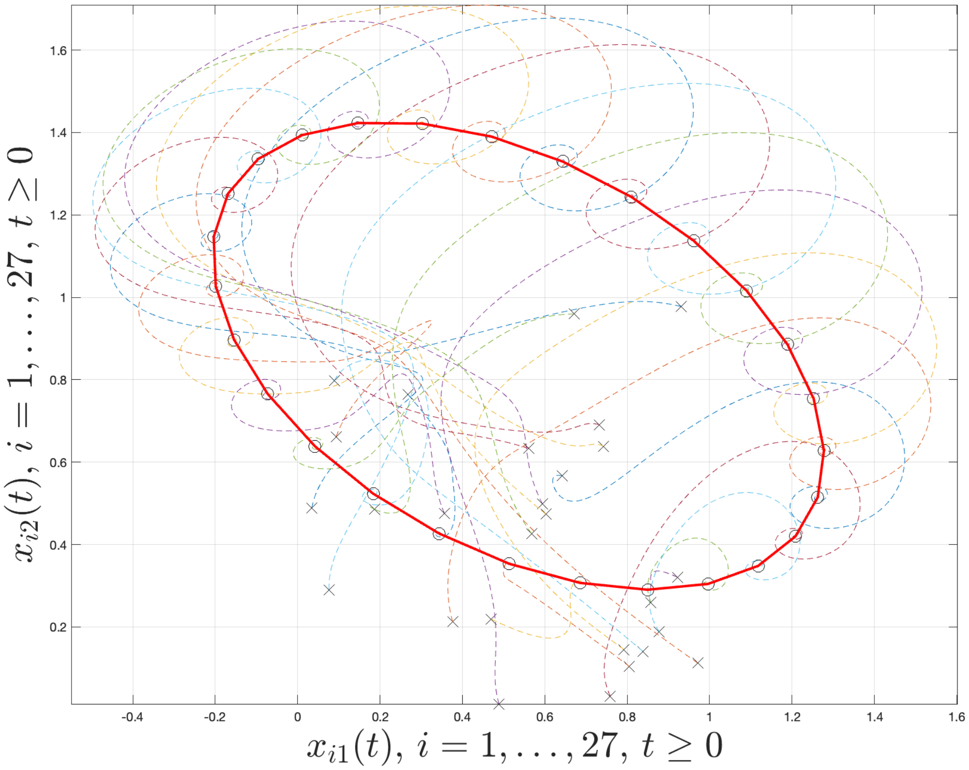


Figure 8.8: Twenty-seven agents converging to an affine formation of unit circle (×: initial position; ○: final position)

affine formation control, IEEE Transactions on Automatic Control, vol.61, pp.2877–2891, 2016

Extension to affine formation maneuver control is reported in:

- S. Zhao, Affine formation maneuver control of multiagent systems, IEEE Transactions on Automatic Control, vol.63, pp.4140–4155, 2018

Chapter 9

Localization in Arbitrary Dimensional Space

In this chapter, we extend the distributed localization problem of multi-agent systems in Chapter 6 from two-dimensional space to arbitrary dimensional space. This extension is practically useful because many applications of localization using (wireless) sensor networks are not limited in a 2D plane. For example, air quality monitoring and underwater information collection are instances in 3D space.

To solve localization in arbitrary dimensions, we develop an approach based on signed Laplacian matrices (as in Chapter 8 for arbitrary dimensional affine formation control). Note that the approach for solving localization in Chapter 6 based on complex Laplacian matrices was limited to 2D space, and cannot be used for higher dimensional localization.

We nevertheless adopt the same *distributed* localization scheme introduced in Chapter 6. Namely we consider a sensor network composed of a minority of *anchor* nodes that know their positions in the global reference frame (e.g. using a GPS), and the rest majority of *free* nodes that need to determine their global positions based on their local frames and locally sensed information (e.g. distances and bearing angles with respect to neighboring nodes).

Modeling the interacting sensor nodes by digraphs, we show that a necessary graphical condition to achieve d-dimensional localization ($d \geq 2$) is that the digraph contains a *spanning* $(d+1)$*-tree* whose $d+1$ roots are anchor nodes. This condition is the same as the one for achieving d-dimensional affine formation in Chapter 8. However, in the special case of $d = 2$, this condition differs from the one (i.e. spanning 2-tree) for achieving 2D localization in Chapter 7. This difference is due to distinct graphical requirements on designing appropriate signed and complex Laplacian matrices. Under the above graphical condition, we present a distributed algorithm to achieve localization in arbitrary dimensions.

9.1 Problem Formulation

Consider a network of n (> 1) agents that are stationary in d-dimensional space ($d \geq 2$), and a global reference frame Σ which is unknown to the agents. The agents labeled $1, \ldots, d+1$ (renumbering if necessary) are the *anchor agents*, whose positions $\xi_1, \dots, \xi_d \in \mathbb{R}^{d+1}$ in Σ are known. The rest agents labeled $d+2, \dots, n$ are the *free agents*, whose positions $\xi_{d+2}, \dots, \xi_n \in \mathbb{R}^d$ in Σ are unknown and need to be determined by these individual free agents. Let

$$\xi_a := \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_{d+1} \end{bmatrix} \in \mathbb{R}^{(d+1)d}, \quad \xi_f := \begin{bmatrix} \xi_{d+2} \\ \vdots \\ \xi_n \end{bmatrix} \in \mathbb{R}^{(n-d-1)d}$$

be the aggregated positions of the anchor and free agents, respectively. Write ξ in terms of ξ_a and ξ_f as follows:

$$\xi = \begin{bmatrix} \xi_a \\ \xi_f \end{bmatrix} \in \mathbb{R}^{nd}$$

and call ξ the *configuration* of the agents.

To determine its own position, each free agent i ($\in [d+2, n]$) is equipped with a *state* variable $x_i(k) \in \mathbb{R}^d$, which is a d-dimensional real vector and denotes the *estimate* of agent i's position ξ_i under the global frame Σ. The time $k \geq 0$ is a nonnegative integer and denotes the *discrete* time. Let

$$x_f(k) := \begin{bmatrix} x_{d+2}(k) \\ \vdots \\ x_n(k) \end{bmatrix} \in \mathbb{R}^{(n-d-1)d}$$

be the aggregated state of the free agents at time k. It is desired that

$$x_f(k) \to \xi_f \text{ as } k \to \infty.$$

For convenience, also let

$$x_a(k) := \begin{bmatrix} x_1(k) \\ \vdots \\ x_{d+1}(k) \end{bmatrix} \in \mathbb{R}^{(d+1)d}$$

be the aggregated state vector of the anchor agents, such that $x_a(k) = \xi_a$ for all $k \geq 0$ (i.e. the anchor agents know their positions in the global frame Σ from the initial time $k = 0$ and never upddatte their estimates). Write $x(k) := [x_a(k)^\top \; x_f(k)^\top]^\top \in \mathbb{R}^{nd}$. Hence the aim of d-dimensional localization is to achieve

$$\lim_{k\to\infty} x(k) = \xi.$$

We model the interconnection structure of the networked agents by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Each *node* in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each directed *edge* (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent i can obtain the relative state information from agent j. The *neighbor set* of agent i is $\mathcal{N}_i := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. For the $d+1$ anchor nodes (numbered $1, \ldots, d+1$), since they do not update their states, even if they had neighbors, the corresponding incoming edges would be associated with weight 0. This is equivalent to considering that the anchor nodes do not have neighbors. For this reason, henceforth in this chapter we consider that $\mathcal{N}_i = \emptyset$ for all $i \in [1, d+1]$.

Moreover, consider that digraph $\mathcal{G}$ is weighted: each edge $(j, i) \in \mathcal{V}$ is associated with a real-valued weight $a_{ij} \in \mathbb{R}$. Hence the adjacency matrix $A = (a_{ij})$, degree matrix $D = \mathrm{diag}(A\mathbf{1})$, and Laplacian matrix $L = D - A$ are all real matrices. Note that the adjacency matrix A is not a nonnegative matrix in general; thus L is a *signed Laplacian*. Since $\mathcal{N}_i = \emptyset$ for the anchor nodes $i \in [1, d+1]$, the signed Laplacian matrix L has the following structure:

$$L = \begin{bmatrix} L_{aa} & L_{af} \\ L_{fa} & L_{ff} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix}. \tag{9.1}$$

Here $L_{fa} \in \mathbb{R}^{(n-d-1)\times(d+1)}$ and $L_{ff} \in \mathbb{R}^{(n-d-1)\times(n-d-1)}$.

To achieve localization in d dimensions, consider the distributed control

$$u_i(k) = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j(k) - x_i(k)), \quad i \in [1, n]. \tag{9.2}$$

Here the control gain w_{ij} satisfies

$$\text{(i)} \sum_{j\in\mathcal{N}_i} w_{ij}(\xi_j - \xi_i) = 0 \tag{9.3}$$

$$\text{(ii)}\ w_{ij} = \epsilon_i a_{ij}, \quad \epsilon_i \in \mathbb{R}, \epsilon_i \neq 0. \tag{9.4}$$

This control (9.2) is in the same form as that for the 2D localization in Chapter 7: the gains w_{ij} are not simply the edge weights $a_{ij} \in \mathbb{R}$, but are real multiples of a_{ij} (9.4) and satisfy linear constraints with respect to the target configuration ξ (9.3). In contrast with Chapter 7, on the other hand, here the gains w_{ij} are real numbers rather than complex ones.

Moreover, substituting (9.4) into (9.3) and removing the common multiple ϵ_i yield

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0. \tag{9.5}$$

This in matrix form is $(L \otimes I_d)\xi = 0$. In view of (9.1) we have

$$\begin{bmatrix} 0 & 0 \\ L_{fa} \otimes I_d & L_{ff} \otimes I_d \end{bmatrix} \begin{bmatrix} \xi_a \\ \xi_f \end{bmatrix} = 0$$

Hence the following equation ensues:

$$(L_{ff} \otimes I_d)\xi_f = -(L_{fa} \otimes I_d)\xi_a \tag{9.6}$$

which relates the configuration of the free agents to that of the anchor agents through appropriate multiplications of submatrices of the signed Laplacian.

Arbitrary Dimensional Localization Problem:

Consider a network of agents (stationary in d-dimensional space) interconnected through a digraph and a configuration $\xi := [\xi_a^\top \ \xi_f^\top]^\top \in \mathbb{R}^{nd}$, which represents the fixed positions of the agents under the global reference frame Σ. Here $\xi_a \in \mathbb{R}^{(d+1)d}$ is known but $\xi_f \in \mathbb{R}^{(n-d-1)d}$ is unknown. Design a distributed algorithm using the control in (9.2) such that

(i) $\text{rank}(L) = n - d - 1$

(ii) $(\forall x_f(0) \in \mathbb{R}^{(n-d-1)d}) \lim_{k \to \infty} x_f(k) = \xi_f$.

The first requirement (i) implies $\text{rank}(L_{ff}) = n - d - 1$; namely L_{ff} is invertible. This implies that $(L_{ff} \otimes I_d)$ is also invertible. Thus it follows from (9.6) that $\xi_f = -(L_{ff} \otimes I_d)^{-1}(L_{fa} \otimes I_d)\xi_a$. Hence the second requirement (ii) becomes:

$$(\forall x_f(0) \in \mathbb{R}^{(n-d-1)d}) \lim_{k \to \infty} x_f(k) = -(L_{ff} \otimes I_d)^{-1}(L_{fa} \otimes I_d)\xi_a.$$

Example 9.1 *We provide an example to illustrate the localization problem in $d(=3)$ dimensions. As displayed in Fig. 9.1, eight agents are interconnected through a digraph; agents 1,2,3,4 are anchor agents while the rest five are free nodes. The neighbor sets of the agents are $\mathcal{N}_1 = \mathcal{N}_2 = \mathcal{N}_3 = \mathcal{N}_4 = \emptyset$, $\mathcal{N}_5 = \{1, 2, 6, 7\}$, $\mathcal{N}_6 = \{3, 4, 7, 8\}$, $\mathcal{N}_7 = \{1, 5, 6, 8\}$, and $\mathcal{N}_8 = \{4, 5, 6, 7\}$.*

Let the configuration $\xi = [\xi_1^\top \ \cdots \ \xi_8^\top]$ of the agents be the vector of eight (three-dimensional)

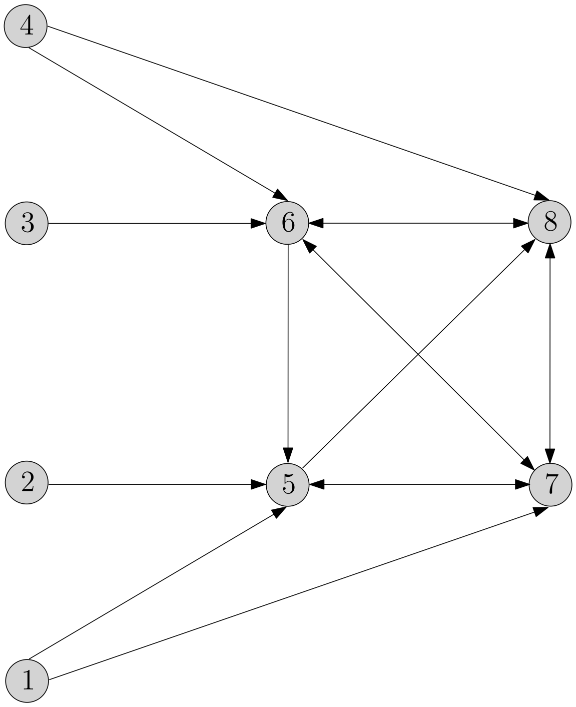


Figure 9.1: Illustrating example of eight agents

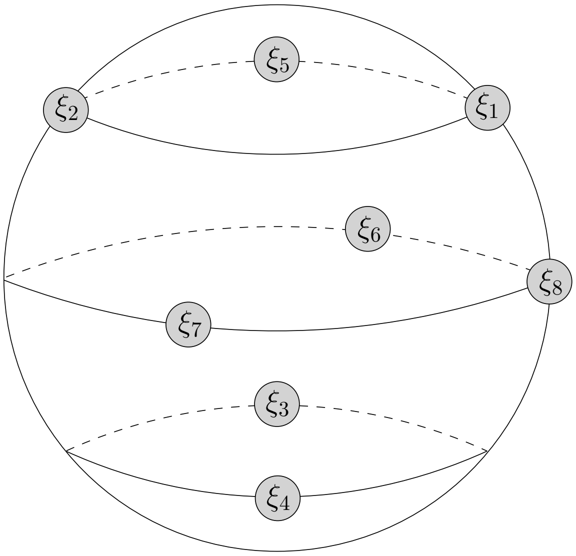


Figure 9.2: Illustrating example of a configuration of eight 3D points on the unit sphere

points on the unit sphere (refer to Fig. 9.2), where

$$\xi_1 = \begin{bmatrix} \cos\frac{\pi}{4} \\ 0 \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_2 = \begin{bmatrix} -\cos\frac{\pi}{4} \\ 0 \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_3 = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix}, \xi_4 = \begin{bmatrix} 0 \\ \cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix},$$
$$\xi_5 = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_6 = \begin{bmatrix} \cos\frac{\pi}{3} \\ -\sin\frac{\pi}{3} \\ 0 \end{bmatrix}, \xi_7 = \begin{bmatrix} -\cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} \\ 0 \end{bmatrix}, \xi_8 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The position vector of the anchor agents $\xi_a = [\xi_1^\top \ \xi_2^\top \ \xi_3^\top \ \xi_4^\top]^\top$ is known, and that of the free agents $\xi_f = [\xi_5^\top \ \xi_6^\top \ \xi_7^\top \ \xi_8^\top]^\top$ is unknown and needs to determined.
The localization problem in 3D is to design a distributed algorithm using the control in (9.2) such that the rank of the signed Laplacian L be $n-4$, and moreover the free agents' state vector asymptotically converges to ξ_f.

A necessary graphical condition for solving the d-dimensional localization problem is given below.

Proposition 9.1 *Suppose that there exists a distributed control in (9.2) that solves the d-dimensional localization problem. Then the digraph contains a spanning $(d+1)$-tree whose $d+1$ roots are the $d+1$ anchor agents.*

Proof. Suppose that there exists a distributed control in (9.2) that solves the d-dimensional localization problem, but that the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ does *not* contain a spanning $(d+1)$-tree whose $d+1$ roots are the $d+1$ anchor agents. We will derive a contradiction that $\text{rank}(L) < n-d-1$, thereby proving that after all $\mathcal{G}$ must contain a spanning $(d+1)$-tree whose $d+1$ roots are the $d+1$ anchor agents.

There are two cases that need to be considered separately. First, the digraph contains a spanning $(d+1)$-tree but at least one of the $d+1$ roots is a free agent. In this case, the subdigraph of free agents contains at least a spanning tree (and at most a spanning $(d+1)$-tree). Hence $\text{rank}(L_{ff}) < n-d-1$. Since the anchor agents do not have neighbors, $\text{rank}(L) < n-d-1$.

The second case is that the digraph does not contain a spanning $(d+1)$-tree. Then it follows similarly to the proof of Proposition 8.1 that $\text{rank}(L) < n-d-1$.

Therefore in both cases above, a contradiction is derived to the solvability of the d-dimensional localization problem. The proof is now complete. □

Owing to Proposition 9.1, we shall henceforth assume the following graphical condition.

Assumption 9.1 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning $(d+1)$-tree whose $d+1$ roots are the $d+1$ anchor agents.*

Even if Assumption 9.1 holds, not every configuration $\xi \in \mathbb{R}^{nd}$ may be determined by a distributed control in (9.2). Similar to Example 8.2, if ξ is not generic, it is possible that $\text{rank}(L) < n-d-1$ for all signed Laplacian matrices satisfying $(L \otimes I_d)\xi = 0$. This means that the d-dimensional localization problem is not solvable. For this reason, and also the fact that the set of all non-generic configurations has Lebesgue measure zero after all, we assume that the configuration ξ is generic.

Assumption 9.2 *The configuration $\xi = [\xi_a^\top \ \xi_f^\top]^\top \in \mathbb{R}^{nd}$ is generic.*

9.2 Distributed Algorithm

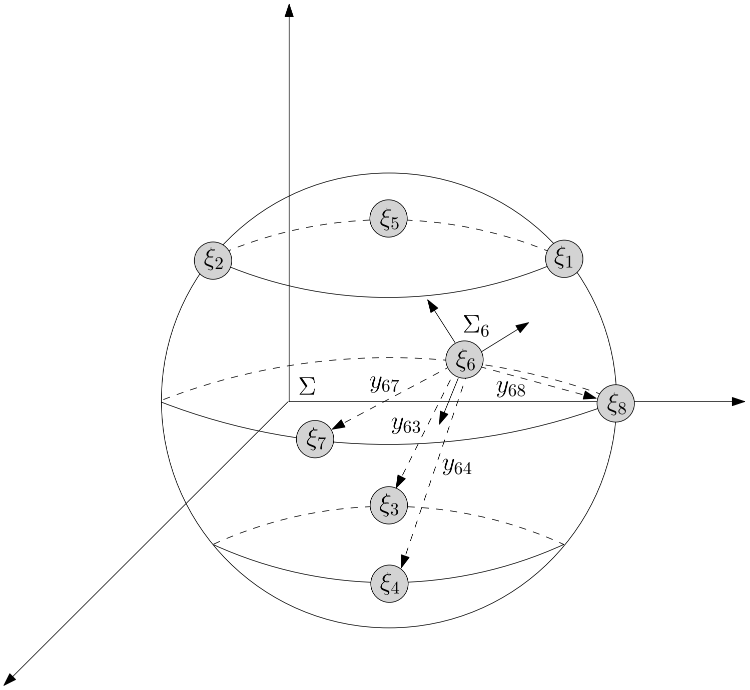


Figure 9.3: Illustration of design of real weights

Example 9.2 *Consider again Example 9.1, where the configuration $\xi = [\xi_1^\top \ \cdots \ \xi_8^\top]^\top$ of* *the agents consists of eight (three-dimensional) points on the unit sphere:*

$$\xi_1 = \begin{bmatrix} \cos\frac{\pi}{4} \\ 0 \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_2 = \begin{bmatrix} -\cos\frac{\pi}{4} \\ 0 \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_3 = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix}, \xi_4 = \begin{bmatrix} 0 \\ \cos\frac{\pi}{4} \\ -\sin\frac{\pi}{4} \end{bmatrix},$$
$$\xi_5 = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \end{bmatrix}, \xi_6 = \begin{bmatrix} \cos\frac{\pi}{3} \\ -\sin\frac{\pi}{3} \\ 0 \end{bmatrix}, \xi_7 = \begin{bmatrix} -\cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} \\ 0 \end{bmatrix}, \xi_8 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

This configuration ξ is generic.
The anchor agents' configuration $\xi_a = [\xi_1^\top \ \xi_2^\top \ \xi_3^\top \ \xi_4^\top]^\top$ is known, and the free agents' configuration $\xi_f = [\xi_5^\top \ \xi_6^\top \ \xi_7^\top \ \xi_8^\top]^\top$ is to be determined. To this end, we consider using the simplest form of distributed control (9.2) by setting all $\epsilon_i = 1$:

$$x_i(k+1) = x_i(k) + \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [1, 8] \tag{9.7}$$

where $a_{ij} \in \mathbb{R}$ are real weights to be designed to satisfy (9.5):

$$\sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) = 0, \quad i \in [1, 8].$$

In the following we illustrate how the real weights may be designed locally to satisfy the above linear constraints. Each free agent $i \in [5, 8]$ has a local reference frame Σ_i, whose origin is the (stationary) position of agent i. The orientation of Σ_i is fixed, but the three offset angles $\alpha_i, \beta_i, \gamma_i$ (counterclockwise) with respect to the global reference frame Σ are unknown. These offset angles give rise to a (fixed) rotation matrix R_i relating the local frame Σ_i to the global Σ. For each neighbor (free or anchor) $j \in \mathcal{N}_i$, we assume that agent i can measure the relative position y_{ij} in Σ_i as

$$y_{ij} := R_i(\xi_j - \xi_i). \tag{9.8}$$

Since R_i is unknown, even though the relative position y_{ij} in Σ_i is known, $\xi_j - \xi_i$ in Σ is unknown. Substituting $\xi_j - \xi_i = R_i^{-1} y_{ij}$ into (9.5) and multiplying R_i from the left, we derive

$$\sum_{j \in \mathcal{N}_i} a_{ij} y_{ij} = 0. \tag{9.9}$$

Hence the weights a_{ij} may be designed based on the relative position y_{ij} under the local reference frame Σ_i.

For example, Fig. 9.3 provides an illustrative example. For agent 6, it has four neighbors $3, 4, 7, 8$. Thus we must find weights $a_{63}, a_{64}, a_{67}, a_{68}$ such that

$$a_{63}y_{63} + a_{64}y_{64} + a_{67}y_{67} + a_{68}y_{68} = 0.$$

The relative positions, as displayed in Fig. 9.3, are

$$y_{63} = \begin{bmatrix} 0 \\ -\cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \end{bmatrix}, y_{64} = \begin{bmatrix} \cos\frac{\pi}{3} \\ -\sin\frac{\pi}{3} \\ 0 \end{bmatrix}, y_{67} = \begin{bmatrix} -\cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} \\ 0 \end{bmatrix}, y_{68} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The local frame Σ_6 has (fixed) offset angles from the global Σ: $\alpha_6 = \frac{\pi}{4}$, $\beta_6 = \frac{\pi}{6}$, and $\gamma_6 = \frac{\pi}{3}$ (all counterclockwise with respect to Σ). Then the corresponding rotation matrix is

$$R_6 = \begin{bmatrix} \cos(\frac{\pi}{3}) & -\sin(\frac{\pi}{3}) & 0 \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\frac{\pi}{6}) & 0 & \sin(\frac{\pi}{6}) & 0 \\ 0 & 1 & 0 \\ -\sin(\frac{\pi}{6}) & 0 & \cos(\frac{\pi}{6}) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\frac{\pi}{4}) & -\sin(\frac{\pi}{4}) \\ 0 & \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \end{bmatrix}.$$

It is verified that

$$y_{6j} = R_6(\xi_j - \xi_6), \quad j = 3, 4, 6, 7.$$

Substituting the relative positions $y_{63}, y_{64}, y_{67}, y_{68}$ into the equation $a_{63}y_{63} + a_{64}y_{64} + a_{67}y_{67} + a_{68}y_{68} = 0$ yields

$$a_{63} \begin{bmatrix} -0.8437 \\ -0.2367 \\ -0.0857 \end{bmatrix} + a_{64} \begin{bmatrix} -1.4598 \\ 0.6964 \\ 0.7803 \end{bmatrix} + a_{67} \begin{bmatrix} -1.1875 \\ 0.3927 \\ 1.5607 \end{bmatrix} + a_{68} \begin{bmatrix} -0.1607 \\ 0.9464 \\ 0.2803 \end{bmatrix} = 0.$$

The above reduces to a system of linear equations, with four unknowns (the weights) and three equations. Thus there are infinitely many solutions (indeed the solution space is one dimensional). One solution is $a_{63} = -1, a_{64} = 1, a_{67} = -0.4082, a_{68} = -0.8165$.

Similarly we design other real weights to satisfy (9.9), and write (9.7) in vector form:

$x(k+1) = ((I - L) \otimes I_3)x(k)$ *where*

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -3.7321 & 0 & 0 & 4.7321 & -1.9319 & 1.9319 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1.2247 & -0.4082 & -0.8165 \\ -1 & 0 & 0 & 0 & 1 & -0.9659 & -0.1494 & 1.1154 \\ 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 \end{bmatrix}.$$

It is verified that the signed Laplacian matrix L *has zero row sums and satisfies* $(L \otimes I_3)\xi = 0$. *Moreover, partition the matrix* L *according to anchor agents and free agents:*

$$L = \begin{bmatrix} L_{aa} & L_{af} \\ L_{fa} & L_{ff} \end{bmatrix}.$$

Thus $L_{aa} = L_{af} = 0$; $L_{fa} \in \mathbb{R}^{4\times 4}$ *and* $L_{ff} \in \mathbb{R}^{4\times 4}$. *It is checked that rank*$(L_{ff}) = 4$, *and thus* L_{ff} *and* $(L_{ff} \otimes I_3)$ *are invertible. Therefore the first condition in the arbitrary dimensional localization problem is satisfied.*

It is left to verify the second condition that the state vector of the free agents $x_f(k)$ *converges to* $-(L_{ff} \otimes I_3)^{-1}(L_{fa} \otimes I_3)\xi_a$ *(when* $x_a(k) = \xi_a$ *for all* $k \geq 0$*). Fix* $\xi_a \in \mathbb{R}^{12}$. *First note that*

$$\bar{x} = \begin{bmatrix} \bar{x}_a \\ \bar{x}_f \end{bmatrix} = \begin{bmatrix} \xi_a \\ -(L_{ff} \otimes I_3)^{-1}(L_{fa} \otimes I_3)\xi_a \end{bmatrix}$$

is the unique fixed point of (9.7). To see this, substituting $\bar{x}$ *into (9.7) yields* $\bar{x}$, *which means that* $\bar{x}$ *is a fixed point of (9.7). Moreover, let*

$$\bar{x}' = \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}$$

be another fixed point of (9.7), namely

$$\begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \left(\left(\begin{bmatrix} I_4 & 0 \\ 0 & I_4 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix} \right) \otimes I_3 \right) \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \begin{bmatrix} I_4 & 0 \\ -L_{fa} & I_4 - L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}.$$

From the above we derive

$$\bar{x}'_f = -(L_{ff} \otimes I_3)^{-1}(L_{fa} \otimes I_3)\xi_a = \bar{x}_f.$$

This shows that $\bar{x}$ is the unique fixed point of (9.7), which in turn implies that starting from an arbitrary initial condition $x(0) = [\xi_a^\top \ x_f^\top(0)]^\top \in \mathbb{R}^{24}$, $x_f(k)$ converges to $-(L_{ff} \otimes I_3)^{-1}(L_{fa} \otimes I_3)\xi_a$ if and only if all the eigenvalues of $I_4 - L_{ff}$ lie inside the unit circle. Unfortunately, the eigenvalues of matrix $I_4 - L_{ff}$ are

$$-0.0967 + 0.2167\mathrm{j}, -0.0967 - 0.2167\mathrm{j}, 2.3807, -3.9946.$$

The last two eigenvalues lie outside of the unit circle. Hence (9.7) is unstable and $x_f(k)$ diverges. To stabilize $x_f(k)$ to the desired fixed point $-(L_{ff} \otimes I_3)^{-1}(L_{fa} \otimes I_3)\xi_a$ (to satisfy the second requirement in the arbitrary dimensional localization problem), the unstable eigenvalues of $I_4 - L_{ff}$ must be moved inside the unit circle. This shows that simply setting all $\epsilon_i = 1$ in (9.2) does not work in general. In fact, ϵ_i need to be properly chosen in order to stabilize $I_4 - L_{ff}$.

Remark 9.1 *As illustrated in Example 9.2 for 3D localization, it is important for each agent to have at least four neighbors to guarantee existence of (infinitely many) appropriate weights a_{ij} such that the signed Laplacian L satisfies $(L \otimes I_3)\xi = 0$. If an agent only had three or fewer neighbors, appropriate weight a_{aj} need not exist in general. This is why for general d-dimensional localization, the digraph needs to contain a spanning $(d+1)$-tree. Specializing to the case of $d = 2$, we need a digraph containing a spanning 3-tree for solving 2D localization based on signed Laplacian. This is in contrast with the result of Chapter 7: based on complex Laplacian, 2D localization is solvable over a digraph containing a spanning 2-tree.*

In the following we describe a distributed algorithm using (9.2) in vector form, and will analyze its stability in relation to the values of ϵ_i in the next section.

Arbitrary Dimensional Localization Algorithm (ADLA):

Each anchor agent $i \in [1, \ldots, d+1]$ has a state variable $x_i(k) \in \mathbb{R}^d$ whose initial value is set to be $x_i(0) = \xi_i$ (which is known). Every free agent $i \in [d+2, \ldots, n]$ also has a state variable $x_i(k) \in \mathbb{R}^d$ whose initial value is an arbitrary d dimensional real vector. Offline, each free agent i computes weights $a_{ij} \in \mathbb{R}$ based on the measured relative positions $y_{ij} = R_i(\xi_j - \xi_i)$ in (9.8) by solving

$$\sum_{j\in\mathcal{N}_i} a_{ij}y_{ij} = 0.$$

Then online, at each time $k \geq 0$, while each anchor agent stays put, i.e.

$$x_i(k+1) = x_i(k), \quad i \in [1, d+1]$$

each free agent i updates its $x_i(k)$ using the following local update protocol:

$$x_i(k+1) = x_i(k) + \epsilon_i \sum_{j\in\mathcal{N}_i} a_{ij}(x_j(k) - x_i(k)), \quad i \in [d+2, n] \tag{9.10}$$

where $\epsilon_i \in \mathbb{R} \setminus \{0\}$ is a (nonzero) real control gain.

Let $x := [x_1^\top \ \cdots \ x_n^\top]^\top$ be the aggregated state of the networked agents, and

$$E = \text{diag}(\epsilon_1, \ldots, \epsilon_n)$$

the (invertible diagonal) control gain matrix. Then the n equations (9.10) become

$$x(k+1) = ((I - EL) \otimes I_d)x(k). \tag{9.11}$$

9.3 Convergence Result

The following is the main result of this section.

Theorem 9.1 *Suppose that Assumptions 9.1 and 9.2 hold. There exists an invertible diagonal control gain matrix $E = \text{diag}(\epsilon_1, \ldots, \epsilon_n)$ such that the ADLA solves the arbitrary dimensional localization problem.*

To prove Theorem 9.1, we will analyze the eigenvalues of the matrix $(I - EL) \otimes I_d$ in (9.11). For this, the following fact is useful (which is the real counterpart of Lemma 7.1 and the discrete counterpart of Lemma 8.1).

Lemma 9.1 *Consider an arbitrary square real matrix $M \in \mathbb{R}^{n\times n}$. If all the principal minors of M are nonzero, then there exists an invertible diagonal matrix $E = \text{diag}(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{R}^{n\times n}$ such that all the eigenvalues of $I - EM$ lie inside the unit circle.*

Proof: The proof is based on induction on n. For the base case $n = 1$, $M = m_{11}$ is a nonzero real scalar (as the principal minor of M is nonzero). Let $\epsilon_1 \in \mathbb{R}$ be such that $\epsilon_1 \in (0, \frac{1}{m_{11}})$. Then $EM = \epsilon_1 m_{11} \in (0,1)$. Hence $1 - EM \in (0,1)$, which lies inside the unit circle.

For the induction step, suppose that the conclusion holds for $M \in \mathbb{R}^{(n-1)\times(n-1)}$. Now consider $M \in \mathbb{R}^{n\times n}$, with all of its principal minors nonzero. Let M_1 be the submatrix of M with the last row and last column removed. Then all the principal minors of M_1 are nonzero, and by the hypothesis there exists an invertible diagonal matrix $E_1 = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_{n-1})$ such that all the eigenvalues $1-\lambda_1, \ldots, 1-\lambda_{n-1}$ of $I - E_1 M_1$ lie inside the unit circle. Now write

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix}$$

where m_{nn} is a nonzero scalar (since all the principal minors of M are nonzero). Also let

$$E = \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix}$$

for some real ϵ_n. Thus

$$I - EM = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} E_1 & 0 \\ 0 & \epsilon_n \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & m_{nn} \end{bmatrix} = \begin{bmatrix} I - E_1 M_1 & -E_1 M_2 \\ -\epsilon_n M_3 & 1 - \epsilon_n m_{nn} \end{bmatrix}$$

If $\epsilon_n = 0$, then

$$I - EM = \begin{bmatrix} I - E_1 M_1 & -E_1 M_2 \\ 0 & 1 \end{bmatrix}$$

which means that all the eigenvalues of $I - EM$ lie inside the unit circle except for a simple eigenvalue 1. Since eigenvalues are continuous functions of matrix entries, for sufficiently small $|\epsilon_n|$, $I - EM$ still has $n-1$ eigenvalues $1-\lambda'_1, \ldots, 1-\lambda'_{n-1}$ which are inside the unit circle.

Now we consider the last eigenvalue $1-\lambda'_n$. If $1-\lambda'_n$ is complex, then it must be a conjugate to an existing eigenvalue inside the unit circle. Hence $1-\lambda'_n$ is also inside the unit circle. If $1-\lambda'_n$ is real, it follows from Lemma 8.1 that ϵ_n may be chosen such that λ'_n is a sufficiently small positive number. Hence the last eigenvalue $1-\lambda'_n$ lies within the unit circle. This proves the induction step, and thereby completes the proof. □

The above proof suggests an algorithm (Algorithm 9.1 below) to compute an invertible diagonal matrix $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_n)$ such that all the eigenvalues $I - EM$ lie inside the unit circle. This algorithm is identical to Algorithm 8.1 in Chapter 8, because appropriate $\delta_1, \ldots, \delta_n$ in line 1 can always be chosen to render the eigenvalues of EM with sufficiently small positive real parts, which in turn ensures that the eigenvalues of $I - EM$ lie inside the unit circle.

Algorithm 9.1 Diagonal Stabilization Algorithm (case of real matrix, inside unit circle)

Input: square real matrix $M \in \mathbb{R}^{n\times n}$ with nonzero principal minors
Output: invertible diagonal matrix $E \in \mathbb{R}^{n\times n}$

```
set $\delta_1, \dots, \delta_n$ to be small positive real numbers
$\epsilon_1 = \frac{\delta_1}{M(1,1)}$
$E_1 = \mathrm{diag}(\epsilon_1)$
for $i = 2, \dots, n$ do
    $\epsilon_i = \frac{\delta_i}{\det(E_{i-1})\det(M(1{:}i,1{:}i))}$
    $E_i = \mathrm{diag}(\epsilon_1, \dots, \epsilon_i)$
end for
$E = \mathrm{diag}(\epsilon_1, \dots, \epsilon_n)$
```

Lemma 9.1 provides a sufficient condition under which the eigenvalues of a real matrix may be moved inside the unit circle using an invertible diagonal real matrix. It then follows from Proposition 8.2 (recalled below for convenience) that under Assumptions 9.1 and 9.2 (Assumption 9.1 implies Assumption 8.1 and Assumption 9.2 is the same as Assumption 8.2), the sufficient condition holds for the submatrix L_{ff} of the signed Laplacian L. Hence there exists an invertible diagonal matrix $E_f = \mathrm{diag}(\epsilon_{d+2}, \dots, \epsilon_n)$ such that all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle.

Proposition 8.2 *Suppose that Assumptions 9.1 and 9.2 hold. Let $\mathcal{R}$ be the set of $d+1$ roots and $L_{\mathcal{R}}$ the submatrix of L by removing the $d+1$ rows and $d+1$ columns corresponding to $\mathcal{R}$. Then for almost all signed Laplacian L satisfying $(L \otimes I_d)\xi = 0$, all principal minors of $L_{\mathcal{R}}$ are nonzero.*

With this preparation, we are ready to prove Theorem 9.1.

Proof of Theorem 9.1: Let Assumptions 9.1 and 9.2 hold. On one hand, it follows from Proposition 8.2 that for almost all signed Laplacian L of $\mathcal{G}$ satisfying $(L \otimes I_d)\xi = 0$ (where ξ is generic), $\mathrm{rank}(L) \geq n - d - 1$. On the other hand, since the first $d+1$ rows of L corresponding to the $d+1$ anchor agents are zero, we have $\mathrm{rank}(L) \leq n - d - 1$. Therefore for almost all signed Laplacian L satisfying $(L \otimes I_d)\xi = 0$, we have $\mathrm{rank}(L) = n - d - 1$, which establishes the first condition in the arbitrary dimensional localization problem.

For the second condition, first note again from Proposition 8.2 that for almost all signed Laplacian L satisfying $(L \otimes I_d)\xi = 0$, all principal minors of L_{ff} are nonzero. It then follows from Lemma 9.1 that there exists an invertible diagonal matrix $E_f = \mathrm{diag}(\epsilon_{d+2}, \dots, \epsilon_n)$ such that all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle. Let

$$E_a := \begin{bmatrix} \epsilon_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_{d+1} \end{bmatrix}, \quad E := \begin{bmatrix} E_a & 0 \\ 0 & E_f \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 \\ L_{fa} & L_{ff} \end{bmatrix}.$$

Here $\epsilon_1, \ldots, \epsilon_{d+1} \neq 0$. Then E is invertible and

$$I - EL = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ E_f L_{fa} & E_f L_{ff} \end{bmatrix} = \begin{bmatrix} I & 0 \\ -E_f L_{fa} & I - E_f L_{ff} \end{bmatrix}.$$

Hence the spectrum (i.e. set of eigenvalues) of $I - EL$ is the union of the spectrum of $I - E_f L_{ff}$ (all inside the unit circle) and $\{1, \ldots, 1\}$ (set of $d+1$ ones).

It is left to verify that for arbitrary initial states of the free agents $x_f(0) \in \mathbb{R}^{(n-d-1)d}$, $x_f(k)$ converges to $-(L_{ff} \otimes I_d)^{-1}(L_{fa} \otimes I_d)\xi_a (= \xi_f)$ when $x_a(k) = \xi_a$ for all $k \geq 0$. Fix $\xi_a \in \mathbb{R}^{(d+1)d}$. First note that

$$\bar{x} = \begin{bmatrix} \bar{x}_a \\ \bar{x}_f \end{bmatrix} = \begin{bmatrix} \xi_a \\ -(L_{ff} \otimes I_d)^{-1}(L_{fa} \otimes I_d)\xi_a \end{bmatrix}$$

is the unique fixed point of (9.11). To see this, substituting $\bar{x}$ into (9.11) yields $\bar{x}$ (thanks to the fact that both E_f and L_{ff} are invertible), which means that $\bar{x}$ is a fixed point of (9.11). Moreover, let

$$\bar{x}' = \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}$$

be another fixed point of (9.11), namely

$$\begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix} = \begin{bmatrix} I & 0 \\ -E_f L_{fa} & I - E_f L_{ff} \end{bmatrix} \begin{bmatrix} \xi_a \\ \bar{x}'_f \end{bmatrix}.$$

From the above we derive

$$\bar{x}'_f = -(L_{ff} \otimes I_d)^{-1}(L_{fa} \otimes I_d)\xi_a = \bar{x}_f.$$

This shows that $\bar{x}$ is the unique fixed point of (9.11). Moreover, since all the eigenvalues of $I - E_f L_{ff}$ lie inside the unit circle, we derive

$$(\forall x_f(0) \in \mathbb{R}^{(n-d-1)d}) \lim_{k \to \infty} x_f(k) = -(L_{ff} \otimes I_d)^{-1}(L_{fa} \otimes I_d)\xi_a (= \xi_f)$$

Namely, the second condition in the arbitrary dimensional localization problem is established. This completes the proof. □

9.4 Simulation Examples

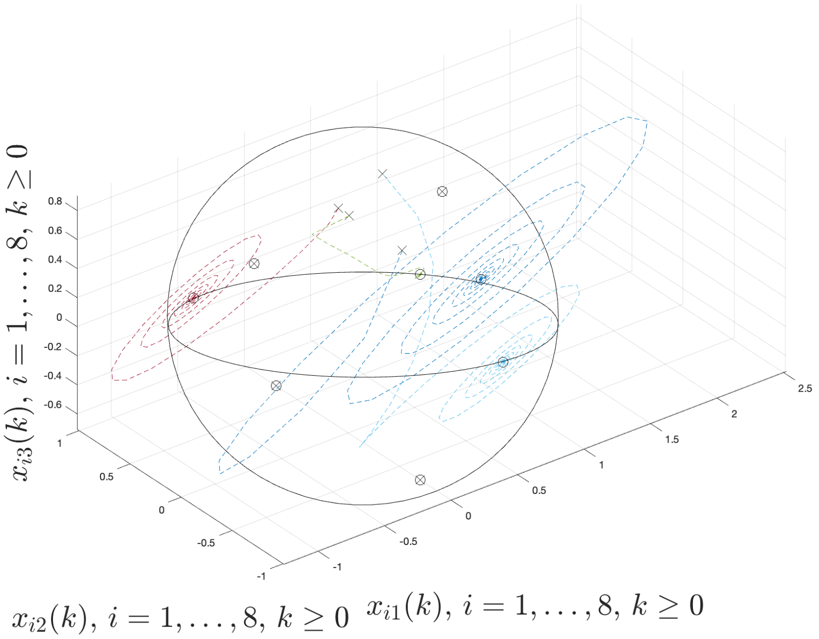


Figure 9.4: Estimations of four free agents converge to their true positions (×: initial estimation; ○: final estimation)

Example 9.3 *Let us consider again Example 9.2, where the (generic) configuration ξ consists of eight (three-dimensional) points on the unit sphere. We have designed a signed*

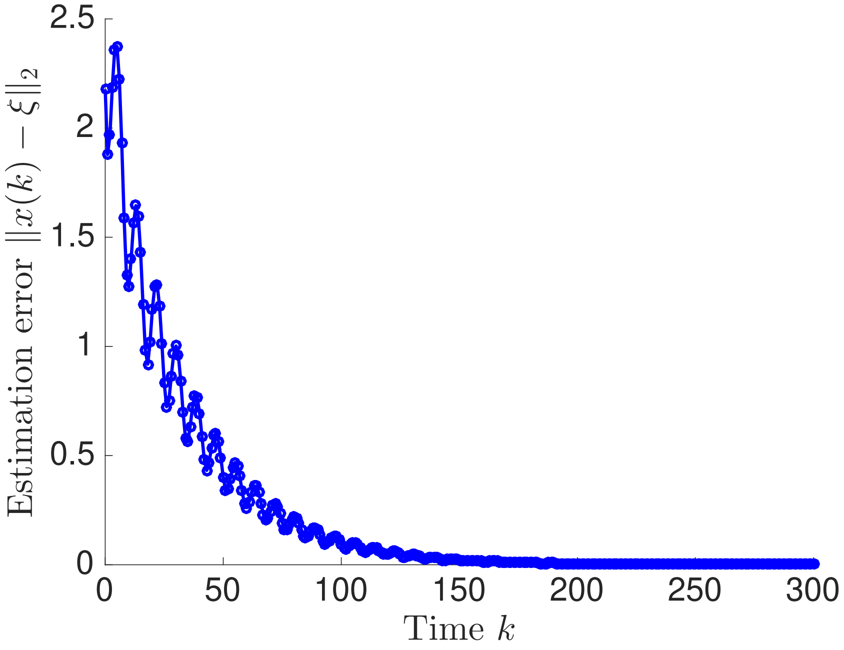


Figure 9.5: Estimation error of eight networked agents asymptotically converges to zero

Laplacian L (copied below for convenience)

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -3.7321 & 0 & 0 & 4.7321 & -1.9319 & 1.9319 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1.2247 & -0.4082 & -0.8165 \\ -1 & 0 & 0 & 0 & 1 & -0.9659 & -0.1494 & 1.1154 \\ 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 \end{bmatrix}.$$

While it is satisfied that rank$(L) = 4$, two of the eigenvalues of $I - L$ are unstable (i.e. outside the unit circle). Thus we need to design an invertible diagonal matrix E such that, except for the four eigenvalues 1, all the other four eigenvalues of $I - EL$ are stable (i.e. inside the unit circle).

Since the configuration ξ is generic and the digraph $\mathcal{G}$ contains a spanning 4-tree whose *four roots are the anchor agents* $1, 2, 3, 4$, *all the principal minors of the submatrix* L_{ff} *are nonzero. Therefore by Lemma 9.1, there exists an invertible diagonal matrix* E_f *such that all the eigenvalues of* $I - E_f L_{ff}$ *lie inside the unit circle. For computing such* E_f, *we apply Algorithm 9.1 and obtain*

$$E_f = \mathrm{diag}(0.2113, 0.2449, -0.1487, 0.4).$$

Then an invertible diagonal matrix E *such that, except for four eigenvalues* 1, *all the other four eigenvalues of* $I - EL$ *lying inside the unit circle is:*

$$E = \mathrm{diag}(1, 1, 1, 1, 0.2113, 0.2449, -0.1487, 0.4).$$

Indeed, the eigenvalues of $I - EL$ *are:*

$$1, 1, 1, 1, 0.903 + 0.3549\mathrm{j}, 0.903 - 0.3549\mathrm{j}, 0.7854, 0.0864.$$

With the initial condition $x_a(0) = \xi_a$ *of the four anchor agents and a random initial condition* $x_f(0) \in \mathbb{R}^{12}$ *of the four free agents, a simulation of the ADLA (i.e.* $x(k+1) = ((I - EL) \otimes I_3)x(k)$*) yields the trajectories displayed in Fig. 9.4. In the figure,* $\times$ *denotes the initial estimated positions, while* $\circ$ *the final estimated positions. First observe that the four anchor agents never change their estimations of their positions, because these global positions are already known and never need to be updated. For the four free agents, they start from some random estimations of their positions, and it is observed that these estimations converge to their true positions.*

Let $e(k) := \|x(k) - \xi\|_2$ *be the total estimation error of the networked agents. Then Fig. 9.5 shows that* $e(k)$ *converges to zero asymptotically.*

Example 9.4 *Consider a network of* 12 *agents in Example 8.5 (Fig. 8.5 is copied here for convenience). Agents* $1, 2, 3, 4$ *are anchor agents, and the rest are free agents. This digraph contains a spanning* 4*-tree whose four roots are the four anchor agents.*

Let us consider a configuration ξ *which is a 3D cuboid with*

- *an added random perturbation* $[p_1\ p_2\ p_3]^\top$, *where* $p_1, p_2, p_3 \in (0, 0.1)$
- *a* $\frac{\pi}{3}$ *rotation along the* x*-axis*
- *a* 3*-time scaling along all three dimensions*
- *a translation:* 1 *along the first dimension,* -1 *along the second dimension, and* 2 *along*

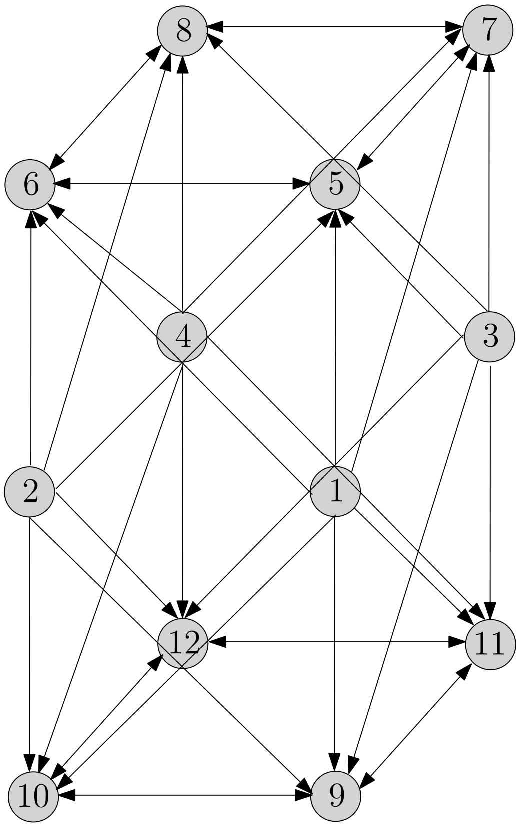


Figure 9.6: Twelve networked agents

the third dimension.

It is verified that this ξ is generic.
Now let $\xi_a = [\xi_1^\top \ \xi_2^\top \ \xi_3^\top \ \xi_4^\top]^\top$ and $\xi_f = [\xi_5^\top \ \cdots \ \xi_{12}^\top]^\top$. We design a signed graph Laplacian L such that rank$(L) = 8$, and compute by Algorithm 9.1 an invertible diagonal matrix E such that all the eigenvalues (except for four eigenvalues 1) of $I - EL$ are stable (i.e. inside the unit circle). With the initial condition $x_a(0) = \xi_a$ of the four anchor agents and a random initial condition $x_f(0) \in \mathbb{R}^{24}$ of the eight free agents, a simulation of the ADLA (i.e. $x(k+1) = ((I - EL) \otimes I_3)x(k)$) yields the trajectories displayed in Fig. 9.7. Observe that the estimations of the free agents converge to their true positions. The estimation error $e(k) := \|x(k) - \xi\|_2$ is displayed in Fig. 9.8, which converges to zero asymptotically.

Example 9.5 *Consider a network of* 27 *agents as displayed in Fig. 9.9. Agents* $1, 2, 3$ *are anchor agents, and the rest are free agents. This digraph contains a spanning* 3*-tree whose three roots are the three anchor agents.*
Consider a configuration ξ *which is a 2D ellipsoid obtained from the unit circle by*

- *a* 2*-time scaling along the second dimension*
- *a* 1*-unit translation along the first dimension.*

This ξ *is generic.*
Let $\xi_a = [\xi_1^\top \ \xi_2^\top \ \xi_3^\top]^\top$ *and* $\xi_f = [\xi_4^\top \ \cdots \ \xi_{27}^\top]^\top$*. We then design a signed graph Laplacian* L *such that* $\mathrm{rank}(L) = 24$*, and compute by Algorithm 9.1 an invertible diagonal matrix* E *such that all the eigenvalues (except for three eigenvalues 1) of* $I - EL$ *are stable (i.e. inside the unit circle). With the initial condition* $x_a(0) = \xi_a$ *of the three anchor agents and a random initial condition* $x_f(0) \in \mathbb{R}^{48}$ *of the twenty-four free nodes, a simulation of the ADLA (i.e.* $x(k+1) = ((I - EL) \otimes I_2)x(k)$*) yields the trajectories displayed in Fig. 9.10. Observe that the estimations of the free agents converge to their true positions. The estimation error* $e(k) := \|x(k) - \xi\|_2$ *is displayed in Fig. 9.11, which converges to zero asymptotically.*

9.5 Notes and References

The arbitrary dimensional localization algorithm (ADLA) is originated here, as a natural extension of 2D localization in Chapter 7 and arbitrary dimensional affine formation control in Chapter 8.

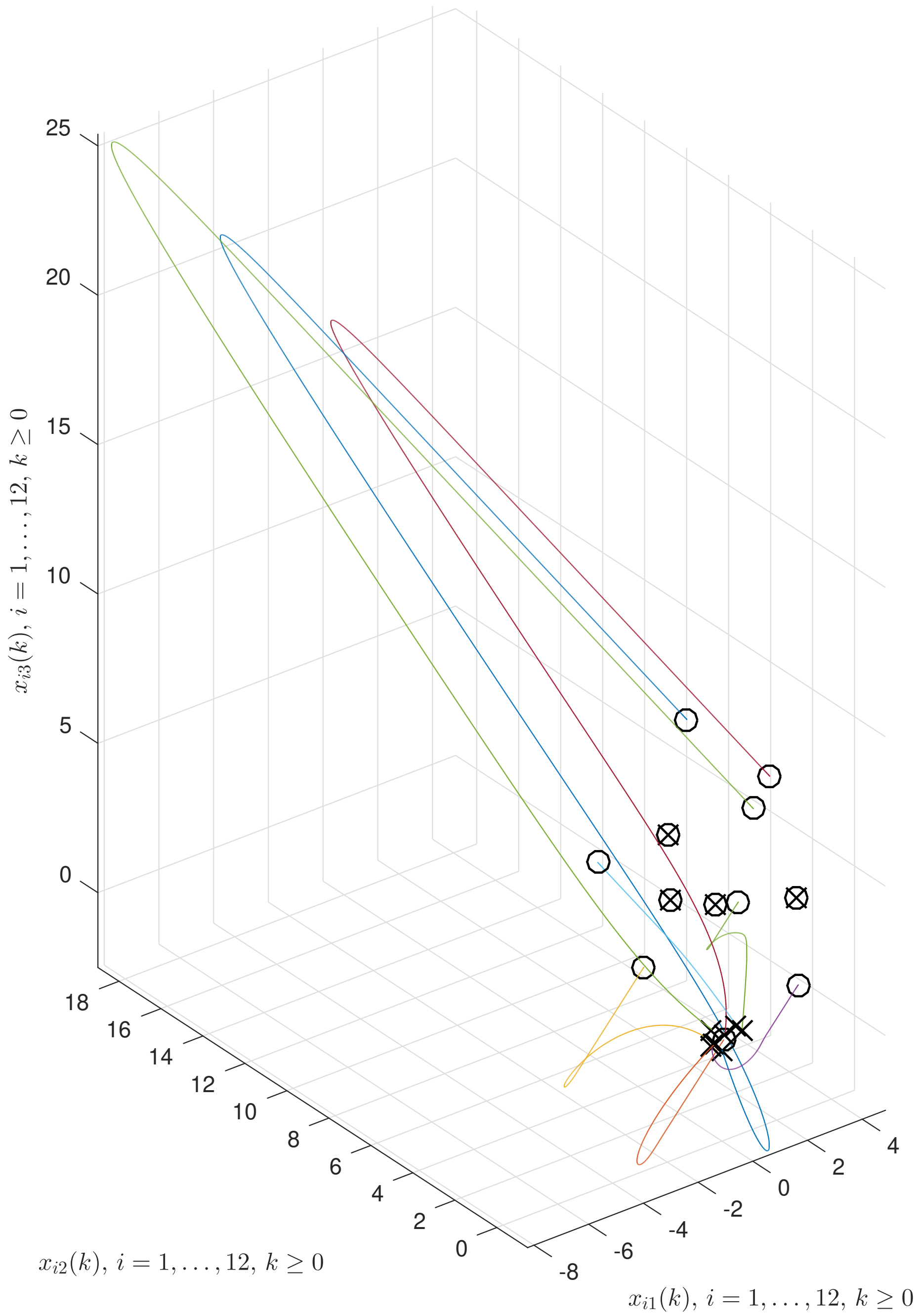


Figure 9.7: Generic configuration: estimations of eight free agents converge to their true positions (×: initial estimation; ○: final estimation)

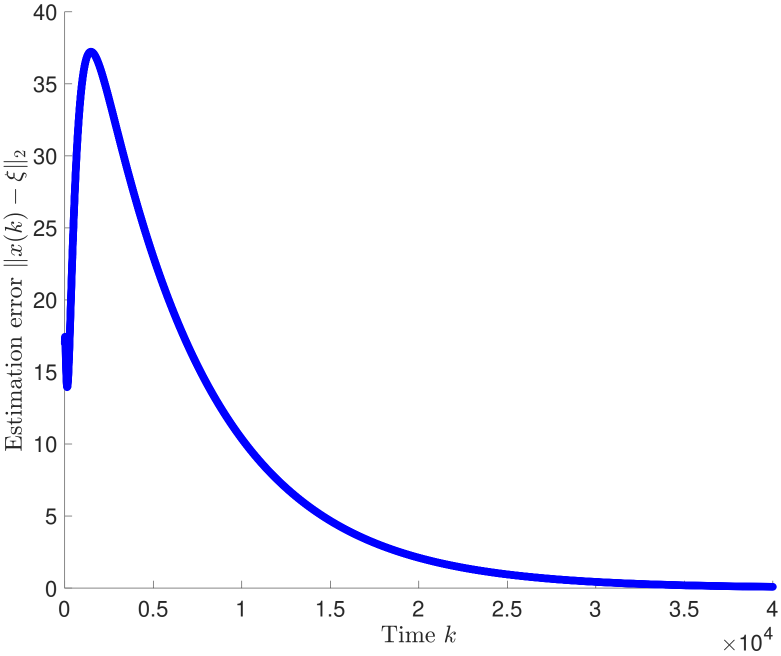


Figure 9.8: Generic configuration: estimation error of twelve networked agents asymptotically converges to zero

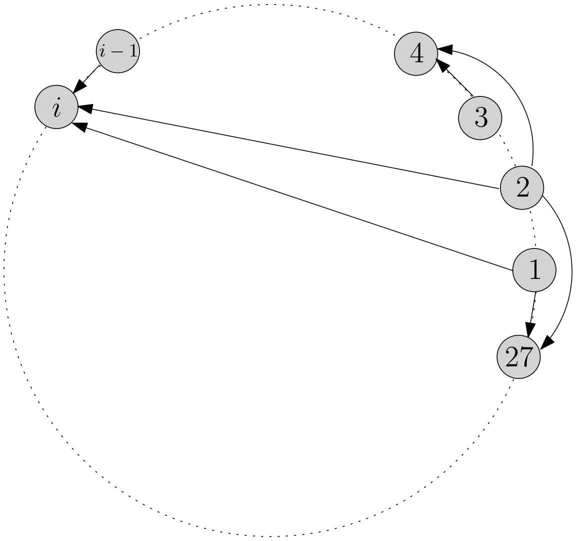


Figure 9.9: Twenty-seven networked agents

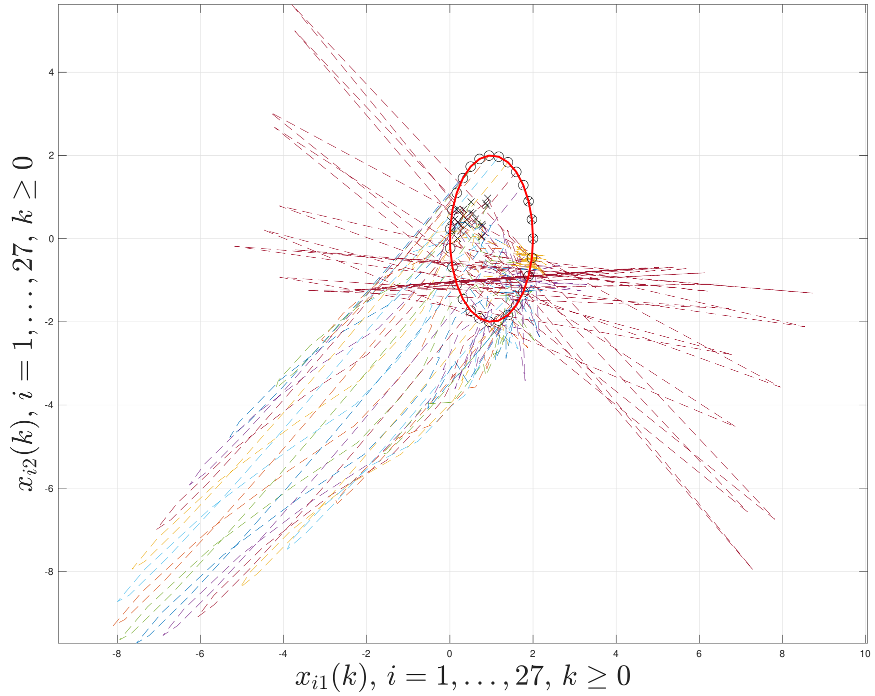


Figure 9.10: Generic configuration: estimations of twenty-four free agents converge to their true positions (×: initial estimation; ○: final estimation)

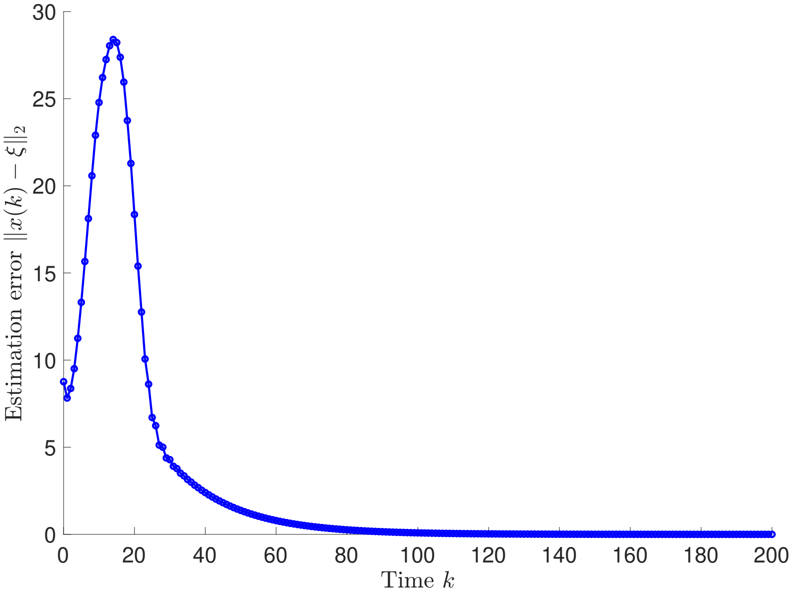


Figure 9.11: Generic configuration: estimation error of twenty-seven networked agents asymptotically converges to zero

Bibliography

[AL15] C. Altafini and G. Lini. Predictable dynamics of opinion forming for networks with antagonistic interactions. *IEEE Trans. Autom. Control*, 60(2):342–357, 2015.

[Bap10] R. B. Bapat. *Graphs and Matrices.* Springer, 2010.

[BAW11] H. Bai, M. Arcak, and J. Wen. *Cooperative Control Design.* 2011.

[Bul22] F. Bullo. *Lectures on Network Systems.* 2022.

[CAYM15] K. Cai, B. D. O. Anderson, C. Yu, and G. Mao. Local average consensus in distributed measurement of spatial-temporal varying parameters: 1d case. *Automatica*, 52(2):135–145, 2015.

[CI11] K. Cai and H. Ishii. Quantized consensus and averaging on gossip digraphs. *IEEE Trans. Autom. Control*, 56(9):2087–2100, 2011.

[CI12] K. Cai and H. Ishii. Average consensus on general strongly connected digraphs. *Automatica*, 48(11):2750–2761, 2012.

[CLC$^+$16] W. Chen, J. Liu, Y. Chen, S. Z. Khong, D. Wang, T. Basar, L. Qiu, and K. H. Johansson. Characterizing the positive semidefiniteness of signed laplacians via effective resistances. In *Proc. IEEE Conf. on Decision and Control*, pages 985–990, 2016.

[CWL$^+$17] W. Chen, D. Wang, J. Liu, T. Basar, and L. Qiu. On spectral properties of signed laplacians for undirected graphs. In *Proc. IEEE Conf. on Decision and Control*, pages 1999–2002, 2017.

[CWRKG20] J. Chen, H. Wang, M. Rubenstein, and H. Kress-Gazit. Automatic control synthesis for swarm robots from formation and location-based high-level specifications. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 8027–8034, 2020.

[CYRC13] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans Industrial informatics*, 9(1):427–438, 2013.

[DB10] F. Dorfler and F. Bullo. Synchronization and transient stability in power networks and non-uniform kuramoto oscillators. *SIAM J. Control and Optimization*, 50(3):1616–1642, 2010.

[DB14] F. Dorfler and F. Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539–1564, 2014.

[DCB13] F. Dorfler, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. *Proc. National Academy of Sciences*, 110(6):2005–2010, 2013.

[FJB16] N. E. Friedkin, P. Jia, and F. Bullo. A theory of the evolution of social power: Natural trajectories of interpersonal influence systems along issue sequences. *Sociological Science*, 3:444–472, 2016.

[FM16] B. A. Francis and M. Maggiore. *Flocking and Rendezvous in Distributed Robotics.* 2016.

[GR00] C. Godsil and G. Royle. *Algebraic Graph Theory.* Springer, 2000.

[GS10] F. Garin and L. Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. In A. Bemporad, M. Heemels, and M. Johansson, editors, *Networked Control Systems, Lecture Notes in Control and Information Sciences, Springer*, pages 75–107, 2010.

[HLZ+17] T. Han, Z. Lin, R. Zheng, Z. Han, and H. Zhang. A barycentric coordinate based approach to three-dimensional distributed localization for wireless sensor networks. In *Proc. IEEE Int. Conf. on Control & Automation*, pages 600–605, 2017.

[INK19] T. Ikeda, M. Nagahara, and K. Kashima. Maximum hands-off distributed control for consensus of multiagent systems with sampled-data state observation. *IEEE Trans. Control of Network Systems*, 6(2):852–862, 2019.

[JLM03] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents sing nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988–1001, 2003.

[KBG14] A. Khanafer, T. Basar, and B. Gharesifard. Stability properties of infection diffusion dynamics over directed networks. In *Proc. IEEE Conf. on Decision and Control*, page 6215–6220, 2014.

[KCK20] S. Kawamura, K. Cai, and M. Kishida. Distributed output regulation of heterogeneous uncertain linear agents. *Automatica*, 2020.

[LDY$^+$13] Z. Lin, W. Ding, G. Yan, C. Yu, and A. Giua. Leader-follower formation via complex laplacian. *Automatica*, 49:1900–1906, 2013.

[LFD15] Z. Lin, M. Fu, and Y. Diao. Distributed self localization for relative position sensing networks in 2d space. *IEEE Trans. Sig. Proc.*, 63(14):3751–3761, 2015.

[LHZF16] Z. Lin, T. Han, R. Zheng, and M. Fu. Distributed localization for 2-d sensor networks with bearing-only measurements under switching topologies. *IEEE Trans. Sig. Proc.*, 64(23):6345–6359, 2016.

[Lun12] J. Lunze. Synchronization of heterogeneous agents. *IEEE Trans. Autom. Control*, 57(11):2885–2890, 2012.

[LWC$^+$16] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han. Necessary and sufficient graphical conditions for affine formation control. *IEEE Trans. Autom. Control*, 61(10):2877–2891, 2016.

[LWHF14] Z. Lin, L. Wang, Z. Han, and M. Fu. Distributed formation control of multi-agent systems using complex laplacian. *IEEE Trans. Autom. Control*, 59(7):1765–1777, 2014.

[LWHF16] Z. Lin, L. Wang, Z. Han, and M. Fu. A graph laplacian approach to coordinate-free formation stabilization for directed networks. *IEEE Trans. Autom. Control*, 61(5):1269–1280, 2016.

[MC19] T. Motoyama and K. Cai. Top-down synthesis of multi-agent formation control: an eigenstructure assignment based approach. *IEEE Trans. Control of Network Systems*, 6(4):1404–1414, 2019.

[ME10] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. 2010.

[OGNK13] K. Oles, E. Gudowska-Nowak, and A. Kleczkowski. Efficient control of epidemics spreading on networks: Balance between treatment and recovery. *PLoS ONE*, 8(6):e63813, 2013.

[OPA15] K. K. Oh, M. C. Park, and H. S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[OS06] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control*, 51(3):401–420, 2006.

[OS07] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proc. IEEE Conf. on Decision and Control*, pages 5492–5498, 2007.

[OSFM07] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE*, 95(1):215–233, 2007.

[PR11] A. Pikovsky and M. Rosenblum. Dynamics of heterogeneous oscillator ensembles in terms of collective variables. *Physica D: Nonlinear Phenomena*, 240(9):872–881, 2011.

[RB08] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. 2008.

[Ren08] W. Ren. Synchronization of coupled harmonic oscillators with local interaction. *Automatica*, 44(12):3195–3200, 2008.

[SS08] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 45(11):2557–2562, 2008.

[SVC$^+$16] M. Saska, V. Vonasek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *J. Intelligent & Robotic Systems*, 84:469–492, 2016.

[WSA11] P. Wieland, R. Sepulchre, and F. Allgower. An internal model principle is necessary and sufficient for linear output synchronization. *Automatica*, 47(5):1068–1074, 2011.

[XHC$^+$17] Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, and M. Fu. A distributed algorithm for resource allocation over dynamic digraphs. *IEEE Trans. Sig. Proc.*, 65(10):2600–2612, 2017.

[YLA$^+$18] M. Ye, J. Liu, B. D. O. Anderson, C. Yu, and T. Basar. Evolution of social power in social networks with dynamic topology. *IEEE Trans. Autom. Control*, 63(11):3793–3808, 2018.

[YLAC21] M. Ye, J. Liu, B. D. O. Anderson, and M. Cao. Applications of the Poincare-Hopf theorem: Epidemic models and Lotka-Volterra systems. *IEEE Trans. Autom. Control*, 2021.

[Zha18] S. Zhao. Affine formation maneuver control of multiagent systems. *IEEE Trans. Autom. Control*, 63(12):4140–4155, 2018.

[ZYC20] J. Zhang, K. You, and K. Cai. Distributed conjugate gradient tracking for resource allocation in unbalanced networks. *IEEE Trans. Sig. Proc.*, 68:2186–2198, 2020.