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 its neighbors in $\mathcal{N}_i$.

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.

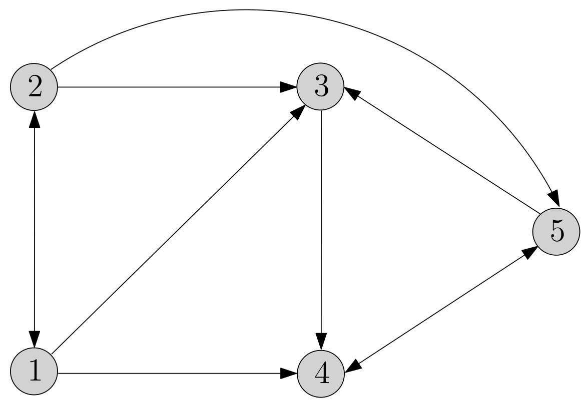

Figure 4.1: Illustrating example of consensus problem with five agents

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.

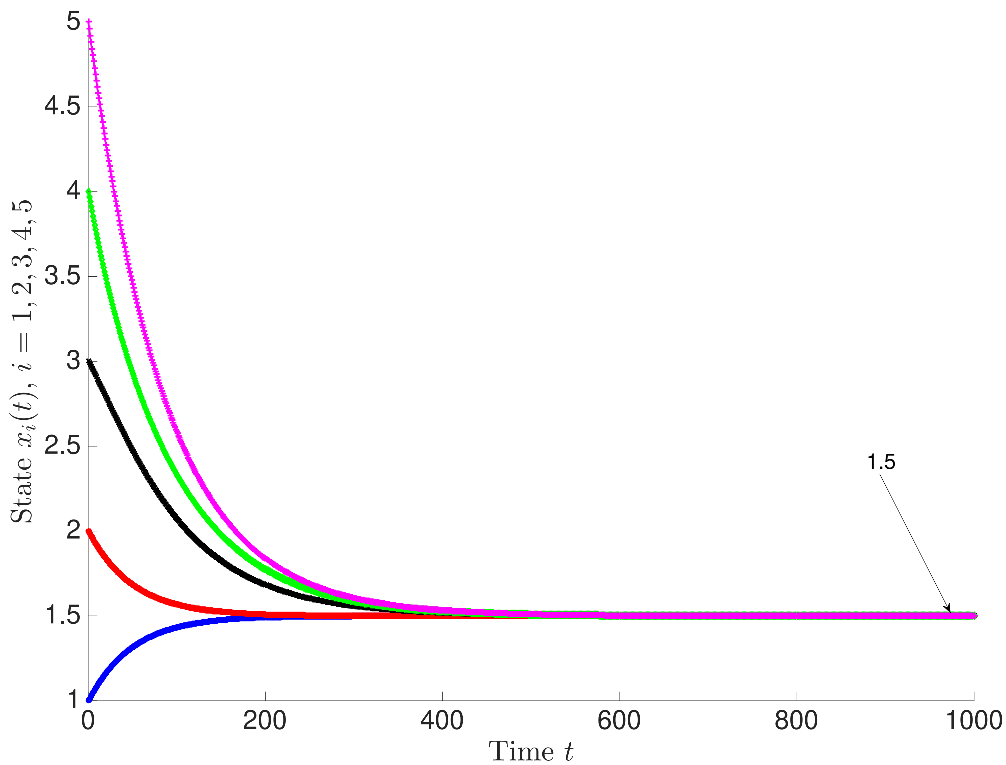

Figure 4.2: Success of achieving consensus

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 edges (j, i) (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 $x := [x_1 \cdots x_n]^\top \in \mathbb{R}^n$ 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:

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